

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra systémového inženýrství**



**Diplomová práce**

**Agilní metodiky v hardwarových projektech**

**Bc. Kateřina Tošerová**

© 2023 ČZU v Praze

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Kateřina Tošerová

Projektové řízení

Název práce

**Agilní metodiky v hardwarových projektech**

Název anglicky

**Agile in Hardware Projects**

### Cíle práce

Cílem závěrečné práce je zhodnocení, zda se agilní metodiky dají využít na hardwarové projekty. Pomocí agilních metodik bude simulován již řešený projekt zabývající se implementací IT infrastruktury. Výsledná řešení budou porovnána za pomoci agilních metodik s výsledky z realizace projektu pomocí interní metodiky vybrané společnosti.

### Metodika

Závěrečná práce bude rozdělena na teoretickou a analytickou část. V teoretické části budou podrobněji popsány teoretické poznatky potřebné pro řešení zpracovávané problematiky v této práci. Jedná se zejména o problematiku projektového řízení vedené klasickým a agilním způsobem. Taktéž budou popsány i vybrané agilní metodiky. Podkladové údaje budou čerpány z relevantních zdrojů dat, knih, odborných článků a prací.

V analytické části práce bude věnována pozornost interní metodice vybrané společnosti a její následné porovnání s řešeným HW projektem. Na základě výsledku projektu budou stanoveny ohrožení a nedostatky, které se pomocí interní metodiky neošetřily. Stejně tak budou vyzdviženy i aspekty, kterých bylo dosaženo. Následně bude HW projekt simulován s využitím vybraných agilních metodik. Poté bude provedeno porovnání a diskuze, zda je vhodné využívat agilní metodiky na HW projekty či nikoliv. Na základě výše zmíněných poznatků bude vyhodnoceno, zda a do jaké míry lze využít agilní metodiky pro řešení HW projektů.

## Doporučený rozsah práce

60 stran

## Klíčová slova

projektové řízení, agilní přístup, hardwarové projekty, PRINCE2, agilní manifesto, agilní metodiky

---

## Doporučené zdroje informací

Apke, L. (2015) Understanding the Agile Manifesto. null edition. lulu.com. ISBN 978-1-312-86391-0.

AXELOS LIMITED. Managing successful projects with PRINCE2®. Norwich: TSO, 2017. ISBN 978-0-11-331533-8.

Beck, K. and Andres, C. (2005) Extreme programming explained: embrace change. 2nd ed. Boston, MA: Addison-Wesley. ISBN 0321278658.

Bentley, C. (2015) The Prince 2 practitioner: from practitioner to professional. Third edition. London; New York: Routledge, Taylor & Francis Group. ISBN 9781138824119.

Buchalcevoá, A. (2005) Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky. 1. vyd. Praha: Grada. ISBN 80-247-1075-7.

Campbell, A. (2020) Agile: Essentials of Team and Project Management. Manifesto for Agile Software Development. Amazon Services LLC – KDP Print US (Agile Project Management with Kanban). ISBN 979-86-631-8258-4.

KERZNER, H. Project management : a systems approach to planning, scheduling, and controlling. Hoboken: John Wiley & Sons, 2013. ISBN 978-1-118-02227-6.

---

## Předběžný termín obhajoby

2022/23 LS – PEF

## Vedoucí práce

doc. Ing. Tomáš Šubrt, Ph.D.

## Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 30. 11. 2022

**doc. Ing. Tomáš Šubrt, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 17. 2. 2023

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 14. 03. 2023

## **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci "Agilní metodiky v hardwarových projektech" jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 21.3.2023

---

## **Poděkování**

Ráda bych touto cestou poděkovala vedoucímu práce doc. Ing. Tomáši Šubrtovi, Ph.D. za vedení diplomové práce, odborný přístup a podnětné návrhy, které práci obohatily. Dále bych chtěla poděkovat konzultantce diplomové práce Ing. Petře Pavlíčkové, Ph.D. za profesionální přístup během konzultací, které rovněž práci obohatily o podnětné návrhy.

# Agilní metodiky v hardwarových projektech

## **Abstrakt**

Diplomová práce se zabývá problematikou, zda se agilní metodiky dají využít na hardwarové projekty. Tato problematika je simulována pomocí vybraných agilních metodik na konkrétním projektu, který spočívá v implementaci IT infrastruktury.

V literární rešerši jsou popsány teoretické poznatky zabývající se klasickým a agilním typem řízení projektů a taktéž jsou popsány vybrané agilní metodiky jako Kanban, Lean development, Crystal metodiky, Extrémní programování a SCRUM. Po zpracování literární rešerše je proveden rozbor projektového řízení ve zvolené společnosti včetně konkrétního hardwarového projektu. Dále se práce věnuje simulacím hardwarového projektu na zvolené agilní metodiky, které jsou podrobně popsány v literární rešerši. Zároveň se práce odkazuje i na projekty, které se agilním přístupem nedají ošetřit. Díky těmto simulacím, diskuzi a potřebných znalostí dané problematiky se dojde k závěru, zda je možná agilní metodiky aplikovat na hardwarové projekty či nikoliv.

**Klíčová slova:** projektové řízení, agilní přístup, PRINCE2, agilní manifesto, agilní metodiky, SCRUM, ZZS JČK IT Technologie, hardwarový projekt

# Agile methods in hardware projects

## **Abstract**

The thesis deals with the issue of whether agile methodologies can be applied to hardware projects. This issue is simulated using selected agile methodologies on a specific project, which consists in the implementation of IT infrastructure.

The literature search describes the theoretical knowledge dealing with classical and agile types of project management and also describes selected agile methodologies such as Kanban, Lean development, Crystal methodologies, Extreme Programming and SCRUM. After the literature search, an analysis of project management in the selected company including a specific hardware project is made. Further, the paper discusses the simulations of the hardware project on the selected agile methodologies which are detailed in the literature search. At the same time, the thesis also refers to projects that cannot be treated by the agile approach. Through these simulations, discussion and the necessary knowledge of the subject matter, it is concluded whether agile methodologies can be applied to hardware projects or not.

**Keywords:** project management, agile approach, PRINCE2, agile manifesto, agile methodologies, SCRUM, ZZS JČK IT Technologie, hardware project

# Obsah

<b>1. Úvod.....</b>	<b>10</b>
<b>2. Cíl metodika.....</b>	<b>11</b>
2.1 Cíl práce .....	11
2.2 Metodika .....	11
<b>3. Teoretická východiska .....</b>	<b>13</b>
3.1 Projektové řízení .....	13
3.1.1 Definice projektového řízení .....	13
3.1.2 Definice projektu .....	14
3.1.2.1 Projektový board.....	15
3.1.2.2 Cíl projektu .....	16
3.1.2.3 Projektový trojimperativ .....	17
3.1.2.4 Životní cyklus projektu.....	18
3.2 Standardy projektového řízení .....	20
3.2.1 Standard projektového řízení podle PMI PMBoK.....	20
3.2.2 Metodika řízení projektů podle PRINCE2.....	21
3.2.3 Standard projektového řízení podle IPMA ICB .....	22
3.3 Vodopádový přístup řízení projektů.....	22
3.4 Agilní přístup řízení projektů .....	24
3.4.1 Vznik a definice.....	24
3.4.2 Agilní manifest .....	27
3.4.3 Agilní metody řízení projektů .....	28
3.4.3.1 Kanban .....	29
3.4.3.2 Lean development.....	31
3.4.3.3 Extrémní programování (XP) .....	33
3.4.3.4 Crystal metodiky .....	35
3.4.3.5 Metoda SCRUM .....	38
3.5 Porovnání vodopádového a agilního přístupu řízení projektů .....	45
<b>4. Vlastní práce.....</b>	<b>48</b>
4.1 Profil společnosti.....	48
4.1.1 Organizační struktura .....	49
4.1.2 Projektové řízení ve Společnosti XY .....	50
4.1.3 Životní cyklus zakázky .....	50
4.1.4 Interní metodika .....	51
4.1.5 Interní metodika vs. PRINCE2 .....	55



4.2 Vybraný projekt – ZZS JČK IT Technologie .....	58
4.2.1 Problémová místa v projektu, která nebyla dostatečně ošetřena.....	62
4.3 Aplikace agilních metodik na projekt ZZS JČK IT Technologie .....	64
4.3.1 Kanban .....	65
4.3.2 Lean development.....	68
4.3.3 Extrémní programování .....	70
4.3.4 Crystal metodiky.....	72
4.3.5 SCRUM .....	75
4.4 Typy projektů, které se agilním přístupem nedají ošetřit.....	79
<b>5. Výsledky a diskuse .....</b>	<b>81</b>
<b>6. Závěr.....</b>	<b>84</b>
<b>7. Seznam použitých zdrojů .....</b>	<b>86</b>
Knižní zdroje.....	86
Internetové zdroje.....	88
<b>8. Seznam obrázků, tabulek, grafů a zkratk .....</b>	<b>90</b>
Seznam obrázků .....	90
Seznam tabulek .....	90
Seznam použitých zkratk.....	90
<b>9. Přílohy .....</b>	<b>91</b>

# 1. Úvod

Nové produkty a služby jsou vyvíjeny tak, aby splňovaly jedinečné potřeby zákazníků, uspokojovaly požadavky trhu a dosahovaly předem stanovených cílů. Hlavní součástí tohoto procesu je projektové řízení. Plánování, koordinace, dohled a řízení všech činností potřebných k efektivnímu dokončení projektu jsou hlavními cíli projektového řízení. Účelem projektového řízení je zajistit, aby byl projekt dokončen včas, v odpovídající kvalitě, v rámci přiděleného rozpočtu a v souladu s přáními a potřebami klienta.

Existují dva základní typy projektového řízení, a to tradiční a agilní přístup. Tradiční neboli waterfallový přístup řízení projektů se zaměřuje na pečlivé plánování a řízení, krok po kroku, s velkým důrazem na stanovené cíle, harmonogram a kvalitu. Tento typ řízení je vhodný pro projekty, které se v průběhu nemění a neočekává se od projektového týmu žádná flexibilita a přizpůsobivost.

Naopak agilní přístup klade velký důraz na flexibilitu a přizpůsobivost a zároveň využívá vysokou efektivitu k dosažení cílů klienta. Agilní metodiky si získávají oblibu nejen u vývojových týmů, ale i mezi projektovými manažery. Agilní metodiky jako jsou např. SCRUM, Kanban, Lean development, Crystal metodiky či Extrémní programování, využívají inkrementální přístup k vývoji, který umožňuje rychlé přizpůsobení změnám požadavků a podmínek projektu. Jsou tak vhodné zejména pro vývoj softwaru.

Problematika agilního přístupu řízení projektů má stále rostoucí význam, nejen v oblasti softwarového vývoje, ale také v oblasti hardwarových projektů. Diplomová práce se věnuje možnosti využití agilních metodik na hardwarové projekty. Ačkoliv každá metodika má své výhody a nevýhody, postupy, nástroje a praktiky, může tato práce sloužit jako návod, jak použít agilní metodiky při řízení hardwarových projektů.

## **2. Cíl metodika**

### **2.1 Cíl práce**

Cílem diplomové práce je zhodnocení, zda se agilní metodiky dají využít na hardwarové projekty. Pomocí agilních metodik bude simulován již řešený projekt zabývající se implementací IT infrastruktury. Výsledná řešení získaná aplikací jednotlivých metodik budou porovnána jak mezi sebou, tak s výsledky z realizace projektu pomocí interní metodiky vybrané společnosti.

### **2.2 Metodika**

Diplomová práce bude rozdělena na teoretickou a analytickou část. V teoretické části se bude vycházet ze studia odborných článků a publikací. Nejprve se teoretická část bude věnovat popisu projektu a jeho životního cyklu, dále budou krátce zmíněny standardy či metodiky projektového řízení jako PRINCE2, PMBoK a ICB. Následující část definuje vodopádový přístup řízení projektů. Posléze se práce přesune k definování agilního přístupu a podrobněji se zaměří na vybrané agilní metodiky jako SCRUM, Kanban, Lean development, Extrémní programování a Crystal metodiky. V závěru teoretické části dojde k porovnání obou přístupů k řízení projektů, v čem tkví jejich přínosy a naopak nedostatky.

Analytická část se zpočátku bude věnovat rozboru projektového řízení ve Společnosti XY, včetně interní metodiky a její rozbor postupu při řízení hardwarových projektů. Po tomto rozboru a získání tak potřebných informací ohledně interní metodiky bude využit konkrétní hardwarový projekt, který spočívá v implementaci IT infrastruktury a bude na něm zkoumáno, do jaké míry se během jeho řízení projektový manažer řídil interní metodikou. Dále dojde k porovnání interní metodiky s metodikou PRINCE2, jelikož se na ni Společnost XY v interní metodice odkazuje.

Jako další oblast řešená v praktické části bude stanovení problémů a nedostatků u konkrétního projektu, které během jeho realizace vznikly a nebyly ničím ošetřeny. Tyto nedostatky a problémy se dále využijí při samotné simulaci jednotlivých agilních metodik na konkrétní HW projekt.

Pro simulaci budou vybrány agilní metodiky SCRUM, Kanban, Lean development, Extrémní programování a Crystal metodiky a pomocí jejich principů, nástrojů a postupů proběhne aplikace na konkrétní hardwarový projekt ZZS JČK IT Technologie. Aplikace každé z metodik na tentýž projekt zapříčiní jejich snadnější porovnání. Zároveň proběhne i identifikace nedostatků každé z nich. Na základě toho může být posléze vybrána ta nejvhodnější.

V neposlední řadě bude zmíněno, jaké typy projektů se agilním přístupem nedají v žádném případě vést.

Posledním krokem celé práce bude zhodnocení problematiky aplikace agilních metodik na hardwarové projekty na základě znalostí z teoretické části a simulací, které proběhly v praktické části. Tato práce může sloužit jako návod, jak uplatnit agilní metodiky na hardwarové projekty.

## 3. Teoretická východiska

### 3.1 Projektové řízení

#### 3.1.1 Definice projektového řízení

Projektové řízení je podobor managementu, který se zabývá řízením realizace projektů. Jedná se tedy o proces, ve kterém se sám člověk, nebo organizace snaží využít svoje zdroje k realizaci projektů. Jde se o způsob plánování a realizace činností, které je potřeba uskutečnit v požadovaném termínu s plánovanými náklady tak, aby bylo dosaženo nadefinovaných cílů.

Účelem projektového řízení je tedy úspěšná realizace projektu. Projekty jsou nástrojem pro vykonání změny, jsou svou povahou jedinečné, a proto se vyznačují vyšší mírou rizika oproti běžným provozním činnostem organizace. Jako pomůcky při řízení projektů existují různé metodologie a standardy.

Projektové řízení má mnoho definic, nicméně všechny mají obdobnou podstatu.

Lock (2007) říká, že projektové řízení je nástrojem k zavedení určité změny, kterou nelze jinak zaručit než právě projektem, který je souhrn prováděných činností utvářejících cestu od počátečního stavu k formulovanému cílovému stavu.

Jiný pohled, avšak stejný význam, je od Project Management Institute (2013), kdy je projektové řízení aplikace znalostí, schopností, nástrojů a technologií na aktivity projektu tak, aby splnily požadavky projektu.

*„Projektovým řízením (project management) se rozumí soubor norem, doporučení a best of practice zkušeností, popisujících, jak řídit projekt. Vzhledem k různorodosti projektů jako takových se veskrze jedná spíše o všeobecně platné skutečnosti, určitou filozofii přístupu k řešení dané problematiky než o konkrétní a podrobné směrnice, návody apod“ (Doležal, 2016).*

Project Management Institute (2013) dělí řízení projektů do pěti základních oblastí:

- zahájení (definování) – definování projektových cílů a účelu, zahájení aktivity
- plánování – naplánování, jak budou splněny požadavky a cíle projektu (které metody a postupy budou použity); specifikace provedení, časového plánu a finančního rozpočtu;
- vykonání – realizace výstupů a dodávek naplánovaným způsobem;
- sledování (monitorování) – kontrola stavu a postupu projektových prací, aby byly včas zjištěny odchylky od plánu, a ty mohly být zavčas korigovány;

- ukončení – ověření, že hotový úkol odpovídá aktuální definici toho, co se mělo udělat (odpovídá specifikaci v zadání), a uzavření všech nedokončených prací, např. dokumentace (včetně dokumentace vyhodnocení průběhu projektu).

*„Rozdělení do pěti manažerských oblastí pokrývá všechny potřebné aktivity a představuje užitečnou koncepční posloupnost. Takto lze pohlížet na celý průběh řízení projektu nebo i na jeho dílčí etapy a případně i jednotlivé činnosti“* Project Management Institute (2013).

### 3.1.2 Definice projektu

Definovat pojem projekt není až tak jednoduché. Samotný pojem lze chápat mnoha směry. Znalejší v oboru projektového řízení jistě tuší význam, ale mnoho jiných si pod slovem projekt představí spíše aktivitu ze stavebnictví, kterou má na starosti projektant, a proto je důležité nezaměňovat pojmy projektant a projektový manažer.

Bentley (2015) definuje projekt jako dočasnou organizaci, která je vytvořena za účelem dodání jednoho nebo více obchodních produktů dle sjednaného Business Case.

Svozilová (2016) definuje pojem projekt takto: *„Projekt je dočasné úsilí vynaložené na vytvoření unikátního produktu, služby nebo určitého výsledku.“*

A v neposlední řadě dle Kerznera (2017) je projekt sled aktivit, které:

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

S definicemi projektu je to tedy obdobné, jako tomu bylo u definic projektového řízení. Není jedna pevně daná, ale je jich mnoho se stejným či podobným významem.

Na první pohled nejvíce srozumitelná definice projektu je taková, že je projekt unikátní posloupnost aktivit za účelem dosažení specifického cíle při omezených zdrojích a také v omezeném časovém rámci. Tzn., že v projektu musí být jasně definovaný čas (začátek a konec projektu), zdroje a náklady, aby mohlo být dosaženo požadovaného cíle. Ne vždy je to takto jednoduché a proveditelné.

Dále je nutné rozdělovat projekt a klasické rutinní práce. Projekt je vymezen dle Doležala a Krátkého (2016) pomocí zdrojů, času a peněz. Od rutinní práce se liší komplexností a složitostí. Zároveň je projekt jedinečný a rizikový.

Projekt lze tedy zapsat i takto: = funkce (cíl, čas, zdroje). Znamená to, projekt je ovlivňován každou z veličin, přičemž vztah mezi veličinami a důležitost nelze zobecnit.

### 3.1.2.1 Projektový board

Za exekutivní a programové řízení je zodpovědný výbor, odborně nazývaný projektový board. Ten zároveň komunikuje s projektovým týmem a klientem. Výše zmíněné řízení je důležité pro strategické rozhodování na úrovni cílů projektů. Oproti tomu projektový tým je skupina všech osob, jenž se podílejí na realizaci samotného projektu. Spadá do něj vedení projektu, řešitelé, členové řídicího výboru i osoby jinak aktivně podílející se na jeho realizaci. Je obvykle jmenován napříč organizační strukturou, jinými slovy jsou v něm členové ze všech hierarchických úrovní firmy. Každý člen projektového týmu má své vymezené úkoly a pravomoci. (AXELOS, 2017).

Aby mohl být sestaven dobrý projektový tým je nejdůležitějším faktorem to, že se musí všichni jeho členové soustředit na dosažení cíle projektu a zároveň musí mít každý člen respekt k základním pravidlům týmové spolupráce.

Velikost projektového týmu není pevně dané, a je to z toho důvodu, že každý projekt je unikátní, a právě tak se k němu i přistupuje ze strany projektového týmu. Tzn., někdy se projektový tým skládá z několika desítek lidí, kdy jde o mohutný projekt, naopak někdy stačí i 5 lidí, kteří tvoří projektový tým a jeho jednotlivci zastupují i více rolí najednou (Kerzner, 2017).

Projektový tým by se měl skládat z odborníků, které určí projektový manažer. Schopný tým je složen nejen z profesně zdatných jedinců, ale je i potřeba, aby byli schopni spolupráce mezi sebou, dokázali se rozhodovat. Pokud toto bude projektový tým splňovat, je vysoká šance, že vytvoří hodnotný výstup práce a tím pádem úspěšný projekt (Máchal a kol, 2015).

Osoba, která je zodpovědná za koordinaci a sjednocení činností přes oblasti rozhodování v organizaci, je projektový manažer. Je tím hlavním, kdo má za úkol stanovit jasné a dosažitelné cíle projektu a zároveň jejich dosažení v rámci řízení projektu. Z toho plyne, že mezi jeho nejběžnější aktivity patří ty, které vedou k vývoji projektového plánu,

jeho uskutečnění a zároveň také aktivity, které napomáhají k aplikaci změn v plánu. Měl by to být člověk, který má nejen technický základ v dané oblasti, ale zároveň musí umět vést lidi.

Projektový manažer má vůči svému týmu závazky v podobě jeho sestavení, dále v sestavení plánu činností, které se svým týmem projedná, následně organizaci týmové práce, doporučení vhodných řešitelů na jednotlivé úkoly a v neposlední řadě také hodnocení jednotlivých členů a udržování motivace k dokončení projektu (Máchal a kol, 2015).

### 3.1.2.2 Cíl projektu

Stanovení cíle u jakéhokoliv projektu je jedním z klíčových věcí, bez kterého se projekt nemůže obejít. Pokud projektový manažer správně nedefinuje se zákazníkem cíl, nikdy nemůže být projekt úspěšně dokončen a předám spokojenému zákazníkovi. Toto je jeden ze stěžejních úkolů projektového manažera, aby od zákazníka získal informace o tom, co by si přál a následně pak tyto informace převedl na realizovatelný cíl projektu, tzn. aby tento cíl odpovídal co nejpřesněji požadavku zákazníka.

Cíl projektu je tedy důvod, proč projektový tým produkuje výstupy. Za výstupy jsou označovány produkty, resp. dodávky, dále také realizovatelné služby nebo výsledky, které zákazník požaduje. Nutno podotknout, že zákazník je taktéž vlastník či sponzor projektu.

Jeden z klíčových úspěchů projektu je tedy správně definovaný cíl. Existuje metoda SMART, která je velice populární a efektivní pro tvorbu reálných a dostupných cílů.

Dle Doležala a kol. (2012) by měl být cíl stanoven pomocí mnemotechnické pomůcky SMART, kdy každé písmeno vyjadřuje popis toho, jak by měl být cíl stanoven.

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

I když je metoda SMART velice rozšířená a je vynikajícím nástrojem pro tvorbu cílů v projektech, existuje již její upgradovaná „chytřejší“ verze SMARTER. Písmeno E lze chápat jako evaluate – hodnocený, a poukazuje na to, že nestačí pouze měřitelně stanovit cíl,



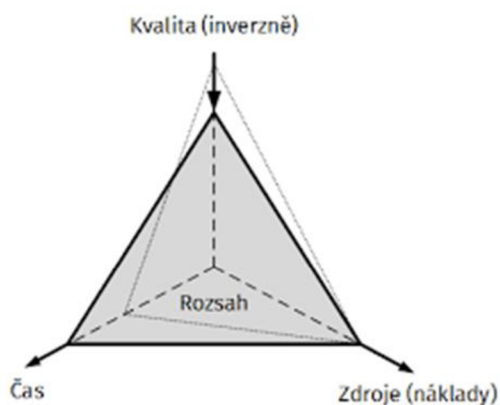
ale je taktéž potřeba ho na konci vyhodnotit. Druhé z písmen, tedy R, znamená reward, tedy odměnit. Význam je obdobný jako v prvním případě. SMARTER metoda oproti SMART verzi vzbuzuje u řešitelů díky motivaci v podobě odměny iniciativu pro kvalitnější výsledky vytvoření v co nejkratším čase (Dcosta, 2012).

Mimo těchto metodik ještě existuje tuzemská alternativa se zkratkou KARAT. Její význam je obdobný, jako mají předešlé dvě metodiky:

- Konkrétní
- Ambiciózní
- Reálný
- Akceptovatelný/akceptovaný
- Termínovaný

### 3.1.2.3 Projektový trojimperativ

Vymezenost projektu nejlépe znázorňuje projektový trojimperativ. Ten ve svých vrcholech zřetelně znázorňuje, čím a jakou hodnotou je projekt omezen. Každý z jeho vrcholů je závislý na těch zbývajících, mezi všemi vrcholy tedy platí vztah ekvivalence. Pokud změním kvalitu (cíl), změní se nám i čas a náklady. Nicméně jejich směr působení je odlišný.



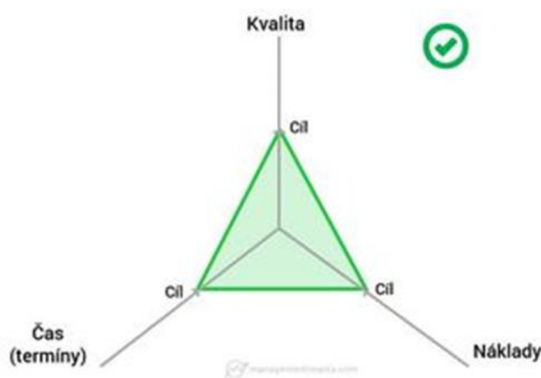
Obrázek 1: Projektový trojimperativ (zdroj: ManagementMania.com, 2023)

Svozilová (2016) tvrdí, že pro úspěšné ukončení projektu musí být kvalita, náklady a čas udrženy v rovnováze.

Trojimperativ tvoří cíl projektu, který je vyjádřený veličinami: výsledek (kvalita), čas a náklady (zdroje).

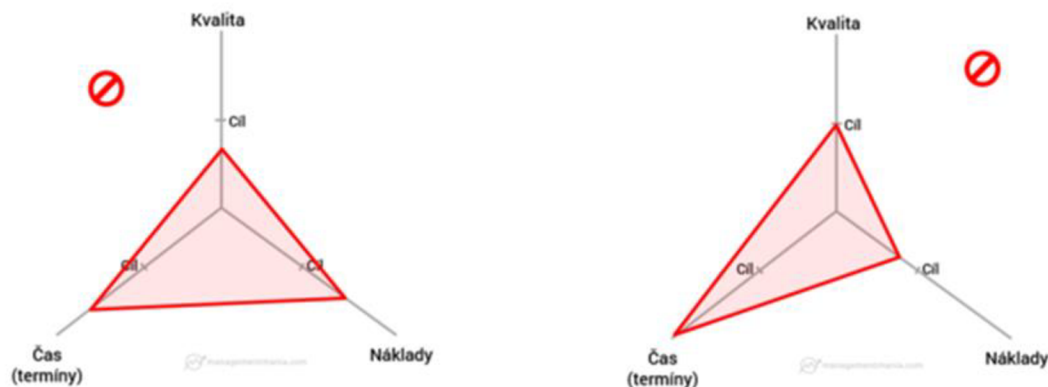
Všechny tři zmíněné veličiny jsou tedy provázané a navzájem se ovlivňují, a to jak pozitivně, tak negativně a nikdy není možné změnit pouze jednu z nich. V nejlepším případě

je cílem dosáhnout maximálního výsledku za minimálního času a nákladů. A to je dalším úkolem projektového manažera.



Obrázek 2: Správné vedení projektu (zdroj: ManagementMania.com, 2023)

Na obrázku 2 je poměrně viditelné, že je projekt veden správně, oproti obrázku 3, kde jsou problémy již zřetelné. Opět se zde propisuje již zmíněné pravidlo, že s jakoukoliv změnou jedné veličiny, se mění i ty zbylé, tzn. se zvyšujícími se náklady a časem se snižuje kvalita projektu. Nebo v jiném případě i jen samotné snížení nákladů může vést k prodloužení projektu, což je nežádoucí.



Obrázek 3: Špatné vedení projektu (zdroj: ManagementMania.com, 2023)

#### 3.1.2.4 Životní cyklus projektu

Za životní cyklus projektu jsou považovány fáze, resp. jde o nějakou posloupnost několika fází, kterými každý projekt prochází. Tyto fáze jsou v odborné literatuře rozděleny a popisovány různě, ale jako tomu již bylo s definicemi dříve, tak se opět ukazuje to, že si jsou podobné. Obvykle se jejich počet pohybuje mezi čtyřmi až osmi s ohledem na velikost a náročnost projektu. Mezi základní členění projektu patří fáze koncepční, plánu, realizace a předání (Fiala, 2004).

Ze zkušeností v praxi je známo, že nemůže nastat situace, kdy bude naprosto stejný životní cyklus a průběh jednoho projektu totožný s jiným projektem. Je to dáno tím, že každý projekt je unikátní a má specifický cíl, a proto nelze aplikovat jeden životní cyklus na všechny druhy projektů. Navíc, každá společnost si uzpůsobuje životní cyklus projektu svým způsobem, aby to vyhovovalo jejich standardům (Kerzner, 2009).



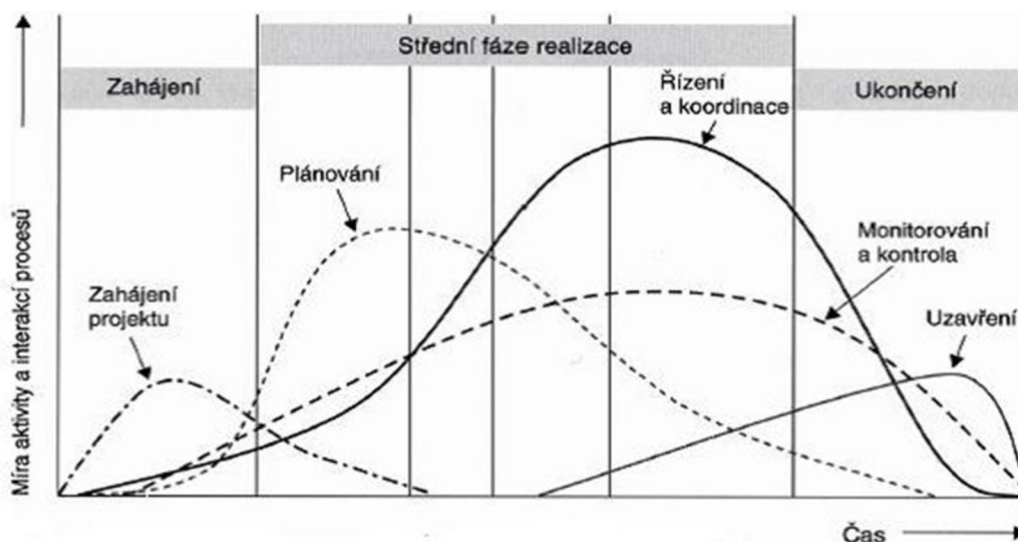
Obrázek 4: Projektové fáze (zdroj: vlastní zpracování)

Ačkoliv byly nastíněny různé definice o životním cyklu projektu, obrázek 4 poukazuje na tři hlavní fáze, které mají ještě své podfáze. Než začne samotný projekt, je důležité se zaměřit na předprojektovou fázi. Zde je potřeba založit business case či případové studie. Doležal a kol. (2012) zmiňují, že tato fáze je od toho, aby prozkoumala příležitost pro daný projekt a usoudila, zda je záměr proveditelný. Má vlastně poskytnout odpovědi na důležité otázky týkající se projektu.

Samotná projektová fáze je asi ta nejrozsáhlejší – obsahuje 3 podfáze a lze je nazvat přípravnou, realizační a ukončovací fází.

- Přípravná fáze – dochází ke stanovení cíle, sestavení projektového týmu a upřesnění stanovených výstupů
- Realizační fáze – celý projektový tým formuluje rozsah projektu, zde se využije již zmíněná WBS, vytvoří se plán, harmonogram a sleduje se průběh projektu
- Ukončovací fáze – zde se předávají výstupy, podepisují se akceptační protokoly a faktury. Dále se vypracuje závěrečná zpráva o projektu.

Poprojektová fáze slouží k analyzování celého projektu, jsou zde vyzdvihnuta pozitiva a negativa, které projekt přinesl. Tvoří se tzv. Lessons learned, tedy poučení získaná z minulých činností (Doležal a kol, 2012).



Obrázek 5: Životní cyklus projektu (zdroj: Svozilová, 2016)

## 3.2 Standardy projektového řízení

Standardy projektového řízení vnášejí do problematiky projektového managementu své poznatky, myšlenky a zkušenosti. Je tedy velice vhodné tyto standardy vnímat jako nápad, jak se s projektovým řízením v určité organizaci vyrovnat.

Mezi nejznámější světové standardy patří PMI PMBoK, IPMA ICB, PRINCE2 a do jisté míry i ISO 21 500. Všechny zmíněné standardy mají velice podobnou filozofii, vyskytují se v nich podobné metody a principy. To přináší výhodou právě v tom, že lidé, kteří v projektovém managementu pracují, vždy porozumí, co se jim daný standard snaží říci a dokáží tak efektivně pracovat. Avšak je důležité zmínit, že každý standard se liší místem vzniku i způsobem, jakým byl zpracován a vypracován.

Níže v kapitolách jsou rozpracovány jen některé z nich a porovnány mezi sebou.

### 3.2.1 Standard projektového řízení podle PMI PMBoK

PMBoK je celosvětově uznávaný standard a jeho vlastníkem je PMI (Project Management Institute). Project Management Institute (2013) popisuje PMBoK jako souhrn předpisů a rad, jak nejlépe řídit projektové řízení vyplývající z best practices.

PMBoK se opírá o 5 procesních skupin a 9 znalostních oblastí. Do procesních skupin spadá initiating, planning, executing, monitoring a controlling a nakonec closing.

Co se týče planningu, tak ten je mnohem lépe a podrobněji popsán, než je tomu v PRINCE2. Nicméně exectuing a monitoring + controlling není popsáno až tak dopodrobna.

Lze tedy vyčíst, že PMBoK není úplně metodikou, ale spíše návodem, a proto oproti PRINCE2 má bohatší oblast znalostí a také se pouští do větší šířky. To lze také vidět v oblasti řízení lidských zdrojů na projektu. V PMBoK je kladen poměrně velký důraz na profesionální působení, zodpovědnost a také etiku.

Projekt je dle standardu PMBoK definován jako: „*A project is a temporary endeavor undertaken to create a unique product, service, or result*“ (Project Management Institute, 2013). Volně tedy přeloženo: „*Projekt je dočasné úsilí, jehož cílem je vytvořit jedinečný produkt, službu nebo výsledek*“.

### **3.2.2 Metodika řízení projektů podle PRINCE2**

Metodika PRINCE2 je jednou z nejrozšířenějších metodik projektového řízení v Evropě. Byla založena několika společnostmi s několikaletými zkušenostmi právě z projektového řízení.

Podstata PRINCE2 závisí na 7 principech, 7 procesech a taktéž popisuje 7 témat. Je velice nutné porozumět všem principům, protože právě principy tvoří tzv. páteř celé metodiky. Aby projekt mohl být projektem PRINCE2, musí být dodržováno právě těchto 7 principů, pak je projekt v kontrolovaném prostředí. Velkou předností (oproti PMBoK) této metodiky je to, že ji lze tailorovat na aktuální projekt, znamená to tedy, že veškeré dokumenty, jež PRINCE2 nabízí, lze buďto využít, pozměnit nebo i dokonce úplně vynechat (AXELOS, 2017).

Další prioritou, kterou se PRINCE2 pyšní, je popis procesů i dokumentů. Projektový manažer má při výběru této metodiky poměrně volné ruce, nicméně musí projít klíčovými činnostmi, jež mu metodika stanovuje.

Dalším bodem odlišujícím se od PMBoK je ten, že podpora přizpůsobení metodiky je zahrnuta v příručce.

Großbritannien (2012) ve své publikaci *Managing successful projects with PRINCE2* popisuje projekt jako: „*A project is a temporary organization that is created for the purpose of delivering one or more business products according to an agreed Business Case*“. Volně přeloženo jako: „*Projekt je dočasná organizace, která je vytvořena za účelem dodání jednoho nebo více obchodních produktů podle dohodnutého obchodního případu*“.

### 3.2.3 Standard projektového řízení podle IPMA ICB

ICB je mezinárodní standard kompetencí projektového řízení a jejím vydavatelem je IPMA (International Project Management Association). Oproti metodice PRINCE2 a standardu PMBoK, které se zaměřují na podoby procesů v projektovém řízení, se ICB zaměřuje na 3 základní okruhy kompetencí. Jedná se o tzv. oko kompetencí, a to se skládá z technických, behaviorálních a kontextových kompetencí (Schoper et al., 2017).

Projekt je definován následovně: „*A project is a time and cost constrained operation to realize a set of defined deliverables (the scope to fulfill the project's objectives) up to quality standards and requirements*“ (Caupin and International Project Management Association, 2006). Volně přeloženo jako: „*Projekt je časově a nákladově omezená operace, jejímž cílem je realizovat soubor definovaných výstupů (rozsah pro splnění cílů projektu) v souladu s normami a požadavky na kvalitu*“.

<b>PRINCE2</b>	<b>PMBoK</b>	<b>ICB</b>
Metoda řízení projektů	Souhrn best practises pro řízení projektů	Kompetence projektových manažerů a členů týmu
Z jednotlivých oblastí není možné čerpat nezávisle na ostatních	Z každého tématu se dá čerpat nezávisle od druhých	---
Pokrývá všechny role řízení projektů	Zaměřen na projektové manažery	Zaměřen na projektové manažery a členy týmu
Neobsahuje mezilidské vztahy a soft skills	Soft skills	Odborné, behaviorální a kontextové kompetence
Odvolává se na techniky	Popisuje techniky řízení projektů	Lze využívat s dalšími globálními standardy

Tabulka 1: Porovnání metodik (zdroj: vlastní zpracování)

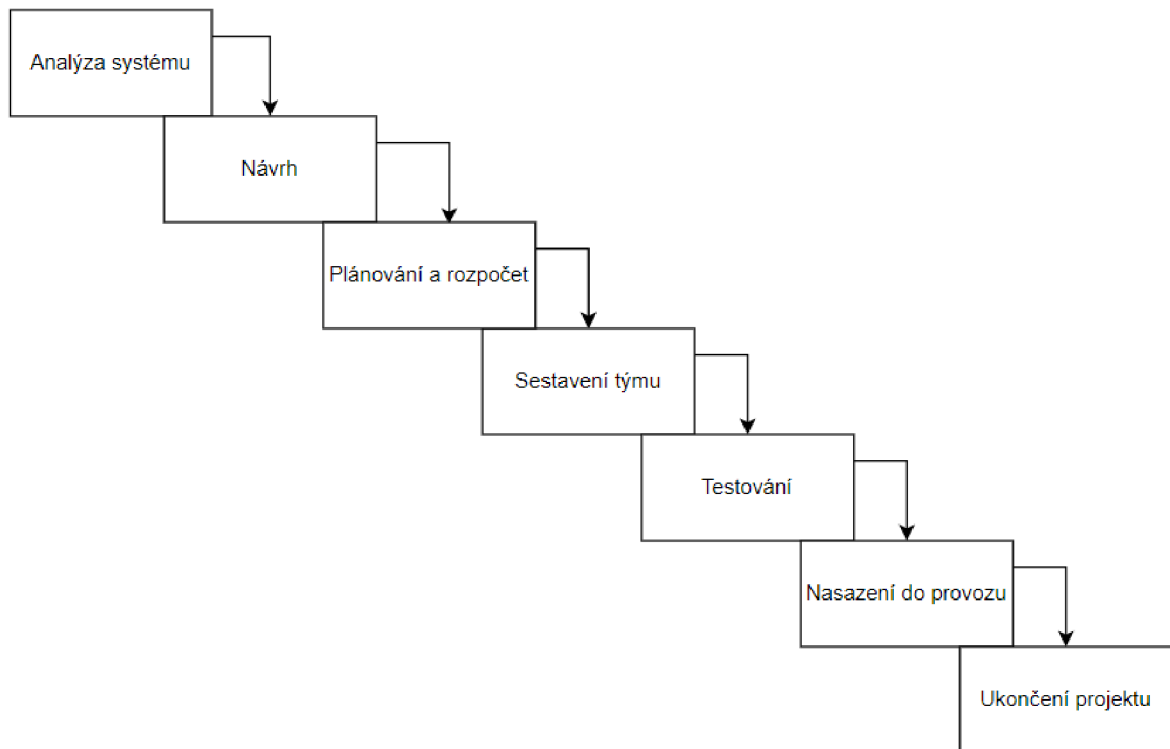
### 3.3 Vodopádový přístup řízení projektů

V projektovém řízení existují dva přístupy vedení projektů. Jedná se o vodopádový neboli „Waterfall“ a agilní přístup. Každý z nich je specifický a je vhodné jej použít jen na některé typy projektů. Nicméně není žádné psané a pevně dané pravidlo, kdy, jaký přístup, na jaký projekt použít.

Waterfall spadá do klasického přístupu řízení projektů a znamená to, že je zastoupený tradičními metodikami, které jsou orientované na procesy a detailně popisují metody,

nástroje, vstupy a výstupy. Je to přístup, který předpokládá předem jasně daný plán, resp. jak se bude postupovat od počáteční analýzy.

Prvotním a nejstarším modelem životního cyklu projektů je vodopádový model (waterfall). Jednotlivé fáze jsou seřazené za sebou a připomínají tak vodopád.



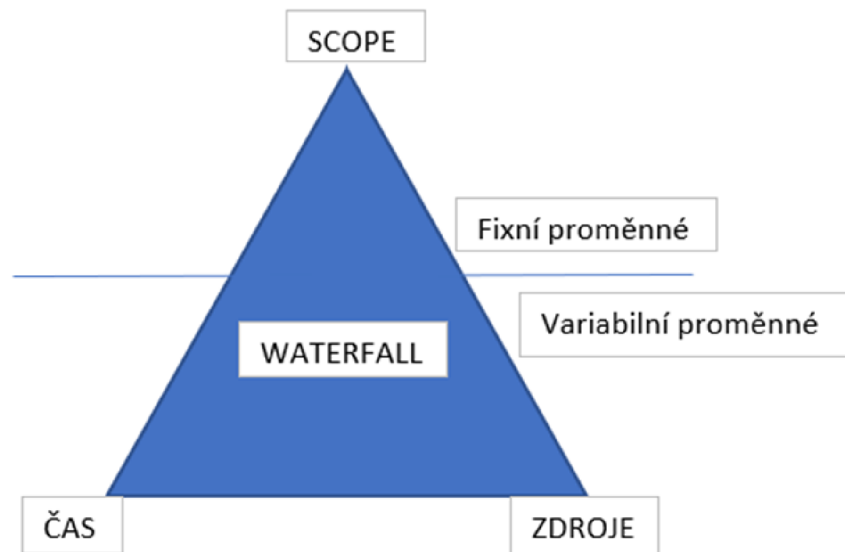
Obrázek 6: Model životního cyklu u waterfallu (zdroj: vlastní zpracování)

Název vychází z přirovnání posloupnosti jednotlivých fází projektu k protékání vody vodopádem. Tento přístup poprvé publikoval Winston W. Royce v roce 1970 a to pomocí sedmi základních fází. (Landry, 2011)

V praxi funguje vodopádový přístup tak, že klade velký důraz na termíny, časový rozvrh prací a plánování. Zaobírá se detailním naplánováním jednotlivých úkolů a jejich dodržování. Co se týká projektového týmu, tak mají minimální možnosti reagovat na jakékoliv změny během průběhu projektu, proto vodopádový přístup není vhodný např. pro vývoj nového software. Naopak je užitečný v projektech, které mají jasně stanovený cíl a definovaný postup a rozdělení prací, např. stavební projekty. Prvotní myšlenka vodopádového přístupu vychází z posloupnostního přístupu k jednotlivým fázím – je charakteristický tím, že vstoupit do další fáze je možné až tehdy, pokud je předchozí fáze celkově dokončena a uzavřena.

Waterfall představuje opak agilního přístupu – ten potřebuje velkou míru inovací a upřesňování v průběhu vývoje.

Vodopádový přístup je vhodný v případě, kdy je nezbytně nutné dodržet termíny a rozpočet. Značná část byrokracie je tedy vyvážena tím, že přístup jasně definuje role, kompetence a částečně i časový harmonogram (Myslín, 2016).



Obrázek 7: Vodopádový trojimperativ (zdroj: vlastní zpracování)

Z obrázku 7 lze vyčíst, že je vodopádový trojimperativ rozdělen na fixní a variabilní proměnné. Je to z toho důvodu, že ve vodopádovém přístupu je pevně daná funkcionality a měnit se mohou jen zdroje a čas, tzn. cíl projektu musí být takový, jaký se nastavil na začátku projektu, ale náklady a časové rozvržení jednotlivých úkolů se může měnit.

### 3.4 Agilní přístup řízení projektů

Vedle vodopádového stylu řízení projektů existuje ještě jeden styl, jak projekty úspěšně a efektivně řídit. Jedná se o agilní přístup, též známý jako „Agile“. Tento typ přístupu vznikl na základě neúspěšných IT projektů, ke kterým se musí přistupovat jinak než klasickým vodopádovým stylem. Agile v překladu znamená hbitý či mrštný, a proto je právě tento přístup řízení projektů používán zejména na ty, které jsou hodně dynamické a těžko předvídatelné, tzn. že obsahují nejasné požadavky, které se v průběhu projektu budou s největší pravděpodobností měnit.

Součástí této kapitoly bude nastíněn vznik a definice agile a agilní manifest. Dále budou zmíněny i vybrané agilní metody, které jsou často aplikované na řízení projektů.

#### 3.4.1 Vznik a definice

Tzv. odlehčené metodiky byly vyvinuty až v polovině devadesátých let, a to z důvodu, že vodopádový model přestával vyhovovat rychle se měnícímu světu. Jak již bylo



popsáno výše, tak vodopádový přístup není schopný rychle reagovat na změnu a potřebuje mít již na začátku fixně nadefinované spolupráce. Nicméně s rozvojem umělé inteligence a zejména počítačů, končilo pouze 13 % projektů hodnocených jako úspěšných z hlediska ceny, času a smluvního cíle. Proto bylo nezbytné rychle reagovat a začaly se tak objevovat odlehčené (agile) metodiky. Tyto metody se dle jejich tvůrců vracejí zpět k vývojovým praktikám z počátků vývoje softwaru. Jedná se o metody jako jsou Scrum, Kanban, Lean, Extrémní programování či metody z rodiny Crystal.

Agilní metody jsou zejména využívány pro projekty, které jsou zaměřeny na vývoj software – jsou dynamické a požadavky se mění každým okamžikem.

Definice agilního přístupu dle Václava Kadlece (2004) zahrnují primární myšlenku agilních přístupů:

- *„velmi rychlá reakce na nové hrozby a změny na trhu při intenzivním zapojení zákazníka,*
- *využití iterativních, inkrementálních a evolučních dodávek za účelem konvergence k optimálnímu řešení, které bude dodáno zákazníkovi,*
- *správné nastavení velikosti týmů, dokumentací a just-in-time procesů pro maximalizaci business hodnoty,*
- *vytvoření zisku v turbulentním globálním prostředí na základě schopnosti reagovat a vytvořit změnu,*
- *schopnost rychle měnit priority při využívání zdrojů při změnovém požadavku, změny na technologii a znalosti“.*

PMBOK formuluje agilní přístupy řízení projektu, jako adaptivní životní cyklus, který očekává, umožňuje a vyžaduje změny. Taktéž vyžaduje úzkou a dlouhotrvající spolupráci se stakeholdery (zainteresovaná strana). Adaptivní životní cykly jsou inkrementální a iterativní. Iterace jsou velmi krátké, obvykle od 2 do 4 týdnů, jsou fixované v čase a zdrojích (Svozilová, 2011).

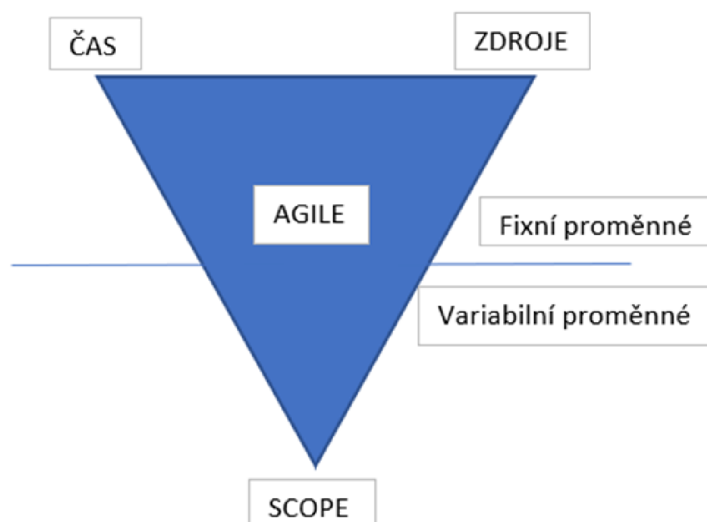
Další definice popisuje dodávku projektu agilním způsobem řízení projektů. Znamená to, že se projekt dodává po menších, samostatně funkčních částech neboli iteracích, a to má velkou výhodu v tom, že je umožněna okamžitá kontrola a předchází se tak zbytečným nesrovnalostem. Všechny iterace dohromady tvoří funkční celek a každá z nich je jinak dlouhá (Highsmith, 2010).

Z hlediska dodavatele vyvstává mnoho důvodů, proč zvolit pro řízení projektu agilní přístup. Hlavními důvody dle Šochové a Kunce (2014) jsou:

- agile pomáhá budovat tým, který je motivovaný, zodpovědný a organizovaný
- úspěch je jasně definovaný
- očekávání, která jsou sdílena mezi všemi podílejícími se osobami, jsou předem známá
- agilním typ umožňuje zaměření týmu na dodávání měřitelných výsledků
- zákazníci komunikují přímo s týmem
- tým cítí více zodpovědnosti za dosažení cíle
- při agilním typu řízení dochází k efektivnějšímu řízení lidských zdrojů
- dochází k úsporám času a nákladů.

Šochová a Kunce (2014) zmiňují výhodný pohled ze strany zákazníky na využití agile, jedná se například o:

- existence produktového vlastníka
- okamžitá a průběžná zpětná reakce
- vysoká projektová viditelnost
- výhodnější sladění IT s byznysem
- efektivnější snížení nadbytečného úsilí
- více adaptivní Agile Change Management
- časté a inkrementální dodávky.



Obrázek 8: Agilní trojimperativ (zdroj: vlastní zpracování)

Z obrázku 8 lze vyčíst, že jde o pravý opak k vodopádovému přístupu. Čas a zdroje by se měnit neměly, nicméně scope je tak variabilní, jak moc si zákazník bude přát během realizace projektu.

### 3.4.2 Agilní manifest

Struktura, jednotný přístup, efektivní vývoj softwaru-to vše vedlo k sepsání základních principů agilního přístupu.

Na základě setkání oborových kapacit v Utahu vznikl nápad na sepsání Manifesta agilního vývoje software. Mezi odborníky patřili např. Alistair Cockburn, Mike Beedle a Jim Highsmith a každý z nich představoval zástupce jedné metodiky. Současně byla založena aliance pro agilní vývoj s názvem Agile Alliance. Co se týče samotného manifesta, tak je popsáno 12 základních principů agilního vývoje software, a právě agilní metodiky by se těmito principy měly při řízení projektů řídit (Campbell, 2020).

Principy stojící za agilním manifestem:

- Nejvyšší prioritou je uspokojit zákazníka průběžnými a rychlými dodávkami kvalitního software
- Změnové požadavky jsou vítány, dokonce i v průběhu vývoje. Agilní procesy je zpracují tak, aby zákazníkovi přinášely konkurenční výhody
- Dodávejte fungující software často, v intervalech týdnů až měsíců. Upřednostňujte kratší intervaly dodání
- Lidé z businessu a vývojáři musí spolupracovat každý den během celého projektu
- Pro práci na projektu vybírejte motivované jedince. Dejte jim prostředí a podporu, kterou potřebují, a důvěřujte jim, že práci dokončí
- Nejúčinnější metoda sdílení informací vývojářskému týmu (i uvnitř tohoto týmu) je osobní setkání
- Fungující software je hlavním měřítkem postupu vývoje
- Agilní procesy podporují udržitelný vývoj. Sponzoři, vývojáři i uživatelé by měli být schopni dodržovat stálý výkon, dokud je třeba
- Průběžná pozornost věnovaná technické dokonalosti a dobrému návrhu posiluje agilní přístup
- Základem je jednoduchost – umění co nejvíce práce vůbec nedělat
- Nejlepší architektury, požadavky a návrhy vznikají v týmech, které se samy organizují

- Tým v pravidelných intervalech vyhodnocuje svou práci a upravuje své postupy tak, aby byl co nejefektivnější (Apke, 2015).

Nicméně ze základních principů dle Šochové a Kunce (2014) vyvstaly 4 hlavní hodnoty neboli myšlenky, které popisují, co znamená být agilní:

- Oceňování jednotlivců a jejich interakcí více než nástrojů a procesů – jednotný spolupracující tým má vyšší pravděpodobnost na úspěch, než jakýkoliv definovaný proces či nástroj. Nicméně procesy a nástroje mají pomáhat těmto týmům dosahovat cíle.
- Kladení většího důrazu na funkční software než na podrobnou dokumentaci – jakákoliv dokumentace by neměla převládat nad dodanou přidanou hodnotou produktu (projektu). Manifest sám doporučuje, aby všechny informace, které nejsou interní, byly dokumentované přímo v kódu a ty interní mají být co nejstručnější.
- Upřednostňovat spolupráci se zákazníkem před vyjednáváním o smlouvě – smlouva se zákazníkem patří k těm důležitým a nepostradatelným věcem, nicméně stává se, že zákazník neví, co přesně chce, a proto je dost často formulace smlouvy nepřesná. Proto je tedy důležitá neustálá komunikace se zákazníkem a nehledět jen na to, co je podepsáno a odsouhlaseno ve smlouvě.
- Reakce na změnu by měla být prioritou před dodržováním stanovených plánů – ačkoliv jsou plány důležitou součástí jakéhokoliv projektu, tak je velice důležité se umět přizpůsobovat změně, a to jak ze strany zákazníka, který si pokaždé může domyslet něco dalšího, tak i ze strany dodavatele, kdy není vhodné koukat jen na plány, ale přizpůsobovat se věcem za chodu.

### **3.4.3 Agilní metody řízení projektů**

Agilní metody jsou v současné době velice rozšířené a hojně využívané a je to z toho důvodu, že je agilní přístup bezpečnější jak pro zákazníka, tak pro realizační (agilní) tým. Nicméně existují i nějaké nevýhody agile přístupu v tom, že si zákazník musí být vědom faktu, že bude po celou dobu projektu zapojen do jeho realizace a taktéž se od něj očekává alespoň základní technické a metodické znalosti.

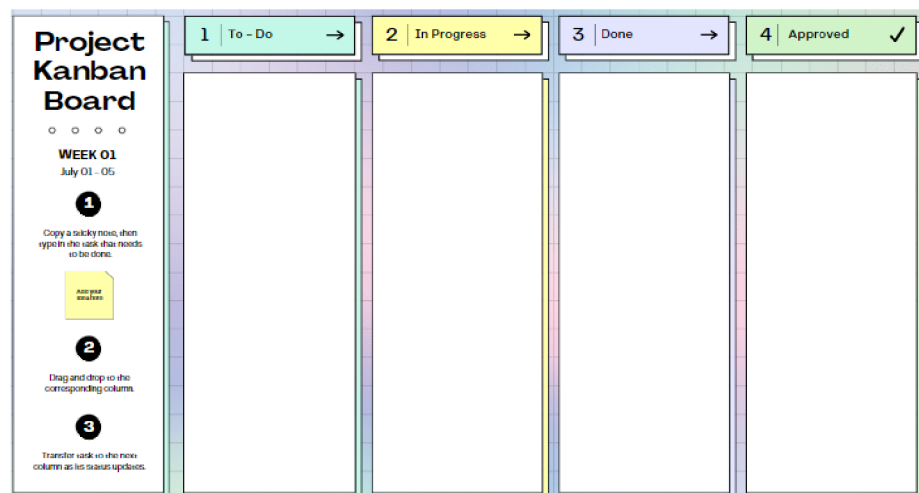
V této kapitole budou představeny agilní metody jako je Kanban, Lean development, Extrémní programování, Crystal metodiky a SCRUM. Existují ještě další metody, ale ty zmíněny nebudou.

### 3.4.3.1 Kanban

Metoda Kanban byla vyvinuta společností Toyota a počátek vývoje se datuje do období 20. století. Metoda byla součástí jejího produkčního systému a sloužila k naplnění výrobního přístupu JIT (= just in time). Jde tedy o metodu řízení toku informací a materiálu, jejímž cílem je uspokojení zákazníka dodáním včas toho pravého produktu, ve správném množství a také v požadované kvalitě. Samotný Kanban bere inspiraci ze supermarketů, kde jsou police doplňovány dalším zbožím až v případě, že to předešlé zboží už nebude.

Pojem Kanban lze také přeložit jako tabule a je tak využíván pro vizualizaci postupu dílčích činností pomocí lístků a tabule. Lístky představují požadované funkcionality, které jsou rozloženy do co nejmenších kusů a následně jsou tyto lístky během řešení posouvány po tabuli, dle stádia, ve kterém se nachází.

*„Kanban je vizuální systém správy úkolů, který se používá k zefektivnění správy projektů, projektových úkolů, ale i operativních, strategických, či brainstormingových plánů. Slovo Kanban pochází z japonštiny v doslovném překladu znamená něco jako "karty, které jsou vidět" (Co je to Kanban, 2022).*



Obrázek 9: Příklad Kanbanová tabule (zdroj: Canva.com, 2023)

Na obrázku 9 je obecně nastíněno, jak taková kanbanová tabule může vypadat. Nicméně mezi základní stádia by mělo patřit:

- Backlog – zde jsou všechny lístky, ze kterého jsou vybírány a následně posouvány dál;

- In progress – rozpracované funkcionality;
- To test – vývoj funkcionality je dokončen a následně je testována, aby správně fungovala;
- Done – hotová a otestovaná funkcionality, může být předána zákazníkovi (Co je to Kanban, 2022).

Aby ale Kanban mohl úspěšně fungovat, musí být dodržována následující pravidla:

- *„Každý proces vydává svým dodavatelům požadavky (kanban), když spotřebovává své zásoby*
- *Každý proces produkuje podle množství a posloupnosti přichozích požadavků*
- *Žádné položky se nevyrábějí ani nepřpravují bez žádosti*
- *K žádosti je vždy připojen požadavek spojený s položkou*
- *Procesy nesmí odesílat chybné položky, aby se zajistilo, že hotové výrobky budou bezchybné*
- *Omezení počtu čekajících žádostí činí proces citlivějším a odhaluje neefektivnost“* (Kanban, 2022).

Výčet možných kroků pro praktické využití metodiky Kanban:

- Definice procesu: Je třeba definovat jasný a detailní proces pro vývoj produktu, který bude využívat metodiku Kanban. Toto by mělo zahrnovat definici fází vývoje, povinností týmu a používaných nástrojů. A v neposlední řadě i cíl projektu.
- Vytvoření Kanban tabule: Tvorba fyzické nebo digitální Kanban tabule, která bude sloužit jako hlavní nástroj pro řízení projektu. Tato tabule by měla být uspořádána do sloupců odpovídajících jednotlivým fázím vývoje. Nejčastěji “TO DO,“ IN PROGRESS“,“DONE“.
- Přiřazení úkolů: Přiřadit jednotlivé úkoly k jednotlivým sloupcům na Kanban tabuli. Tyto úkoly by měly být popsány jasně a jednoznačně, aby bylo možné je snadno identifikovat a sledovat.
- Sledování a aktualizace úkolů: Sledování a aktualizování stavu jednotlivých úkolů na Kanban tabuli pro sledování postupu projektu a možná identifikace potenciálních problémů.
- Implementace Work-in-Progress (WIP) limity: Nastavit Work-in-Progress (WIP) limity pro jednotlivé sloupce na Kanban tabuli, aby bylo možné omezit počet

úkolů, které jsou současně v procesu vývoje. Toto umožní týmu se soustředit na jednotlivé úkoly a zajistit jejich co nejrychlejší dokončení.

- Zlepšování procesu: Neustálé sledování a zlepšování procesů vývoje produktu, aby bylo možné zajistit co nejlepší výkon a efektivitu. Toto může zahrnovat úpravy WIP limitů, přiřazení nových úkolů nebo změny v postupech pro vývoj produktu.
- Vyhodnocování výkon: Pravidelné vyhodnocování výkonu projektového týmu.

Po vykonaných krocích by měl být projekt řízený metodikou Kanban efektivnější a mělo by být možné snadno sledovat a vyhodnocovat postup projektu. Je důležité pečlivě sledovat a upravovat proces tak, aby vyhovoval potřebám týmu a projektu a umožňoval dosahovat co nejlepších výsledků.

### 3.4.3.2 Lean development

Lean development klade velký důraz na optimalizaci efektivity a minimalizaci plýtvání v procesu vývoje softwaru. Tento přístup má kořeny v hnutí tzv. štíhlé výroby v 80. letech 20. století. V současné době je však považován za integrální součást metodiky agilního vývoje softwaru.

Hlavní cíl metodiky Lean je to, že se snaží co nejvíce eliminovat plýtvání bez ztráty kvality. Znamená to tedy, že se metoda snaží, aby za chyby neplatil zákazník, když je způsobuje firma. Metodiku lze využít na všech úrovních procesu, kde je uplatňována efektivita a kde se nakládá s tzv. odpady. Mezi tyto úrovně procesu patří jak individuální úroveň, tak i na odděleních, v celé organizaci a také i mezi zákazníky a dodavateli.

Poppendieck (2010) popisuje Lean přístup, který je založen na sedmi přístupech spočívající v:

- Eliminaci odpadu – po každé iteraci musí projektoví manažeři definovat úzká místa, plýtvání a vypracují plán na jeho odstranění. Tzn., vše, co se v danou chvíli nevytváří pro zákazníka je odpad a ten musí být eliminován. Jedná se třeba o:
  - nepotřebný kód či softwarové funkce;
  - více úkolů, než lze dokončit v protokolu úkolů;
  - byrokratické procesy či problémy s kvalitou.
- Stavět v kvalitě – k zajištění kvality se používají všemožné postupy, např. párové programování či vývoj řízený testováním.

- Zesilovat učení – závěrem tohoto přístupu je zefektivnění (zrychlení a zjednodušení), jak šířit informace mezi zákazníkem a vývojovým týmem.
- Rozhodovat se co nejpозději – vývoj softwaru je propojen s poměrně velkou nejistotou, co se týče požadavků od zákazníka, prostředí a dalších faktorů. Tím pádem rozhodnutí, která jsou vytvořena co nejpозději, tak vlastně snižují míru nejistoty a rizika.
- Doručovat co nejrychleji – čím dříve se produkt dodá zákazníkovi, tím dříve vývojový tým dostane zpětnou vazbu a případné nesrovnalosti může vyřešit v další iteraci.
- Respektovat lidi (tým) – vývojový tým musí mít respekt mezi sebou, tím bude předurčena vynikající spolupráce. Lean development podporuje zdravé konflikty, proaktivní komunikaci a také neustálou zpětnou vazbu.
- Soustředit se na celkový výsledek – celý tým zkoumá proces od začátku do konce, aby byly hodnoty Lean co nejúčinnější. Tzn. je zapotřebí, aby se uvažovalo ve velkém, začalo se u malých drobných věcí, konalo se rychle a zároveň se analyzovaly chyby a vývojáři se z nich poučovali.

Výčet možných kroků pro praktické využití metodiky Lean development:

- Identifikace zdrojů: Analyzovat všechny zúčastněné zdroje na projektu – lidské, technické, finanční.
- Identifikace zdrojů plýtvání: Analyzovat, kde se v průběhu vývoje produktu vyskytují zdroje ztrát, jako například nadbytečné kroky, duplicitní činnosti a nedostatečný přístup k informacím.
- Zlepšení procesu: Identifikované zdroje ztrát využijte k implementaci změn, které vedou k zefektivnění procesu.
- Definice cílů: Stanovení si cíle pro projekt, a to jak krátkodobé, tak dlouhodobé. Tyto cíle pomohou určit směr a zaměření pro zlepšování procesu.
- Sledování výsledků: Sledování a měření výsledků zlepšovacích opatření. Pomocí těchto měření lze zjistit, zda je projekt na správné cestě a zda se dosahuje stanovených cílů.
- Kontinuální zlepšování: Neustále hledat nové způsoby, jak zlepšit proces.

Tento proces by měl být opakován neustále, aby bylo možné dosáhnout trvalého zlepšení a vedlo to tak k efektivním výrobním procesům.



### 3.4.3.3 Extrémní programování (XP)

Tuto metodu začal vyvíjet Kent Beck v 90. letech minulého století pro softwarový vývoj. Slovo extrémní vyjadřuje to, že tato metoda převádí všechny běžné postupy a principy do extrému. Sám Kent Beck to popsal jako: „*Je to jako, když máte ovládací panel s otočným knoflíkem, od jedné do deseti, pro každou osvědčenou metodu. Extrémní programování nastává, pokud všechny knoflíky otočíte na desítku*“ (Beck, Andres, 2005).

Metoda je vhodná pro menší vývojové týmy, které se skládají ze 2 až 10 lidí a staví na dvou předpokladech. Programátorům či vývojářům je umožněno pracovat jen na úkolech, které jsou pro projekt důležité a nebudou nuceni dělat rozhodnutí, na které nemají kvalifikaci a tím pádem se mohou maximálně soustředit na projekt. XP přinese vlastníkům produktu a projektovým manažerům vysokou efektivitu skrze sledování pokroků a provádění jakýchkoliv změn bez zbytečného zvyšování nákladů (Beck, Andres, 2005).

Extrémní programování uznává základní čtyři hodnoty podle Landry (2011):

- Komunikace – kvalitní komunikace je základním pilířem. Je nezbytné, aby tuto činnost měl na starosti týmový kouč a snažil se tak udržovat vývojový tým.
- Jednoduchost – to zaručuje úspěch ve vývoji. Tým nedělá nic navíc, píše jen základní kód, dokumentaci a také jen základní procesy. Výhoda je v tom, že sám zákazník neví, co by měl vše kód obsahovat, a proto se vývojáři nemusí zabývat s něčím navíc.
- Zpětná vazba – projevuje se hned na několika místech v projektu. Nejdříve programátoři stále tvoří jednotkové testy, takže v poměrně krátkých časových intervalech dokáží ověřit, zda vše funguje a že se nic během poslední změny kódu nerozhodilo. Další je zpětná vazba pro zákazníky. Ti provádí akceptační testy a zjišťují, zda jim to takto vyhovuje či nikoliv.
- Odvaha – tím je myšleno to, že pokud programátor narazí na problém v kódu, který dělal klidně i několik hodin, musí být schopný uznat, že je to celé k ničemu a klidně ho i zahodit. Ač dojde ke zdržení, samotného programátora to posune dále a do budoucna se podobným chybám vyvaruje.

Vedle hodnot existují ještě principy XP. Hodnoty jsou tedy takové, ke kterým se musí během projektu dojít a udržovat je. Nicméně principy představují to, co se musí dodržovat, aby se mohlo jednat o metodu XP.

Principy extrémního programování dle Buchalceové (2005):

- Plánovací hra – tvorba plánu dle představy zákazníka, na které se podílí všichni členové týmu
- Malé verze – co nejčastější uvolňování nových verzí vede k uspokojení zákazníka
- Metafora – pomocí metafor se popíše systém, jak bude komplexně vypadat
- Jednoduchý návrh – nejlepší je maximální jednoduchost systému
- Testování – během vývoje jsou vytvořeny automatické testy
- RefaktORIZACE – vývojáři restrukturalizují systém, aby odstranili duplicitu, zjednodušili systém nebo přidali větší flexibilitu, to vše, aniž by se změnil systém
- Párové programování – veškeré programování je prováděno ve dvou, tzn. dva programátoři na jednom počítači
- Společné vlastnictví – všichni vývojáři znají kódy ostatních a mohou je kdykoliv přepsat či opravit
- Nepřetržitá integrace – kdykoliv se dokončí další úkol, je systém znovu stavěn a integrován
- 40hodinový pracovní týden – nejsou uznávány přesčasy, ba jsou skoro zakázány, a to z důvodu zbytečného přepínání vývojářů
- Zákazník na pracovišti – zapojení zákazníka do vývoje, aby mohl poskytovat zpětnou vazbu.

Výčet možných kroků pro praktické využití metodiky XP:

- Týmový přístup: Sestavení týmu, který bude odpovědný za realizaci projektu. V týmu by měli být zastoupeni vývojáři, testéři, analytici a uživatelé.
- Komunikace: Zajistit, aby byla komunikace mezi členy týmu co nejlepší. XP vyžaduje týmovou spolupráci a otevřenou komunikaci.
- Příprava plánu: Připravit plán projektu s ohledem na požadavky zákazníka a technické specifikace. Tento plán by měl být přizpůsobený a upravován během průběhu projektu.
- Krátké cykly vývoje: XP vyžaduje krátké cykly vývoje, aby bylo možné co nejdříve zjistit a opravit chyby. Tyto cykly by měly trvat jen několik dní až týdnů.

- **Automatické testování:** Automatické testování je klíčovým prvkem XP. Tyto testy by měly být spouštěny automaticky po každé změně kódu, aby bylo možné zajistit kvalitu a integritu software.
- **Reflektování:** XP vyžaduje, aby tým pravidelně reflektoval a hodnotil svůj proces. Tyto hodnocení by měla být použita k identifikaci možností zlepšení a ke kontinuálnímu zlepšování procesu.
- **Adaptace:** Vyžaduje se, aby byl tým schopen rychle reagovat na změny a přizpůsobit svůj proces těmto změnám. Tyto změny mohou být vyvolané změnami v požadavcích zákazníka nebo změnami v technickém prostředí. Tyto změny by měly být brzy identifikovány a integrovány do procesu vývoje.
- **Integrace:** XP vyžaduje pravidelnou integraci kódu, aby bylo možné zajistit, že se jednotlivé části software spolu správně interagují. Tyto integrace by měly být prováděny co nejčastěji, aby bylo možné co nejdříve odhalit a opravit chyby.
- **Práce s požadavky:** Potřeba jasných a srozumitelných požadavků. Tyto požadavky by měly být neustále hodnoceny a upravovány, aby odpovídaly aktuálním potřebám zákazníka.
- **Spolupráce se zákazníkem:** XP vyžaduje blízkou spolupráci se zákazníkem, aby bylo možné zajistit, že se software vyvíjí v souladu s jeho požadavky. Tyto spolupráce by měly být pravidelné a přímé, aby bylo možné co nejdříve reagovat na změny.

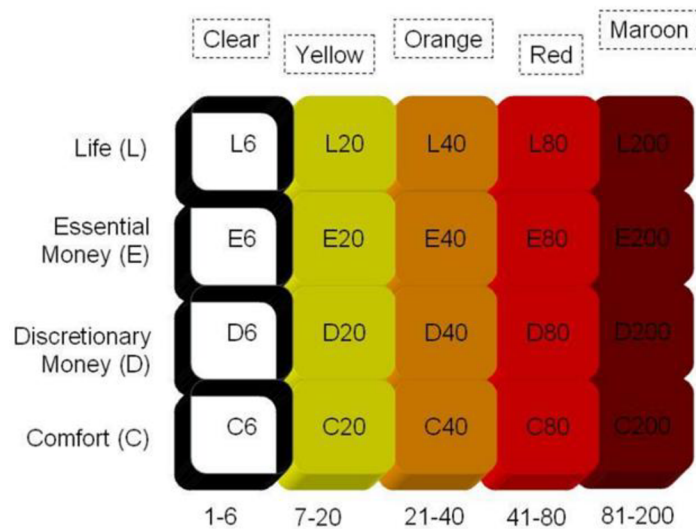
#### 3.4.3.4 Crystal metodiky

Crystal není jméno jediné metodiky, ale slouží jako označení pro celou rodinu metodik, které vymyslel Alistair Cockburn. Každá z metodik je pojmenovaná podle barvy. Navzájem se od sebe tyto metodiky odlišují svou vhodností pro různě velké týmy a také pro různě složité projekty. Čím je barva metodiky tmavší, tím je metodika mohutnější a je tak vhodná pro velké a kritické projekty. Pomocí třech vlastností projektu se následně vybere vhodná metodika. Mezi tyto vlastnosti patří velikost vývojového týmu, kritičnost projektu a také to, jakou má samotný projekt prioritu.

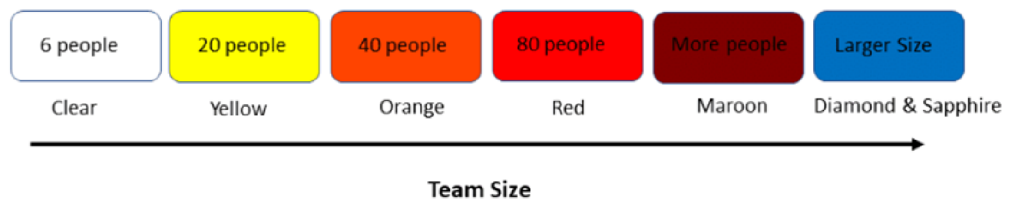
V obrázku č. 10 lze vidět i písmena C, D, E a L. Tyto písmena představují kritičnost projektu, tedy to, jak velký dopad to bude mít na selhání systému. Kdy podle Kadlec (2004):

- **C...** ztráta komfortu (nejmenší dopad), systém lze nadále používat, ale uživatele to stojí větší úsilí.

- D...malá ztráta peněz (větší dopad), zde již firma ztrácí finance.
- E...velká ztráta peněz (větší dopad), u tohoto však firma ztrácí velkou část financí.
- L...ztráta lidských životů je nejhorší možný dopad.



Obrázek 10: Barevné rozlišení Crystal metodik (zdroj: Singh, 2019)



Obrázek 11: Další pohled na rozdělení Crystal metodik (zdroj: Singh, 2019)

Jako tomu bylo i u předešlých metod, tak se Crystal metodiky skládají z principů, postupů, rolí a samotného životního cyklu.

Crystal metodiky se dle Singh (2019) skládají ze 7 principů a jejich stručný výčet je následující:

- Časté dodávky – časté uvolňování iterací softwaru
- Reflektivní vylepšování – je založeno na přerušení práce a nalezení lepšího řešení procesů
- Blízká nebo osmotická komunikace – všichni vývojáři programují v jedné místnosti
- Osobní bezpečnost – důvěra a svobodné vyjadřování v týmu
- Zaměření – to se týká jak zaměření se na konkrétní jednu úlohu, tak i zároveň na směr, kterým se projekt ubírá
- Snadný přístup k expertům – specialista je v dosahu vývojářů a podává jim pomocnou ruku

- Technické prostředí s automatizovanými testy, managementem konfigurace a časté integrace – časté testování a integrace pro včasné odhalení nedostatků.

Vedle principů je na místě uvést i postupy, kterými se Crystal metodiky řídí. Pro definování postupu je vybrána metodika Crystal Clear z důvodu, že využívá pojem strategie či techniky a snaží se vyhýbat tzv. praktikám. Je to z důvodu, že metoda Crystal Clear dává naprostou svobodu vývojovému týmu a spíše slouží jako doporučení, jak se k danému projektu postavit. Pak už záleží na samotném vývojovém týmu, zda zůstanou u naučených postupech, popřípadě použijí něco jiného ba dokonce zkombinují několik metod dohromady.

Pro usnadnění použití Crystal Clear metodiky jsou však doporučeny postupy/strategie, kterých se může vývojový tým držet. Spadají do nich např dle Cockburn (2005):

- Průzkum – má projekt smysl?
- Motivující úspěch – nejdříve se dělají jednodušší úkoly a tím se udržuje motivovanost týmu.
- Běžící kostra – co nejdříve vytvořit kostru fungujícího systému.
- Inkrementální architektura – nutnost zdokonalování běžící kostry.
- Informační tabule – např. Kanban board pro vizualizaci.
- Nastavení metodiky – na začátku projektu se nastaví parametry metodiky.
- Věštění doby trvání – využití expertních odhadů na celkovou dobu trvání.
- Denní schůzky – během pár minut se projde u každého člena, co se dělo, děje a bude dít.
- Přírůstkový graf – technika pro přehlednost a pokrok v projektu.
- a další.

Projekt, jež je řízen metodikou crystal clear, je tvořen skupinou cyklů, které jsou do sebe vnořeny a mají různé délky. Tyto cykly se skládají z vývoje, iterace, dodávky, a nakonec celého projektu.

Projekt ale začíná etapou mapování, kde probíhá sestavení týmu a rozdělení rolí a na to pak následuje průzkum týkající se metodiky. Závěrem tohoto cyklu je vytvořen prvotní plán projektu. Další etapa se věnuje dodávce, která se skládá z jedné či několika iterací. Výsledkem této etapy by měl být dodán některým uživatelům nový inkrement k otestování funkcionalit. Během dodávky se může uskutečnit i zdokonalující workshop. Co se týče délky iterací, tak může být délky od jednoho týdne až po 2 měsíce a délka integračního cyklu závisí

na schopnostech celého týmu. Poslední z cyklů se věnuje vývoji, který je součástí integračního cyklu. V tomto cyklu se ale většinou projednává návrh, programování, testování a hledání chyb (Highsmith, 2010).

Výčet možných kroků pro praktické využití metodiky crystal clear:

- Sestavení týmu: Před zahájením vývoje software pomocí metodiky crystal clear je nutné sestavit tým, který bude pracovat na projektu.
- Mapování: Je nutné definovat jasné a srozumitelné požadavky na výsledek toho, co zákazník chce, aby bylo možné zajistit, že bude v souladu s potřebami zákazníka. Tyto požadavky by měly být neustále hodnoceny a upravovány, aby odpovídaly aktuálním potřebám.
- Sestavení plánu projektu: Následujícím krokem je sestavení plánu projektu, který bude obsahovat informace o tom, co se bude v průběhu projektu dít, kdy se to stane a kdo bude za danou část projektu zodpovědný.
- Zajištění komunikace: Crystal clear vyžaduje silný důraz na komunikaci, a to jak mezi členy týmu, tak i mezi zákazníkem a týmem.
- Vývoj inkrementu: Následujícím krokem je samotný vývoj inkrementu. Je nutné dbát na to, aby byl vyvíjen s co nejvyšší kvalitou a aby odpovídal požadavkům zákazníka.
- Testování funkcionalit: Po dokončení vývoje je nutné pečlivě otestovat funkcionality inkrementu, aby bylo možné odhalit a opravit chyby.
- Integrace inkrementů: Po úspěšném testování je nutné vše integrovat do stávajícího technického prostředí.
- Zajištění podpory: Po dokončení projektu je nutné zajistit, aby byla zákazníkovi poskytnuta dostatečná podpora.
- Hodnocení a zlepšování: Využití přírůstkového grafu či denních schůzek pro hodnocení postupu projektu.

#### 3.4.3.5 Metoda SCRUM

Jako dalším agilním typem řízení projektů je metoda SCRUM. Jde o iterativní a inkrementální způsob řízení vývoje software. Iterace je základem SCRUMu. Pro vývoj software je typická vysoká flexibilita a požadavky zákazníka se mohou v průběhu projektu měnit. Oproti waterfall přístupu zde není veden klasický plán jako celek, ale vše je

soustředěno na řízení jednotlivých iterací. Proto se tato metoda soustředí zejména na maximální schopnost realizačního týmu – ten se snaží dodat část s nejvyšší prioritou co nejrychleji a případně reagovat na nové požadavky.

SCRUM využívá empirický přístup, aby se přizpůsobil poměrně častým měnícím se požadavkům zákazníka. „Empirický přístup znamená pracovat způsobem založeným na faktech, zkušenostech a důkazech, a zejména pokrok je založen na pozorování reality, nikoli na fiktivních plánech založených na velkém množství počátečních požadavků“ (Singh, 2019).

Lze tím tedy chápat, že se vývojový tým má učit a také se snažit zlepšovat své minulé chyby a zkušenosti.

Metoda SCRUM dle Verheyen (2017) je založena na třech pilířích a to transparentnosti, kontrole a adaptaci:

1. **Transparentnost** spočívá v zadání a komunikaci. Znamená to, že všechny zainteresované osoby musí používat stejná názvosloví, stejný jazyk. Všichni musí vědět, co vidí, slyší. Transparentnost může být uskutečňována pomocí nástrojů, jimiž SCRUM disponuje. Jedná se např. o produktové backlogy, denní stand-upy, sprintové recenze či retrospektiva. Tyto nástroje se využívají pro přenos toku práce skrze cross-funkční tým.

2. **Kontrola neboli inspekce**, spočívá v pravidelných kontrolách SCRUM artefaktů vzhledem k cíli sprintu. Slouží to k tomu, aby se včas daly odhalit nevhodné odchylky a změny. Tyto kontroly by měly být pouze tak časté, aby nic nestálo v průběhu práce na celém projektu.

3. **Adaptace** je proces, který má zamezit stavu, který by mohl projekt ohrožit a stát se tak neakceptovatelným. Jde tedy o to, že se nevhodný stav musí adaptovat do správného stavu a k tomu právě slouží i již zmíněné kontroly.

#### 1.4.1.1.1 SCRUM tým

Tým SCRUMU se skládá jak z Product Ownera, SCRUM Mastera, tak z vývojového týmu. SCRUM týmy jsou tzv. sebe-řízené a znamená to, že je tým sebe-navržený a každý člen si zvolí kdo, jak a na čem bude pracovat. Tým nemá žádné vedení či řízení z vnějšku.

#### **SCRUM Master**

SCRUM master je role, která je zodpovědná za správné pochopení a přijetí metody SCRUM vývojovým týmem. Toho lze docílit jen tím, že se vývojový tým bude snažit dodržovat SCRUM teorii, praktiky a pravidla. Role SCRUM Mastera je mnohdy vnímána

jako role kouče, protože je týmu vždy nápomocný a nebrání jim v jednotlivých krocích. Jeho úkolem je směřovat tým hlavně k efektivní práci, spokojenosti a dosažení cílů. Měl by zejména pomáhat k agilní transformaci – poslouchat tým a snažit se jim navrhnout efektivnější řešení.

Stručné vytyčení nejdůležitějších bodů práce SCRUM Mastera:

- Opora vývojovému týmu, agilní transformace a vytvoření vhodného prostředí
- Redukce problémů a překážek, se kterými se tým často potýká
- Udržování dobrých vztahů v týmu

SCRUM Master má být součástí vývojového týmu a měl by se věnovat pouze své roli, která mu je určena z důvodu, aby nevznikla kolize v cílech a zájmech v případě, kdy by zastupovat i jinou roli, např. product ownera.

### **Product Owner**

*„Taktický člen týmu pro vývoj produktů se účastní každodenních schůzek Scrum a upřednostňuje nevyřízené položky. Kromě toho se od nich očekává, že zajistí, aby vývojáři pracovali efektivně a na správných položkách“ (Verheyen, 2017).*

Vlastník produktu (product owner) je role, která určuje produktovou vizi – co se bude vyvíjet v rámci produktu a zároveň určuje i priority. Je to role, která musí rozumět a být za jedno s vizí produktu, zákazníkem a komunikací. Musí být správně definován cíl a také porozumění požadavkům.

Hlavní činnosti product ownera:

- Definování úkolů v product backlogu a následně jejich uspořádání dle priorit
- Zajištění transparentnosti product backlogu pro všechny členy týmu

Úspěšnost product ownera i samotného projektu závisí na respektování rozhodnutí ze strany vývojového týmu a dodavatele služeb. Jeho hlavní úlohou je orientace na efektivitu výroby produktu a zejména to, zda je vyráběný produkt ten pravý pro zákazníka. Taktéž se stará o správu backlogu a na konci každé iterace má poslední slovo ve schvalování dodaných inkrementů. Je také osobou, co zodpovídá členům týmu otázky týkající se produktu – co, proč, jak (Singh, 2019).



## **Vývojový tým**

Vývojový tým je složen z profesionálů, kteří jsou zodpovědní za dodání přírůstku produktu na konci každého sprintu. Jak už bylo popsáno výše, SCRUM týmy jsou sebeorganizující a sebeřídící se a potažmo díky tomu je vývojový tým velice efektivní. Co se týče velikosti vývojového týmu, tak se většinou pohybuje od 3 do 9 lidí, tzn., že tým není moc velký a díky tomu funguje veškerá komunikace a rozdělení práce (Verheyen, 2017).

### **3.4.3.5.1 Artefakty SCRUMu**

Artefakty SCRUMu jsou v podstatě seznamy, kterými se sdílí informace napříč celým průběhem řízení projektu. Existují 4 hlavní SCRUM artefakty, díky kterým je práce na projektu usnadněna. Všechny tyto artefakty dodávají sprintu přidanou hodnotu. Jedná se o artefakty product backlog, sprint backlog, produktový přírůstek a burndown chart.

#### **Product backlog**

Singh (2019) popisuje produktový backlog, jako výčet pracovních položek (požadavky, funkce, vlastnosti, opravy chyb), které mají být v průběhu času dokončeny.

Každá jedna položka v seznamu má přiřazenou prioritu a odhad, jak dlouho by měla trvat. Tyto priority jsou přiřazovány product ownerem, který za ně nese zodpovědnost, nicméně odhady pracností jednotlivých úkolů provádí sám vývojový tým. V produktovém backlogu mají být vydefinovány vlastnosti a možnosti daného vyvíjeného produktu. Avšak tento seznam se v průběhu času neustále mění dle priorit, potřeb a náprav chyb.

Typická vlastnost product backlogu je ta, že s vývojem produktu narůstá a product owner ho musí pravidelně kontrolovat či revidovat. Stará se tedy o to, aby byly k dokončení vybrány nejdůležitější úkoly, následně rozepsány do detailu a tím pádem byly připraveny k přemístění do sprintu.

#### **Sprint backlog**

Do sprint backlogu (zadání iterace) jsou vybrány jednotlivé položky z product backlogu, které spadají do současného sprintu. Sprint backlog je nižší řád product backlogu a skládá se z úkolů, na kterých musí vývojový tým během sprintu pracovat. Každá položka musí být vyvinuta, otestována a také zdokumentována. Sprint backlog má podobu úkolové tabule (třeba jako je Kanban) a její obsah vybírá product owner (Singh, 2019).

Na konci každého sprintu by měly být ideálně všechny úkoly hotové, ale pokud tomu tak není, jsou úkoly vráceny zpět do product backlogu a budou vybrány pro další sprint na dodělení.

### **Produktový přírůstek**

Dalším artefaktem je produktový přírůstek. Jedná se o seznam všech položek z product backlogu, které již byly dokončeny během předešlých sprintů. *„Aby sledování tohoto přírůstku mohlo fungovat, je třeba mít jasno v tom, co znamená “dokončená položka”.* Tato definice se však může v průběhu času měnit tak, aby odpovídala agilním principům, tak jak je u vás hlídá Scrum master“ (Singh, 2019).

### **Burndown chart**

Burndown chart je graf, díky kterému lze vidět, jak vývojový tým rychle postupuje napříč produktovým backlogem. Taktéž lze vyčíst i to, jaký je vztah udělané práce oproti jejímu celkovému množství. Úkolem burndown chartu je monitoring postupu termínů a zajištění správného postupu projektu.

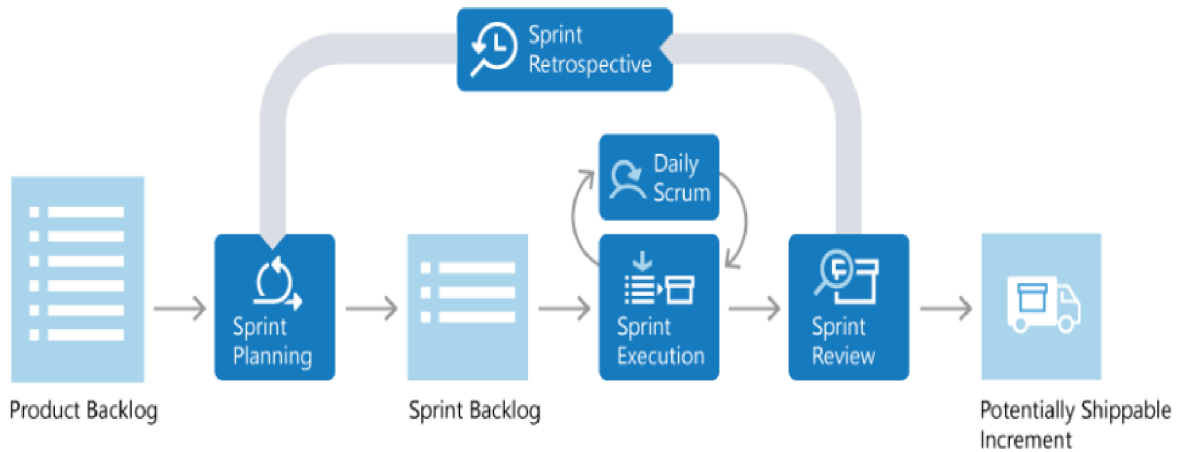
*„Tempo postupu při odbavování story-pointů z user – stories neboli velocity, vám pak kromě trackingu času současného projektu může sloužit i během plánování dalších projektů“ (Verheyen, 2017).*

#### 3.4.5.2 SCRUM aktivity

SCRUM aktivity byly vytvořeny za účelem dodržování pravidelnosti a minimalizování požadavků na porady, které nejsou ve SCRUM metodě formulovány. Každá z těchto aktivit má pevně daný časový úsek s maximální délkou trvání.

### **Sprint**

Definice sprintu je, že se jedná o jednu časově ohraničenou iteraci nepřetržitého vývojového cyklu. Během sprintu musí vývojový tým dokončit plánované množství práce a připravit se ke kontrole. Jednotlivé sprinty jsou většinou krátké, v rozmezí několika dní, a v žádném případě nejsou delší než 1 měsíc. Každý sprint začne hned poté, co se předešlý sprint ukončil.



Obrázek 12: Sprint (zdroj: LearnMicrosoft.com, 2023)

### **Sprint planning**

Sprint planning probíhá vždy na počátku sprintu a popisuje vše, co se během sprintu bude dělat a jakým způsobem. Doba plánování by měla být úměrná délce sprintu. Sprint planning vždy řídí SCRUM master.

*„Délka schůzky plánování sprintu je úměrná délce sprintu. Čtyřtýdenní sprint by měl být naplánován za nejvýše 8 hodin“ (Sprint. Ideal Modeling & Diagramming Tool for Agile Team Collaboration, 2022).*

### **Daily SCRUM**

Daily SCRUM jsou tzv. denní stand-upy, které trvají maximálně 15 minut a během této krátké doby musí být definován plán na dalších 24 hodin. Je to z toho důvodu, aby na správných věcech pracovali správní lidé ve správném čase.

Každá osoba ve vývojovém týmu má za úkol zodpovědět 3 jednoduché otázky:

*„Co jsem udělal včera, abych pomohl dosáhnout cíle sprintu?“*

*Co dnes udělám, abych dosáhl cíle sprintu?“*

*Co když něco brání nebo blokuje pokrok směrem k cíli sprintu?“ (Sprint. Ideal Modeling & Diagramming Tool for Agile Team Collaboration, 2022).*

### **Sprint review**

Sprint review je neformální schůzka, která probíhá na konci každého sprintu spoluprací SCRUM týmu a všech dalších zainteresovaných osob včetně product ownera. Během těchto schůzek se zkoumá přírůstek. Dále jsou zde prezentovány zpětné vazby,

výsledky a spolupráce. Závěrem vyhodnocení sprintu je pak revidovaný či pozměněný product backlog, který dále slouží jako podklad pro další sprint (Singh, 2019).

### **Sprint retrospective**

Sprint retrospective slouží k vyhodnocení práce SCRUM týmu. Retrospektiva je přibližně 3 hodiny dlouhá schůzka, kde se vytváří plán pro zlepšení týmu na další projekty. Slouží ke zhodnocení všech lidí a prací na jednotlivých úkolech. Na tyto schůzky dohlíží SCRUM Master a zároveň je jeho úkolem transparentnost těchto sezení – každý musí rozumět otázkám typu co, proč a jak. Dále dochází ke kontrole průběhu sprintu z hlediska procesů, nástrojů, ale také vztahů mezi jednotlivými členy týmu (Verheyen, 2017).

Výčet možných kroků pro praktické využití metodiky SCRUM:

- Sestavení týmu: Před zahájením vývoje software pomocí metodiky crystal clear je nutné sestavit tým, který bude pracovat na projektu.
- Product backlog: Výčet jednotlivých položek, které mají být v průběhu projektu dokončeny a dodány.
- Sprint planning: Během plánování sprintu se tým setkává a rozhoduje o tom, co bude během sprintu dělat. Výsledkem plánování sprintu je sprint backlog, který obsahuje seznam úkolů, které se mají během sprintu dokončit.
- Sprint: Sprint je obvykle dvoutýdenní časové období, během kterého tým pracuje na určité části projektu. Během sprintu musí vývojový tým dokončit plánované množství práce a připravit se ke kontrole.
- Daily Scrum: Daily Scrum je krátké (15minutové) setkání, které se koná každý den během sprintu. Cílem je sdílet informace o tom, co bylo dokončeno od posledního setkání, co se bude dělat dnes a jaké jsou případné problémy, které by mohly bránit dokončení úkolů.
- Sprint review: Po dokončení sprintu se tým setkává, aby prezentoval dokončenou práci a získal zpětnou vazbu od zákazníka nebo jiných zainteresovaných stran. Tým také hodnotí, co se podařilo dokončit během sprintu a co bylo obtížné.
- Retrospektiva sprintu: Retrospektiva sprintu je setkání týmu, kde se hodnotí, co se podařilo, co se nepodařilo, co bylo obtížné a jakým způsobem by se mohlo zlepšit. Tento krok je důležitý pro postupné zdokonalování procesu a dosahování lepších výsledků.

### 3.5 Porovnání vodopádového a agilního přístupu řízení projektů

V dřívějších kapitolách byly podrobně popsány metody, jak efektivně řídit projekty. Tato kapitola se již tedy nebude zabývat podrobným popisem metod waterfall a agile, ale bude sloužit k jejich porovnání.

Waterfall a agile jsou světoznámé metodiky projektového řízení, nicméně každá se hodí na jiný typ projektů. Největší rozdíl v těchto metodách tkví v tom, že waterfall je typický lineární systém práce, kde je vyžadováno dokončení jednoho úkolu před začátkem následujícího. Zatímco agile je založen na současném plnění všech možných fází projektu. A v tomto je ten největší rozdíl mezi metodami.

*„Metoda waterfall je lineární forma projektového řízení ideální pro projekty, kde je konečný výsledek jasně stanoven od začátku projektu. Očekávání od projektu a výstupy každé fáze jsou jasné a jsou nutné pro postup do další fáze.*

- *Přístup: ruce pryč; cíle a výsledky stanovené od začátku*
- *Pružnost: Nízká*
- *Vyžaduje: Dokončení výstupů pro postup do další fáze“ (Hoory, 2022).*

*„Agilní metoda byla vyvinuta jako reakce na tužší strukturu Waterfall. Ve výsledku je to mnohem plymulejší forma projektového řízení. Dokončení projektu vývoje softwaru může trvat roky a technologie se během této doby může výrazně změnit. Agile byla vyvinuta jako flexibilní metoda, která vítá začlenění změn směru i pozdě v procesu, stejně jako zohlednění zpětné vazby zúčastněných stran v průběhu procesu.*

- *Přístup: Častá interakce se zúčastněnými stranami*
- *Pružnost: Vysoká*
- *Požadujeme: Týmovou iniciativu a krátkodobé termíny (Hoory, 2022).*

Nejdůležitější rozdíly shrnuje tabulka 4, nejsou zde uvedeny všechny, ale jen ty nejvíce rozdílné a signifikantní pro danou metodu.

<b>AGILE</b>	<b>WATERFALL</b>
Životní cyklus projektu je rozdělen do sprintů	Životní cyklus projektu je rozdělen do jednotlivých fází
Inkrementální přístup	Sekvenční návrhový proces
Velice flexibilní	Neumožňuje flexibilitu, vše je pevně dáno
Provádět změny lze v jakékoliv fázi	Nelze měnit požadavky po zahájení vývoje
Testovací plán se reviduje po každém sprintu	Zřídka se diskutuje o testování během testovací fáze
Od agility se očekává, že se požadavky budou měnit a vyvíjet	Ideální pro projekty, kde se požadavky a změny vůbec neočekávají
Malé a oddané týmy, dobře koordinované a spolupracující	Tým je značně omezen na koordinaci a synchronizaci
Očekává se velké zapojení zákazníka	Neočekává se zapojení zákazníka
Rozpočet je flexibilní	Rozpočet je pevně dán

*Tabulka 2: Agile vs. Waterfall (zdroj: vlastní zpracování)*

Jako výhody se do waterfall metody jednoznačně řadí následující body:

- lepší plánování, koordinace a řízení
- schopnost doručení projektu i s méně kvalitním týmem
- jasné určení zodpovědnosti
- je přirozenější

Nicméně vedle výhod jsou i nevýhody a jejich výčet je zde:

- neefektivní způsob řízení
- statisticky méně úspěšný
- nemá dostatek zpětné vazby od zákazníka

Do výhod agilní metody lze zařadit:

- vysoká úspěšnost projektů
- více zpětné vazby od zákazníka
- poměrně velké zaměření na spokojenost týmu a dosahování výsledků

Nevýhody agile metody:

- obtížnost při škálování a řízení skrze velké celky
- vynikající práce v týmu, selhává však ve spolupráci mezi týmy
- může se zdát obtížná a nepřirozená

Waterfall a agile jsou dvě různé metody řízení a každá se hodí na jiný typ projektů. Pokud jsou již od počátku jasné výsledky, je waterfall tím nejlepším možným způsobem řízení daného projektu. Waterfall je ideální pro projekty, které musí splňovat přesné předpisy (např. stavební projekty).

Dále je pak agile, když není vhodné aplikovat waterfall. Tato metoda je vhodnější pro týmy, kde je rychlý pohyb, hodně se experimentuje a úplně přesně nikdo neví, jak má konečný produkt vypadat. Zároveň je agile hodně flexibilní, tým spolu hodně spolupracuje a je vyžadovaná častá kontrola se zákazníkem a dalšími zainteresovanými osobami, které jsou spojeny s projektem.

Pokud se na projekt nehodí ani jedna výše zmíněná metodika, je možné na projekty aplikovat i tzv. hybrid metodiku. Ta spočívá ve využití kombinace waterfallu a agilu.

## 4. Vlastní práce

Vlastní práce nejprve podrobněji popisuje proces projektového řízení ve zvolené společnosti. Následuje rozbor konkrétního HW projektu a porovnání jeho řízení s interní metodikou. Nejstěžejnější část se věnuje aplikaci vybraných agilních metodik právě na výše zmíněný HW projekt. V neposlední řadě jsou vyzdvihnuty aspekty a závěry, které vyplývají z aplikace agilních metodik na projekt. Dále jsou přiloženy odpovědi vybraných RDM v rámci dotazníku týkajícího se využití agilních metodik na HW projekty.

### 4.1 Profil společnosti

Vzhledem k zachování anonymity zvolené společnosti, ve které bude podrobněji rozebráno projektové řízení a konkrétní projekt, bude používán název Společnost XY.

Společnost XY je soukromá společnost, která působí v oblasti dodávek zboží, služeb a řešení na poli informačních a komunikačních technologií od roku 2001. Během prvních let působení na trhu přispěly k jejímu rychlému rozvoji výroba vlastní značky osobních počítačů a také budování portfolia, které se zaměřovalo na obchodní a servisní sítě. Následně se Společnost XY zaměřila na poskytování souhrnných IT řešení a služeb pro firmy a také pro státní správu. Tento krok tak umožnil to, že se Společnost XY vypracovala mezi nejvýznamnější firmy ve sféře objemných a složitých ICT projektů.

Dále se Společnost XY zaměřila na vylepšování odborných znalostí a schopností, monitoring zákazníků a taktéž na samotnou dekompozici vnitřní organizace firmy. Díky všem zmíněným krokům se tak stala největším a nejvýznamnějším českým dodavatelem informačních a komunikačních technologií.

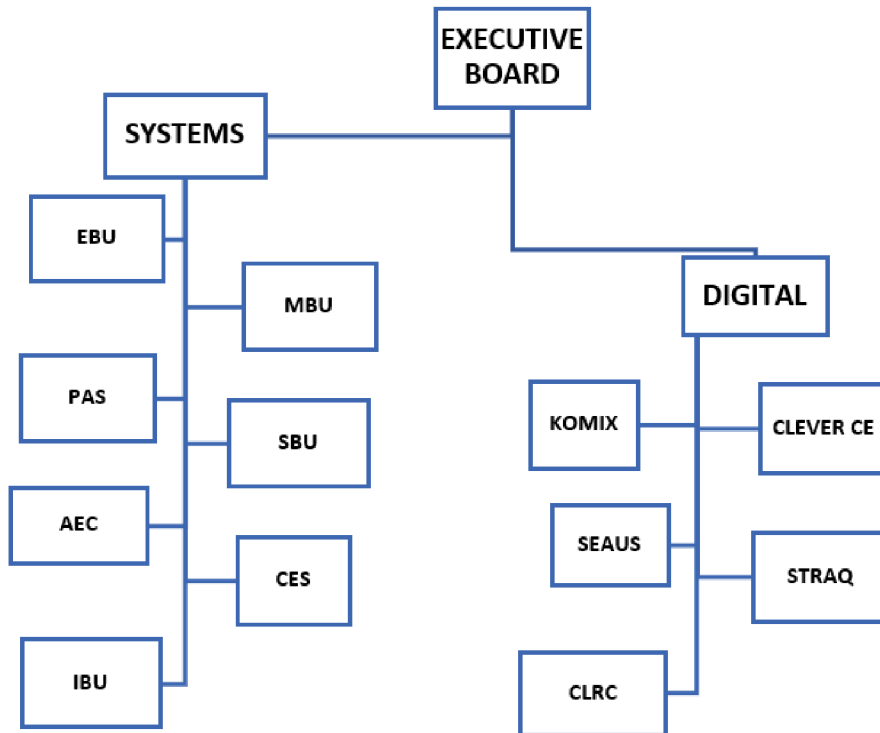
Společnost XY se pyšní i svým posláním, které klade velký důraz na zavádění a provozování informačních technologií v organizacích. A co se týče firemních hodnot, tak ty jsou založené zejména na odbornosti, týmové spolupráci a v neposlední řadě na osobní odpovědnosti.

Během posledních několika let, které byly významně dynamické, se představenstvo Společnosti XY rozhodlo pro akvizici několika dalších firem a následně se stala součástí jedné nejmenované skupiny, která je zaměřena na dodávky IT produktů, služeb a poradenství.



#### 4.1.1 Organizační struktura

Organizační struktura společnosti je interním dokumentem a z tohoto důvodu bude následné vyobrazení lehce pozměněno. Nicméně stále bude odrážet realitu a jednotlivé úseky a divize Společnosti XY.



Obrázek 13: Organizační struktura Společnosti XY (zdroj: Společnost XY, 2023)

Jak lze z nákresu organizační struktury vidět, Společnost XY je rozdělena na dvě divize Systems a Digital, nicméně obě spadají pod Executive board. Dále jsou jednotlivé divize rozděleny na úseky a každý z nich se specializuje v oblasti dodávek IT na něco jiného. Do těchto specializací se řadí IT infrastruktura, outsourcing IT, kybernetická bezpečnost, podnikové aplikace, cloudové služby a v neposlední řadě i architektura a strategie IT.

K této diplomové práci bude ale nejdůležitější divize Systems a zejména úsek MBU. Tento úsek se zabývá IT infrastrukturou. Tato problematika je poměrně rozsáhlá, jelikož začíná u kabelů, které jsou položeny v zemi a naftových generátorech, které slouží k zajištění napájení během výpadku až po výstavbu datových center, včetně možností využití technologií cloud computingu, datových uložišť, virtualizací aplikací a desktopů až po zajištění bezpečnosti.

#### 4.1.2 Projektové řízení ve Společnosti XY

Projektové řízení ve Společnosti XY nevychází z celosvětově známého standardu či metodiky. Vzhledem k různorodému portfoliu Společnost XY, si každá divize a úsek volí svůj přístup k řízení projektů. Většina těchto přístupů je souhrn best practices a lessons learned za několik let působení na trhu s ICT projekty a jejich úspěšným ukončením. Majorita těchto přístupů je popsána v interních dokumentech, které mají podobu textového dokumentu, kde je popsáno, jak přistupovat k projektům a jaké kroky by se při řízení jednotlivých projektů měly dodržovat, resp. co nikdy nevynechat. Nicméně stejně jako u metodiky PRINCE2, jsou tyto dokumenty brány jako podpora pro řízení projektů, a ne jako striktní dokument, který se musí dodržovat každým krokem.

Následné kapitoly se budou týkat projektového řízení v úseku MBU a od toho se budou odvíjet i interní metodiky na efektivní řízení hardwarových projektů, protože MBU se zaobírá ICT infrastrukturou.

#### 4.1.3 Životní cyklus zakázky

Pro pochopení, jak ve Společnosti XY funguje projektové řízení, je důležité porozumět všem procesům, které se týkají životního cyklu zakázky, protože právě toto porozumění je klíčem k interním metodikám a stylu řízení projektů.

Životní cyklus zakázky se skládá hned z několika fází, a to z obchodní, realizační a poskytování servisních služeb. Tento cyklus však ještě může obsahovat i další, podpůrné procesy.

##### **Obchodní fáze**

V této fázi je nejdůležitější obchodník, který je zodpovědný za objednávky u zákazníků a nezáleží, zda se jedná jen o prodej HW, služeb či implementace ICT infrastruktury. Výstupem této fáze je nabídka a návrh smlouvy. Ještě před samotnou realizační fází musí dojít ze strany obchodníka a zákazníka k podpisu smlouvy. Obchodník tedy zodpovídá za iniciaci toho, aby byl požadavek předán do realizace a zároveň měl připravenou potřebnou obchodní dokumentaci v odpovídající kvalitě.

##### **Realizační fáze**

V realizační části je nejdůležitější osobou projektový manažer, resp. Společnost XY používá raději roli Regionální delivery manager (RDM). V této fázi musí RDM dostat od

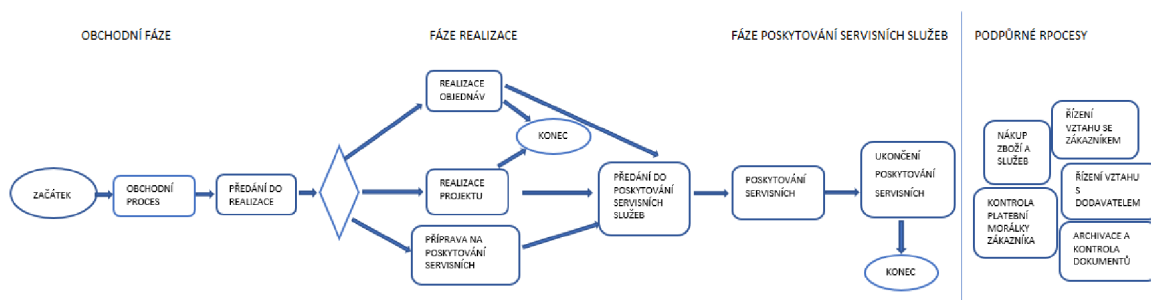
obchodníka obchodní dokumentaci a zkontrolovat ji, aby byla v pořádku, protože v realizační fázi přebírá RDM odpovědnost realizaci projektu.

### Fáze poskytování servisních služeb

Po fázi realizace může nastat konec projektu, ale pokud se přijme jen požadavek na realizaci jakéhokoliv charakteru či požadavek na servisní služby, tak následuje ještě fáze poskytování servisních služeb.

### Podpůrné procesy

V neposlední řadě přichází na řadu podpůrné procesy, které nemusí souviset s předchozími fázemi. Jedná se o podporu jako: nákup zboží a služeb, archivace a kontrola dokumentů, kontrola platební morálky zákazníka a řízení vztahu s dodavatelem či zákazníkem.



Obrázek 14: Životní cyklus zakázky (zdroj: Společnost XY, 2016)

Na obrázku 16 je vizualizace životního cyklu zakázky. Následně bude tato diplomová práce věnována realizaci konkrétního projektu, nejdříve je však nutné zmínit kroky interní metodiky.

#### 4.1.4 Interní metodika

V interní metodice je zmíněno, že je zapotřebí sladit poznatky a zkušenosti, které byly získány z úspěšně ukončených projektů, a právě tyto nabyté znalosti provázat se základními pravidly, které definuje např. metodika PRINCE2. Ačkoliv už bylo zmíněno, že Společnost XY nevede projekty dle konkrétního standardu či metodiky, tak většina RDM musí splnit základní certifikaci PRINCE2 a mít tedy ponětí, jak se projekty mohou řídit.

Je důležité brát v potaz, že MBU je úsek, který je zaměřen na ICT infrastrukturu, tedy zejména na hardwarové projekty a ve většině případů jsou projekty řízeny klasickým, tedy waterfallovým způsobem. Dále je nezbytné přimět klienta či zákazníka hned na začátku, aby jasně vyjádřil, co od projektu očekává, resp. co bude považovat za dokončený a splněný

cíl. Toto definování cíle je v praxi velmi složité a dost často i obtížně dosažitelné. Není tedy výjimkou, že se zákazník na konci projektu nedokáže spokojit s tím, co bylo dodáno. Takovéto nepříjemnosti s sebou nese bohužel tzv. waterfallové řízené projektů. Nicméně v interní metodice je jasně popsáno, že se takovýmto nepříjemnostem musí RDM vyhnout na maximální možné úrovni.

Dalším doporučením v interní metodice je fakt, že řadu dokumentů je třeba výrazně přizpůsobit aktuální situaci a projektu, včetně obsazení pracovních pozic, formálního nastavení a personálního obsazení orgánů projektu, jakož i formy a míry podrobnosti řady dokumentů. Přehlednost je nutná i u malého projektu a je nutné vědět (Interní metodika Společnosti XY, 2019):

- Kdo je zodpovědný za jeho správu a řízení?
- Jak je projekt evidován v našem IS a kde je projektová dokumentace?
- Je-li projektový tým více než dvoučlenný, co se od členů očekává a za co zodpovídají?
- Kdo odpovídá za dodání správného řešení, které splní naše požadavky a požadavky zákazníka?

A v neposlední řadě je v interní metodice přehled platných pravidel a šablon, které je třeba dodržovat a respektovat v celém životním cyklu projektu. Všechny jsou publikované na interním portále a je třeba vždy vycházet z aktuální publikované verze.

V následující části budou detailněji popsány jednotlivé fáze projektu, kterým se interní metodika věnuje detailněji:

### **Plánování projektu**

Plánování projektu je nejdůležitější z toho pohledu, aby byla převzatá obchodní dokumentace od obchodníka a byla tak vhodná k přípravě podkladů pro celý projekt. Je tedy potřebná smlouva se zákazníkem, nebo závazná objednávka. V dalším kroku je nezbytné definovat klíčové role a odpovědnosti. Jde zejména o cross-úsekové řízení projektů. Někdy RDM potřebuje vedoucího úseku, aby mu pomohl najít vhodné řešitele pro projektový úkol dle kompetencí, které zastává. Nejen u řešitelů je potřeba stanovit role a odpovědnosti, jako další se musí stanovit vedoucí projektu, hlavní architekt (může jich být více) a sponzor. Nevyhnutelná fáze v plánování projektu je samotné založení čili vytvoření projektu v informačním systému Společnosti XY. Informační systém, který je využíván pro řízení

projektů, je aplikace od Microsoft, tzv. D365. Tato aplikace je velice podobná MS Project, ale je již rovnou propojena s finančním modulem.

Založení projektu se tedy skládá z dekompozice plánovaných prací (projekt), které mají splnit očekávaný cíl zákazníka, ale je také potřeba k tomu doplnit potřebnou dokumentaci, analýzy a taktéž interní kalkulace a podíl jednotlivých týmů na kalkulaci a harmonogram.

### **Zahájení projektu**

V této fázi se jedná hlavně o interní aktivity typu – co je potřeba upřesnit, aby bylo jasné, co je nutné dodat. Do těchto aktivit spadá cíl projektu a hodnocení jeho úspěchu, zainteresované osoby a případná koordinace s ostatními projekty u zákazníka.

Cíl projektu a hodnocení jeho úspěchu je brán z pohledu zákazníka a obsahuje tyto body:

- Co se má skutečně změnit – výstup / výsledek / přínos;
- Kdo to stanoví/hodnotí, jaká má na tom KPI;
- Co se musí změnit, aby bylo dosaženo cíle;
- Kdo jsou zainteresované osoby, které mají na projekt vliv;
- Za jakých podmínek se bude projekt hodnotit;
- Do které skupiny patří zainteresované osoby – investor X uživatel X dodavatel.

Co se týče zainteresovaných osob, tak je doporučena komunikační matice, tedy, kdo s kým a v jaké intenzitě bude komunikovat. A dále je doporučena mapa těchto osob, protože každá z nich může pocházet úplně z jiného útvaru či mimo organizaci.

Vedle aktivit, které stanovují jasnost a definovanost projektu jsou další, nezbytné aktivity, které musejí být vykonány dle interní metodiky. Jako první z aktivit je interní kick-off, kde je prezentace všech podkladů, jsou stanovena akceptační kritéria, potřebná role obchodníka, a nakonec role a odpovědnosti. Následně navazuje aktivita, kde probíhá seznámení všech členů projektového týmu s klíčovými parametry projektu, jejich rolí a odpovědností, a nakonec se smluvními podmínkami. Jako poslední aktivita, která předchází samotné realizační fázi, je příprava na externí kick-off. Zde je potřeba mít vše interně vyjasněné jako požadavky na součinnost, skutečný cíl projektu a podrobný plán prací projektu. Taktéž je zapotřebí definovat pravidla komunikace na projektu a rizika projektu (opět záleží na velikosti projektu).

## Realizace projektu

V realizační části projektu má největší zodpovědnost RDM o následující činnosti:

- Plánování práce, alokace projektového týmu, kontrola práce projektového týmu a kontrola vykazování práce a zveřejňování v příslušných termínech;
- Vedení projektové dokumentace – zápisy, protokoly a evidence, koordinace návštěv, součinnost se zákazníkem;
- Zajištění přípravy potřebné technické dokumentace;
- Dodávky zboží (HW, SW), poskytování služeb subdodavatelem, akceptační protokoly a předávání protokolů obchodníkovi.

Vedle těchto administrativních činností se ještě očekává velká interakce se zákazníkem a jednou takovou interakcí je právě externí kick-off. Zde je nutné si potvrdit cíle projektu se zákazníkem (včetně hodnotících kritérií) a také je doporučeno si vytvořit mapy zainteresovaných osob u zákazníka. Dále se vyjasní, kdo bude hlavní uživatel, který bude rozhodovat o vhodnosti dodaného výsledku či nikoliv. Další důležitou součástí kick-offu je prezentace RDM o přístupu Společnosti XY k projektu (technologie, implementace), forma a způsob pravidelného reportování, akceptační kritéria, harmonogram projektu a v neposlední řadě požadavek na nutnou součinnost zákazníka (ideálně konkrétních lidí) a případná rizika a dopady, pokud tato součinnost nebude poskytnuta.

V realizační fázi projektu jsou taktéž zmíněny řídicí výbory, a to interní a externí. V interním řídicím výboru se diskutuje zejména nad základním statusem projektu, jedná se o konfrontaci stavu projektu se smlouvou/harmonogramem/rozpočtem. Dále i ekonomické hodnocení a jak se plán čerpání zdrojů odlišuje od skutečnosti. Dalšími záležitostmi, kterým se řídicí výbor věnuje, jsou také rizika a plán jejich eliminace.

Zatímco se externí řídicí výbor věnuje spíše struktuře podkladů k projektu, jsou pro něj nejdůležitější informace jako stav projektu, stav plnění úkolů a plánované úkoly na nejbližší období. Dále řídicí výbor schvaluje akceptaci dosažených milníků. Taktéž je pro něj důležitý časový scénář přípravy podkladů a způsob jejich předložení. Mezi další aktivity řídicího výboru se řadí ověřování, zda je zákazníkovo očekávání kompatibilní s očekáváním Společnosti XY, sledování rizik a také kontrola vstupů nových lidí do projektu.

Další části, které se řídicí výběr věnuje, je interní měření, resp. hodnocení projektu:

- Plnění klíčových milníků;
- Čerpání zdrojů – porovnání plánu a skutečnosti;
- Ekonomické hodnocení – standardní projektový report, příp. plán cashflow;

- Stav smluvní a projektové dokumentace;
- Co zbývá udělat k dosažení cílů projektů.

### **Ukončení projektu**

Aby mohl být projekt ukončen, je nutné splnit následující podmínky:

- Realizace v dohodnutém rozsahu a kvalitě;
- Musí být vypořádány implementační issue-logy a všechny výhrady k akceptaci;
- Mít kompletní sady originálně podepsaných akceptačních protokolů nebo dodávkových listů;
- Musí být vystaveny všechny faktury;
- Je nutné vypořádání se všemi subdodavateli;
- Je třeba mít zaúčtovány všechny náklady (interní + externí).

Nicméně za správné ukončení projektu je odpovědný RDM a jeho úkoly spočívají mimo jiné i v předání akceptačních a předávacích protokolů obchodníkovi, zajištění vypracování případové studie, pomocí konkrétní šablony musí sestavit hodnocení projektu, archivovat veškeré dokumenty dle smluvních podmínek a jako poslední uzavřít projekt v interním informačním systému.

Po úspěšném ukončení projektu se v mnoha případech stane, že zákazník, který si projekt objednal a následně byl spokojen s jeho výsledkem, si nechá od Společnosti XY poskytovat servisní služby.

Závěrem je k interní metodice důležité napsat, že se Společnost XY řídí svojí ozkoušenou metodikou a zároveň ani není nikde definováno, zda se projekty řídí klasickým způsobem (waterfallově) či agilně, nebo dokonce hybridním způsobem.

#### 4.1.5 Interní metodika vs. PRINCE2

Vzhledem k tomu, že Společnost XY požaduje po svých RDM minimálně základní certifikaci PRINCE2, budou proto v následujících řádcích popsány možné podobnosti či naprosté odlišnosti mezi interní metodikou a metodikou projektového řízení PRINCE2.

Metodika projektového řízení PRINCE2 nabízí hned několik výhod podle AXELOS (2017):

- Strukturovaný přístup: nabízí jasně definovaný proces pro řízení projektu, který poskytuje jasnou strukturu a kontrolu projektu od počátku do konce;

- Flexibilita: PRINCE2 je flexibilní metodika, která může být použita pro různé typy projektů a různé velikosti týmů;
- Řízení rizik: zahrnuje procesy pro identifikaci a řešení rizik, což pomáhá minimalizovat negativní dopady na projekt;
- Řízení změn: nabízí procesy pro řízení změn, což umožňuje týmu rychle reagovat na změny v požadavcích nebo okolnostech projektu;
- Řízení kvality: zahrnuje procesy pro řízení kvality, což pomáhá zajistit, že projekt splňuje požadavky na kvalitu;
- Řízení týmu: nabízí jasně definované role a odpovědnosti pro členy týmu, což pomáhá zajistit, že projekt je řízen efektivně;
- Ukončení projektu: zahrnuje procesy pro ukončení projektu, což pomáhá zajistit, že projekt je úspěšně dokončen a že se z něj plyne maximum benefitů;
- Zlepšení komunikace: zahrnuje procesy pro řízení komunikace, což pomáhá zajistit, že všichni zúčastnění v projektu jsou informováni o důležitých informacích a že jsou vytvořeny jasné kanály pro komunikaci;
- Zlepšení plánování: nabízí procesy pro plánování projektu, což pomáhá týmu lépe plánovat a koordinovat činnosti projektu;
- Zlepšení kontroly: nabízí procesy pro kontrolu projektu, což pomáhá týmu sledovat postup projektu a zajistit, že se projekt vydává správným směrem;
- Zlepšení dokumentace: PRINCE2 zahrnuje procesy pro dokumentaci projektu, což pomáhá zajistit, že důležité informace jsou zaznamenány a uchovány pro budoucí reference.

Oproti tomu má interní metodika poměrně volný postup a figuruje spíše jako sada doporučení, jak řídit infrastrukturní projekty v ICT. Podobnosti či rozdílnosti vůči metodice PRINCE2 jsou:

- Strukturovaný přístup: zde je značný rozdíl v tom, že interní metodika nedefinuje jasný proces řízení projektu. Naopak se odvolává na skutečnost, že každý projekt je jiný a tím pádem se ke každému přistupuje jinak.
- Flexibilita: zde je přístup podobný u obou metodik, je vhodné si metodiku tailarovat pro daný projekt.
- Řízení rizik: PRINCE2 zahrnuje procesy pro identifikaci a řízení rizik, interní metodika se o rizicích sice zmiňuje a přisuzuje jim určitou váhu, není však nikde definován postup a jejich řízení.



- Řízení změn: zde je opět velký rozdíl, kde interní metodika nepopisuje řízení změn, kdežto PRINCE2 ano.
- Řízení kvality: v interní metodice je toto téma prisuzováno akceptačním kritériím.
- Řízení týmu: tento bod definují obě metodiky stejně a obě tomu přidávají důležitou váhu.
- Ukončení projektu: interní metodika nutí dodržovat podmínky, aby mohl být projekt správně dokončen, PRINCE2 má své procesy. Lze z toho vyvodit závěr, že obě metodiky mají své kroky, které musí při ukončení projektu splnit.
- Zlepšení komunikace: PRINCE2 má jasně definované procesy pro řízení komunikace (komunikační matice, kanály, atd), naopak interní metodika se sice odvolává na řízení komunikace, ale nejsou definovány žádné konkrétní komunikační kanály nebo komunikační matice.
- Zlepšení plánování: zde jsou obě metodiky na podobné úrovni a podotýkají, že dobré plánování je základ.
- Zlepšení kontroly: PRINCE2 má opět své procesy, kdežto interní metodika podotýká, že kontrolu si provádí sám RDM a je na něm, koho si k tomu kdy přizve, ale většinou je u toho projektový tým a zákazník.
- Zlepšení dokumentace: obě metodiky zahrnují procesy, které se skládají i z šablon a dokumentů, které jsou vhodné k vylepšování dokumentace.

## 4.2 Vybraný projekt – ZZS JČK IT Technologie

Pro praktický výzkum v této práci je zvolen hardwarový projekt, který byl skutečně implementován Společností XY. Projekt ZZS JČK IT Technologie, jak už vypovídá sám název, je orientován zejména na dodávku IT technologie, včetně souvisejících montážních prací. Zákazníkem je firma, která působí ve farmaceutickém a zdravotnickém sektoru. Výslednou dodávkou projektu, resp. cílem projektu je rozmístění serverů na jednotlivé lokality dle požadavků zákazníka a s tím spojené činnosti jako je např. instalace, konfigurace, virtualizace, povýšení AD (*active directory*) na nejvyšší možnou úroveň a další. Ačkoliv se cíl projektu zdá velice jednoduchý, samotná implementace veškerého HW musí splňovat určité posloupnosti, aby naplňoval funkční požadavky, které si zákazník objednal a zaplatil.

Aby mohl být projekt implementován, musí RDM dostat nabídku a objednávku, které jsou dobře zpracované od obchodníka, tzn. RDM si musí vždy zkontrolovat, aby objednávka HW odpovídala s nabídkou prací. Dle těchto dokumentů je tedy smluvně dohodnuto, že Společnost XY dodá IT technologie spočívající zejména v implementaci nových boxů hlavního firewallu, přenesení konfigurace z původních boxů, přesunutí LAN konektivity, kabeláže a konfigurace switchů. Dále je potřeba nahradit stávající fyzické servery a reinstalovat virtuální servery na nejvyšší možnou úroveň. A jako poslední je instalace VMWaru na servery virtualizačního clusteru.

Zároveň se od dodavatele HW zařízení očekává, že provede vzdálenou instalaci. Zákazník přislíbil, že si sám rozveze servery na jednotlivé lokality a bude spolupracovat s implementačními týmy Společnosti XY a RDM.

### Plánování projektu ZZS JČK IT Technologie

Při plánování projektu ZZS JČK IT Technologie je prvním nezbytným krokem příjem obchodní dokumentace od obchodníka (objedávka a nabídka). Během tohoto kroku si přiřazený RDM (dle kraje působnosti) překontroluje dokumenty a následně vytvoří směrný plán projektu v interním informačním systému D365. Směrný plán je vytvořen z důvodu, aby mohl RDM reportovat zákazníkovi aktuální stav projektu a jeho odlišnosti od původního naplánování, viz obrázek 17. Zároveň směrný plán slouží i samotnému RDM pro sledování projektu. Ve směrném plánu jsou tedy definovány jednotlivé úkoly, předchůdci či následovníci, doba trvání v hodinách, datum, kdy má být úkol realizován a jako poslední je k jednotlivému úkolu přiřazen jeden nebo více řešitelů.

Dalším důležitým krokem je definování klíčových rolí a odpovědností. U tohoto projektu je tedy definován RDM jako hlavní vedoucí projektu, projektový asistent (volitelná role), konkrétní řešitelé pro dané úkoly a hlavní architekt, resp. řešitel. Je důležité zmínit, že je tento projekt velice jednoduchý na implementaci, a proto není potřeba přiřazovat každý úkol jinému řešiteli. Také to má velkou výhodu v tom, že pokud jednomu řešiteli navazuje na sebe vícero úkolů, snižuje se tím náročnost koordinace RDM, protože řešitel si úkoly navazuje na sebe ihned po dokončení předcházejícího úkolu. Sponzorem je v tomto případě samotný zákazník.

Dle interní metodiky je dále doporučeno doplnit potřebnou dokumentaci a analýzy. Nicméně vzhledem k tomu, že se nejedná o komplikovaný projekt a jde o implementaci u stávajícího zákazníka, odpadá tak potřeba před implementační dokumentace či analýza rizik, protože prostředí je pro řešitele Společnosti XY dobře známé.

	Režim	Jméno	Předchůdce	Úsilí	Počáteční datum	Koncové datum	Doba trvání
	Automati...	ZS JČK - IT Technologie II - 2022		175,00	01.11.2022	10.02.2023	72
1	Ruční	Zahájení projektu a kick-off		1,00	01.11.2022	01.11.2022	1
2	Ruční	Instalace 2ks nových boxů hlavního firewallu	1	40,00	16.11.2022	09.12.2022	17
3	Ruční	Instalace VMWaru na nové servery R650 + zařazení do virtualiza	1	16,00	16.11.2022	16.12.2022	22
4	Ruční	Instalace Dell monitorů	1	8,00	21.11.2022	10.02.2023	59
5	Ruční	Nahrazení stávajících šesti fyzických a reinstalace dvou virtuálníc	1	32,00	05.12.2022	02.01.2023	20
6	Ruční	Povyšení struktury AD + přenos rolí na nové servery (DC, radius, 5	5	56,00	03.01.2023	16.01.2023	10
7	Ruční	Integrace všech dodaných technologií do PRTG		8,00	12.12.2022	16.01.2023	25
8	Ruční	Dokumentace	1	12,00	02.11.2022	23.12.2022	37
9	Ruční	Vedení projektu a projektová rezerva	5,7,8	1,00	17.01.2023	19.01.2023	3
10	Ruční	Ukončení a akceptace projektu		1,00	15.12.2022	15.12.2022	1

Obrázek 15: Směrný plán projektu (zdroj: Společnost XY)

### Zahájení projektu ZZS JČK IT Technologie

Ve fázi zahájení projektu je nejdůležitější interní kick-off. Avšak ještě před samotným kick-off musel RDM upřesnit aktivity, které vedly k definování konkrétního cíle projektu (u HW projektů je toto velice důležitý bod) a k jeho hodnocení, dále musel kontaktovat všechny zainteresované osoby, kterých se projekt týká a domluvit termíny implementace u zákazníka.

Interní metodika doporučuje vytvoření komunikační matice pro zainteresované osoby, nicméně pro tento projekt není vytvořena, a právě toto vyjmutí poměrně důležitého nástroje pro pomoc řízení projektu je v realizační části vnímáno jako chybné rozhodnutí. Dále není sestavena ani mapa zainteresovaných osob.

Realizace samotného interního kick-off probíhá na platformě MS Teams z důvodu, protože většina realizátorů přiřazena na konkrétní úkoly je rozmístěna po celé České republice, jedná se tedy o cross-úsekové řízení projektu. Této schůzky se účastní RDM, projektový asistent, dodavatel HW a pouze nejdůležitější řešitelé, takže ne všichni pak mají k dispozici aktuální informace o prostředí, ve kterém budou HW implementovat. Na programu je definování si rolí a odpovědností, dále cíl projektu, tedy čeho se má dosáhnout a definování akceptačních kritérií z pohledu Společnosti XY. Dále jsou vyjasněny požadavky na součinnost a podrobný plán projektu. Vzhledem k tomu, že projektový plán sestavoval RDM s pomocí projektového asistenta, je nutné si na interním kick-off vyjasnit, zda pracovní podmínky a termíny úkolů odpovídají reálnému času, ve kterém je jsou řešitelé schopni implementovat. Poté, co je veškerá agenda prodiskutována, pošle po ukončení kick-off projektový asistent zápis na všechny členy projektového týmu a zároveň domluví termín na externím kick-off se zákazníkem.

### **Realizace projektu ZZS JČK IT Technologie**

Samotná realizace projektu je největší odpovědnost RDM a projektového asistenta, pokud je v projektovém týmu. RDM má na starosti jak administrativní činnosti, tak zároveň musí být vždy k dispozici zákazníkovi a řešitelům projektového týmu. V projektu ZZS JČK IT Technologie se RDM zabývá administrativní prací jako je plánování práce, alokace řešitelů na jednotlivé úkoly a jejich následná rezervace v kalendáři, schvalování vykázané práce řešitelů, koordinace projektového týmu a dodržování termínů plnění. Dále je RDM zodpovědný za příslušnou dokumentaci, kterou musí delegovat na hlavního architekta řešení a dále zajišťovat komunikaci s dodavatelem HW a spolupracovat s obchodníky, kteří jsou zodpovědní za dodávky HW v požadovaném čase.

Vedle administrativních činností spadá do realizace projektu i externí kick-off, ten je již se zákazníkem. Během této schůzky, opět na platformě MS Teams, se společně se zákazníkem upřesní cíl projektu včetně hodnotících kritérií. Dále musí být vyjasněno, kdo bude hlavní uživatel, který rozhodne o vhodnosti dodaného výsledku, v tomto případě se jedná o zákazníka. Jsou potvrzeny termíny dodávek a řešitelé, se kterými je možné komunikovat. V neposlední řadě musí RDM prezentovat celkový přístup Společnosti XY k projektu ZZS JČK IT Technologie včetně technologií, reportování, harmonogramu projektu a jako poslední definovat požadavek na součinnost zákazníka, resp., jak by projektový tým potřeboval kooperovat se zákazníkem.

Interní metodika v realizační fázi zmiňuje řídicí výbory – interní a externí. Do interního řídicího výboru se řadí RDM a jeho nadřízení, resp. ředitelé divize, kteří s ním konzultují smlouvu, harmonogram a rozpočet. Je důležité mít prostudovanou smlouvu, protože velice často se v ní objevují penále za nedodání produktů včas či nedodání výsledného produktu v dohodnuté specifikaci, a právě toto je často diskutováno mezi RDM, obchodníkem a ředitelem divize.

Externí řídicí výbor se skládá ze zákazníka a RDM. Pro zákazníka jsou důležité reporty, jež jsou zpracovány na týdenní bázi. Tyto reporty zpracovává RDM a zároveň informuje zákazníka o úkolech, které jsou plánovány na nejbližší období. Dalším požadavkem, který externí řídicí výbor vyžaduje, je ověřování, zda je zákazníkovo očekávání kompatibilní s očekáváním Společnosti XY. Spolu s tím se sledují rizika, nicméně u tohoto konkrétního projektu byla veškerá komunikace na jednoduché bázi a rizika se už vůbec neřešila. Ačkoliv RDM pravidelně vypracovává reporty, tak se 1x za 14 dní uskuteční schůzka na MS Teams, kde se vše ještě jednou prodiskutuje a připomenou se termíny plnění.

### **Ukončení projektu ZZS JČK IT Technologie**

Ukončení projektu probíhá téměř bez výhrad dle interní metodiky a obsahuje všechny v ní zmíněné body. Tzn. RDM je zodpovědný za včasné dodání realizace v dohodnutém termínu a kvalitě. Projektový asistent vypracovává akceptační protokol a dává ho k podpisu obchodníkovi, který ho následně se zákazníkem podepisuje, pouze za předpokladu, že se dodalo vše, oč zákazník žádal, a to v požadované kvalitě. Tomuto kroku znovu předchází schůzka na MS Teams, kde si zákazník, RDM a hlavní realizátor vyjasní, co bylo dodáno a co naopak ještě chybí dodat či implementovat.

Ačkoliv byl akceptační protokol podepsán oběma stranami a schválen ke dni předání akceptačního protokolu, tak nebyly dodány všechny služby a HW. A to z důvodu, že dodávky HW se neustále posunují a nikdo nedokáže s přesným datem říci, kdy dorazí. To má pak za následek zpoždění služeb na instalaci těchto HW zařízení. S akceptačním protokolem byl ještě sepsán dodatek, který definoval HW a služby, které budou ještě vykonány. Každopádně se jedná jen o minoritní úkoly – dodávka monitorů, jejich implementace a nastavení monitoringu PRTG. Výsledkem projektu ve fázi podepsání dodatku je fakt, že je projekt dokončen z 90 % (potvrzeno od zákazníka).

Nicméně projekt je formálně uzavřen, jsou podepsány všechny protokoly, ale obě strany jsou dohodnuté, že až dorazí HW, tak se musí implementace co nejrychleji dokončit.

RDM a projektový asistent jsou tedy ve fázi, kdy tento projekt už není veden jako aktivní, ale ve stavu odloženého řešení. Až po úplném dokončení projektu schválí RDM zbytek vykázaných prací realizátorů v interním systému, dále zavře projekt a bude požadovat po hlavním architektovi řešení vypracovanou technickou dokumentaci.

#### 4.2.1 Problémová místa v projektu, která nebyla dostatečně ošetřena

U projektu ZZS JČK IT Technologie není zpracována analýza rizik a ani dokument, který by se zabíral problematikou možných ohrožení projektu. I když má Společnost XY interní metodiku, která slouží pouze jako sada doporučení, jak projekt vést, tak se během tohoto konkrétního projektu objevily problémy, kterým se mohlo předejít větším využitím interní metodiky nebo i dalších metodik, které nejsou v interním dokumentu popsány.

Problémy, které se vyskytly kvůli nesestavení komunikační matice:

- projektový asistent nebyl dostatečně respektován ze strany řešitelů;
- celková komunikace s realizátory (účast i na jiných projektech).

Projektový asistent nebyl dostatečně respektován ze strany řešitelů z důvodu, že je tato role a jí přiřazené odpovědnosti nepovinnou rolí v projektovém týmu a většina řešitelů s ní nemá žádné zkušenosti. Dalším důvodem je fakt, že tuto roli vykonává juniorní zaměstnanec, který má s pozicí RDM pouze krátkou zkušenost. Ze strany řešitelů se tedy nejedná o dostatečnou autoritu. I když se hlavní manažer projektu, tedy RDM, snaží řešitele na tento problém upozorňovat, není jednoduché tuto skutečnost zásadněji ovlivnit.

Dalším problémem je to, že jsou realizátoři alokováni na několik souběžně běžících projektů a z pohledu RDM je značný problém zajistit kapacity řešitelů na konkrétní termín a úkol. Tento problém se bohužel nedá nijak vyřešit a je to zodpovědnost RDM, koho si, na jaký úkol a v konkrétním termínu alokuje. Nicméně tato problematika je řešena řediteli úseku MBU a snaží se najít nějaké vhodné řešení, aby se tyto problémy často nevyskytovaly.

Problémy, které nelze nijak ošetřit za využití interní metodiky:

- RDM řídí více projektů, než jsou schopni zvládnout;
- dodávky HW jsou nepředvídatelné a tím je způsobeno nedodržení včasného doručení projektu;
- vysoká odbornost řešitelů a znalost zákaznickova prostředí – řešitele není jednoduché zaměňovat s nikým dalším;

- absence retrospektivy (co se podařilo a nepodařilo na projektu) – lze ji dělat, ale postrádá smysl pro další projekty, protože je na každý projekt jiný tým řešitelů.

Ve Společnosti XY není jednomu RDM přiřazen pouze jeden projekt. Jeden RDM vždy působí ve více krajích České republiky, které musí pokrýt. Tím, je mnohdy způsobeno, že projekty nejsou vedeny efektivně a v mnoha případech se zpožďují (v těchto případech v tom nehraje pozdní dodávka HW).

Co se týče dodávek HW, tak je to velmi nepředvídatelná oblast a vždy je potřeba zákazníka informovat o možných rizicích s tím souvisejících.

Pro Společnost XY je typické, že vedle projektů a drobných realizací, udržuje u jistých zákazníků i pravidelné služby v IT sféře. A právě u těchto zákazníků je tvořen tým řešitelů ze Společnosti XY, který zodpovídá za funkčnost IT prvků u zákazníka. V případě, že od stávajícího zákazníka přijde nabídka na projekt, je vhodné pro jednotlivé úkoly na projektu zvolit ty řešitele, kteří jsou v samotném týmu řešitelů. Je to vhodné z toho důvodu, že zákazníka znají a mají dostatečné schopnosti a kompetence na to úkoly vyřešit. Nicméně ne vždy jsou projekty od stávajících zákazníků. Pokud přijde požadavek od nového zákazníka na specifické implementace, je velice náročné sehnat konkrétního řešitele nejen kvůli kompetencím, ale také kvůli lokalitě, na které působí. Proto se někdy stává, že se musí čekat na konkrétního řešitele, protože je úkol tak specifický, že ho nemůže vykonat nikdo jiný.

Posledním problémovým faktorem je absence retrospektivy. Ta je vhodná pro týmy, které spolu pracují na více projektech a dobře se navzájem znají a dokáží spolu spolupracovat. Naneštěstí u tohoto projektu chybí, ale pokud by byla aplikována, byť i v malé míře, tak by to členům projektového týmu mohlo být k užitku a do budoucna by bylo možno předejít některým chybám, zde např. vyvarovat se nerespektování projektového asistenta a vnímat ho jako plnohodnotnou součást týmu.

Naopak, i když má projekt více problematických míst, je důležité vyzdvihnout i aspekty, které se povedly:

- projekt byl doručen v požadované kvalitě a ceně;
- bezproblémová komunikace RDM a projektového asistenta se zákazníkem;
- zákazník je nadále pravidelným odběratelem služeb od Společnosti XY;
- úkoly, které byly řešitelům přiděleny, byly ve většině případů včas vyřešeny a dodány bez ohledu na HW.

Ačkoliv je akceptační protokol podepsán s dodatkem, je projekt ZZS JČK IT Technologie doručen v požadované kvalitě a rozpočtu. Vždy to ale závisí na dohodě mezi zákazníkem a RDM.

I když se ve problémových oblastech objevila absence komunikační matice, je důležité podotknout, že ač komunikace uvnitř projektového týmu nebyla nejlepší, tak komunikace od RDM a projektového asistenta směrem k zákazníkovi byla na velice dobré úrovni. I to mohlo přispět k následnému faktu, že se zákazník stal pravidelným odběratelem IT služeb poskytovaných Společností XY.

S komunikační maticí souvisí další bod a to, že ačkoliv nebyly jasně ujednané odpovědnosti a respekt vůči všem členům projektového týmu, tak pokud šlo o přidělené úkoly jednotlivým řešitelům, nebyl problém s jejich realizací a byly tak včas vyřešeny a komunikovány směrem k zákazníkovi.

### **4.3 Aplikace agilních metodik na projekt ZZS JČK IT Technologie**

Další část praktického výzkumu se věnuje aplikaci vybraných agilních metodik, které jsou podrobněji popsány v literární rešerši této práce.

Problematika se vztahuje k výzkumu, zda je možné na hardwarové projekty aplikovat agilní metodiky a ošetřit tak problémová místa, která se během realizace projektu objevila. Cílem této kapitoly je zjištění, v jakém rozsahu lze využít nástroje a procesy z konkrétně zvolených agilních metodik v hardwarovém projektu ZZS JČK IT Technologie.

Rekapitulace HW projektu:

- cílem je rozmístění serverů na jednotlivé lokality dle požadavků zákazníka a s tím spojené činnosti jako je např. konfigurace switchů, virtualizace, povýšení AD (*active directory*) na nejvyšší možnou úroveň a další;
- dle projektového plánu začne souběžně instalace nových boxů hlavního firewallu, instalace VMWaru na nové servery + zařazení do virtualizačního clusteru, instalace DELL monitorů, nahrazení stávajících 6 fyzických a reinstalace dvou virtuálních serverů, souběžně s těmito úkoly je psána dokumentace hlavním architektem, a posledním úkolem, který se má udělat, je povýšení struktury AD a přenos rolí na nové servery;
- všechny tyto úkoly musí být monitorovány skrze PRTG.



### 4.3.1 Kanban

Metodika Kanban byla vyvinuta pro vývoj produktu a používá se k řízení pracovních toků a je často používána v agilních projektech. Kanban se zaměřuje na průběžnou optimalizaci procesů a zlepšování efektivity pracovního toku k dosažení výsledného produktu.

Aplikace metodiky Kanban na hardwarové projekty by mohla být užitečná pro zlepšení efektivity a průhlednosti v pracovním toku. Například použitím kanbanové tabule mohou být tak sledovány úkoly v různých fázích. Díky tomu tak mohou být rychle identifikovány a řešeny potenciální problémy nebo prodlevy.

Kanban je také zastáncem toho, že jsou úkoly přiděleny na základě kapacity týmu, namísto plánování všech úkolů dopředu. Toto může umožnit týmu rychle reagovat na změny v požadavcích nebo problémy, které vzniknou během projektu.

Možná aplikace Kanban metodiky by mohla vypadat takto:

Definovat proces: Zde je pro začátek projektu důležité definovat jasný a detailní proces, jak bude projekt probíhat – v tomto případě se musí nejprve definovat cíl a na něj následně napasovat detailní proces postupu. Cíl lze definovat následně: rozmístění serverů na jednotlivé lokality dle požadavků zákazníka a s tím spojené činnosti jako je např. konfigurace switch, virtualizace, povýšení AD na nejvyšší možnou úroveň a další činnosti. Detailní proces postupu dle projektového plánu je tedy:

- Instalace 2 ks nových boxů hlavního firewallu – znamená to, že na počítačovém systému nebo síti je nainstalován software, který má za úkol monitorovat a kontrolovat příchozí a odchozí síťový provoz. Firewall slouží jako ochrana proti neautorizovaným přístupům a útokům z internetu nebo z jiných sítí. Tato instalace vyžaduje řešitele na místě u zákazníka, jedná se tedy o instalaci on-site (na místě).
- Instalace VMWaru na nové servery + zařazení do virtualizačního clusteru – znamená, že na nových fyzických serverech je nainstalován software virtualizační platformy VMWare. Po instalaci VMWare na serverovou infrastrukturu a konfiguraci virtualizačních parametrů je možné vytvářet virtuální stroje, na kterých se dají provozovat operační systémy a aplikace tak, jako by byly na samostatných fyzických serverech. Vytvoření virtualizačního clusteru umožňuje řízení a správu virtuálních strojů jako celku a zajišťuje vyšší dostupnost služeb. Tato funkcionalita je přínosná pro organizace, které potřebují provozovat více

služeb a aplikací a současně zajišťovat jejich dostupnost a výkonnost. Tento úkol se může dělat off-site, tedy nemusí k tomu docházet u zákazníka, ale řešitel to může dělat v kanceláři na pobočce a po dokončení fyzicky u zákazníka nainstalovat.

- Instalace DELL monitorů – připojení počítačových monitorů k počítačovému systému. Musí být u zákazníka, tedy on-site.
- Nahrazení stávajících 6 fyzických a reinstalace dvou virtuálních serverů-kompletní výměna a aktualizace serverové infrastruktury organizace. U zákazníka on-site.
- Povýšení struktury AD a přenos rolí na nové servery znamená, že organizace nahrazuje starší servery, které sloužily jako řadiče domény, novými a modernějšími servery s větší kapacitou a spolehlivostí. Přenos rolí na nové servery zahrnuje přesunutí zodpovědnosti za určité funkce a úlohy v rámci AD z jednoho serveru na jiný. Lze dělat off-site.
- PRTG monitoring je softwarové řešení pro monitorování sítě a infrastruktury v reálném čase. Je užitečný v tom, že provádí monitoring v reálném čase pro minimalizování výpadku služeb a zlepšení výkonu sítě. Off-site monitoring.

Definovat úkoly: Je zapotřebí stanovení úkolů, které jsou součástí implementace IT infrastruktury (viz. obrázek 17). Dále je nutné si vymezit i rozvržení úkolů dle životního cyklu projektu. Tyto úkoly by měly být rozděleny do kategorií, jako je návrh, implementace, testování a dokončení.

<b>NÁVRH</b>	Kick-off – definování si cíle projektu, resp. cíl implementace a detailní proces postupu
<b>IMPLEMENTACE</b>	Boxy hlavního firewallu, instalace VMWaru a virtualizace, instalace DELL monitorů, nahrazení fyzických serverů a reinstalace, povýšení struktury AD
<b>TESTOVÁNÍ</b>	Otestování funkčnosti jako celku včetně PRTG monitoringu ze strany Společnosti XY
<b>DOKONČENÍ</b>	Ukončení projektu a akceptace

*Tabulka 3: Definování jednotlivých úkolů dle Kanban (zdroj: vlastní zpracování)*

Vytvoření tabule Kanban: Po stanovení úkolů, které musí být na implementaci provedeny, je vhodné vytvořit tabuli Kanban. Může být buď fyzická nebo lze využít online nástroje. Tabule se skládá ze tří sloupců: „To do“, „Doing“ a „Done“. Na začátku implementace jsou všechny úkoly ve sloupci „To do“ a jejich následné posouvání velice

záleží na předchůdcích a následovnicích. Zde lze zpozorovat první velký rozdíl – hardwarové projekty závisí na určité posloupnosti, kdežto softwarové ne. U projektu ZZS JČK IT Technologie, jak už bylo zmíněno výše, začíná souběžně několik úkolů, protože nejsou fixovány na žádného předchůdce. Proto je vhodné využití Kanban tabule pro vizualizaci projektu. Postupem dodávání jednotlivých částí IT infrastruktury se jednotlivé úkoly posouvají do sloupce „Doing“. Poslední sloupec „Done“ obsahuje úkoly, které jsou dokončeny.

Konkrétní aplikace projektu ZZS JČK IT Technologie je viz. Příloha.

Sledovat a aktualizovat: Pravidelné sledování a aktualizování výkonu a postupu je klíčem k úspěšné implementaci IT infrastruktury a pomocí metodiky Kanban je toto dosažitelné, nicméně toto již RDM dělá, jen to není na tak časté bázi.

Implementace Work-in-Progress: Nastavení maximálního počtu úkolů pro jednotlivé sloupce je v tomto případě minoritní problém, jelikož je projekt velice jednoduchý a nehrozí tak přetížení jednotlivých realizátorů.

Zlepšování procesu: Tento krok je dobré vynechat. U implementace IT infrastruktury, a to konkrétně u projektu ZZS JČK IT Technologie se nedá proces celého projektu urychlit či zefektivnit. Je to z toho důvodu, že jakékoliv instalace, konfigurace atd. se dělají už tak efektivně. Tyto práce vyžadují určité kroky a postupy, které se nedají nikterak zkracovat či vylepšovat, protože by pak nemusely splňovat funkční požadavky s dalšími kompatibilními součástmi HW.

Vyhodnocování výkonu: Ačkoliv metodika Kanban doporučuje pravidelné hodnocení týmu, tak to v tomto případě nedává smysl. Tým spolu zcela úzce nespolupracuje na jednotlivých úkolech a tato pravidelná vyhodnocování by vyžadoval i čas, který RDM a samotní realizátoři mnohdy nemají.

Tímto demonstrativním způsobem lze metodiku Kanban aplikovat na implementaci IT infrastruktury. Nicméně zde ale platí to, že je vhodné si metodiku tailorovat pro konkrétní projekt a použít, či vynechat některé kroky, které nejsou úplně vhodné pro zakomponování. Každopádně pro úspěšné aplikování Kanban metodiky platí, že ji musí celý projektový tým znát a respektovat ji.

Metodika závisí na pečlivé implementaci a pravidelné revizi. Je třeba dbát na to, aby byly úkoly správně přiřazeny k jednotlivým sloupcům, aby byl dostatečně sledován postup a aby bylo možné včas reagovat na jakékoliv potíže nebo překážky, které mohou v průběhu

implementace nastat. Zde se může vyskytnout problém v pravidelnosti sledování a aktualizace. RDM mají více projektů v realizaci, a ačkoliv se Kanban snaží předcházet tímto bodem potížím a překážkám během realizace, zaneprázdněnost RDM by mohla být v tomto ohledu problematická. V neposlední řadě Kanban nedokáže ošetřit problém s dodávkou hardwarových součástí. Co se týče napravení nerespektování projektového asistenta ze strany realizátorů, tak se naskytuje pouze nutnost vytvoření komunikační matice před začátkem samotného projektu a zdůraznění role RDM s tím, že projektový asistent je stejně důležitý jako samotný RDM. Problematika s celkovou komunikací realizátorů se metodika snaží vyřešit tím, že jasně definuje úkoly a jim přiřazené realizátory. Nicméně i tak to může být problém, protože realizátor je alokovaný na více úkolech v nespočet projektech a bude tak mnohdy zaneprázdněn a komunikace tak nemusí být optimální. S tím je nadále spojena problematika vysoké odbornosti řešitelů a znalost zákazníka – řešitelé se nedají jednoduše zaměřovat nejen kvůli svým kompetencím, ale také kvůli znalosti zákazníka a místa jejich pracoviště.

#### 4.3.2 Lean development

Metodika Lean development se zaměřuje na minimalizaci plýtvání a maximalizaci hodnoty pro zákazníka. Toho se dosahuje prostřednictvím zlepšování procesů, snižováním nadbytečného inventáře a eliminací neefektivních procesů, tzv. odpadů. Opět se ale jedná o agilní metodiku a její využití se předpokládá zejména pro vývoj softwarového produktu.

Metodika Lean development by mohla být úspěšně aplikována na implementaci IT infrastruktury s cílem zlepšit efektivitu a produktivitu práce, avšak až po samotné aplikaci je možné dojít k závěru, zda je vhodné ji aplikovat či nikoliv.

Veškeré principy a nástroje metodiky Lean development byly rozebrány v literární rešerši této práce.

Zde je několik kroků, jak lze metodiku aplikovat:

Definovat cíle: Je nutné jasně definovat cíl, kterého je potřeba dosáhnout při implementaci IT infrastruktury v projektu ZZS JČK IT Technologie. Tento cíl by měl být jasně formulovaný a dostatečně podrobný, aby bylo možné jej splnit. Jeho znění je – rozmístění serverů na jednotlivé lokality dle požadavků zákazníka a s tím spojené činnosti jako je např. instalace serverů, firewallů, konfigurace switchů, virtualizace, povýšení AD na nejvyšší možnou úroveň a další činnosti.

Identifikace zdrojů: V dalším kroku je potřeba určit si všechny zdroje, které se na projektu budou účastnit. Jedná se zejména o RDM, řešitele, projektového asistenta, obchodníka a jaký hardware bude doručován (firewally, monitory, fyzické servery). Toto jsou hlavní zdroje, které jsou v projektu ZZS JČK IT Technologie identifikovány. Následně jsou s těmito zdroji spojené práce.

Zjednodušení procesů: Ačkoliv se zjednodušováním procesů metodika Lean development odlišuje od ostatních agilních metodik, tak se tím bohužel u projektů, které se zabývají implementací IT infrastruktury, nedá řídit. Tyto projekty jsou velice technicky náročné na implementaci a pokud by se eliminovaly některé kroky, tak by bohužel finální provedení implementace nebylo kompatibilní s tím, o čem zákazník žádal a zejména by mohlo dojít k narušení funkčnosti finálního řešení. Projekt ZZS JČK IT Technologie je vhodným příkladem pro demonstrativní ukázkou, že se opravdu nedají procesy v HW projektu zjednodušit. Už i v Kanban metodice bylo nastíněno u kroku zlepšování procesů, že to nelze dělat, jelikož IT infrastruktura (PC, servery, firewally, atd) jsou komponenty, které by bez požadovaných procesů (konfigurace, instalace, reinstalace) nefungovaly a nebyly by kompatibilní s dalšími IT prvky. Tyto procesy či kroky jsou jasně stanoveny, jak se mají dělat a jsou tam i posloupnosti, které se musejí dodržovat. Jejich zjednodušení, či případná eliminace by vedla k fatálním výsledkům, a proto takový přístup nepřipadá v úvahu využívat.

Eliminace plýtvání: Zde je vhodné odstranit veškeré činnosti a procesy, které nejsou nezbytné pro dosažení cíle projektu. Může se jednat jak o zdroje, tak procesy či schůzky. Nicméně u tohoto konkrétního projektu, ZZS JČK IT Technologie, nebyly shledány žádné zbytečné procesy, zdroje či schůzky, které by mohly být vypuštěny a projekt by tím mohl být časově efektivnější. Viz proces zjednodušování.

Sledování a revidování: Opět podobné jako u Kanban metodiky. Pravidelné sledování a revidování průběhu implementace IT infrastruktury slouží k včasnému odhalení překážek a nejasností, které mohou v projektu nastat. V projektu ZZS JČK IT Technologie provádí RDM monitoring projektu na týdenní bázi, nicméně pravidelné sledování či revidování, jak to Lean metodika popisuje, by v tomto případě k ničemu nevedlo. Překážky či problémy se u implementací IT infrastruktury projevují během instalací či následného monitoringu a do této doby nedokáže ani nejzkušenější řešitel říci, co všechno se může stát a jak tomu případně předcházet.

Procesy, které nebyly zahrnuty v aplikaci na konkrétní projekt, ale kterým se metodika Lean development dále věnuje, jsou např. zesilování učení, rozhodování se co nejpozději, doručovat co nejrychleji, respektovat tým a soustředit se na celkový výsledek. Zde je důležité zdůraznit, že téměř většina nezahrnutých procesů v aplikaci na konkrétní projekt, viz. výše, nedávají ani z hlediska IT infrastruktury smysl. Rozhodování se co nejpozději je pro hardwarové využití naprostý nesmysl z důvodu, že většina procesů musí být definována dopředu a s tím související nákup potřebného HW či případné před-práce. S tím souvisí i doručování co nejrychleji. Zákazníci u HW projektů chtějí, aby byl projekt dodán jako celek a hlavně, aby tento celek fungoval, nepotřebují průběžné dodávky. Zesilování učení je taktéž zbytečný proces, protože většina zákazníků si ani neuvědomuje veškeré náležitosti spojené s implementací IT infrastruktury a mnohdy je jednodušší z pohledu RDM i samotných realizátorů, že procesu realizace zákazník nerozumí a nevyžaduje se tak nadbytečná interakce ze strany zákazníka.

Jediné, co je vhodné zahrnout a zahrnuto nebylo, je respektování lidí v týmu, tzn. všech členů. Jak již bylo zmíněno, v případě projektu ZZS JČK IT Technologie se objevil problém s nerespektováním projektového asistenta. Tento aspekt by tedy proces řízení projektu v tomto ohledu zlepšil.

Aplikace metodiky Lean development na implementaci IT infrastruktury není vhodné z důvodu zeštíhlování procesů či jejich eliminací. To bohužel u implementace IT infrastruktury nelze dělat, jelikož by výsledná dodávka nefungovala tak, jak má. V porovnání s Kanban metodikou je Lean slabší pro využití u HW projektů, které jsou zaměřeny na implementaci IT infrastruktury a je zřejmé, že je více vhodný pro projekty, u kterých je žádoucí eliminace některých procesů. Nicméně slabá místa projektu, která byla na konkrétním projektu ZZS JČK IT Technologie objevena (problémy s komunikací) lze ošetřit Lean metodikou, protože obsahuje princip respektování lidí v týmu. Avšak problémy jako dodávky HW, nezaměnitelnost realizátorů či absence retrospektivy vyřešit nelze.

#### 4.3.3 Extrémní programování

Agilní metodika extrémního programování je určena pro vývoj softwaru. Je vhodná pro menší vývojové týmy. Metodika staví na 4 hodnotách, a to na komunikaci, jednoduchosti, zpětné vazbě a odvaze. Dále se odvolává i na principy, jak již ovšem bylo uvedeno v kapitole této problematice věnované. Hlavní charakteristikou metodiky XP je

fakt, že veškeré postupy a principy převádí do extrému, proto může být pro HW projekty zaměřené na implementaci IT infrastruktury nevhodná či velice těžko aplikovatelná.

Aplikace agilní metodiky XP na projekt ZZS JČK IT Technologie v tomto případě nedává smysl, z důvodu principů, které jsou založeny primárně pro programování softwaru a menších vývojových týmech, které jsou na sebe zvyklé a umí spolu spolupracovat. Dále se mezi principy řadí malé verze, metafora, jednoduchý návrh, testování, refaktorizace a párové programování, nepřetržitá integrace či zákazník na pracovišti. Veškeré tyto principy jsou neuplatnitelné pro projekt ZZS JČK IT Technologie, a to z mnoha důvodů. Jedním z těchto důvodů je fakt, že všechny výše zmíněné principy jsou určeny pro vývoj softwaru (malé verze, jednoduchý návrh, testování, refaktorizace a párové programování). U projektu ZZS JČK IT Technologie nelze dělat malé verze, které slouží k uvolňování nových verzí výsledného produktu, nelze testovat po každém uvolnění nové verze, nelze dělat jednoduché návrhy, protože typ projektu zaměřený na implementaci IT infrastruktury musí obsahovat vše hned od začátku (kde budou umístěny firewally, servery, kde jsou zásuvky, jaké kabely pro připojování budou potřeba atd), taktéž refaktorizace nedává smysl, protože u toho typu projektů se nevyskytují nechtěné duplicity. A jako posledním je zmíněné párové programování, i samotný název se nehodí pro HW projekt ZZS JČK IT Technologie nejen kvůli programování, ale také kvůli hlavní myšlence XP (veškeré programování je prováděno ve dvou lidech), jak už bylo popsáno v předešlých aplikacích agilních metodik, některé úkoly z projektu ZZS JČK IT Technologie jsou tak specifické, že je pro ně vhodný pouze jeden řešitel s danou kompetencí a není možné je přiřadit někomu jinému.

XP uznává 4 hodnoty, které jsou založeny na komunikaci, jednoduchosti, zpětné vazbě a odvaze. Vzhledem k definovaným slabým místům projektu ZZS JČK IT Technologie by bylo žádoucí si z této metodiky vzít hodnotu komunikace a mohl by se tak vyřešit problém s projektovým asistentem, kde by byl zaveden jak komunikační kanál, skrze který by procházela veškerá komunikace k projektu od všech realizátorů + RDM + projektového asistenta, ale i vytvoření komunikační matice by jistě pomohlo.

Závěrem lze k metodice XP dodat, že ji opravdu nedává smysl využívat na technické IT infrastrukturální projekty kvůli častým iteracím, testování či programování. Tato metodika je oproti již popsanému Kanbanu či Lean developmentu více zaměřené na software. Ani za pomoci tailorování by se XP nedala využít. Jinými slovy, pro uplatnění metodiky XP pro projekt ZZS JČK IT Technologie by se muselo téměř od všech procesů, principů a hodnot upustit a tím pádem by vznikla úplně nová metodika.

#### 4.3.4 Crystal metodiky

Mezi další agilní metodiky využitě pro aplikaci na konkrétní HW projekt jsou tzv. crystal metodiky, nicméně pro tento účel aplikace je zvolena metodika crystal clear z důvodu, že je projekt velice jednoduchý a projektový tým se skládá maximálně z 6 lidí.

Všechny metodiky crystal jsou založeny na principech, jež vyzdvihují blízkou komunikaci, časté dodávky, vylepšování a další. Co je ale předností crystal metodik, tak to, že nedefinují praktiky, ale spíše strategie či techniky, jak k projektům přistupovat a řídit je. Pak je na samotném týmu, zda zvolí svou cestu vývoje produktu, či se budou inspirovat z metodik rodiny crystal.

Možná aplikace metodiky crystal clear na projekt ZZS JČK IT Technologie:

Sestavení týmu: V počátku projektu je nutné dle metodiky stanovit počet lidí, kteří se budou projektu účastnit. Jedná se zejména o RDM, řešitele (architekt, specialista), projektový asistenta a osoba zastupující zákazníka. Tým se skládá z maximálně 6 lidí, proto je vhodné použít metodiku crystal clear a také proto, že je projekt poměrně jednoduchý.

Mapování: Zde je žádoucí definovat jasné a srozumitelné požadavky na výsledek toho, co zákazník chce. Pro připomenutí se jedná o:

- cílem je rozmístění serverů na jednotlivé lokality dle požadavků zákazníka a s tím spojené činnosti jako je např. konfigurace switchů, virtualizace, povýšení AD (*active directory*) na nejvyšší možnou úroveň a další;
- dle projektového plánu začne souběžně instalace nových boxů hlavního firewallu, instalace VMWaru na nové severny + zařazení do virtualizačního clusteru, instalace DELL monitorů, nahrazení stávajících 6 fyzických a reinstalace dvou virtuálních serverů, souběžně s těmito úkoly je psána dokumentace hlavním architektem, a jako posledním úkolem, který se má udělat je povýšení struktury AD a přenos rolí na nové servery;
- všechny tyto zmíněné úkoly musí být monitorovány skrze PRTG.

Metodika crystal clear doporučuje průběžné kontrolování a následné upravování nesrovnalostí (opět z důvodu, že je doporučena pro vývoj software). V případě HW projektu ZZS JČK IT Technologie lze dělat průběžnou kontrolu, ale nelze přijímat požadavky na změny ze strany zákazníka. Cíl projektu, respektive výsledný produkt, je u HW infrastruktury přesně dán a nelze jej měnit během implementace.



Sestavení plánu projektu: Po sestavení konkrétního týmu realizátorů a fázi mapování je na řadě sestavení plánu projektu:

- Instalace 2 ks nových boxů hlavního firewallu.
- Instalace VMWaru na nové severy + zařazení do virtualizačního clusteru.
- Instalace DELL monitorů.
- Nahrazení stávajících 6 fyzických a reinstalace dvou virtuálních serverů.
- Povýšení struktury AD a přenos rolí na nové severy.
- PRTG monitoring

Podrobnější popis plánu projektu, resp. toho, co veškeré úkoly zahrnují za práce, je popsáno v kapitole 4.3.1 Kanban. Crystal clear metodika je opět spíše pro vývoj softwaru a očekává se, že se celý tým bude podílet na sestavení projektového plánu. V aplikaci na konkrétní projekt to lze také využít, protože RDM nemá všechny znalosti a informace, co bude daná implementace, za jaké práce vyžadovat, další výhodou je v definování si pracností u jednotlivých úkolů. V neposlední řadě se RDM doptá řešitelů u konkrétních úkolů, jaká pracnost tomu odpovídá.

Zajištění komunikace: Crystal clear metodika vyžaduje silný důraz na komunikaci, a to jak mezi členy týmu, tak i mezi zákazníkem a týmem. Tento krok by mohl velice pomoci při vyřešení problému s nerespektováním projektového asistenta a zároveň celkovou komunikaci mezi členy týmu. Co se týče zákazníka, tak ten je primárně v kontaktu s RDM, popř. může být zákazník přeměrován napřímo na realizátora pro podrobnější informace.

Vývoj inkrementu: U tohoto kroku, jež je doporučován samotnou metodikou, je lehce zavádějící jeho využití pro HW infrastrukturální projekt. V tomto případě se žádný HW nevyrobí, jen je implementován na dané lokality (firewally, severy, monitory atd). Proto je možné vnímat pojem inkrement například ve smyslu dokončení instalace firewallů a serverů do racků. Nicméně z pohledu metodiky zaměřené na vývoj softwaru to má k jejím doporučeným postupům daleko.

Testování funkcionalit: Tento krok je následníkem vývoje inkrementu a očekává se, že se právě daný inkrement bude testovat na jeho funkcionality a zda je splňuje. V konkrétním případě HW infrastrukturálního projektu by se otestovalo, zda zapojení firewallů funguje a tím pádem je zabráněno neautorizovaným přístupům a útokům zvenčí. Také zapojení serverů do racku je možné otestovat, další aspekty však nikoli. Otestování plné funkčnosti a nepřetržitého fungování celého systému je u tohoto projektu možné až po úplném dokončení všech úkolů s potřebným HW. Dell monitory a jejich zpožděná dodávka

na to ale nemá žádný vliv, jelikož má zákazník jiné monitory pro dočasné překlenutí doby, než dorazí DELL monitory.

Integrace inkrementů: V případě projektu ZZS JČK IT Technologie je to naprosto zbytečný krok a lze jej vypustit. Je to dáno tím, že inkrementy, které se používají ve vývoji softwaru (inkrement = část z celkového produktu softwaru) se nevyskytují u projektů, které jsou založeny na implementaci IT infrastruktury. Tyto projekty fungují pouze jako celek, kdy je veškerý HW dodán a nainstalován a práce s tím spojené (instalace, konfigurace, virtualizace) jsou hotové.

Zajištění podpory: Jak už bylo popsáno v ukončovací fázi projektu ZZS JČK IT Technologie, projekt byl neoficiálně dodán jako úspěšný a jen se čeká na DELL monitory. Každopádně i v tomto případě je zajištěna další podpora ze strany Společnosti XY ve vztahu nadáleho poskytování servisních služeb zákazníkovi.

Hodnocení a zlepšování: Metodika crystal clear doporučuje využití denních schůzek a přírůstkového grafu pro sledování postupu na projektu, nicméně ve Společnosti XY a projektech, jimiž se zabývá, to nedává smysl. IT infrastrukturální projekty nefungují na stejné bázi, jako vývoj softwaru, tzn. ne každý den se eviduje posun v projektu. U projektu ZZS JČK IT Technologie by tak nebyly žádoucí denní schůzky, kde by se řešilo, jaký řešitel pracuje, na kterém úkolu, a to z důvodu daných posloupností jak časových, tak i instalačních. Nedává ani smysl tvořit přírůstkový graf, ale pokud ano, musely by v něm být zaneseny veškeré práce na jednotlivých úkolech, př. natažení kabelů, rozmístění serverů na jednotlivé lokality, servery nainstalovány v racku na všech lokalitách, virtualizace, přenos rolí atd. Přírůstkový graf by pak obsahoval příliš mnoho informací a byl by nejen zbytečný, ale i poměrně zmatečný. Pro tento případ slouží v interním systému jen vykazování práce jednotlivých realizátorů, kde popíšu, co daný den udělali, případně pak komunikace realizátorů s RDM dá jasný pohled, v jaké fázi odpracovaných prací se projekt nachází.

Shrnutí pro využití metodiky crystal clear na projekt ZZS JČK IT Technologie je pro tento výzkum pozitivní. Je to z důvodu, že metodiku crystal clear lze využít pro HW projekt s podmínkou, jež sama metodika vyzdvihuje. Smyslem crystal metodik je úprava postupu a strategií pro potřebu u konkrétního projektu. Nicméně obsahuje i kroky, pomocí kterých lze postupovat v případě, kdy si vývojový tým neví rady.

Co se týče problémů, se kterými se projekt ZZS JČK IT Technologie potýkal během realizace, tak by využití crystal clear metodiky mohlo vést k eliminaci nerespektování všech

členů týmu a zároveň i zlepšení veškeré komunikace s realizátory. Zde ale vyvstává fakt, že jsou realizátoři alokováni i na jiných projektech, s čímž metodika crystal clear zcela úplně nepracuje (jeden tým má jeden projekt).

Dalšími problémy, jimiž projekt disponoval (viz. RDM řídí více projektů najednou, dodávky HW jsou nepředvídatelné, vysoká odbornost řešitelů a absence retrospektivy) by se využitím crystal clear metodiky zcela nevyřešily, nicméně tyto problémy by nejspíše nevyřešila žádná metodika. Za předpokladu, kdy by jeden RDM měl pouze jeden projekt, by muselo dojít k přenastavení ve Společnosti XY. Bohužel dodávky HW se nedají ošetřit žádným způsobem. Obdobný problém souvisí s vysokou odborností řešitelů, nelze je zaměňovat a absence retrospektivy by metodika crystal clear neošetřila, ač se snaží využívat denní schůzky a přírůstkové grafy, opět by to pro tento případ projektu nedávalo smysl.

Závěrem lze dodat, že je možnost uplatnění této metodiky na HW infrastrukturální projekt, ale musela by se přizpůsobit potřebám daného projektu. Nicméně by dokázala ošetřit problémová místa projektu, jež se vyskytla, ale taktéž (jako předešlé metodiky) by nedokázala ošetřit problémy, které není schopna ošetřit ani interní metodika Společnosti XY.

#### 4.3.5 SCRUM

Poslední zvolenou agilní metodikou pro projekt ZZS JČK IT Technologie, je metodika SCRUM. Její využití je vhodné spíše pro vývoj softwaru, ale může být aplikována i na jiné typy projektů. Metodika se zaměřuje na iterativní a inkrementální vývoj produktu, přičemž celý tým spolupracuje na dosažení společného cíle. SCRUM vyniká svou flexibilitou a schopností rychle reagovat na změny a požadavky zákazníka. Skládá se z několika klíčových prvků, jako jsou plánování sprintů, daily scrum, sprint review a retrospektiva sprintu. Tyto prvky zajišťují, že tým pracuje soustředěně na dokončení úkolů během daného sprintu a současně se průběžně učí a zlepšuje své procesy.

Možná aplikace SCRUM metodiky na projekt ZZS JČK IT Technologie:

Definování týmu: Ačkoliv to metodika sama nedefinuje, je vhodné začít se sestavením týmu, který se bude na implementaci projektu podílet a bude se řídit metodikou SCRUM. Definování týmu bylo na projektu ZZS JČK IT Technologie zvoleno, protože jednotliví řešitelé spolu nepracují na všech projektech, ale RDM je vybírají dle jejich kompetencí pro různé projekty. Definovaný tým se skládá z RDM, řešitelů (hlavní architekt,

specialisté), projektového asistenta a osoby zastupující zákazníka. Tým se skládá z maximálně 6 lidí.

**Product backlog:** Výčet jednotlivých položek, které mají být v průběhu projektu dokončeny a dodány, je v productovém backlogu, viz Tabulka 4. Nedefinuje je product owner, protože jimi Společnost XY nedisponuje, ale má to na starosti RDM s pomocí řešitelů, kteří se na projektu budou podílet.

<p><b>Boxy hlavního firewallu</b>  Dovoz na lokality zákazníkem  Zapojení a instalace řešitelem Společnosti XY</p>
<p><b>Instalace VMWaru a virtualizace</b>  Instalace software virtualizační platformy VMWare  Konfigurace virtualizačních parametrů  Vytvoření virtualizačního clusteru</p>
<p><b>Instalace DELL monitorů</b>  Dovoz na lokalitu řešitelem Společnosti XY a následné zapojení</p>
<p><b>Nahrazení fyzických serverů a reinstalace</b>  Kompletní výměna a aktualizace serverové infrastruktury</p>
<p><b>Povýšení struktury AD a přenos rolí</b>  Nahrazení starých serverů novými  Přenos rolí na nové servery – přesunutí zodpovědnosti za určité funkce a úlohy v rámci AD z jednoho serveru na jiný</p>
<p><b>PRTG monitoring</b>  Instalace software pro monitorování v reálném čase</p>

Tabulka 4: Product backlog ZZS JČK IT Technologie (zdroj: vlastní zpracování)

**Sprint planning:** Dalším krokem po sestavení product backlogu je sprint planning. Z Tabulky 4 si lze povšimnout, že je tento projekt opravdu jednoduchý a není tam mnoho úkolů, které se musejí splnit. Nicméně v této fázi je potřebné určit, co se bude v 1. sprintu realizovat za úkoly.

Sprint backlog pro 1. sprint:

Boxy hlavního firewallu
Instalace VMWaru a virtualizace
Instalace DELL monitorů
Nahrazení fyzických serverů a reinstalace

Tabulka 5: Backlog pro 1. sprint (zdroj: vlastní zpracování)

Tyto úkoly, které obsahuje Tabulka 5, si definoval sám tým a rozhodl o nich, že se na nich bude pracovat během 1. sprintu.

Sprint: Ačkoliv teorie doporučuje, aby jednotlivé sprinty trvaly kolem 2 týdnů, tak se to v případě HW projektů nedá většinou dodržovat. Hraje v tom roli dodávka HW, alokace jednotlivých realizátorů a součinnost zákazníka, popř. dnů, kde je možné provést “odstávky“. V konkrétním případě projektu ZZS JČK IT Technologie 1. sprint potrvá minimálně 1 měsíc, a i tak se nepodaří dodělat všechny požadované úkoly. Celkově vychází projekt na 3 sprinty, nicméně u 3. sprintu není možné jasně definovat ukončení, respektive délky trvání, protože dodávka HW je nepředvídatelná.

Další sprint backlog pro nadcházející sprinty viz. Příloha.

Daily Scrum: Daily Scrum je krátké setkání, které se koná každý den během sprintu a cílem je zjištění stavu, co bylo dokončeno, na čem se pracuje a pracovat bude. U projektu ZZS JČK IT Technologie to nedává smysl stejně tak, jako u předešlých metodik, které denní setkání doporučují. Je to z důvodu, že ne každý den všichni realizátoři mají něco za úkol, a ne vždy se na něčem pracuje. Důvodů je hned několik – realizátor pracuje na jiném projektu, nebo se čeká na zákazníka, nebo na dodávku HW.

Sprint review: Po dokončení každého sprintu by se měl projektový tým setkat a odprezentovat dokončenou práci zákazníkovi a dostat tak zpětnou vazbu k práci, která je hotová. U projektu ZZS JČK IT Technologie by využití sprint review nedávalo smysl. Ačkoliv by se práce rozdělila do jednotlivých sprintů, tak by jejich dokončení nepotřebovalo souhlas od zákazníka, protože veškerým pracím, které se udělaly v daném sprintu zákazník nemusí rozumět. Zákazníkův požadavek je dodání celého fungujícího celku a průběžné review by byly zbytečné jak pro zákazníka, tak pro projektový tým. Ale tímto se nevylučují průběžné schůzky jak s projektovým týmem, tak se zákazníkem, jen by to probíhalo jinou formou.

Retrospektiva sprintu: Retrospektiva sprintu je setkání týmu, kde se hodnotí, co se podařilo, co se nepodařilo, co bylo obtížné a jakým způsobem by se to mohlo zlepšit. Retrospektiva je vhodným nástrojem pro týmy, které jsou zvyklé spolu spolupracovat na více projektech a dostatečně znají své dovednosti a návyky při práci. Bohužel ve Společnosti XY se realizátoři jen zřídka setkávají na více projektech, takže nemají žádné společné návyky v rámci spolupráce. Nicméně absence retrospektivy je definována jako jeden z problémů, které nedokáže interní metodika ošetřit. Zakomponování retrospektivy sprintu do projektu ZZS JČK IT Technologie je zajímavým podnětem pro další vylepšení řízení projektu. Spočívalo by to tedy nejen v rozdělení prací do jednotlivých sprintů, ale zároveň i seznámení projektového týmu s těmito pojmy a chování s tím související. Rozhodně by retrospektiva přinesla ucelenější pohled pro RDM, co se ve sprintu odpracovalo, či co ještě

chybí a díky tomu by realizátoři tyto informace měli k dispozici. Tím by se zamezilo přebytné komunikaci mezi nimi a RDM.

Závěrem vyplývajícím z aplikace agilní metodiky SCRUM na projekt ZZS JČK IT Technologie je skutečnost, že ačkoliv se metodika zaměřuje na inkrementální a iterativní vývoj produktu a tedy projekty, které jsou velice flexibilní, tak tato praktická aplikace dokazuje, že se hodí i na HW projekty, ač s určitým tailorováním. Jediný bod, který by se pro implementaci HW projektů musel z metodiky SCRUM vyřadit, je daily scrum, protože to nedává z časového hlediska pro žádného člena týmu smysl.

Nicméně metodika SCRUM dokáže vyřešit vyvstálé problémy, kterými projekt ZZS JČK IT Technologie disponuje, tedy převážně problémy s komunikací. Vzhledem k tomu, že metodika požaduje od týmu úzkou spolupráci na definování si jednotlivých sprintů neboli sprint backlogů, tak zde odpadá nutnost role projektového asistenta. Tím by se tedy nerespektování projektového asistenta úplně vyřadilo a problém s celkovou komunikací s realizátory prakticky také. Realizátoři by si řídili jednotlivé úkoly téměř sami a RDM by jen dával informace, kdy je možné začít s jakou částí implementace. Nebo je zde možnost RDM roli více zredukovat a mohl by tak RDM vystupovat pouze jako podpora pro tým, když by na začátku projektu jasně definoval komunikační kanál a sami realizátoři by si přebírali odpovědnosti za posloupnost úkolů. Nicméně tento krok by vyžadoval změnu v interní metodice a v celkovém přístupu jednotlivých realizátorů k práci a komunikaci.

Co se týče problému absence retrospektivy, tak by byla vhodná pro využívání u HW projektů, ale musela by se taktéž zimplementovat do interní metodiky Společnosti XY a být důležitým bodem u vyhodnocování projektu. Aby ale retrospektiva měla nějaký význam do budoucna, je potřeba, aby ji RDM zakomponovali do svých projektů. To bude vyžadovat velkou součinnost u realizátorů, nicméně do budoucna by to mohlo přinést zlepšení v podobě sociálních a profesních návyků.

Vysoká odbornost řešitelů by byla klíčovým bodem, kdy by se RDM musel hodně zaměřit na vhodnost sestaveného týmu pro konkrétního zákazníka a zároveň by se musel zabývat i časovými možnostmi jednotlivých realizátorů.

Dodávky HW se nedají opět ošetřit ani touto metodikou a zůstávají tak u všech metodik nejrizikovější a neošetřenou součástí projektu.

## 4.4 Typy projektů, které se agilním přístupem nedají ošetřit

Agilní přístup řízení projektů je vhodný pro projekty, které nemají jasně definovaný cíl a zároveň je vývojový tým velice flexibilní a je ochotný rychle reagovat na změny, které během projektu přichází. Nicméně předešlý výzkum ukazuje, že je tento přístup vhodný i pro HW projekty, které mají jasně definovaný cíl.

Avšak i agilní vedení projektů má svá omezení, a proto není vhodné pro všechny typy projektů. Jedná se zejména o projekty, které mají pevné požadavky již od samého začátku plánování a také pro projekty, které mají vysoký stupeň regulace a compliance.

### Projekty s pevnými požadavky

Projekty s pevně stanovenými požadavky mají předem stanovené výstupy a výsledky, které musí být splněny v souladu s plánem. Tyto projekty se často vyskytují v odvětvích s vysokými specifikacemi a standardy, jako je stavebnictví, zdravotnická technika nebo letecký průmysl, kde je splnění těchto požadavků vyžadováno zákonem a bezpečnostními předpisy.

Použití agilních přístupů u projektů s pevně stanovenými potřebami může být náročné, velice rizikové a dost možná i nelogické. Bylo by velice obtížné začlenit změny požadavků nebo plánů, aniž by byl ohrožen celý projekt. Projektový manažer a tým u těchto typů projektů například nemůže jednoduše změnit specifikace či zahrnout další úkoly, pokud projekt musí dodržet stanovenou právní normu. Podobně platí, že jakmile je jednou stanoven termín realizace projektu, nelze jej snadno posunout kvůli změně specifikací projektu.

Za těchto okolností je tedy nezbytné zvolit přístup k řízení projektu, který je vhodný pro projekty s pevnými požadavky. Pro projekty se stanovenými požadavky je vhodnější klasický (waterfallový) přístup k řízení projektů.

### Projekty s vysokým stupněm regulace a compliance

Projekty, které se musí řídit přísnými směrnici a normami stanovenými právními předpisy, vládními orgány nebo mezinárodními regulačními organizacemi, jsou považovány za vysoce regulované a náročné. Tyto projekty se často vyskytují v oblastech jako farmaceutický průmysl, bankovníctví, zdravotnictví, automobilový průmysl a letectví, kde jsou přísná pravidla a normy důležité pro bezpečnost a ochranu veřejného zdraví a životů.

Tyto typy projektů mohou být náročné na koordinaci, protože musí dodržovat přísná pravidla a normy a být také pečlivě monitorovány, aby byl zajištěn soulad s regulacemi.

Každý krok v projektu musí být pečlivě zdokumentován a zaznamenán tak, aby bylo možné prokázat, že projekt dodržuje požadované standardy.

Vzhledem k tomu, že projekty s vysokým stupněm regulace a compliance jsou náročné na řízení a často mají vysokou míru komplexity, je důležité mít v týmu projektu odborníky v oblasti regulací a normativních požadavků. Tito odborníci by měli být schopni porozumět požadavkům regulace a identifikovat potenciální rizika a problémy, které mohou být spojeny s dodržováním přísných standardů a předpisů.

I v tomto případě nedává smysl využít agilního řízení, nejen, že by to bylo moc riskantní, ale také by se s největší pravděpodobností nedodržovaly agilní praktiky, postupy či nástroje.



## 5. Výsledky a diskuse

Praktická část diplomové práce přinesla celou řadu zajímavých zjištění, která byla získána za pomoci komparace interní metodiky Společnosti XY a PRINCE2, a rovněž pomocí analýzy zjišťující, jak moc se RDM při řízení projektu řídí interní metodikou. Díky podrobně rozebranému HW projektu ZZS JČK IT Technologie pomocí interní metodiky Společnosti XY a definování si tak slabých míst v projektu, která se během jeho realizace objevila, se pak mohl samotný výzkum nadále věnovat problematice využitelnosti agilních metodik na hardwarové projekty za účelem eliminace zmíněných problémových míst.

Z prvního výzkumu, který se týkal interní metodiky a jejího porovnání s metodikou PRINCE2, vyvstalo, že ačkoliv se interní metodika odkazuje na PRINCE2 a RDM mají povinnou minimální úroveň certifikace (PRINCE2 Foundation), nelze jednoznačně určit, do jaké míry je ve Společnosti XY metodika PRINCE2 využívána ve smyslu procesů či nástrojů. Nicméně z následného, a tudíž detailnějšího porovnání se došlo k závěru, že obě metodiky doporučují pro každý projekt zvolit tailorování, tzn. přizpůsobení si metodiky pro konkrétní projekt.

Pro hlavní výzkum bylo nutné stanovení nedostatků čili problémů, které se během realizace projektu ZZS JČK IT Technologie objevily. Jednalo se zejména o problémy s komunikací směrem k řešitelům jednotlivých úkolů, nerespektování projektového asistenta, absence retrospektivy, velké vytíženosti RDM na vícero projektech najednou, vysoké odbornosti řešitelů a s tím související nezaměnitelnost s nikým jiným, a v neposlední řadě pozdní dodávky HW.

Na základě těchto problémových míst bylo navrženo řešení pomocí aplikace zvolených agilních metodik. Do těchto metodik byly zařazeny Kanban, Lean development, Extrémní programování, SCRUM a Crystal metodiky. Ačkoliv jsou všechny tyto metodiky vhodné pro dynamické prostředí projektů a také pro iterativní a inkrementální vývoje (nejvíce software), tak bylo v této diplomové práci dokázáno, že i tak je zde možnost využití agility u projektů, které mají pevně stanovený scope. Každopádně je nutné zmínit, jak k využití konkrétní agilní metodiky na HW projekt je vhodné přistupovat.

Jako první metodika, která byla využita pro aplikace na projekt ZZS JČK IT Technologie, je Kanban. Ta je vhodná zejména pro vizualizaci, ale je aplikovatelná na

všechny druhy projektů a určitým způsobem by s její pomocí byly vyřešeny problémy, jimiž projekt disponuje. Komunikace s realizátory by byla ošetřena komunikační maticí, do toho spadá i problém s projektovým asistentem. Každopádně ale metodika doporučuje procesy, které jsou zbytečné pro konkrétní projekt, např. zlepšování procesu. Závěrem k metodice lze říci, že je vhodná k aplikaci na HW projekty, ale je nutné ji tailorovat.

Další metodika, jež byla aplikována, byl Lean development. Tato metodika se specializuje zejména na zeštíhlování procesů a eliminaci tzv. odpadu. U tohoto konkrétního HW projektu je naprosto zbytečné využívat Lean přístup z toho důvodu, že u HW projektů jsou většinou dost pevně dané postupy a jakákoliv eliminace čili zeštíhlení by vedla k fatálním dopadům. S tím souvisejí i další přístupy, které nejsou vhodné pro aplikování na tento projekt, a to dodávky co nejrychleji a rozhodovat se co nejpozději. Jediné pozitivum této metodiky je přístup, který se věnuje respektování lidí v týmu, jež pomáhá k lepší a efektivnější komunikaci mezi všemi členy. Oproti metodice Kanban je Lean slabší a nedá se z jeho procesů či přístupů mnoho využít. Ostatní problémy jsou podobně řešitelné jako u metodiky Kanban.

Co se týče metodiky Extrémní programování, tak ta byla označena jako absolutně nevhodná pro HW projektu ZZS JČK IT Technologie. Je to hlavně z toho důvodu, že už samotný název vypovídá jistý směr pro její aplikaci. Na HW projekt není vhodná zejména kvůli jejím procesům a principům, na které se odvolává, viz kontinuální integrace, refactoring, programování či testování. To u HW projektu ztrácí smysl.

U Crystal metodik byla zvolena metodika Crystal clear, protože je vhodná pro menší týmy, což tento projekt splňuje. Ačkoliv se na první pohled zdá, že je neaplikovatelná na HW projekty, tak se ukázala jako vhodná metodika. Znamenalo by to ale určité tailorování, což sama metodika doporučuje. Neudává žádné striktní postupy, jen doporučení, jak postupovat. Díky kroku sestavení týmu a mapování v praktické aplikaci metodiky Crystal clear, by mohl vypadnout problém s komunikací s nerespektováním projektového asistenta. Dále i samotný krok zajištění komunikace by toto tvrzení potvrdil. Další vyvstálé problémy by bohužel metodika Crystal clear opět vyřešit nedokázala, stejně tak jako předešlé metodiky.

Poslední metodikou, jež byla prakticky aplikována na projekt ZZS JČK IT Technologie, je metodika SCRUM. Díky samotnému výzkumu se tato metodika ukázala jako nejvhodnější pro implementaci IT infrastruktury, což je zajímavý fakt, vzhledem k tomu, že SCRUM metodika je zaměřena na velice flexibilní prostředí s inkrementálním a iterativním vývojem. Opět by se zde vyřešil problém s komunikací a respektem

k projektovému asistentovi. Jednotlivé implementace by byly vhodně rozvrhnuty do sprintů, celý tým by věděl, na čem se aktuálně pracuje, stejně jako u Kanban metodiky. Jediné, co by nebylo vhodné z této metodiky zahrnout, jsou tzv. daily scrum, protože by na tento typ činnosti nikdo z projektového týmu neměl čas a u HW projektů tohoto typu by to nedávalo ani smysl. Dalším bodem výzkumu u SCRUM metodiky byla sprint retrospective, to se ukázalo v samotném projektu jako problémové místo a s pomocí využití ať už sprint retrospective nebo retrospektivy na konci projektu by se mohlo do budoucna předcházet problémům, které se v tomto projektu vyskytly a mohly by se tak využít u dalších projektů, které jsou zaměřeny na IT infrastrukturu.

Celkové shrnutí k aplikaci agilních metodik na hardwarové projekty je, že je zde možnost využití jednotlivých metodik pro zlepšení procesů a efektivity na projektech zabývajících se implementací IT infrastruktury, ale nesmí se opomíjet tailoring, aby celek dával logiku. Smyslem agility v této práci, a i na základě výzkumu slabých míst v projektu ZZS JČK IT Technologie je to, že agilita je zaměřená na soft skills, a to se u konkrétního HW projektu ukázalo jako nejvíce slabé místo, které nebylo ošetřeno, ale díky těmto poznatkům se pro nadcházející projekty mohou RDM ve Společnosti XY snadno vyvarovat. Nicméně závěrečná práce obsahuje i kapitolu, která se věnuje projektům, které se v žádném případě nedají vést agilně z důvodu striktních požadavků, regulací či compliance.

Doporučením této práce pro Společnost XY je zamyšlení se nad stávající interní metodikou a návrhnutí možných vylepšení včetně soft skills nástrojů a procesů pro usnadnění práce RDM a zároveň zvýšení efektivity u samotných projektů. Díky tailoringu jednotlivých agilních metodik na HW projekty by se tak mohlo předejít mnohým zbytečným komplikacím. Nejvhodnějším řešením by tak mohlo být tzv. hybridní řízení projektů, kdy by se do jasně definovaných kroků postupu projektu vložili agilní procesy a nástroje a vytvořil by se tím tak efektivnější styl vedení projektů. Dále by tato závěrečná práce mohla pokračovat v dalším výzkumu, a to jak u samotných RDM, tak u realizátorů Společnosti XY. Výzkum by nejspíše spočíval v polo-strukturovaných rozhovorech, kde by se tak zjistily výsledky, jaký má obecně Společnost XY postoj k řízení projektů a jeho zlepšování a posouvání se s dobou. Pokud by dotazovaní byli nakloněni k agilitě, rozhodně by bylo vhodné zapojení agilních koučů a vybudování týmů, které spolu budou pracovat na více než jednom projektu.

## 6. Závěr

Na závěr této diplomové práce lze konstatovat, že je projektové řízení nezbytným prvkem každé organizace, která se zabývá projekty jakéhokoliv typu. Pokud budou projekty zaměřené na vývoj aplikací, tedy softwaru, je důležité, aby bylo v dané firmě agilní centrum. Jestliže to budou naopak projekty typické pro waterfallové řízení, je vhodné ve firmě založit projektovou kancelář, která se poté bude rozpadat na další úrovně. Nicméně oba tyto přístupy vedení projektů jsou důležité pro správné a efektivní vedení projektů.

Cíl této práce, tedy zhodnocení, zda se dají agilní metodiky využít pro HW projekty, byl splněn díky výzkumu založenému na simulacích HW projektu ZZS JČK IT Technologie a vybraných agilních metodik.

První část práce se věnovala teoretickým poznatkům z oblasti waterfallového a agilního přístupu řízení projektů, kde byly nadále podrobněji rozepsány vybrané agilní metodiky SCRUM, Extrémní programování, Kanban, Lean development a Crystal metodiky. Tyto poznatky byly důležité jako základ pro samotnou praktickou část.

V praktické části byl zprvu dán prostor pro rozbor Společnosti XY a její interní metodice, která se odvolávala na metodiku PRINCE2. Následně byly obě metodiky porovnány pro zjištění, jak moc je metodika PRINCE2 využívána pro řízení projektů. Zde se potvrdil fakt, že obě metodiky mají určité odlišnosti, ale shodují se na tom, že je nutné pro každý projekt zvolit vhodné tailorování metodiky, a to jak interní, tak dle PRINCE2.

Následně byl podrobněji popsán a dle interní metodiky rozebrán konkrétní HW projekt ZZS JČK IT Technologie. Taktéž byly vyzdviženy problémy, které se během jeho realizace objevily. Tyto problémy se tak staly testovacím prostorem pro následně zvolené agilní metodiky.

Samotná simulace projektu ZZS JČK IT Technologie pomocí vybraných agilních metodik přinesla mnoho zjištění. Lze tedy konstatovat, že lze agilitu využít pro technické projekty, ale je nutné jednotlivé kroky, procesy a nástroje přizpůsobit danému projektu. Část agilních metodik se ukázala jako úspěšná pro simulovaný HW projekt, ale není vyloučeno, že pro jiný typ projektu by byla vhodnější jiná metodika. Nicméně agilitu je třeba se

vyvarovat u projektů, které jsou náročné na řízení kvůli jasně daným požadavkům a také u těch, které se musí řídit dle standardů a compliance.

Závěrem lze konstatovat, že aplikace každé metodiky závisí na její znalosti, pečlivé implementaci a sledování všech kroků a procesů. Zároveň je volba metodiky klíčovým faktorem pro úspěšné vedení projektů dle rozsahu, nákladů a času. Stále je ale potřebné hledat způsoby pro zlepšování procesů a zvyšování efektivity a produktivity týmů. Proto je nutné neustálého vzdělávání a zlepšování znalostí a dovedností nejen u projektových manažerů či SCRUM masterů/agilních koučů, ale také u týmů, které na samotných projektech pracují.

## 7. Seznam použitých zdrojů

### Knížní zdroje

Apke, L. (2015) *Understanding the Agile Manifesto*. null edition. lulu.com. ISBN 978-1-312-86391-0.

AXELOS Limited (ed.) (2017) *Managing successful projects with PRINCE2*. 6th edition. London Norwich: TSO. ISBN 978-0-11-331533-8.

Beck, K. and Andres, C. (2005) *Extreme programming explained: embrace change*. 2nd ed. Boston, MA: Addison-Wesley. ISBN 0321278658.

Bentley, C. (2015) *The Prince 2 practitioner: from practitioner to professional*. Third edition. London; New York: Routledge, Taylor & Francis Group. ISBN 9781138824119.

Buchalceková, A. (2005) *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. 1. vyd. Praha: Grada. ISBN 80-247-1075-7.

Campbell, A. (2020) *Agile: Essentials of Team and Project Management. Manifesto for Agile Software Development*. Amazon Services LLC – KDP Print US (Agile Project Management with Kanban). ISBN 979-86-631-8258-4.

Cockburn, A. (2005): *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, 0-201-69947-8.

Dcosta, A. (2012) *Project Management for the Beginner – Part 2*. Booktango. ISBN 978-1-4689-0901-2.

Doležal, J. (2016) *Projektový management: komplexně, prakticky a podle světových standardů*. Praha: Grada Publishing, Expert (Grada). ISBN 978-80-247-5620-2.

Doležal, J., Krátký, J. (2016) *Projektový management v praxi*. Praha: Grada. Expert (Grada). ISBN 978-80-247-5693-6.

Doležal, J., Máchal, P., Lacko, B. (2012) *Projektový management podle IPMA*. 2., aktualiz. a dopl. vyd. Praha: Grada. Expert (Grada). ISBN 978-80-247-4275-5.

Fiala, P. (2004) *Projektové řízení modely, metody, analýzy*. Praha: Professional Publishing, 2004, ISBN 80-86419-24-X.

Großbritannien (ed.) (2012) *Managing successful projects with PRINCE2*. 5. ed., 2. impr. London: TSO. ISBN 978-0113310593

Highsmith, J.A. (2010) *Agile project management: creating innovative products*. 2nd ed. Upper Saddle River, NJ: Addison-Wesley (The agile software development series). ISBN 978-0-321-65839-5.

Kadlec, V. (2004) *Agilní programování: metodiky efektivního vývoje softwaru*. Brno: Computer Press. ISBN 80-251-0342-0.

Kerzner, H. (2009) *Project management: a systems approach to planning, scheduling, and controlling*. 10th ed. Hoboken, N.J: John Wiley & Sons. ISBN 978-0-470-27870-3.

Kerzner, H. (2017) *Project management: a systems approach to planning, scheduling, and controlling*. Twelfth edition. Hoboken, New Jersey: Wiley. ISBN 978-11-191-6535-4.

Landry, N. (2011) *The Huddle Method Iterative & Agile Implementation Methodologies in Business Intelligence Software Development*. 2nd edition. Nat Landry Books. ISBN 978-1-257-50327-8.

Lock, D. (2007) *The essentials of project management*. 3rd ed. Aldershot, England; Burlington, VT: Gower. ISBN-13: 978-0566088056

Máchal, P., Ondrouchová, M., Presová, R. (2015) *Světové standarty projektového řízení: pro malé a střední firmy: IPMA, PMI, PRINCE2*. Praha: Grada. ISBN 978-80-2475-321-8.

Myslín, J. (2016) *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press. ISBN 978-80-251-4650-7.

Poppendieck, M. and Poppendieck, T. (2010) *Lean software development: an agile toolkit*. Nachdr. Boston: Addison-Wesley (The agile software development series). ISBN 978-0-321-15078-3.

Project Management Institute (ed.) (2013) *A guide to the project management body of knowledge (PMBOK guide)*. Fifth edition. Newtown Square, Pennsylvania: Project Management Institute, Inc. ISBN 978-1-933890-51-7.

*Singh, A. (2019) Agile & Scrum Patna Bihar India, ACT. ISBN 978-1-09-982031-1.*

Schoper, Y. et al. (eds) (2017) *Individual Competence Baseline für Portfoliomanagement*. 1. Auflage. Nürnberg Berlin: GPM Deutsche Gesellschaft für Projektmanagement e. V (IPMA Global Standard). ISBN 978-94-92338-00-6

Svozilová, A. (2011) *Projektový management*. 2., aktualiz. a dopl. vyd. Praha: Grada. Expert (Grada). ISBN 978-80-247-3611-2.

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

Šochová, Z. and Kunce, E. (2014) *Agilní metody řízení projektů*. 1. vyd. Brno: Computer Press. ISBN 978-80-251-4194-6.

Verheyen, G. (2017) *Scrum: a pocket guide*. First edition, sixth impression, July 2017. Zaltbommel: Van Haren Publishing.

## **Internetové zdroje**

Kanban – A brief introduction | Atlassian. Atlassian | Software Development and Collaboration Tools [online]. Copyright © 2022 Atlassian [cit. 24.08.2022]. Dostupné z: <https://www.atlassian.com/agile/kanban>



Co je to Kanban | TAYLLORCOX. Manažerské kurzy, školení na míru, certifikace | TAYLLORCOX [online]. Copyright © AXELOS Limited 2009. Copyright [cit. 11.08.2022]. Dostupné z: <https://www.tx.cz/kanban/co-je-kanban>

Hoory, L. (2022) Agile Vs. Waterfall: Which Project Management Methodology Is Best For You? [online] [cit. 13.11.2022]. Dostupné z: <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>.

Magický projektový trjúhelník Collaboration [online] [cit. 24.08.2022]. Dostupné z: *ManagementMania.com* <https://managementmania.com/cs/magicky-trojuhelnik-projektoveho-rizeni>.

Sprint. Ideal Modeling & Diagramming Tool for Agile Team Collaboration [online] [cit. 24.08.2022]. Dostupné z: <https://www.visual-paradigm.com/learning/handbooks/agile-handbook/sprint.jsp>

Životní cyklus Scrumu [online] [cit. 23.02.2023]. Dostupné z: *Learn.Microsoft.com* <https://learn.microsoft.com/cs-cz/devops/plan/what-is-scrum>.

## 8. Seznam obrázků, tabulek, grafů a zkratk

### Seznam obrázků

Obrázek 1: Projektový trojimperativ (zdroj: ManagementMania.com, 2023) .....	17
Obrázek 2: Správné vedení projektu (zdroj: ManagementMania.com, 2023).....	18
Obrázek 3: Špatné vedení projektu (zdroj ManagementMania.com, 2023) .....	18
Obrázek 4: Projektové fáze (zdroj: vlastní zpracování).....	19
Obrázek 5: Životní cyklus projektu (zdroj: Svozilová, 2016) .....	20
Obrázek 6: Model životního cyklu u waterfallu (zdroj: vlastní zpracování).....	23
Obrázek 7: Vodopádový trojimperativ (zdroj: vlastní zpracování) .....	24
Obrázek 8: Agilní trojimperativ (zdroj: vlastní zpracování) .....	26
Obrázek 9: Příklad Kanbanová tabule (zdroj: Canva.com, 2023) .....	29
Obrázek 10: Barevné rozlišení Crystal metodik (zdroj: Singh, 2019).....	36
Obrázek 11: Další pohled na rozdělení Crystal metodik (zdroj: Singh, 2019).....	36
Obrázek 12: Sprint (zdroj: LearnMicrosoft.com, 2023) .....	43
Obrázek 13: Organizační struktura Společnosti XY (zdroj: Společnost XY, 2023) .....	49
Obrázek 14: Životní cyklus zakázky (zdroj: Společnost XY, 2016) .....	51
Obrázek 15: Směrný plán projektu (zdroj: Společnost XY).....	59

### Seznam tabulek

Tabulka 1: Porovnání metodik (zdroj: vlastní zpracování) .....	22
Tabulka 2: Agile vs. Waterfall (zdroj: vlastní zpracování) .....	46
Tabulka 3: Definování jednotlivých úkolů dle Kanban (zdroj: vlastní zpracování).....	66
Tabulka 4: Product backlog ZZS JČK IT Technologie (zdroj: vlastní zpracování).....	76
Tabulka 5: Backlog pro 1. sprint (zdroj: vlastní zpracování) .....	76
Tabulka 6: Kanban tabule začátek projektu (zdroj: vlastní zpracování) .....	91
Tabulka 7: Kanban tabule postup (zdroj: vlastní zpracování) .....	91
Tabulka 8: Kanban tabule postup (zdroj: vlastní zpracování) .....	92
Tabulka 9: Kanban tabule postup (zdroj: vlastní zpracování) .....	92
Tabulka 10: Kanban tabule postup (zdroj: vlastní zpracování) .....	93
Tabulka 11: Kanban tabule postup (zdroj: vlastní zpracování) .....	93
Tabulka 12: Backlog pro 2. sprint (zdroj: vlastní zpracování) .....	93
Tabulka 13: Backlog pro 3. sprint (zdroj: vlastní zpracování) .....	93

### Seznam použitých zkratk

RDM – regionální delivery manažer (jiný název pro projektového manažera)

HW – hardware

XP – extrémní programování

WIP – Work in Progress

## 9. Přílohy

### Aplikace projektu ZZS JČK IT Technologie dle metodiky Kanban

<b>TO DO</b>	<b>DOING</b>	<b>DONE</b>
Kick-off		
Boxy hlavního firewallu		
Instalace VMWaru a virtualizace		
Instalace DELL monitorů		
Nahrazení fyzických serverů a reinstalace		
Povýšení struktury AD a přenos rolí		
PRTG monitoring		
Ukončení projektu a akceptace		

*Tabulka 6: Kanban tabule začátek projektu (zdroj: vlastní zpracování)*

<b>TO DO</b>	<b>DOING</b>	<b>DONE</b>
	Kick-off	
Boxy hlavního firewallu		
Instalace VMWaru a virtualizace		
Instalace DELL monitorů		
Nahrazení fyzických serverů a reinstalace		
Povýšení struktury AD a přenos rolí		
PRTG monitoring		
Ukončení projektu a akceptace		

*Tabulka 7: Kanban tabule postup (zdroj: vlastní zpracování)*

<b>TO DO</b>	<b>DOING</b>	<b>DONE</b>
		Kick-off
	Boxy hlavního firewallu	
	Instalace VMWaru a virtualizace	
	Instalace DELL monitorů	
	Nahrazení fyzických serverů a reinstalace	
Povýšení struktury AD a přenos rolí		
PRTG monitoring		
Ukončení projektu a akceptace		

*Tabulka 8: Kanban tabule postup (zdroj: vlastní zpracování)*

<b>TO DO</b>	<b>DOING</b>	<b>DONE</b>
		Kick-off
		Boxy hlavního firewallu
		Instalace VMWaru a virtualizace
	Instalace DELL monitorů	
		Nahrazení fyzických serverů a reinstalace
	Povýšení struktury AD a přenos rolí	
	PRTG monitoring	
Ukončení projektu a akceptace		

*Tabulka 9: Kanban tabule postup (zdroj: vlastní zpracování)*

TO DO	DOING	DONE
		Kick-off
		Boxy hlavního firewallu
		Instalace VMWaru a virtualizace
	Instalace DELL monitorů	
		Nahrazení fyzických serverů a reinstalace
		Povýšení struktury AD a přenos rolí
	PRTG monitoring	
	Ukončení projektu a akceptace	

Tabulka 10: Kanban tabule postup (zdroj: vlastní zpracování)

TO DO	DOING	DONE
		Kick-off
		Boxy hlavního firewallu
		Instalace VMWaru a virtualizace
	Instalace DELL monitorů	
		Nahrazení fyzických serverů a reinstalace
		Povýšení struktury AD a přenos rolí
	PRTG monitoring	
		Ukončení projektu a akceptace

Tabulka 11: Kanban tabule postup (zdroj: vlastní zpracování)

## Sprint backlog pro další sprinty

Instalace DELL monitorů
Povýšení struktury AD a přenos rolí
PRTG monitoring

Tabulka 12: Backlog pro 2. sprint (zdroj: vlastní zpracování)

Instalace DELL monitorů
PRTG monitoring

Tabulka 13: Backlog pro 3. sprint (zdroj: vlastní zpracování)