



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

APLIKACE PRVKŮ AGILNÍCH METOD PROJEKTOVÉHO ŘÍZENÍ VÝVOJE SOFTWARE VE FIRMĚ

APPLICATION OF ELEMENTS OF AGILE PROJECT MANAGEMENT METHODS FOR SOFTWARE
DEVELOPMENT IN A COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

František Kudrna

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lenka Širáňová, Ph.D.

BRNO 2023

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **František Kudrna**
Vedoucí práce: **Ing. Lenka Širáňová, Ph.D.**
Akademický rok: 2022/23
Studijní program: Manažerská informatika

Garant studijního programu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Aplikace prvků agilních metod projektového řízení vývoje softwaru ve firmě

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Návrh řešení a přínos návrhů řešení
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Cílem bakalářské práce je aplikace teoretických poznatků, nástrojů a prvků agilních metod, vedoucí ke zefektivnění procesů řízení vývoje softwaru v daném podniku.

Základní literární prameny:

BUCHALCEVOVÁ, Alena. Metodiky vývoje a údržby informačních systémů. Praha: Grada Publishing, 2005. ISBN 80-247-1075-7.

DOLEŽAL, Jan. Agilní přístupy vývoje produktu a řízení projektu. Praha: Grada Publishing, 2022. ISBN 978-80-271-6606-0.

DOLEŽAL, Jan. Projektový management podle IPMA. Praha: Grada Publishing, 2009. ISBN 978-80-247-2848-3.

DOUČEK, Petr. Řízení projektů informačních systémů. Praha: Professional publishing, 2004. ISBN 80-86419-71-1.

SVOZILOVÁ, Alena. Projektový management. Praha: Grada Publishing, 2006. 356 s. ISBN 80-247-1501-5.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně dne 5.2.2023

L. S.

Ing. Jiří Kříž, Ph.D.
garant

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Bakalářská práce je zaměřena na aplikaci vybraných prvků agilních metodik do procesů projektového řízení vývoje softwaru vybrané firmy. Práce je tvořena třemi kapitolami. V první části je zpracován teoretický podklad, který se ve svém úvodu zaměřuje na základy projektového řízení a postupně podrobněji popisuje agilní metodiky a jejich zástupce Scrum. Ve druhé části je představena vybraná firma společně s analýzou současného stavu projektového řízení a jsou popsány aktuální procesy, které ve firmě probíhají v rámci řízení projektů. Z výsledků analýzy jsou identifikovány oblasti, pro které je zpracováno řešení formou návrhu s využitím vybraných prvků agilních metodik.

Abstract

The bachelor thesis focuses on the application of selected elements of agile methodologies to the project management processes of software development in a selected company. The thesis consists of three chapters. In the first part, the theoretical background is developed, which in its introduction focuses on the basics of project management and gradually describes agile methodologies and their representative Scrum in more detail. In the second part, a selected company is introduced together with an analysis of the current state of project management and the current processes that take place in the company within the framework of project management are described. From the results of the analysis, areas are identified for which a solution is developed in the form of a proposal using selected elements of agile methodologies.

Klíčová slova

Projektové řízení, projektový manažer, Kanban, Scrum, agilní metody

Key words

Project management, project manager, Kanban, Scrum, agile methodology

Bibliografická citace

KUDRNA, František. *Aplikace prvků agilních metod projektového řízení vývoje softwaru ve firmě* [online]. Brno, 2023 [cit. 2023-05-12]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/152547>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Lenka Širáňová, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 12. 5. 2023

František Kudrna

autor

Poděkování

Rád bych touto cestou poděkoval vedoucí mé bakalářské práce, paní Ing. Lence Širáňové Ph.D., za její velmi ochotný přístup, postřehy a cenné rady, které byly vždy velmi přínosné k dokončení mé práce. Velké díky také mým kolegům a vedení vybrané firmy za poskytnuté informace, ochotu a pomoc.

Obsah

Úvod.....	9
Cíle práce, metody a postupy zpracování	11
Cíle práce	11
Metody a postupy zpracování	11
1 Teoretická východiska práce.....	13
1.1 Projekt	13
1.1.1 Work Breakdown Structure.....	14
1.1.2 Trojimperativ	15
1.1.3 SMART cíl	15
1.1.4 Úspěšnost projektu.....	16
1.2 Životní cyklus projektu a fáze projektu	17
1.2.1 Předprojektová fáze	18
1.2.2 Projektová fáze	19
1.2.3 Poprojektová fáze	21
1.3 Životní cyklus vývoje softwaru.....	21
1.3.1 Analýza požadavků.....	22
1.3.2 Design	22
1.3.3 Vývoj	22
1.3.4 Testování.....	22
1.3.5 Údržba.....	22
1.4 Vodopádový model.....	23
1.5 Agilní metodiky	24
1.5.1 Hlavní principy agilního vývoje	24
1.6 Lean development	25
1.7 Kanban	26

1.7.1	Popis Kanbanu	27
1.7.2	Řízení toku práce	27
1.8	Scrum	28
1.8.1	Základy Scrumu	29
1.8.2	Scrum tým	29
1.8.3	Product backlog	32
1.8.4	Definition of done	33
1.8.5	Sprint backlog	34
1.8.6	User story	34
1.8.7	Sprint	35
1.8.8	Denní Scrum	36
1.8.9	Review sprintu	37
1.8.10	Shrnutí Scrumu	37
2	Analýza současného stavu	39
2.1	Představení společnosti	39
2.2	Organizační struktura	39
2.3	Jednání se zákazníkem	40
2.3.1	Získávání zakázek	41
2.4	Produktové portfolio	41
2.5	Používané softwarové nástroje a technologie	42
2.5.1	SAP	42
2.5.2	Visual Studio	42
2.5.3	Microsoft Office	43
2.5.4	Hyper-V	43
2.5.5	Jira	43
2.6	Proces vývoje	46

2.6.1	Životní cyklus nového projektu	46
2.6.2	Projektový tým.....	49
2.7	Řízení projektů.....	50
2.7.1	Pravidelné porady	50
2.7.2	Komunikace	51
2.7.3	Požadavky	51
2.7.4	Řízení vývoje	52
2.7.5	Projektová knihovna	53
2.8	Shrnutí	54
3	Návrh řešení a přínos návrhů řešení	56
3.1	Zavedení role Product owner	56
3.2	Zavedení role Scrum Master	57
3.3	Uživatelské požadavky.....	58
3.3.1	User story	58
3.3.2	Prioritizace	60
3.3.3	Definition od Done	60
3.3.4	Praktická ukázka user story	61
3.4	Evidence chyb softwaru	63
3.5	Schůzky projektového týmu.....	64
3.6	Porada programátorů	67
3.7	Školení.....	68
3.8	Ekonomické zhodnocení	68
3.9	Přínosy navrhovaného řešení	70
	Závěr	71
	Seznam literatury	72
	Seznam Obrázků	74

Seznam Tabulek	75
----------------------	----

Úvod

Na počátku projektového řízení v sedmdesátých letech byl nejčastěji aplikován vodopádový model. Tento model byl dlouhou dobu šablonou většiny realizovaných projektů. Pokrok však graduje ve všech oblastech a tím se projekty stávaly větší a komplexnější než kdy dříve, a právě tehdy začal model narážet na své limity. Větší a komplexnější projekty si žádaly násobně více lidských i finančních zdrojů. S větší náročností se velkým problémem stával i čas. Aby bylo možné takto rozsáhlé projekty spolehlivě řídit, bylo nutné udržovat spolupráci se zadavatelem projektu. Nezbytnost průběžných konzultací ohledně vývoje projektu, úpravy či nových požadavků si žádaly vzestup nových agilních metod projektového řízení. Navzdory nově vzniklým metodám řízení projektů mají základní stavební kameny stejné jako tradiční metody řízení projektů. Společným základem, který sdílí všechny typy metod, jsou cíl, náklady a čas, společně utvářejí tzv. trojimperativ projektu.

Je to právě agilní přístup projektového řízení a jeho implementace do podniku, kterým se v dnešní době zabývá čím dál více společností. Transformace podnikových procesů, zaběhlých v dobách vodopádového modelu na agilní přístup, nemusí být pro tradiční firmu lehký úkol, jak se na první pohled může zdát. Existuje řada zástupců agilních metodik a ani výběr vhodné metody nemusí nutně znamenat konečný úspěch. Je zde několik aspektů, které je nutno před implementací zvážit. Tím jsou procesy uvnitř firmy, spokojenost zaměstnanců, uspokojení potřeb zákazníka, ale je to také bezpečnost produktu, co může implementaci ovlivňovat. Zkrátka ke každé společnosti je třeba přistupovat individuálně a uvažovat o pozitivních i negativních aspektech jednotlivých přístupů. Jmenované důvody jsou motivací k zaměření se na to, jaké má firma očekávání od transformace, a poté důkladně zvážit aplikaci vybraných prvků agilních metod.

Pro tyto účely poskytuje první kapitola práce teoretický základ, který představuje základy projektového řízení. Následně je věnován prostor k představení agilních metod a jejich zástupců. Detailně jsou zde popsány techniky a principy rámce Scrum.

Otázka transformace přístupů projektového řízení je zpracovávána ve vybrané firmě, která podniká v oblasti vývoje informačních systémů pro sektor obrany a z toho důvodu chce zůstat anonymní. Analýze současného stavu podniku je věnována druhá kapitola, která ve svém úvodu popisuje společnost, organizační strukturu a hlavního

zákazníka. Dále následují podkapitoly, které popisují proces realizace a řízení vývoje softwaru a hlubší seznámení se s procesy spjaté s řízením projektů.

Poslední kapitola se věnuje návrhům ohledně řešení identifikovaných oblastí vedoucích ke zlepšení v analytické části. Pro nalezení efektivního řešení pro zlepšení projektového řízení u vybrané firmy jsou vypracovány návrhy aplikace vybraných prvků z agilní metody Scrum. Součástí uvedených návrhů je také závěr shrnující nejdůležitější přínosy. V samotném závěru je zpracována finanční kalkulace nákladů, které vznikají navrhovanými změnami.

Cíle práce, metody a postupy zpracování

Cíle práce

Hlavním cílem této bakalářské práce je aplikace teoretických poznatků, nástrojů a prvků agilních metod, vedoucích k zefektivnění procesů řízení vývoje softwaru v daném podniku.

Hlavního cíle bude dosaženo pomocí několika dílčích cílů reprezentujících hlavní kapitoly této práce:

- Zpracování teoretického podkladu věnujícího se prvkům a nástrojům projektového řízení s hlavním zaměřením na agilní metodiky.
- Vypracování analýzy současného stavu projektového řízení vybrané firmy s cílem identifikovat oblasti ke zpracování návrhů na zlepšení.
- Zpracování návrhů řešení pro aplikaci vybraných prvků agilních metod vedoucích k zefektivnění projektového řízení ve vybrané firmě.

Hlavního cíle práce a jednotlivých dílčích cílů bude dosaženo za pomoci metod a postupů zpracování, které budou detailně popsány v kapitole níže.

Metody a postupy zpracování

Zpracování teoretické části práce je provedeno na základě knižní literatury a odborných článků řešících oblast projektového managementu. V teoretické části jsou uvedeny a následně vysvětleny veškeré odborné pojmy, které budou součástí analytické části společně s částí týkající se návrhu. Uvedená teoretická část je základem řešené problematiky. Začátkem kapitoly je čtenář uveden do obecné oblasti problematiky a jsou mu objasněny základní pojmy společně s principy projektového managementu. Po získání základních znalostí se kapitola orientuje na hlubší poznání přístupů řízení i se specifickým zaměřením na konkrétní zástupce.

Podkladové materiály pro část analýzy současného stavu byly zpracovány formou osobních rozhovorů s vedením podniku i zaměstnanci a na základě osobní pracovní zkušenosti ve vybrané firmě. Získané informace z nabitých zkušeností a poznatků z rozhovorů tvořily základ pro porovnání s teoretickým podkladem agilního projektového řízení v první kapitole práce. Výstupem provedeného porovnání je návrh vhodných řešení

pro optimalizaci projektového řízení ve firmě. Zpracované návrhy na řešení identifikovaných oblastí ke zlepšení se opírají o teoretický základ první části práce.

Analýza i návrhová část se bude týkat menší firmy s několikaletou tradicí, která se věnuje primárně vývoji softwaru. Z důvodu práce firmy s utajovanými daty a informacemi bude v rámci celé práce zachována anonymita. V kapitole analýzy současného stavu podniku se práce detailně zaměří na organizační strukturu firmy, vnitrofiremní procesy vedoucí k uspokojování potřeb zákazníka a nejčastěji řešené problémy související s dosavadním fungováním podniku. Veškeré informace použité v kapitole analýzy a návrhové části jsou čerpány z osobních zkušeností během pracovního poměru ve firmě a osobních pohovorů se zaměstnanci společnosti.

1 Teoretická východiska práce

První kapitola bude věnována teoretickému základu, který bude tvořit osnovu pro vysvětlení odborných termínů v dalších kapitolách. Postupně budou objasněny elementární pojmy projektového řízení včetně základního členění jednotlivých přístupů. Dále se kapitola věnuje bližšímu pohledu na agilní metodiku a její zástupce. V tomto případě Scrum a Kanban, kterým bude věnován prostor v závěrečné části kapitoly.

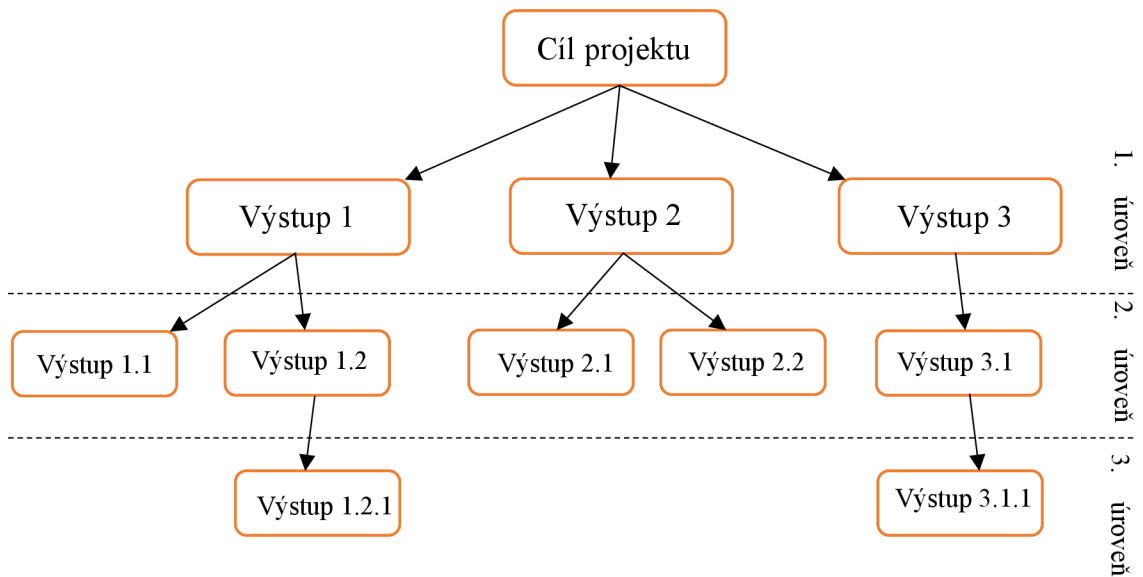
1.1 Projekt

Jasně specifikovat projekt pomocí jedné definice nelze. Definice řady autorů se odlišuje, zkrátka každý může projekt definovat dle svého pohledu. Základem definování projektu je, že projektem se rozumí sled konkrétních činností vedoucích za pomoci určitých pravidel ke stanovenému cíli při čerpání limitovaných zdrojů k jeho realizaci. Cílem může být například vytvoření zcela unikátního produktu či služby nebo také realizace plánované změny. Ve výsledku realizace projektu dochází vždy ke změně. Každý projekt může být jedinečný, a to z důvodu cílů, kterých má být dosaženo, za pomoci nákladů a času na projekt vyhrazených. Právě díky času je každý projekt dočasný. Čas nám definuje, kdy projekt začíná a kdy bude dosaženo cíle. V průběhu projektu dochází k řešení specifických potřeb a cílů, které obstarává jedinečný projektový tým, složený dočasně pouze za existence daného projektu. To ve výsledku znamená neopakovatelnost každého z projektů [1].

Projekty se mezi sebou mohou různě lišit. Jsou projekty, které vznikají na základě vlastní iniciativy podniku, který plánuje dosáhnout nějaké změny. Jsou projekty, které vznikají jako reakce na poptávku zákazníka. Příkladem může být tvorba nového softwaru na přání zákazníka. Podnikové prostředí ale nemusí být pouze místo, kde se s projekty lze setkat. Za projekty se považují i školské práce, i ty mají svůj cíl, své zdroje. Spektrum příkladů a druhů projektů je velmi široké a každý den se s nimi můžeme setkávat [1].

1.1.1 Work Breakdown Structure

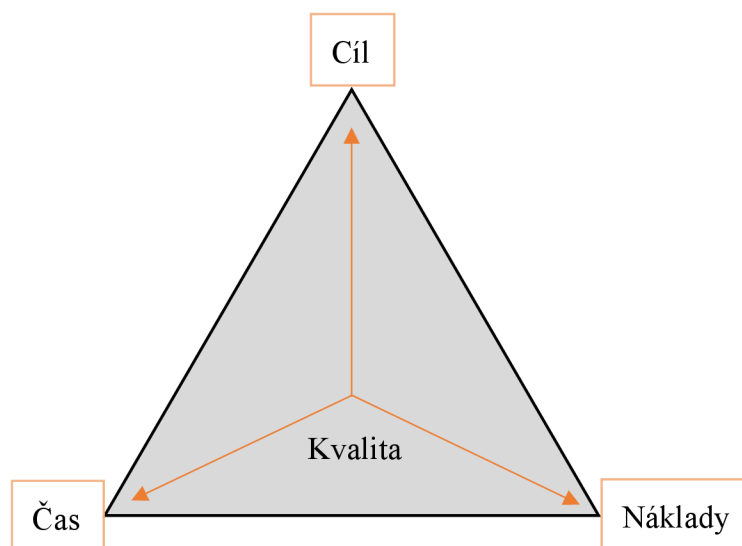
WBS lze chápat jako rozpad stanoveného cíle na hierarchicky provázané menší produkty či činnosti na co nejdetailnější úroveň pracovních balíků, vedoucí k jednomu cíli. Jinými slovy WBS představuje součet všech činností, které se mají v rámci projektu vykonat neboli co se nevyskytuje ve WBS, nebude realizováno. Proces rozpadu cíle je používán k efektivnímu řízení a kontrole postupu na projektu [6].



Obrázek 1 **WBS** (Zdroj: vlastní zpracování)

1.1.2 Trojimperativ

Navzdory různorodosti všech projektů mají prakticky vždy jedno společné. V každém projektu, se kterým se můžeme setkat, se vyskytují tři nejpodstatnější pojmy, které jsou základními veličinami projektu. Jedná se o cíl, náklady a čas. Společně tyto veličiny utvářejí trojimperativ projektového řízení, který si lze představit jako tři spojené body. Protože jsou veličiny mezi sebou přímo provázány, tak nelze měnit jednu, aniž by byl ovlivněn zbytek. To znamená, že jestliže bude změněn čas projektu, ovlivní to náklady i cíl. Cílem každého trojimperativu je najít rovnováhu mezi všemi body a vytvořit tak optimální stav. Prakticky se bude vždy jednat o dosažení maxima z definovaného cíle za pomoci minimálních nákladů a času. Trojimperativ projektu nepůsobí pouze nad projektem jako takovým, ale nachází se i v rámci etap i milníků projektu, dokonce i mezi jednotlivými činnostmi potřebnými k dosažení stanoveného cíle [2].



Obrázek 2 *Trojimperativ projektu* (Zdroj: vlastní zpracování)

1.1.3 SMART cíl

Vynaložené úsilí k dosažení cíle projektu by vždy mělo vést k jasně definovanému cíli. Jestliže nebude stanovený cíl srozumitelně a detailně definován, zvyšujeme pravděpodobnost neúspěšného dokončení odvedené práce a projekt selže. Součástí plánů projektu jsou i zainteresované strany. Tyto strany, stejně jako řešitel, musí znát cíl a musí jim být stejně jako řešiteli zcela jasno, co daný cíl představuje a co bude vyprodukováno na konci [2].

Technika SMART pomáhá s přesnou definicí cíle podle jednoduchého schématu.

- S – specifický a specifikovaný – odpovídá na otázku CO?
- M – měřitelný – určení, čeho bude dosaženo
- A – akceptovaný – společně odsouhlasený všemi stranami
- R – realistický – zda je realizace možná
- T – termínovaný – určení termínu [2]

1.1.4 Úspěšnost projektu

Představa úspěšnosti projektu za podmínky zdárného splnění trojimperativu nemusí být relevantním měřítkem úspěšného dokončení projektu. Jak bylo popsáno výše, trojimperativem se rozumí plánované dodržení základních atributů. Dosažení stanoveného cíle za pomoci vyhrazených zdrojů, nákladů a času. Trojimperativ nevyovídá o reálné možnosti uplatnění vzniklého řešení. Za situace, kdy dle plánů dokončíme projekt, ale výsledná změna již nebude použitelná, nelze považovat projekt za úspěšný [2].

Rozhodnutí o úspěšnosti projektu lze určit pomocí několika kritérií, která mají za úkol nám na tuto otázku odpovědět. Jedná se o kritéria úspěchu projektu a kladou důraz na srozumitelnost, jednoznačnost a měřitelnost. Nejedná se o metodu, kterou lze aplikovat pouze jednou. Je nutné kritéria znovu sestavit, spolu s novým projektem či zákazníkem. Je třeba zhodnotit, analyzovat a dobře komunikovat [2].

Kritéria úspěšnosti:

- Projekt je funkční
- Jsou naplněny požadavky zákazníka
- Uspokojení očekávání zainteresovaných stran
- Výsledek projektu je na trhu včas

Kritéria neúspěšnosti:

- Nedodržení nákladů a času
- Nekvalitní produkt
- Neplánované negativní ovlivnění životního prostředí
- Selhání splnění požadavků zákazníka či zainteresovaných stran
- Produkt nelze prezentovat na trhu

Finanční kritéria jsou třetí a poslední skupinou kritérií. Jedná se o kritéria, která lze predikovat již před zahájením projektu pomocí finančních ukazatelů. Finanční kritéria vyhodnocují návratnost investice. Tedy zda projekt bude ziskový a máme důvod jej realizovat [2].

Prvním finančním ukazatelem je návratnost investice, zkráceně ROI. Pomocí tohoto ukazatele lze určit návratnost investice [2].

$$ROI [\%] = \frac{\text{výnos} - \text{investice}}{\text{investice}} * 100 [\%] = \frac{\text{zisk}}{\text{investice}} * 100 [\%]$$

Druhým ukazatelem je výpočet čisté současné hodnoty, zkráceně NPV. Principem je výpočet na základě čistých současných hodnot ve zvolených časových obdobích. Jestliže $NPV > 0$, pak je investice výhodná. Pokud $NPV = 0$ nebo $NPV < 0$, je investice nevýhodná [2].

$$NPV = \sum_{t=0}^n \frac{CF_t}{(1+r)^t} - IN$$

1.2 Životní cyklus projektu a fáze projektu

Pod pojmem životní cyklus projektu se skrývají jednotlivé fáze, které dohromady utvářejí zmiňovaný životní cyklus. Můžeme jej chápat jako sjednocující termín pro všechny fáze, na které se projekt rozkládá. Slovo cyklus zde není myšleno doslovně a nejedná se o opakující se sled činností, protože čas v projektu plyne pouze jedním směrem. Ze samotné definice projektu vyplývá skutečnost, že se jedná zpravidla o jedinečný a neopakovatelný sled událostí [2].

Fáze, ze kterých se cyklus skládá, jsou z manažerské perspektivy časové úseky, ve kterých jsou vykonávány určité činnosti. Posloupnosti takových činností v časovém úseku jsou natolik odlišné od dalšího cyklu, že je lze rozlišit a zařadit do další fáze projektu [1]. K tomu, aby fáze dodržovaly řád, je třeba vymyslet cíle, kterých má být dosaženo. K efektivnímu řízení a koordinaci činností vedoucích k takovým cílům pomáhají milníky projektu. Ty jsou důležitou součástí každého projektu kvůli přehledné orientaci a kontrole [2].

Obecně lze u projektů stanovit základní 3 fáze. U každé lze zmínit, jaké typy výstupů by měly přinést a také jakým způsobem budou ohodnoceny. S měnicími se

fázemi se také mění charakteristiky cyklu. S plynoucím časem se čerpají dostupné zdroje. V pozdějších fázích také klesá schopnost reagovat na změny. Popisovaný životní cyklus je tedy jen obecnou formou, jak cyklus vypadá. Například u softwaru se životní cyklus více konkretizuje a má své specifické fáze, základ ale zůstává stejný a lze si jej upravit na míru. Model SDLC bude v práci popsán níže [1].

1.2.1 Předprojektová fáze

První fáze se nazývá předprojektová, je to fáze přípravná či definiční. Výstupem je jasná a specifikovaná představa o projektu. Na počátku je třeba analyzovat příležitost, jaká by byla potřebná pro projekt, a přijít s rozhodnutím o proveditelnosti projektu. Je možné v této části přednést vizi a základní myšlenku projektu [2].

Jestliže věnujeme přípravě dostatek času a úsilí, dostaneme i odpovědi na otázky strategického řízení:

- Odkud jdeme?
- Kam chceme dojít?
- Jakou cestou?
- Má to cenu realizovat? [2]

Všechny záležitosti přípravy projektu včetně zhotovení dokumentů má za úkol zhotovit určený tým. Ten zpracuje doporučený postup pro projekt. Rozhodnutí o zahájení je plně v rukou managementu podniku, není-li stanoveno jinak [2].

Studie proveditelnosti

Jedná se o důkladný průzkum nejlepší cesty k realizaci projektu. Cílem je upřesnit obsah projektu a s ním i najít vhodný termín zahájení a ukončení. Dalším důležitým výstupem tohoto dokumentu je rozpis potřebných nákladů a zdrojů k realizaci. Abychom mohli spolehlivě plánovat čas, náklady i zdroje projektu, je třeba vycházet z reálného současného stavu podniku. Jestliže firma nedisponuje potřebnými zdroji, nelze provést studii korektně. Nedílnou součástí je i vytvoření návrhu složení týmu, který se bude projektem zabývat. Nelze začínat projekt, aniž by byla vyřešena otázka lidských zdrojů potřebných k realizaci. Zpracování studie proveditelnosti probíhá na základě studie příležitosti i dalších podkladových materiálů [2].

Analýza rizik

Součástí projektového řízení by mělo být i důkladné vyhledávání a šetření rizik. Protože na světě neexistuje okolí bez rizik, u všech projektů by měla probíhat tzv. analýza rizik [1].

Rizika obvykle nabývají hodnot pravděpodobnosti 0 a 1, buď riziko nastane, nebo nenastane. Rizikem můžeme chápat působení negativního vlivu na cíle projektu. Při jejich analýze manažer postupuje na základě dostupných informací; čím více informací a řádně kvalitních má, tím méně se v analýze bude vyskytovat rizik [1].

V rámci životního cyklu projektu velmi je důležité, v jaké fázi se rizika mohou vyskytovat. Posouzení hodnoty rizika nevychází pouze z identifikace druhu a hodnoty dopadu, ale také je velmi důležité, v jaké části cyklu se může objevit. V případě už téměř hotového projektu je například riziko ztráty dat mnohem nebezpečnější než na začátku [1].

Identifikace rizik neprobíhá pouze v předprojektové fázi. Je třeba rizika řídit i během celého projektu. Tento proces se nazývá proces řízení rizik. Jedná se o aktivity spojené s prevencí a způsobem ošetřením rizik, tak aby byly odstraněny nebo co nejvíce odvráceny možné dopady na projekt [1].

Proces řízení rizik má i svou základní strukturu:

- určení strategie řízení rizik
- identifikace a analýza rizik
- navržení nápravných opatření

Analýza rizik bývá předmětem jednání projektového týmu nebo je to odpovědností výhradně jmenovaného člena týmu na řízení procesu rizik [1].

1.2.2 Projektová fáze

Ve fázi realizace, jak bývá též nazývána, již bylo vedením schváleno a spuštěno zahájení projektu. Na základě příprav předprojektové fáze je zformován projektový tým. Hlavním předmětem této fáze je postupovat dle plánů a dospět k očekávaným výsledkům. Z důvodu velké obsáhlosti této fáze je dělena na několik menších částí [2].

Zahájení

Před oficiálním zahájením projektu proběhne validace výstupů předprojektové fáze. Konkrétně cíle projektu, účel, lidské zdroje, kompetence a vše, co je dále třeba. Současně se zahájením projektu se doporučuje vytvořit identifikační listinu projektu. Jedná se o dokument shrnující základní informace o projektu v této brzké fázi. Definuje základní technické a organizační záležitosti [2].

Plánování

Do této části již přichází vybraný projektový tým a ten je postaven před již předchystané zadání projektu. Jedním z prvních úkolů týmu je vytvořit finální plán projektu, který lze předložit ke schválení vedení společnosti [2].

Realizace projektu

Samotná realizace projektu znamená spuštění vývoje kódu nebo spuštění výroby. Před spuštěním se doporučuje uspořádání „kick-off meeting“. Na toto jednání se dostaví zástupci nezbytných zainteresovaných stran a předmětem je poslední zrekapitulování sestaveného plánu. Dalším pozitivním efektem je osobní kontakt na jednom místě, který může přinést bližší poznání všech stran. Tímto je oficiálně realizace zahájena a projekt se naplno rozbíhá [2].

Nad odváděnou prací je třeba aktivní kontrola, zda probíhající činnosti vedou k očekávaným výstupům dle plánu. Řízení probíhá podle plánu kontroly a řízení stanoveného v plánu projektu. Pokud se vyskytnou odchylky, jsou řešeny také na základě plánu pro řízení změn. Mohou se i vyskytnout změny, které vyvolají změny v plánu projektu. Ty jsou poté projednány a popřípadě odsouhlaseny se zainteresovanými stranami [2].

Ukončení realizace

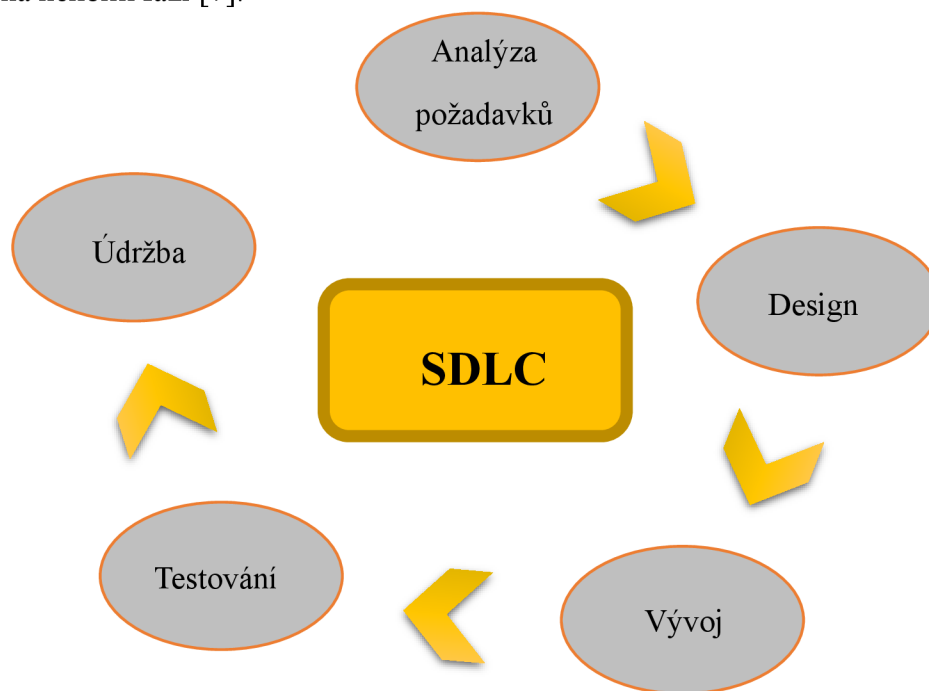
Předání dokončených výstupů z realizační části vede k procesu kontroly a podepisování předávacích i akceptačních protokolů. Součástí bývají podnikové a akceptační zkoušky jak na straně dodavatele, tak i na straně odběratele. Jedná se o závěrečnou kontrolu základního plánu projektu s finálními dodanými výstupy [2].

1.2.3 Poprojektová fáze

S průběhem celého projektu se můžeme setkávat s různými problémy a situacemi. Je třeba pro neustálý rozvoj podniku všechny odvedené projekty v rámci této fáze vyhodnocovat. Vyhodnocení může mít různou podobu. Lze se na základě zkušenosti se spolupracujícími dodavateli rozhodnout, které dodavatele budeme upřednostňovat a které po špatné zkušenosti nebudeme. Každý projekt s sebou přináší řadu nových zkušeností a ty je třeba zpracovat do budoucna a využít ve svůj prospěch. Procesem vyhodnocení lze také dosáhnout efektivního plánování. Můžeme na podobnosti projektů stanovit předpokládaný vývoj dalšího projektu. Výsledným vyhodnocením je možné objevit množství chyb, a naopak dobrých rozhodnutí, které nám do budoucna pomohou rozvinout firmu správným směrem [2].

1.3 Životní cyklus vývoje softwaru

Vývoj počítačových softwarů a informačních systémů měl vždy svůj specifický přizpůsobený model vývoje. Rámec vývoje softwaru obsahuje pravidla a metodiky k plánování, řízení a kontrole procesů vývoje systému. Všeobecně je tento cyklus znám pod zkratkou SDLC, tedy životní cyklus vývoje softwaru. Standartně rozděluje průběh vývoje na několik fází [7].



Obrázek 3 Model SDLC (Zdroj: vlastní zpracování)

1.3.1 Analýza požadavků

V SDLC je první fází analýza požadavků. Zde je pozornost soustředěna zejména na požadavky zákazníka. Při zjišťování klientských požadavků je nutno mít jasno v tom, co se od systému čeká a jaké by měl plnit funkce. Výstupem této fáze je dokumentace obsahující požadavky klienta a souhrn veškerých požadovaných funkcí a chování softwaru [7].

1.3.2 Design

Vstupem pro tuto fázi jsou klientské požadavky na software. Zde se pracuje hlavně s přeměnou požadavků do uceleného plánu struktury produktu. Na plánování řešení projektu se podílejí vývojáři a návrháři, kteří budou na projektu pracovat. Jejich cílem je vnést svůj pohled a přijít s návrhem na řešení softwaru [7].

1.3.3 Vývoj

Fází vývoje se rozumí tvorba softwaru pomocí potřebných nástrojů a programovacího jazyka k dosažení stanoveného cíle. Pro tuto práci je třeba odborných schopností a komunikace vývojářů. Cílem je obvykle vzniklý programový kód [7].

1.3.4 Testování

Po fázi kódování je třeba výslednou práci důkladně zrevidovat. Testování je potřebné k porovnání výsledku odvedené práce na vývoji s výsledkem, jehož mělo být dosaženo na základě klientských požadavků. Jedná se o velmi důležitou fázi, která nám zhodnotí úspěšnost odvedené práce. Efektivní nastavení testovací fáze může vést k spolehlivějším výsledkům a nižším nárokům na budoucí údržbu systému [7].

1.3.5 Údržba

Konečná fáze projektu je poprodejní proces provozování softwaru. Může být nastavený na odpovědnost klienta nebo také jako součástí placených služeb dodavatele, kdy se vyskytnou problémy vzniklé v průběhu ostrého provozu [7].

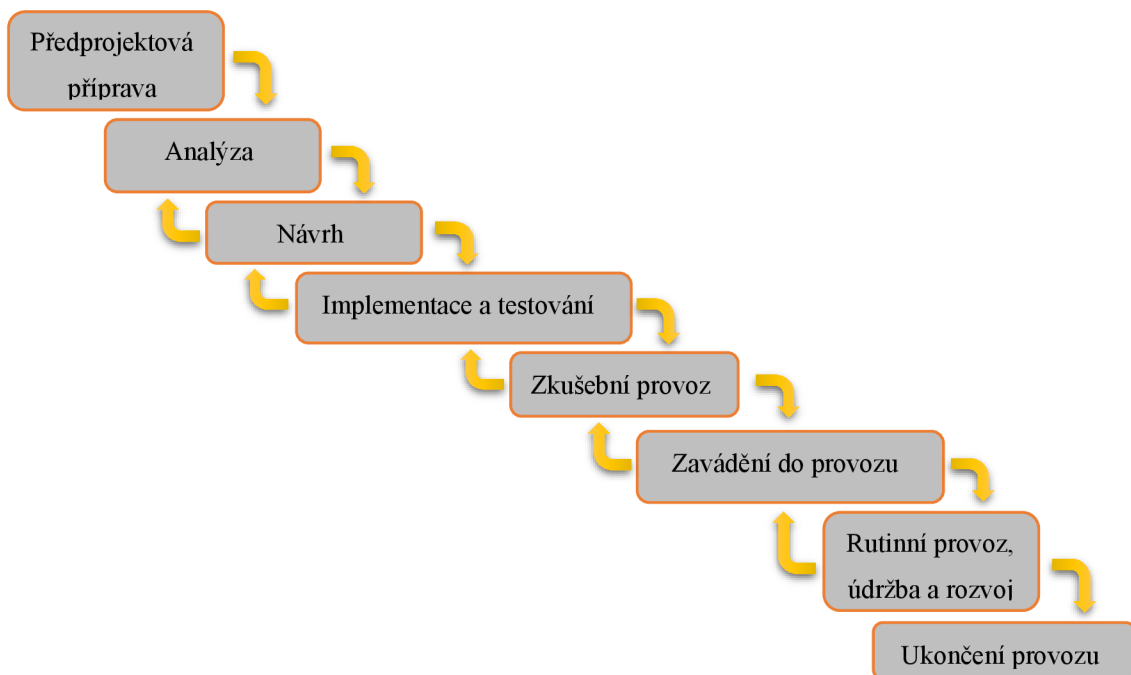
K modelu SDLC se váže několik různých přístupů k vývoji softwaru. Každý z těchto přístupů se řídí dle vlastních stanovených pravidel [7].

Modely procesu vývoje softwaru:

- Vodopádový model
- Iterativní modely
- Spirálový model
- Model tvaru V
- Inkrementální model [7]

1.4 Vodopádový model

Vodopádový model je jeden ze starších modelů klasického období, který přesahuje i do dnešních dob a stále má své využití. Klasické období je vnímáno jako doba, kdy softwarové projekty byly vyvíjeny zejména na míru zákazníka až do samotné „softwarové krize“. V dobách, kdy byl tento model vyvinut, se modely životního cyklu vztahovaly spíše k ideovým představám o projektu a vyznačovaly se spíše obecným konceptuálním přístupem. Dnes již není využíván k operativním činnostem, ale spíše se více hodí na taktické až strategické řízení, a to právě díky svému obecnému modelu. Dnes se s tímto modelem setkáváme u firem a jejich prvních zakázek, kdy se pokouší firma vejít na trh



Obrázek 4 *Vodopádový model* (Zdroj: vlastní zpracování)

s novým produktem. V určitých případech projektového řízení nachází uplatnění díky své srozumitelnosti, jednoduchosti a možnosti skvěle reflektovat stav projektu na časové ose [5].

Důležitou charakteristikou tohoto modelu jsou jednotlivé etapy. Etapy na sebe velmi striktně navazují a nelze začít novou etapu, aniž by byla dokončena předešlá. Při ukončení etapy je třeba výsledky zhodnotit, protože představují vstupy pro následující etapu. Po finanční stránce by dle modelu mělo fungovat plné hrazení projektu od předprojektové fáze po údržbu. Dodavatel odpovídá pouze za to, že zaměstnanci jsou srozuměni s požadavky na realizaci projektu [5].

1.5 Agilní metodiky

Za současného trendu náhlých změn a rychlého nasazování IS/ICT systémů do různých typů podniků klasické tradiční metodiky ztrácejí na efektivitě. V dynamickém prostředí je třeba metodik, které jsou schopny rychle tvořit řešení a pružně reagovat na změny. Jedná se o agilní metodiky, které se začaly formovat ve druhé polovině 90. let a nejedná se o jeden ucelený návod, jak postupovat, ale dnes máme již několik zástupců tohoto charakteru. Ta velká změna oproti tradičním metodikám spočívá v častém předkládání výstupů zákazníkovi a rychlé reakci na jeho požadavky. Zástupci agilních metodik mají tuto základní myšlenku společnou, avšak každý zástupce má své hodnoty a specifika [4].

1.5.1 Hlavní principy agilního vývoje

Při aplikaci agilních metodik je definováno několik společných charakteristik napříč agilními zástupci. Kvůli odlišnostem jednotlivých zástupců se v roce 2001 zformoval „Manifest agilního vývoje softwaru“. Jednání probíhalo za účasti představitelů za agilní zástupce a úspěšně byly dohodnuty a podepsány společné hodnoty a principy. Z principů manifestu lze odvodit několik hlavních pravidel, která je spojují [4].

Základem je pravidelná a včasná dodávka softwaru zákazníkovi. Zákazníka zajímá hlavně dodávaná část, která má pro něj hodnotu. Jeho hlavními požadavky jsou viditelné a fungující části programu či produktu, který si objednal. Pravidelným dodáváním provádíme kontrolu, zda to, co děláme, má pro zákazníka hodnotu [4].

V předprojektové části, kdy probíhá příprava a plánování projektu, jsou u agilních metodik definované jen hrubé požadavky. Důvodem je velmi komplikované sestavení konkrétních požadavků, upřesnění i změny probíhají v rámci často probíhajících konzultací s uživateli. Tím se i výrazně zvyšuje možnost uživatelů spolupracovat přímo na projektu. Dokonce i návrh zde není samostatná fáze, ale návrh je činnost vyskytující se v jednotlivých iteracích průběhu projektu. Kvalita návrhu odpovídá schopnosti reagovat na možné změny [4].

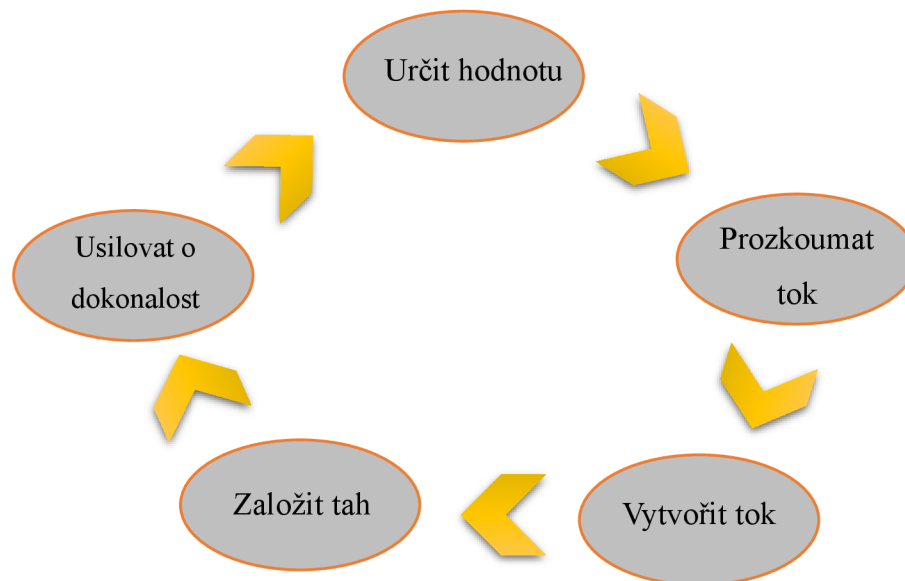
Agilní metodiky si zakládají na osobní komunikaci. Osobní kontakt je považován za hodnotnější a efektivnější než dokumentace, proto je na dokumentaci kladen menší důraz. Systém, splňující požadavky zákazníka, nestojí na dokumentaci ani plnění plánu, ale na fungujícím systému [4].

U hlavních principů agility platí, že se zde více prosazuje lidská kreativita a komunikace než u tradičních metod. Při správném aplikování těchto metod by se měla držet i zdravá morálka v pracovním týmu, čímž se buduje efektivita práce [4].

1.6 Lean development

Jestliže se máme zabývat dnes velmi často diskutovanou agilitou, musíme nejdříve pochopit, kde začal onen nápad. V padesátých letech minulého století se začala vyvíjet metoda řízení se zaměřením na rozvinutí schopností zaměstnanců. Změnu tehdejšího systému představovalo zaměření na zaměstnance, od kterých se dříve neočekávala iniciativa ani osobní rozvoj. Začal se budovat respekt a neustálý „seberozvoj“ zaměstnanců. Manažeři v podniku začali své zaměstnance učit, jak by měli sami přemýšlet nad vzniklými problémy a tím podpořili samostatnost zaměstnanců. Nebylo dále třeba určovat jejich každý jeden krok [3].

Zjednodušená filozofie leanu spočívá ve snaze provádět pouze činnosti, za které zákazník zaplatí, a omezit plýtvání nákladů a času činnostmi, za které nám nezaplatí. Na rozdíl od taktického řízení, kterým se zabývá řada agilních metodik, se lean development dá efektivně uplatnit ve strategickém řízení kvůli vazbě na podnikovou strategii. Lean development byl spíše využíván k vývoji softwaru, podobně jako dnešní populární Scrum. Model lean developmentu se dá znázornit pomocí 5 základních kroků [4].



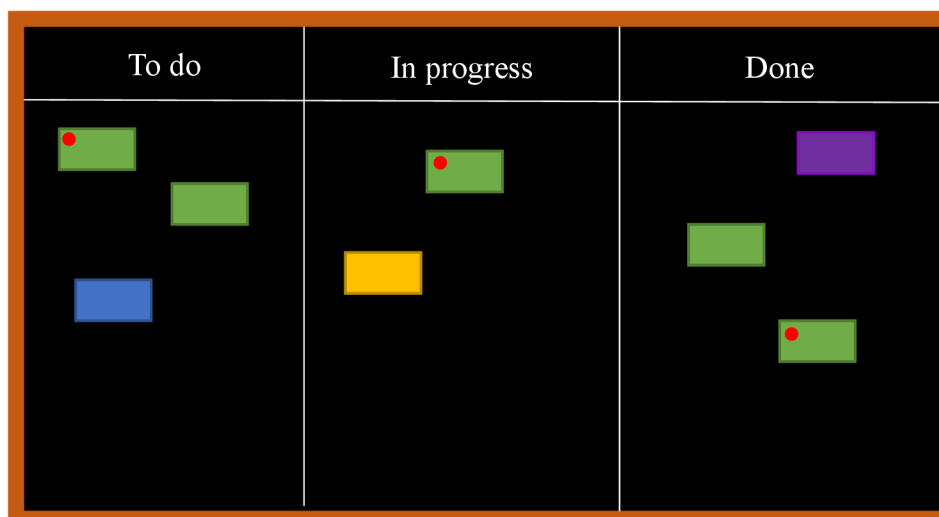
Obrázek 5 *Model lean* (Zdroj: vlastní zpracování)

1.7 Kanban

Kanban je štihlou lean metodou, která pomáhá vizualizovat pracovní úkoly na tzv. Kanban tabuli. Její původ se vztahuje ke kořenům agility, kdy se díky automobilce Toyota formoval lean development. Principem bylo vyrábět podle požadavků zákazníků a hledat nedostatky ve výrobním procesu. Hlavním cílem Kanbanu je vyvážit organizaci práce za pomoci dostupných zdrojů a identifikace slabých stránek. Práce, a to všech různých typů, se organizuje na základě označení priority. Úkoly s vyšší prioritou mají přednost před úkoly s prioritou nižší. Využitelnost metodiky Kanban lze najít u vývoje softwaru. Tuto metodu je možné používat samostatně i v kombinaci s metodikou Scrum.[3].

1.7.1 Popis Kanbanu

Nejlépe lze Kanban popsat jako tabuli, na které jsou štítky. Obsahem štítku je název úkolu, detailnější popis, označení priority a jméno člena týmu, kterému úkol náleží. Struktura Kanbanu se může pro různé typy pracovních postupů lišit. Na obrázku níže je popsána obecná forma Kanbanu. Jednoduchost zobrazení toku položek k určeným odpovědným osobám pomáhá dobře monitorovat obecný pracovní postup na projektu [3].



Obrázek 6 *Kanban board* (Zdroj: vlastní zpracování)

Sloupce označují pracovní postup. První sloupec je seznam úkolů, které čekají na začátek zpracování. Ve druhém sloupci jsou úkoly již v průběhu vývoje. Poslední sloupec je pouze pro dokončené úkoly. Více informací o průběhu a stavu jednotlivých úkolů můžeme najít uvnitř vytvořené štítku. Celková podoba Kanbanu se ve firmách může lišit na základě jejich individuálních potřeb. Podoba Kanbanu bude dále záležet na tom, pomocí kterého softwarového nástroje je metoda zpracována a jaké možnosti přizpůsobení umožňuje [3].

1.7.2 Řízení toku práce

Pracovní štítky jsou v rámci Kanbanu přesouvány v reálném čase podle stavu, ve kterém se úkol nachází. Jako denní Scrum schůzky u metody Scrum, i u Kanbanu je vhodné realizovat pravidelná setkání o aktuálním stavu [3].

Je doporučováno určovat si u sloupců maximální počet WIP. WIP je zkratka pro rozpracovanou výrobu. Znamená to maximální počet rozpracovaných úkolů v rámci sloupců. Když bude nastaven počet 5 WIP u druhého sloupce, znamená to, že do sloupce

nelze přesunout více než 5 štítků. Teprve po dokončení a uvolnění pozice lze do „In-progress“ přesunout další. Toto pravidlo zabraňuje ztrátě kontroly a přehlcení prací. Také v případě, že počet WIP není ideální a spíše průběh projektu omezuje, lze jej změnit. Tento proces je díky principům Kanbanu dobře viditelný a řízený v reálném čase. Omezující pravidlo WIP má zásadní přínos v plynulosti toku práce. Motivuje členy týmu organizovat práci a pomáhat tam, kde je třeba, aby se projekt nezastavoval [3].

Pomocí dostupných informací a metrik máme možnost sestavovat výsledné statistiky projektu. Podmínkou pro tuto analýzu je definovat podobně náročné úkoly o ideálně stejné velikosti. Lze zjistit, kolik podobně náročných úkolů je tým schopný zvládnout za stanovené časové období [3].

Průměrná doba dokončení úkolu pomáhá k výpočtu dodací doby a času potřebnému k dokončení úkolu. Doba trvání je počítána od momentu, kdy zákazník stanoví požadavky. Čas cyklu začíná spolu se začátkem prací na úkolu, znamená dobu, jak rychle je týmem výsledek realizován. Doba dodání je nezbytná ke stanovení času, který bude zákazník čekat na svoji objednávku [3].

1.8 Scrum

Scrum je v dnešní době velmi populární a často vyhledávaným agilním rámcem projektového řízení. S jeho implementací se lze setkat v širokém spektru odvětví. Je aplikován i ve stále přibývajícím množství velmi náročných a složitých projektů různých oblastí. Jeho původ však náleží řízení projektů vývoje softwaru [8].

Byl založen v roce 1995 a jeho principy a základy jsou popsány ve volně šiřitelné příručce Scrum Guide [3]. Tato příručka byla prvně veřejně vydána v roce 2010. Díky ní se Scrum mohl lépe šířit do světa a pomáhal lidem s implementací a porozuměním technik Scrumu. Jakékoliv další úpravy od roku 2010 byly spíše menší změny vedoucí ke zlepšení metody [8].

Na rozdíl od tradičních metod projektového řízení je Scrum popisován jako pozice „mlýnu“ ve hře rugby. Jsou mu připisovány společné vlastnosti jako adaptivita, rychlost a samostatná organizace [4]. Základní myšlenkou je společný cíl týmu společně táhnout jedním směrem. Jedině dobře fungující a organizovaný tým je schopen vytvářet hodnotu odvedené práce [3].

Jedná se o iterativní model, kde jednotlivé iterace jsou nazývány „sprint“. Tým je složen z lidí kompetentních k danému typu práce. Členové týmu mají odborné znalosti i dovednosti a jsou otevřeni novým nápadům a novým zkušenostem [3].

1.8.1 Základy Scrumu

Základy Scrumu jsou založeny na efektivním řízení práce. Cílem je zamezit takovým činnostem vedoucím ke zbytečným pracím a zaměřit se pouze na to podstatné. To znamená omezit plýtvání energií, času a nákladů na minimální úroveň. Ve Scrumu jsou oceňovány osobní zkušenosti vyplývající ze znalostí získaných důkladným pozorováním. Tyto společné znaky lean thinking a empirismu společně utvářejí hodnoty Scrumu, tak jak jsou sepsány v příručce Scrum guide [8].

Podstatou spokojenosti o dobře odvedené práce je, aby práce byla vidět. Je velmi důležité, aby výsledky probíhající pracovní činnosti byly známé jak členům týmu, tak zákazníkovi a zainteresovaným stranám. Všechny tyto skupiny, které jsou v projektu zapojeny, by měly mít přehled o probíhající práci, stavu a budoucímu postupu, na kterém se bude pracovat [3].

Řídit projekt je třeba v krátkých intervalech. Pozorováním klíčových činností spojených s vytvářením požadované hodnoty a stavu průběhu sprintu kontrolujeme průběh projektu. Efektivním vedením týmu analyzujeme odchylky a vytváříme plán pro přizpůsobení [3].

Veškeré odchylky od původního plánu musíme zpracovávat včas, aby nedocházelo zbytečně k dalším. Jestliže výsledné výstupy nesplňují limity nebo je výsledkem nepoužitelný produkt, je třeba reagovat a zahájit změnové řízení. Za pomoci dobře kvalifikovaného týmu lze dosáhnout úpravy dosažených výstupů nebo dalších procesů spojených s řízením změn. Za těchto okolností musí umět Scrum tým agilně reagovat a ponaučit se z nových zkušeností [3].

1.8.2 Scrum tým

Projektový tým u Scrumu je spíše malá skupina lidí, proto se používá výraz Scrum tým. Je složen z několika rolí [3]:

- Scrum master – jeden člen týmu
- Product owner – jeden člen týmu

- Vývojáři – několik osob

Jelikož je Scrum tým složen pouze z menší skupiny lidí, nepanuje zde žádná striktní hierarchie. Tým je složen z dostatečně kvalifikovaných lidí, kteří jsou zaměřeni na cíl produktu a ten je pro všechny společný. Všichni členové mají odpovídající kvalifikace k naplnění hodnot každého sprintu. Nejedná se o princip „všichni jsou odborníci na všechno“, ale ta pravá podstata je vzájemně se doplňující skupinu [3].

Uvnitř týmu se na rozhodování o tom kdy, kdo, co a jak podílejí všichni členové. Jelikož se často týmy skládají z deseti a méně lidí, zůstávají velmi dobře soběstačné a jsou samostatně řízené. Velikost týmu by měla být dostačující vzhledem k pokrytí práce, na kterou je určen a také aby zůstal dostatečně přizpůsobivý. U menších týmů taktéž platí lepší komunikace a vyšší produktivita [3].

Odpovědnost za činnosti spojené s produktem včetně vývoje, údržby, kontroly, spolupráce se zákazníkem nese Scrum tým. Tým funguje v rámci průběhu projektu ve stanovených rytmech neboli sprintech. Ty mají určenou dobu trvání a každý sprint má na konci určitý cíl. Cílem by měl být vždy přírůstek, který přinese hodnotu a bude užitečný pro výsledný produkt. Kontrola a odpovědnost se deleguje na základně tři výše vzpomínaných rolí Scrum mastera, Product ownera a vývojářů [3].

1.8.2.1 Vývojáři

Vývojáři v metodice Scrum nemusejí být zpravidla lidé, kteří píšou kód programu. Jedná se o zažitý význam slova při vzniku Scrumu, kdy se jednalo skutečně o softwarové vývojáře. V dnešní době, kdy je tato metoda velmi populární, tomu tak již doslovně není a termín je pouze obecný. Dnes se metodika Scrum totiž nepoužívá pouze u vývoje softwaru, ale nalezneme ji i u řady dalších oblastí mimo informační technologie. Vývojářem může být myšlen člověk jakéhokoliv zaměření či profese [3].

Vývojáři jsou členy týmu, kteří se aktivně podílejí na výstupech každého sprintu. Aplikují své dovednosti a zkušenosti, které jsou třeba k vytvoření stanoveného cíle. Jsou to lidé s bohatými znalostmi, které lze uplatnit pro různé pracovní oblasti a zaměření produktu [3].

Tým vývojářů se u projektu věnuje tvorbě přírůstků směřujícím k výslednému produktu. Vývojáři mají řadu odpovědností v rámci projektu. Jedna z nich je vést řádně backlog sprintu a vytvořit plán pro sprint. Odvedenou práci musí plnit za dodržení kvality

podle definice „Definition of Done“. Jedná se o tým složený z profesionálů určených k dosažení stanoveného cíle. Jako tým musí komunikovat, spolupracovat a také mají vzájemnou odpovědnost za odvedenou práci [3].

1.8.2.2 Product owner

Role Product ownera ve Scrum týmu znamená mít na starost maximalizaci hodnoty produktu. Klíčem k úspěchu je důvěra vedení podniku vůči práci Product ownera [3].

K zodpovědnostem Product ownera se řadí efektivní řízení produktového backlogu. To zahrnuje tvorbu nových položek a komunikace položek backlogu se svým týmem. Product owner má také na starosti vybírat, které položky jsou v dané situaci prioritou a které mohou ještě počkat. Vyhodnocuje důležitost jednotlivých položek backlogu v dané situaci. Pouze on má pravomoc přidávat, odebírat nebo upřednostňovat položky v backlogu. Není totiž řízený zájmem skupiny, ale vystupuje pouze sám za sebe a za své rozhodnutí. V rámci své práce zastupuje potřeby a zájmy několika zainteresovaných stran [3].

Jestliže má Product owner efektivně řídit produktový backlog, musí usilovat o rozvoj prospěšné komunikace v týmu. Jeho zodpovědností je dobře nastavená komunikace během celého projektu [3].

1.8.2.3 Scrum Master

Scrum master v týmu je osoba, která je v roli odborníka na metodiku Scrum. Jeho největší vědomostní oporou je příručka Scrum Guide, podle které postupuje a řídí tým podle definovaných pravidel. Svě znalosti nabitě z příručky pomáhá přenášet na všechny členy týmu a pomáhat jim pravidla pochopit a aplikovat. Mimo jiné má také velký vliv na aplikaci Scrumu v daném podniku. Jestliže podnik plánuje zavedení Scrum metodiky, je to právě Scrum master, kdo pomáhá tento cíl uskutečnit, včetně plánování, školení a zavádění. Jeho povinností je i uplatňování svých znalostí při komunikaci zainteresovaných stran a Scrum týmem kvůli odstraňování bariér v jejich spolupráci [3].

Jeho zodpovědností je po každém sprintu vyhodnocovat jeho průběh. Hodnocení se skládá z retrospektivy, jak členy bylo třeba řídit a co by se dalo na uplynulých činnostech do budoucna zlepšit. Tímto podporuje vlastní organizovanost Scrum týmu a efektivně řídí průběh projektu. Ke členům může přistupovat i jednotlivě a podporovat je v jejich

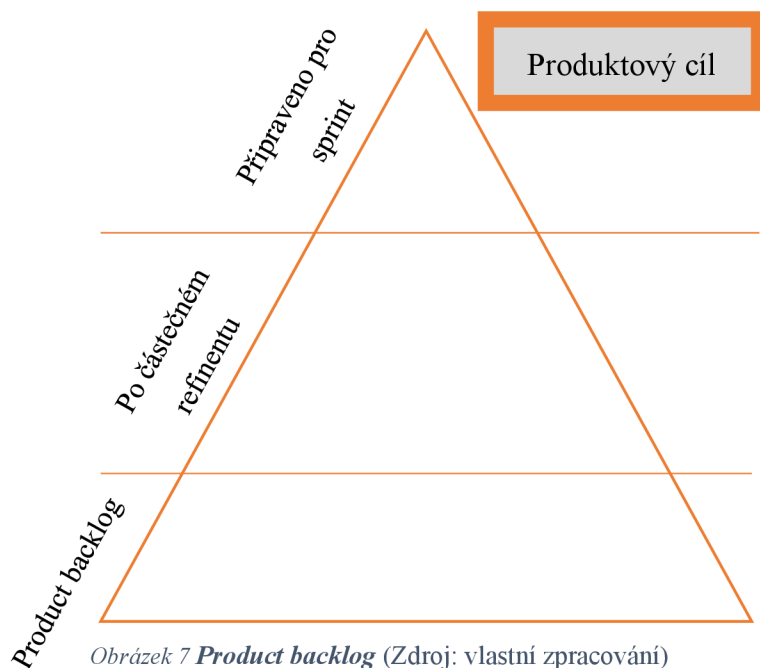
sebeřízení a multifunkčnosti. Scrum tým analyzuje, jestli práce úspěšně přináší hodnotné přírůstky, které jsou podle pravidel „Definition of Done“ [3].

Spolupráci mezi Scrum masterem a Product ownerem spočívá v hledání efektivního plánu pro správu backlogu a spolupráci na definování cíle produktu. Pomáhá Product ownerovi s interpretací položek backlogu, tak aby všem členům bylo řádné vysvětleno, co která položka představuje a jak naplnit její podstatu. Spolu s Product ownerem usiluje o zjednodušení komunikace a spolupráce se zainteresovanými stranami, kde je důraz na plnění jejich požadavků a potřeb [3].

1.8.3 Product backlog

Jeden z prvních kroků ve vedení projektu podle Scrumu je formulace cíle produktu. Jedná se o ujasnění, čeho má být produktem dosaženo, jaká je vize u plánovaného produktu a co má splňovat. To vše je starostí Product ownera a ten je i zodpovědný za produktový backlog [9].

Je to seznam tvořen z položek, co vše je třeba ke splnění zákaznických požadavků. Položky v backlogu jsou seřazeny podle priority a lze tedy mít přehled o prioritních položkách. Tento seznam se od momentu, kdy vznikne, neustále vyvíjí a rozšiřuje o další položky. Svou formou je i plánem projektu, který je v daném okamžiku jedinečným pohledem na produkt. Je to ucelený seznam všech položek, které tým musí pro produkt vykonat seřazený podle priority [9].



Obrázek 7 **Product backlog** (Zdroj: vlastní zpracování)

V každém projektu může existovat pouze jeden backlog. Product owner musí vědět, jakým způsobem označit položky z hlediska důležitosti pro vývoj produktu. Ve Scrum týmu pouze Product owner reprezentuje zájmy zainteresovaných stran, a to včetně zákazníka i Scrum týmu. Musí brát v potaz zájmy všech zainteresovaných stran a podle toho rozhodovat o důležitosti položek v backlogu [9].

Produktový backlog je složen ze širokého spektra položek. Značná část je tvořena novými požadavky zákazníka, ale též je tvořen technologickými poznámkami vývojářů. Vývojáři mohou do backlogu přispívat svými podněty na vylepšení technologických řešení produktu. Obsahově by každý dobře vedený backlog měl být rozsáhlý [9]. Dalo by se tedy říct, že položky produktového backlogu jsou z velké části tvořeny přímo Scrum týmem. V případě položek po refinentu, tedy bližšímu rozboru položek, které je tým schopen zpracovat během jednoho sprintu, jsou označeny jako připraveno pro sprint [3].

1.8.4 Definition of done

Po každém sprintu jsou vyžadovány relevantní výstupy. V ideálním případě po uplynulém sprintu vzniknou výstupy pro přímý přírůstek k produktu. Již před prvním sprintem je třeba si ujasnit kritéria, které výstupy musí splňovat, aby položka backlogu byla přijatelná k možnému odeslání. Tato kritéria si musí dostatečně dopředu stanovit Product owner a Scrum master i včetně týmu, zkrátka celý Scrum tým. Přibývající

zkušenosti Scrum týmu přispívají dovednostnímu rozvoji, a to s sebou nese i možnost časem kritéria DoD vylepšovat a rozšiřovat, nejedná se tedy o permanentní seznam pravidel [9].

Kritéria DoD představují formálně definovaná pravidla, které musí položky po sprintu splňovat, aby dosáhly požadovaných kvalit. Teprve po úspěšném splnění těchto pravidel se dá položka považovat za splněnou. Za situace, kdy položka nesplňuje požadované kvality, vrací se zpět do backlogu a není ani součástí sprint review [3].

Základní DoD kritéria, se kterými se lze nejpravděpodobněji setkat:

- *„Vytvořeno*
- *Otestováno a opraveno*
- *Schváleno Product ownerem*
- *Otestována a opravena integrace s ostatními přírůstky*
- *Vytvořena dokumentace úrovně XYZ*
- *Uloženo v úložišti X a pracovní verze odstraněny*
- *Prošlo sprint review bez připomínek“* [3, s. 119]

1.8.5 Sprint backlog

Sprint backlog je tvořen položkami části produktového backlogu připravených pro týmový sprint. Tyto položky se během přípravy rozkládají na detailnější úroveň. Během probíhajícího sprintu jsou položky sledovány a lze odhadnout, kolik úsilí položka vyžaduje pro dokončení [9].

Tento seznam položek určených ke sprintu je vytvořen na základě rozhodnutí vývojářů. Vývojáři se mezi sebou dohodnou, které položky na sprint backlog zařadí a na těch budou i ve sprintu pracovat. Sprint backlog mimo položky připravené ke sprintu zahrnuje i cíl sprintu, tedy proč se sprint bude konat a plán, jak přírůstku dosáhnout. V případě, že dojde k novým poznatkům během sprintu, je možné sprint backlog aktualizovat. Na schůzce denního Scrumu je poté konzultován pokrok a poznatky. Proto by měl být sprint backlog natolik rozsáhlý, aby sledování pokroku umožňoval [3].

1.8.6 User story

Pro co nejpřesnější pochopení představ zákazníka o funkcionalitách produktu je ve Scrumu implementován prvek user story. Při úvodní analýze uživatelských požadavků

může být sepisování požadavků formou user story velmi efektivní. Zapsané user story mají jasnou strukturu, kterou je třeba dodržet. Dobře zapsaná user story nám odpovídá na klíčové otázky:

- Kdo?
- Co?
- Proč?

Odpověďmi na tyto otázky získáme představu, pro koho je požadavek zpracováván, co je jeho obsahem a k čemu by to mělo být [14].

Je prací Product ownera dokázat přenést ideu zákazníka do výsledného produktu. Mělo by se jednat o jednoduše pochopitelnou funkcionalitu, která je dostatečně malá na to, aby v sobě nesla všechny důležité informace, a ne příliš velká, aby neobsahovala zase nadbytečně mnoho. Nad míru velká user story je pro tým těžce uchopitelná, proto je třeba věnovat úsilí, aby svou podobou přinášela relevantní hodnotu. Lidé dokážou věci vždy lépe a rychleji pochopit pomocí vizuálního kontaktu, proto je-li to možné, měl by se Product owner snažit zachytit představu zákazníka ve vizuální podobě pomocí jednoduchého obrázku [14].

User story je tedy o pohledu zákazníka, v čem spočívají hodnoty a jaké role se v daném user story nacházejí. Jinými slovy user story je nějaký celek, který se skládá z několika dalších úkolů. Tyto úkoly, na kterých vývojový tým pracuje, poté tvoří dohromady potřebné hodnoty user story [14].

1.8.7 Sprint

Sprinty jsou ve Scrumu rytmicky naladěné cykly vývoje produktu, které jsou plánovány na přesné datum s pevně stanovenou dobou trvání. Jednou za měsíc dochází ke kontrole odvedeného postupu směrem k cíli produktu. Při kontrole se zvažuje aktuální stav a tomu je postup řádně přizpůsoben [3]. Hlavními částmi sprintu jsou:

- Produktový cíl
- Plánování sprintu
- Denní Scrum
- Review sprintu
- Retrospektiva sprintu [3]

Do sprintu se tým dostává s již určitým plánem, takže během sprintu nedochází k narušení jeho cíle. Během sprintu probíhá aktivní práce s položkami z produktového backlogu, kdy se upravují samotné položky, tak i jejich označení priority. Do sprintu je sestaven seznam pravidel DoD a k jeho změně během sprintu nedochází. Za situace, kdy se cíl sprintu ukáže jako nežádoucí, je na odpovědnosti Product ownera, aby sprint ukončil [3].

V praxi tým veškerý čas stráví ve sprintu. Sprintem není pouze vývoj produktu, ale jeho část tvoří i opravy chyb, týmové vzdělávání, čas potřebný k refinentu apod. Obecně se dá uvažovat o poměru 60/40, kdy 60 % času je vyhrazen na práci a 40 % na vše ostatní [3].

1.8.8 Denní Scrum

Denní Scrum jsou pravidelně nastavené schůzky v období probíhajícího sprintu. Vznikají za účelem pravidelné kontroly stavu činností na projektu a sledování postupu. Je důležité mít jistotu o činnostech směřující ke správnému cíli sprintu [3].

Jedná se o kratší přibližně 15minutové jednání mezi vývojáři Scrum týmu. Za situace, kdy se na tvorbě sprint backlogu významně podílejí i Scrum master a Product owner, jsou součástí denního Scrumu také [3].

Formu schůzek si vývojáři stanovují sami. Je však důležité dbát na nezanedbatelný obsah schůzky. Tím by vždy měly být dodrženy podmínky, kdy veškerý pokrok vede k plánovanému cíli a vždy by měl být během schůzky vytvořen plán pro další den, který se pak znovu podrobí kontrole. Dodržování těchto základů vede k podpoře samořízení a soustředěnosti na produkt [3].

Ve Scrumu se lze setkat s formou Kanbanu nazývanou Scrum board. Jedná se o efektivní metodu, jak vizualizovat plán a postup. Stejně jako u Kanbanu se jedná o tabuli rozdělenou do několika sloupců. U Scrumu je možné použít označení sloupců „To do“, „In progress“ a „Done“. Ve sloupcích jsou rozřazeny položky z produktového backlogu

a k nim přiřazené vývojářské úkoly. Úkoly na sobě nesou označení iniciálu člena týmu, který na úkolu pracuje. Z výhod Kanbanu i zde vyplývá velmi dobrý přehled v každém momentu sprintu o aktuálním stavu práce. Takto přenesená struktura Kanbanu je velice jednoduchá a v případě využití u Scrumu je vhodné ji doplnit o sloupec

„schváleno Product ownerem“, „testing“ apod. Modifikace Scrum board bude vždy záležet na potřebách podniku a na domluvě Scrum týmu [3].

1.8.9 Review sprintu

Po skončení sprintu je třeba odvedenou práci zpětně zhodnotit. K tomuto účelu je ve Scrumu zaveden sprint review. Jeho účelem je rozebrat výsledek sprintu, který může přinést nové poznatky a podněty ke zlepšení. Získané poznatky z výsledku sprintu poté Scrum tým prezentuje výstupy zainteresovaným stranám. Cílem diskuse je zhodnotit pokrok vedoucí směrem k cíli produktu [3].

Sprint review je nezbytný hlavně ke společnému jednání o věcech, které se podařilo dokončit. Prezentování pokroku stakeholderům je významnou částí review, nikoli však jedinou. Vývojáři odvedenou část představí a účastníci review si dané výstupy prozkoumají. Po prezentaci následuje diskuse nad přírůstkem. Během diskuse je nutné soustředit se na hlavní smysl review. Tím je získat zpětnou vazbu týkající se přírůstku [3].

Je sestaven seznam položek, které splnily kritéria DoD a jsou považovány za hotové. Položky, které tyto pravidla nesplňují, se na sprint review ani neprezentují a jsou vráceny zpět do produktového backlogu a znovu k plánování sprintu. Je velmi důležité, aby tým dodržel zásady Scrum a položky, které nesplní kritéria DoD, se automaticky nepřesouvaly do dalšího sprintu. Každý sprint může mít zcela jiný cíl a také jiné položky připravené ke sprintu. Vše je otázkou plánování a položky, které DoD nesplní, se do sprintu dostanou, až to bude vhodné [3].

Doba trvání review měsíčního sprintu je doporučena pro limit 4 hodin. V případě sprintu, který trvá kratší dobu, je i trvání sprint review kratší [3].

1.8.10 Shrnutí Scrumu

Aplikace metodiky Scrum v projektovém řízení znamená pro podnik rozsáhlé změny. Zejména působí změny ve spolupráci vývojářského prostředí a projektového řízení směrem k agilitě. Předmětem Scrumu nejsou způsoby, jak a za jakých podmínek vzniká tým, nepopisuje, jakým způsobem se projekt řadí do dané organizace ani jak se definují základní informace o produktu. [3].

Jeho obsahem je popis nezbytných prvků a činností patřící k této metodice. Nechává volný prostor firemním potřebám v těch oblastech, které rámeček nepopisuje. Nejčastěji jde o vytváření hodnot produktu jak k zákazníkovi, tak i k uživateli. Konkrétní způsoby, jak vzniká produkt, tým nebo jak se popisují položky backlogu Scrum neřeší, není-li uvedeno jinak [3].

Pro podnik, který razí strategii tradičních metod řízení vývoje softwaru, bude velmi obtížná transformace prostředí, jak popisuje Scrum guide. Na první pohled se může Scrum jevit jako jednoduchý na implementaci. Pokud má vedení firmy pocit, že Scrum poskytuje spíše velký prostor volnosti a je otevřený nápadům, jak si jej upravit dle své libosti, nikdy tato metoda nebude fungovat tak, jak byla navržena a myšlena [3].

2 Analýza současného stavu

Obsah následující kapitoly je věnován analýze současného stavu podniku, jenž zde bude představen. Jak bylo již zmíněno výše, společnost, která je zde popisována, pracuje s utajovanými informacemi, a proto bude zachována její anonymita. V rámci bakalářské práce je zvolen pracovní název firmy XYZ, spol. s r.o. Bude popsána její charakteristika, organizační struktura, proces vývoje a řízení softwaru a využívané nástroje. Zdroje informací pro tuto kapitolu vycházejí z veřejně dostupných zdrojů o společnosti, osobních zkušeností po dobu výkonu práce ve firmě a rozhovorů se zaměstnanci a vedením společnosti.

2.1 Představení společnosti

Popisovaná společnost působí na trhu informačních technologií od roku 1993. Se sídlem v Brně se zaměřuje na vývoj softwarových produktů pro vojenský průmysl. Mimo vývoj vlastních softwarových produktů je i implementátorem a poskytovatelem dlouhodobé technické podpory Armádě České republiky. Svým klientům, jimiž jsou různé části armády, poskytuje i školení a připravuje uživatele na práci s poskytovanými systémy.

Své produkty zaměřuje na několik specifických oblastí. Jedná se o portfolio produktů a služeb v sektoru výzkumu, vývoje a implementace specializovaných řešení pro obranný sektor. Hlavním produktem je systém velení a řízení vojenských štábů a jednotek.

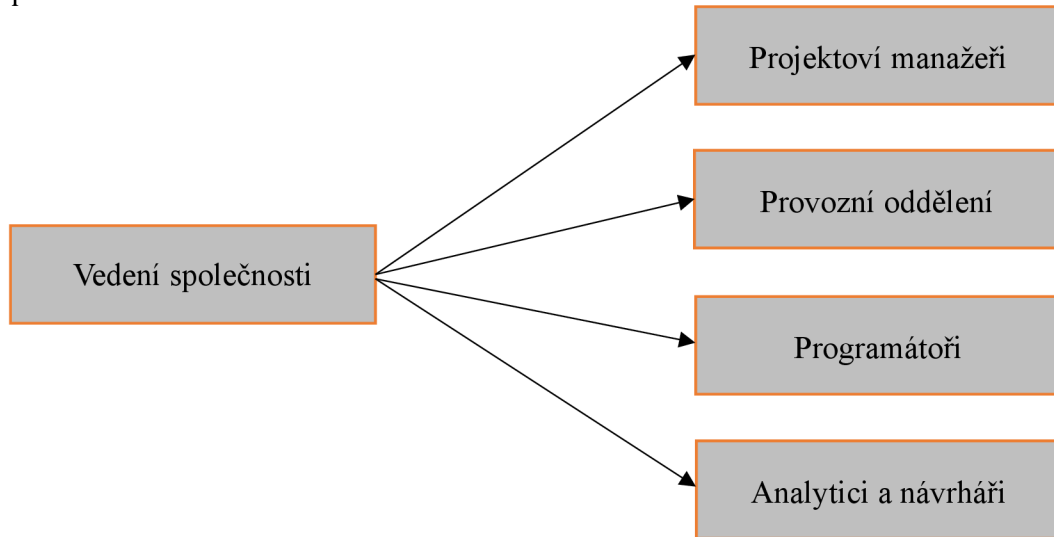
2.2 Organizační struktura

Na trhu se zaměřením na vývoj softwaru se firma XYZ, spol. s r.o. dostává počínaje rokem 1993 jako samostatná firma typu s.r.o. Roku 2008 přichází velká změna a firma se stává součástí větší skupiny firem ABC a.s. mateřské společnosti ABC N.V. V současné době se jedná o skupinu 11 firem.

Za zájmy společnosti XYZ, spol. s r.o. vystupuje a jedná samostatně jeden jednatel. V rejstříku firmy je dále vedena nadřazená společnost ABC Holding a.s. jako jediný společník firmy XYZ, spol. s r.o.

Za vedení podniku je zodpovědný ředitel. Ředitel má i svého zástupce, který do značné míry díky své odbornosti a schopnostem zastává obchodní činnosti. Do vedení

podniku se řadí i asistentka ředitele a vedoucí pobočky. V hierarchii pod vedením podniku jsou na stejné úrovni provozní skupina, programátoři, analytici společně s návrháři a projektoví manažeři. Skupina programátorů, provozu a analytiků s návrháři mají vybrané zástupce, kteří za dané sekce zodpovídají. Ti se za zájmy svých oddělení účastní i projektových rad a mají na starosti rozdělení práce mezi zaměstnance spadající do svých skupin.



Obrázek 8 *Organizační struktura* (Zdroj: vlastní zpracování)

2.3 Jednání se zákazníkem

Pro vybranou firmu XYZ, spol. s r.o. je v roli nejdůležitějšího zákazníka Armáda České republiky. Veškeré projednávané zakázky a jejich financování je pod kontrolou a správou Ministerstva obrany ČR. Zákazníkem je Armáda ČR, která má rozsáhlou strukturu skládající se z mnoha ústavů a oddílů vyskytujících se na různých místech České republiky.

2.3.1 Získávání zakázek

Nabídky a požadavky na dodávku nových produktů jsou realizovány díky dlouhodobé spolupráci se zákazníkem, založené na dodávání produktu IS VŘ PozS – Informačního systému velení a řízení pozemních sil do prostředí Ministerstva obrany ČR.

Zakázky jsou získávány třemi způsoby:

1. Za přímého kontaktování společnosti od Ministerstva obrany ČR v režimu jednacího řízení bez uveřejnění. Zadavatel předloží své požadavky, které souvisejí s rozvojem informačního systému velení a řízení pozemních sil.
2. Ve spolupráci s jinými společnostmi, které se podílejí na dodávání svých produktů nebo při modernizaci bojové techniky do prostředí Ministerstva obrany ČR. Tyto zakázky firma získává kvůli potřebné modernizaci techniky vyžadující integraci IS VŘ PozS za účelem jednosměrné či obousměrné výměny informací se systémem.
3. Zahraniční obchodní příležitosti společnost získává přímou účastí výběrového řízení nebo v rámci spolupráce s místním zastoupením.

2.4 Produktové portfolio

Popisovaná firma působí na velmi úzkém trhu v oblasti výzkumu, vývoje a implementací specializovaných informačních a geografických informačních systémů. Dodává softwarová řešení pro svého hlavního zákazníka, Armádu České republiky.

Mimo vývoj svých vlastních informačních systémů a jejich součástí je také implementátorem. Pro své klienty nabízí dlouhodobou a pravidelnou podporu v podobě zavádění nových aktualizací, přizpůsobování softwaru na míru, vývoji dalších zcela nových vylepšení a také školení pro adminy i uživatele softwaru.

Hlavním produktem firmy je systém velení a řízení pozemních sil C4ISTAR. Ty jsou určeny pro velitele a štáby vojenských jednotek pro plánování a řízení jejich činností. Společnost XYZ, spol. s r. o. má ve svém portfoliu tři vzájemně spolupracující systémy. Jedná se o automatizovaný systém velení a řízení, bojový vozidlový informační systém a systém sesednutého vojáka.

Pro rodinu těchto vzájemně spolupracujících systémů pro štáby, vozidlo i sesednuté vojáky je stěžejní podpora interoperability a integrace senzorů právě do systémů typu C2/C4ISTAR. Díky těmto klíčovým vlastnostem systémů, které firma vyvíjí a dodává, může mít na trhu velmi silné dominantní postavení. Další nepostradatelnou vlastností je podpora specifikací Multilateral Interoperability Programme, zkráceně MIP, v prostoru operace zúčastněných zemí NATO. Jedná se o klíčovou vlastnost systému, který dokáže spolupracovat oboustrannou výměnou dat pomocí sdíleného modelu i se systémy jiných zemí a vývojářů angažovaných také v programu MIP [13].

2.5 Používané softwarové nástroje a technologie

2.5.1 SAP

Pro řízení veškerých podnikových zdrojů a vedení přehledů o projektech je ve firmě XYZ, spol. s r.o. implementován ERP systém SAP. Není však přístupný nikomu jinému než řediteli, který má jako jediný přístup ke všem vedeným datům v systému.

Systém SAP je zaveden i pro vykazování docházky. Řešení je nasazeno v podobě aplikace MS Excel propojené se systémem SAP a zaměstnanci tak vykazují odvedenou práci přímo do projektů vedených v ERP systému. Vykazování práce je v podobě několika typů činností vykonávaných v rámci projektu.

Pro potřeby projektového managementu, čímž mohou být přehledy o plánovaných MD v projektu, detailní rozpis práce v projektu, finanční přehled nazýván IPIS, či podrobnější náhled na informace vedené v projektu, je zřízeno oddělení nadřazenou společností ABC a.s. pro tyto potřeby. Požadované informace jsou zasilány na vyžádání či v dohodnutých pravidelných měsíčních intervalech.

2.5.2 Visual Studio

Veškerá práce programátorů je tvořena ve vývojovém prostředí Microsoft Visual Studio. Jedná se o velmi komplexní pokročilé vývojové prostředí pro různé platformy .NET a pro tvorbu kódu i s možností práci na GUI pomocí grafického designéru. Pro potřeby programátorů je ke tvorbě produktu firmy klíčová i práce s databází a nesmí chybět i podpora verzování [10].

Další technologie a nástroje využívané programátory:

- Jazyk C#
- WPF – Windows Presentation Foundation
- Visual studio CODE
- WSL – Windows Subsystem for Linux
- TFS – Team Foundation Server

2.5.3 Microsoft Office

Velká část činností ve firmě je realizována skrze balík kancelářských programů od firmy Microsoft.

Pro tvorbu textových dokumentů, jako jsou příručky, analýzy, zápisy z jednání, projektová i programová dokumentace, je využíván program Word. Prezentování nových i stávajících produktů či ukázkové materiály při školení jsou vytvářeny v programu Powerpoint. Pro správu pošty je k dispozici na stanicích Outlook.

Zaměstnanci mají k dispozici k určitým činnostem, typicky vyúčtování nákupu nebo cestovní příkaz, dokument Excel, jenž je připravený a propojený přímo se systémem SAP a tyto činnosti jsou řešeny v ERP. Pro vykazování odvedené práce na projektech je též používán dokument Excel zapisující informace do systému SAP a zároveň z něj data o dostupných projektech čerpá.

2.5.4 Hyper-V

V prostorách firmy je zřízeno testovací prostředí pro testování funkčnosti softwaru v simulovaných podmínkách. K těmto účelům je na počítačových stanicích v rámci operačního systému Windows 10 zřízen program Hyper-V. Slouží k virtualizaci prostředí a díky němu je možné efektivně testovat od jednotlivých verzí aplikace až po zcela odlišné vyvíjené systémy na libovolných verzích operačního systému. To vše díky možnostem virtualizace a funkcím, které Hyper-V pro účely podniku poskytuje [11].

2.5.5 Jira

Pro potřeby projektového řízení je ve firmě zaveden nástroj Jira. Jde o nástroj podporující firemní činnosti týkající se projektů a jejich řízení. Pomocí nástroje Jira lze

plánovat a sledovat proces vývoje až po jeho dokončení. Umožňuje nám pohled na celkový životní cyklus produktu [12].

Typické využití lze nalézt při vývoji softwaru, kde Jira umožňuje agilně evidovat nově zaváděné funkcionality, chyby, opravy a zkrátka veškeré úkoly, které se týkají projektu. Každý z úkolů má zavedou strukturu obsahující název, popis, přiřazení, k jakému projektu úkoly patří, komu je úkol přidělen a kdo na úkol může dohlížet a spoustu dalších funkcionalit [12].

Jira se řadí k softwarovým nástrojům, který si lze bohatě přizpůsobit svým potřebám. Společnost XYZ, spol. s r. o. má zkušenosti s tímto softwarovým nástrojem teprve krátce a Jira je zavedena do firmy začátkem roku 2022. Na současném nastavení je tato raná zkušenost znát, avšak s přibývajícím zkušenostmi vznikají nápady na optimalizaci. Jelikož se potřebám projektového řízení tento nástroj velmi osvědčil, jsou do budoucna plánované změny, jak implementovat více firemních procesů do prostředí Jira pomocí dalších rozšiřujících pluginů. Důležité ovšem je nejen implementovat více firemních procesů, ale také optimalizovat současný systém zadávání úkolů, kterému bude věnována část v kapitole návrhu na řešení.

Zavedení nového nástroje Jira v roce 2022 mělo pro procesy projektového řízení zásadní vliv. Nástroj Jira nahradil časově velmi náročně a nepřehledné řízení vývoje softwaru a projektů pomocí programů MS Office. Tento krok vpřed výrazně ovlivnil orientaci v řízení projektů, evidenci chyb a nových funkcionalit systému a postupně se stále zapracovávají další firemní procesy. Zkrátka se tato změna ve firmě velmi osvědčila a brzké budoucnosti je třeba pracovat na optimalizaci. Díky zavádění nových technologií se firma snaží držet krok v této moderní době.

2.5.5.1 Prostředí Jira

Prostředí Jira je uzpůsobeno dle potřeb podniku. V projektech lze nalézt pouze Kanban tabule rozdělené dle kalendářních let. Veškeré probíhající projekty jsou považovány v rámci prostředí za komponenty. Kanban tabule tak obsahuje veškeré úkoly, které lze díky rychlým filtrům uzpůsobovat zobrazení dle komponent značících projekty. Do nástroje byly implementovány i názvy aplikací a softwaru, které firma nabízí pod funkcionalitou s názvem epic. Označení nové úlohy pomocí odkazu na epic pomáhá v přehlednosti a navigaci mezi úlohami.

Součástí zápisu do Jira nejsou pouze nové úkoly, ale i evidence nových chyb a také nápadů na zlepšení, školení, příprava na armádní cvičení a další. Veškerý tento obsah je v nástroji Jira označován jako úloha.

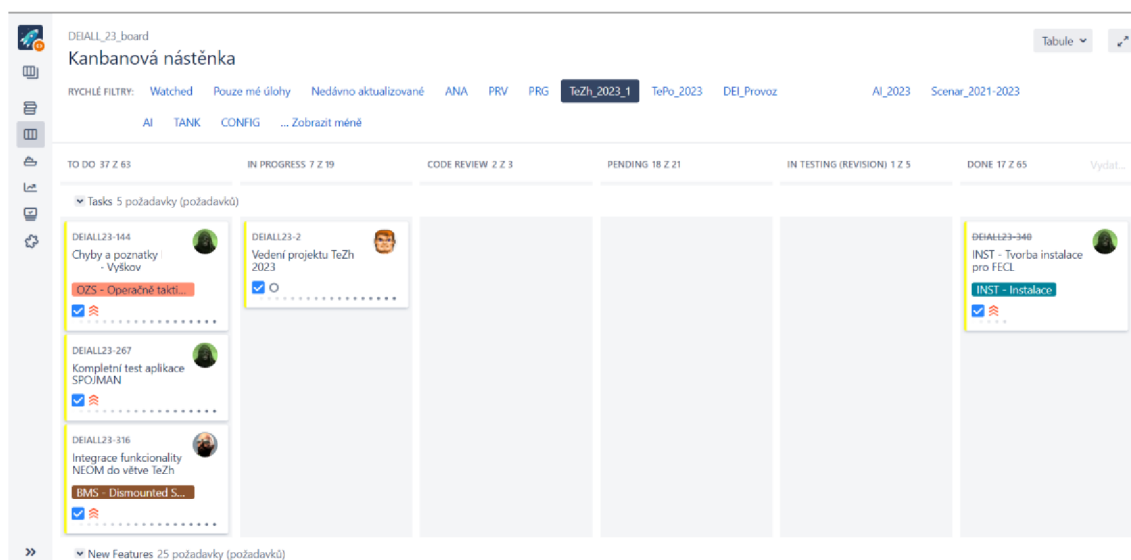
Při zakládání nové úlohy je zvolen

1. projekt, zde tedy Kanban tabule pro daný rok
2. typ úlohy, zda jde o nový úkol nebo chybu
3. název úlohy
4. komponenta, tedy o jaký projekt se jedná
5. je možné přiložit štítky, jaké aplikace se úloha týká nebo se kterou souvisí

Součástí vytváření nové úlohy může být i řada dalších rozšiřujících informací včetně připojených obrázků.

Ve firmě je pro řízení procesu vývoje softwaru využívána Kanban tabule. Ta obsahuje několik fází úlohy:

- To do
- In progress
- Code review
- Pending
- In testing (revision)
- Done



Obrázek 9 *Kanban tabulka JIRA* (Zdroj: Vlastní zpracování)

Manipulovat s úlohami mezi těmito etapami může pouze vedoucí projektu.

2.6 Proces vývoje

V kapitole vývojového procesu bude v práci popsán proces řízení a realizace vývoje softwaru. Dále zde budou popsány druhy projektů, které ve firmě probíhají, a struktura projektového týmu.

2.6.1 Životní cyklus nového projektu

Hlavním klientem a zadavatelem zakázek pro podnik je Armáda České republiky. Zpravidla platí, že z Ministerstva obrany ČR přijde požadavek a od vedení firmy se očekává návrh řešení. V reakci na přijaté požadavky následují jednání s cílem získat co nejpřesnější zadání a představu zadavatelů o konečném řešení. Prvotní jednání jsou v režii vedení společnosti v čele s ředitelem. Výstupem je smlouva obsahující veškeré dohodnuté informace týkající se specifik požadavků na projekt. Projekt začíná, jakmile je smlouva všemi stranami odsouhlasena a podepsána. Až teprve sjednanou a podepsanou smlouvou začínají veškeré práce, bez podpisu smlouvy nelze na projektu začít pracovat.

V průběhu realizace softwaru je s ohledem na zákazníka nutné dodržovat předpisy a standardy definující postupy a vlastnosti systému. Jedná se o vojenské požadavky na proces realizace, které definuje např. AQAP, požadavky národních vojenských norem, standardy a postupy NATO STANAG, doporučení MIP, ADatp a NAF.

Při startu nového projektu je vedením firmy sestaven projektový tým. Vedením je určen vedoucí projektu a také tým vývojářů, kteří se budou na řešení podílet. Vedoucí projektu přebírá odpovědnost za veškeré realizační etapy a volí vhodné nástroje k jeho realizaci a ověření kvality. Organizuje, plánuje a řídí veškeré činnosti a je v čele projektového týmu. Během projektových rad plánuje kapacity a případné zařazení dalších členů do týmu. Proces realizace softwaru má následující realizační etapy:

- Analýza
- Návrh
- Implementace
- Testování
- Finalizace

Po získání nové zakázky a podpisu smlouvy začíná práce na projektu. Zakázky jsou poptávány na základně nových požadavků na úpravu či vytvoření nového softwaru.

Při získávání informací o požadavcích objednatele dochází k nejasným představám o realizaci produktu a často scházejí důležité informace. Součástí poptávky nebývá dostatečně popsání zadání, které popisuje představy o funkcionalitě a návrhu softwaru. Zejména pak vývojáři nemají dostatečně popsání zadání a jsou nuceni často postupovat dle svých představ. Je na vedoucím projektu, aby dostatečně pravidelně konzultoval se zákazníkem, zda odvedená práce splňuje jeho představy a aby dodal při zadávání dostačující množství informací.

2.6.1.1 Analýza

Obsahem první etapy je vypracování potřebné analýzy zahrnující posouzení, hodnocení a rozbor informací souvisejících s požadavky. Na začátku probíhají základní postupy pro tvorbu analýzy. To obsahuje sběr informací, konzultace se zainteresovanými stranami, třídění a klasifikace informací a vyhodnocení informací. Nejzákladnější informace se týkají požadavků na vlastnosti, schopnosti, procesy a funkce, aktivity a bezpečnost softwaru. V popisované společnosti se provádějí dva typy analýz:

- Produkční analýza – souvisí s produkcí díla
- Projektové analýzy – popisují organizaci a řízení samotné produkce

Je běžné, že oba typy analýz se prolínají a jsou řešeny společně s odlišnými výstupy. Vstupy pro analýzy jsou definované smlouvou, zadáním projektu, normami, standarty, metodikami, architektonickými principy nebo firemní dokumentací. Typické výstupy jsou analytické studie, studie proveditelnosti, katalog vyhodnocených požadavků a podklady pro návrh.

2.6.1.2 Návrh

Základem pro realizační etapu návrhu jsou informace plynoucí z provedených analýz, firemních implementačních postupů a doporučení. Návrh je zpracován formou danou smlouvou nebo zadáním projektu a stanovuje způsob realizace řešení zadaného požadavku. Účel návrhové etapy je zpracování požadavků, optimalizaci řešení, eliminace rizik, návrh a vyhodnocení variant.

Během etapy návrhu mohou být stále přijaty nové uživatelské požadavky pro doplnění nebo upřesnění analytických informací, které poté mohou být implementovány. Výstup návrhu by měl být jasný, přehledný a srozumitelný.

2.6.1.3 Implementace

Implementace probíhá programováním kódu dle sestaveného zadání a provedených studií. Před zahájením prací na projektu je obvykle uspořádán meeting, jehož účelem je rozdělení prací a diskuse nad zadáním. V případě nejasností ohledně zadání může programátor své poznatky diskutovat s vedoucím projektu.

Na svých úkolech programátoři pracují obvykle samostatně a pomocí nástrojů a technologií dle svých uvážení. Po rozdělení úkolů každý z programátorů pracuje na svých úkolech a pracuje na jejich řešení dle svých představ v souladu se zadáním projektu. Jakmile je jeho práce u konce, je vytvořený kód zkontrolován a nadchází fáze testování.

2.6.1.4 Testování

Po úspěšném dokončení prací na implementaci je vytvořený software zkontrolován testováním. To probíhá za účelem ověření funkčnosti, zda je vytvořený produkt v souladu se zadáním a plní uživatelské požadavky. O provedení testů a dosažených výsledků je vytvořen protokol z testování a záznamy o chybách.

Za tímto účelem je vypracována testovací dokumentace obsahující kritéria hodnocení funkčnosti, evidenci chyb a testovací případy. Během testování jsou dokončeny, upřesněny nebo upraveny zpracované testovací případy podle potřeby.

2.6.1.5 Finalizace

V poslední etapě realizačního procesu je výsledný produkt doplněn o další části, jsou-li specifikovány. Zpravidla je posledními činnostmi příprava instalačního balíku, který bude zaručovat správný chod systému. K předání je také připravena dokumentace dle podkladů z analýzy, návrhu a implementace. Na závěr může být provedena řada školení a doprovodných ukázek použití systému.

2.6.2 Projektový tým

Při startu nového projektu je vedením podniku určeno, kdo bude stát za vedením projektu, a tým vývojářů, který se bude na projektu podílet. Vedoucí projektu může dále požádat o další vývojáře či zaměstnance z provozu s realizací projektu. Vedoucí projektu komunikuje s členy týmu a odbornými garanty problémy týkající se vlastností softwarového produktu a jeho dokumentaci tak, aby bylo řešení v souladu se zadáním projektu a v zájmu společnosti. Velikosti projektových týmů se mohou výrazně lišit. Týmy se skládají z členů o dvou rolích:

- Vývojové – analytik, návrhář, programátor, manažer produktu
- Podpůrné – tester, provozní pracovník, dokumentarista

Ve firmě XYZ, spol. s r. o. existují odlišné typy projektů. Existují ty, které se týkají dlouholeté technické podpory a zhodnocení pro zákazníka Armádu ČR. Ve vedení takových projektů jsou zkušení odborní zaměstnanci. Obsahem těchto projektů jsou školení, implementace, aktualizace a práce na požadovaných opravách či vývoji softwaru. Tým vývojářů v terminologii Scrumu se v tomto případě skládá ze všech zaměstnanců provozního oddělení a za situace, kdy je třeba oprava chyb či vývoj nového řešení, se do projektu zapojují i programátoři. Ve výsledku se na této smluvené dlouhodobé spolupráci podílí zaměstnanci celé firmy.

Druhým typem projektu jsou nové požadavky od zákazníka, které se mohou týkat vývoje zcela nového systému, aplikace nebo jen implementace. Takové situace mohou nastat při nákupu nových vozidel, vybavení nebo modernizaci starší techniky o nové prvky. Tento typ vyžaduje založení nového projektu a nastává proces pro spuštění nového projektu a jeho životní cyklus.

V prostředí firmy je znát zájem o trendy managementu a projektového řízení. Tomu přispívají i pravidelné nabídky školení svých zaměstnanců na různá odborná témata. Vedení firmy má snahu své zaměstnance dále odborně vzdělávat a aktivně jim pomáhá se svým rozvojem.

2.7 Řízení projektů

Řízení projektů se ve společnosti XYZ, spol. s r. o. skládá z činností plánování, řízení, komunikace a kontroly v rámci softwarového produktu. Za veškeré řízení a realizaci softwaru ve firmě má odpovědnost zástupce ředitele. Koordinace práce na projektech probíhá ve spolupráci s vedoucími projektů, vedoucím projektové rady a manažerem produktu. K udržování přehledu o stavu projektů slouží projektová rada vždy na začátku měsíce.

2.7.1 Pravidelné porady

2.7.1.1 Projektová rada

V podniku probíhá vždy počátkem měsíce projektová rada. Setkání je pořádáno a organizováno pověřenou osobou jmenovanou jako vedoucí projektové rady ředitelem firmy. Vedoucí projektové rady má odpovědnost za volbu vhodného termínu a přípravu dokumentu obsahujícího aktuální analýzu výkonnosti v projektech a sjednocení příspěvků od vedoucích projektů referujících o tom, co se za uplynulý měsíc v projektech odehrálo. Rady se účastní ředitel firmy i jeho zástupce, projektoví manažeři, ostatní vedoucí probíhajících projektů. Přičemž mimo projektové manažery jsou ve vedení projektů zaměstnanci se specifickými odbornými vlastnostmi potřebnými k vedení daného projektu.

Každý vedoucí projektu má povinnost den před projektovou radou odeslat příspěvek o svých projektech vedoucímu projektové rady. Příspěvek obsahuje stručné informace obsažené ve větách a poté jsou jednotlivé události, které se za uplynulý měsíc odehrály, vypsány v bodech. Na závěr by měl vedoucí projektu uvést, co se bude dít během dalšího měsíce. Vedoucí projektu na projektové radě svůj příspěvek osobně komentuje a pokračuje diskuse.

Průběh projektové rady moderuje vedoucí projektové rady. Začíná úvodním slovem o probíhajících projektech, následuje slovo ředitele k finančnímu plánu a plánu plnění kapacit na projekty a dále mají slovo vedoucí projektů a jejich příspěvky. Projektovou radu zakončuje její vedoucí a na závěr pouze vedoucí sekcí nahlašují odhadované rozvržení práce zaměstnanců na projekty.

2.7.1.2 Programátorská porada

Začátkem roku 2023 byla zavedena pravidelná porada programátorů. Ta se koná vždy ze začátku týdne v pondělí. Nejedná se o pravidelná každodenní setkání, jak je tomu u metodiky Scrum, ale svým způsobem je to změna k lepšímu právě díky pravidelně nastavenému intervalu. Po diskusi s programátory je jejich názor na nastavený interval setkání dostatečný a každodenní setkání po vzoru denního Scrum by nebylo vhodné.

Poradu moderuje vedoucí sekce programátorů. Obsahem setkání je diskuse o aktuálně řešených problémech a plánu úkolů pro následující týden. Doba trvání je obvykle kratší, přibližně 15 minut a je závislá na množství a rozsáhlosti úkolů. Tento nově zavedený prvek pomohl zlepšit jednotlivcům přehled o vykonané práci a plánech na budoucí týden. Poradě nepředchází žádný předem stanovený harmonogram ani není jejím výstupem žádný zápis.

2.7.2 Komunikace

Vedoucí projektu odpovídá za veškeré činnosti spojené s realizací projektu. Komunikuje za zájmy společnosti a za úspěšné dokončení projektu v souladu se zadáním definovaným smlouvou.

Hlavním a nejběžnějším nástrojem komunikace projektového manažera se zainteresovanými stranami je mobilní telefon a MS Outlook. Využívání elektronické pošty má výhodu v zaznamenání korespondence s adresátem. Uložené konverzace obsahující důležité informace a dokumenty je možné považovat za součást projektové dokumentace, konkrétně součást konzultací, a tedy jejich obsah přikládat do projektové knihovny.

Pro potřeby hlubší komunikace nad úkoly a řešením může vedoucí projektu sjednat společné jednání, kterého se budou účastnit členové týmu, případně další zaměstnanci i mimo tým. Z jednání poté vedoucí zhotoví zápis, který je odsouhlasen a zařazen do projektové dokumentace.

2.7.3 Požadavky

Na začátku nového projektu proběhne jednání se smluvními partnery, ať už se jedná přímo o zákazníka, subdodavatele nebo i uživatele, a pracuje se na sestavení

vstupních analýz a na náležitostech souvisejících se spuštěním projektu. Vedoucí intenzivně pracuje na sepisování požadavků uživatele, objednatele i požadavků na bezpečnost softwaru. Pro sepsání uživatelských požadavků není zavedená pevná struktura. Proto se mohou formy zápisu lišit v závislosti na zvoleném způsobu získávání informací vedoucím projektu.

Reakce na požadavky, jež nelze realizovat, je součástí dokumentu řešení omezení, které sestávají z čísla požadavku a vysvětlení vzniklého omezení. Soupis takticko-technických požadavků a omezení řešení musí být zakomponován do smlouvy.

Často řešeným problémem vývojářů při začátku vývoje softwaru je nejednoznačnost zadání a popis představ o výsledném řešení produktu. Výsledkem bývá složité a časově náročné sbírání informací, na čem vlastně vývojáři mají pracovat. Primární náplní práce programátorů by měla být tvorba kódu a práce na cílovém produktu. Tato nedostatečná informovanost je dostává do situací, kdy se nemohou práci věnovat a tím vývoj softwaru značně trpí.

2.7.4 Řízení vývoje

Způsob úkolování uvnitř podniku je nastaven prostřednictvím softwarového nástroje Jira. Zadávané úkoly by vždy měly být srozumitelné, dostatečně popsáné a včas zaznamenané. Toto obecně zavedené pravidlo o obsahu úloh je velmi nejednoznačné. Následkem je nesrozumitelné popsání například evidované chyby nebo návrh na vylepšení aplikace. Programátor, který úkol dostane, není schopen úkol vyřešit, aniž by osobně nemusel shánět detailnější informace od zaměstnance, který chybu nahlásil nebo člena týmu, který návrh na zlepšení softwaru navrhnul.

Dle nastavených pravidel je zaměstnanec odpovědný za úkol trvající déle než 5 pracovních dnů povinen průběžně alespoň jednou za týden postup komentovat. Zaměstnanec, který založí novou úlohu, má povinnost označit vedoucího projektu a ten poté obsah úlohy deleguje na vhodného člena. Takto má vedoucí projektu neustále přehled o současné situaci a každý má možnost se i zpětně chronologicky podívat na události, které se v projektu odehrály v minulosti. Pouze vedoucí je oprávněn přesouvat úlohy z jedné fáze do druhé. Tento proces by měl neustále reflektovat aktuální stav a úlohy by se měly nacházet v takové fázi, v jaké reálně jsou.

2.7.5 Projektová knihovna

Úložný prostor pro veškerou dokumentaci tvoří na sdíleném a zálohovaném disku projektová knihovna. Data jsou zde uložena klasicky do adresářů podle názvu. Zde lze najít seznam všech aktuálně běžících projektů i projekty již ukončené.

Každý z projektů má stejnou adresářovou strukturu obsahující položky:

- A_Konzultace
- B_Řízení_plány
- C_Smlouvy
- D_Ostatní
- E_Výstupy
- F_Úkoly
- G_Normy_standardy

Do konzultací jsou ukládány zápisy z jednání s objednatelem, uživateli nebo zápisy ze schůzek projektového týmu. V případě oponentních řízení, zkoušek nebo kontrolních dnů jsou všechna vyjádření a zápisy ukládány také.

Řízení a plány se týkají výkazů ze systému SAP a projektových dokumentů. Ve výkazech formou Excel dokumentů je evidovaná veškerá provedená práce členů týmu včetně měsíční kalkulace plánovaných zbývajících člověkodnů stanovených smlouvou pro aktuální rok. Dále se do této složky ukládají základní projektové dokumenty a ostatní.

Ve smlouvách se nachází konečná podoba smlouvy včetně všech dodatků. Mohou se zde nacházet i předešlé návrhy smluv a dodatků, akceptační a předávací protokoly a jiné právní listiny.

Do složky „Ostatní“ se vkládají například předaná data a jiné dokumenty, které nelze jednoznačně zařadit. Obsah složky se mění podle uvážení vedoucího, ten určuje, která data zde budou uložena.

Výstupy jsou děleny dle etap projektu. Jestliže etapy obsahují nějaké výstupy, které se na konci etapy odevzdávají, jsou zde uloženy a k nim přiloženy podepsané předávací protokoly.

Poslední dvě složky nejsou již moc aktivní. Úkoly byly nahrazeny nástrojem Jira a ve složce Normy a standardy se nacházejí pouze příložené dokumenty, které specifikují způsob vývoje a bezpečnosti dle požadavků.

2.8 Shrnutí

V minulosti se již vedení společnosti aktivně zajímalo o agilní metodiky vývoje softwaru. K úplné implementaci jedné z metodik nedošlo, ale několik prvků agility se zařadit do firemních procesu povedlo. Vedení firmy je otevřeno návrhům a vítá nápady svých zaměstnanců vedoucí k pozitivnímu dopadu na procesy a atmosféru uvnitř firmy. Díky provedené analýze byly nalezeny oblasti pro navržení řady změn, které povedou k větší efektivitě řízení a vývoji ve firmě.

Několik prvků agilních metodik, které jsou implementovány:

- Kanban tabule, využívána pro řízení vývoje softwaru a ostatních firemních procesů
- Pravidelné projektové rady vždy na začátku měsíce
- Pravidelné porady programátorů každé pondělí

Agilní přístup vývoje softwaru je v určité míře již aplikován u projektů týkajících se technické podpory a zhodnocení. Jsou to projekty, kdy ve smlouvou stanovených intervalech je vydána nová verze systému, která reaguje na připomínky a nové požadavky zákazníka. Ty jsou testery podrobeny testovacím procesům a předány k dalšímu testování samotným zákazníkem. Tyto projekty s sebou nesou znaky agility ve zpětné vazbě od zákazníka, tímto způsobem ale nejsou nastaveny veškeré projekty ve firmě.

S ohledem na hlavního zákazníka není vždy možné jít cestou pravidelné kontroly a získání zpětné vazby, i přes tuto skutečnost by měl být kladen větší důraz na uplatnění prvků agilních metod v co nejvíce projektech to jen jde.

Na základě analýzy současného stavu budou v návrhové části popsány návrhy na případná zlepšení v těchto oblastech:

- Při pravidelných poradách je třeba vést zápis z porady.
- Zavádění nových úkolů do Jira by mělo být více strukturalizované.
- Dbát na dostatečně detailní popis evidovaných chyb pro opravy.
- Lépe zvládat řízení požadavků zákazníka.

- Zavedení nových týmových rolí.
- Zlepšení procesu sepisování zákaznických požadavků.

3 Návrh řešení a přínos návrhů řešení

Tato kapitola se věnuje návrhu možných zlepšení v určitých oblastech a implementaci nových prvků do firmy. Navrhované změny povedou ke zlepšení projektového řízení vývoje softwaru a nastaví efektivnější řešení pro procesy probíhající ve společnosti. Podkladem pro tuto kapitolu jsou teoretická východiska a analýza současného stavu podniku.

3.1 Zavedení role Product owner

Roli Product ownera by měl vykonávat člen týmu, který disponuje velmi dobrými komunikačními schopnostmi, měl by mít velmi dobré znalosti o produktu a být odpovědný za svou práci. Tato týmová role by měla být co nejpřesnějším zastoupením zájmů zákazníka. Jeho hlavní úlohou je pochopit zájmy zákazníka a ostatních stakeholderů zapojených do projektu. Aby zmiňovanou úlohu mohl plnit, je nutné udržovat si co nejbližší kontakt se zákazníkem a stakeholdery. Product owner je hlavním členem týmu, který se stará o komunikaci se zainteresovanými stranami a naplnění hodnoty produktu.

Do této role by se za aktuální situace v podniku nejvíce hodil vedoucí skupiny provozu. Jeho role je aktuálně nejvíce zaměřená na komunikaci se zákazníkem, ostatními zainteresovanými stranami a má velmi rozsáhlé odborné znalosti o produktech firmy. Disponuje dostatečnými dovednostmi a odbornými znalostmi potřebnými k případnému navázání a udržování vztahu se zákazníkem. Dalším důvodem jsou bohaté zkušenosti za období více než 10 let jako vedoucí skupiny provozu a vedoucího projektu technické podpory, který bude nadále zachován. Tyto zkušenosti vedly k rozsáhlým znalostem o vyvíjených produktech a technických řešeních za řadu let. Z toho vyplývá předpoklad o správném definování hodnot pro zákazníka a rozhodnutí, co je v danou chvíli důležité a na čem pracovat.

Zavedení role Product ownera by znamenalo v oblasti se zákazníkem a stakeholdery minimum. Tato oblast je nastavena na dostatečně dobré úrovni. Změnu by však představovala forma získávání informací o požadavcích na software, které by měly nově zavedenou strukturu (viz 4.3.4.).

Hlavním přínosem nejen pro zákazníka, ale také i pro společnost by byla vyšší kvalita softwarových produktů. Role Product ownera je významně založena na odvádění

kvalitních výstupů kvůli zájmům a spokojenosti zákazníka. Tím ručí za výstupní kvalitu všech jednotlivých položek v produktovém backlogu, které dohromady utvářejí jeden celek, hlavní produkt společnosti.

Kompetence Product ownera:

- Reprezentace zájmů zákazníka a stakeholderů
- Komunikuje pokrok vývoje projektu
- Tvorba a práce s user story
- Definování priorit položek v produktovém backlogu
- Projektová dokumentace týkající se jednání se zákazníkem
- Udává směr a stará se o hodnoty produktu
- Rozhodne, kdy je položka backlogu splněna
- Testování dokončené práce
- Opora týmu, co se týče návrhu na řešení

3.2 Zavedení role Scrum Master

Z analýzy současného stavu projektového řízení ve firmě vyplývá skutečnost absence role, která pomáhá týmu, dohlíží na jeho správné fungování a vede tým k samostatnosti. Scrum master je druhou rukou Product ownera. Jejich spolupráce vede ke kvalitnějším výstupům práce na user stories díky účasti na přípravě kritérií „Definition of Done“ a pomáhá Product ownerovi s prioritizací položek produktového backlogu.

Z aktuální situace kapacitního stavu zaměstnanců, kdy je pohledávka pouze na rozšíření oddělení programátorů a analytiků, bylo rozhodnuto, že kompetence Scrum mastera bude vykonávat jeden ze současných zaměstnanců, nikoli někdo nový zvenčí.

Pro tuto roli byl navržen současný vedoucí dlouhodobého projektu „Technické zhodnocení“. Je žádoucí, aby jeho dosavadní pracovní povinnosti byly zachovány a pouze byly rozšířeny o povinnosti vyplývající z role Scrum mastera. Aktuálně je jeho odpovědností příprava a moderování projektových rad, na které zasedají vedoucí projektů a vedení společnosti viz. 3.7.1.1. Do role Scrum master byl navržen díky svým dlouholetým znalostem produktu společnosti a vedením projektového týmu.

Jeden z úkolů Scrum mastera je být oporou pro projektový tým. Díky svým hlubokým znalostem o produktech společnosti a dlouhé praxi s vedením projektů může

svému projektovému týmu nabídnout potřebné informace k dosahování lepších výsledků. Vyskytnou-li se v projektu problémy, tak Scrum master je tu od toho, aby přišel s řešením, do kterého se ale snaží zapojit i tým a v ideálním případě je společně s týmem problém řešen. Není totiž od toho, aby problémy řešil sám. Jeho cílem je budovat samostatný tým, který dokáže řešit sám krizové situace a je dostatečně motivovaný a organizovaný, aby dosahoval vždy co nejkvalitnější odvedené práce.

Nově v roli Scrum mastera budou jeho zkušenosti s moderováním projektové rady využity i v rámci programátorské porady a schůzce projektového týmu. Bude udržovat správnou formu a směr těchto meetingů. Je důležité, aby během diskuse všichni členové mluvili k tématu a byl naplněn obsah schůze. Mimo koordinaci probíhající schůze bude vždy včas rozesílat pozvánky a rezervovat zasedací místnosti pro setkání.

Kompetence Scrum mastera:

- Plánování a řízení schůzek
- Vede tým k samostatnosti
- Působí na jednotlivce a vede je k dosažení lepších výsledků
- Spolupracuje s Product ownerem na položkách backlogu
- Podílí se na stanovení pravidel Definition of Done
- Spolu s Product ownerem hodnotí kvalitu odvedené práce
- Stanovuje projektovému týmu správný směr k dosažení výsledků

3.3 Uživatelské požadavky

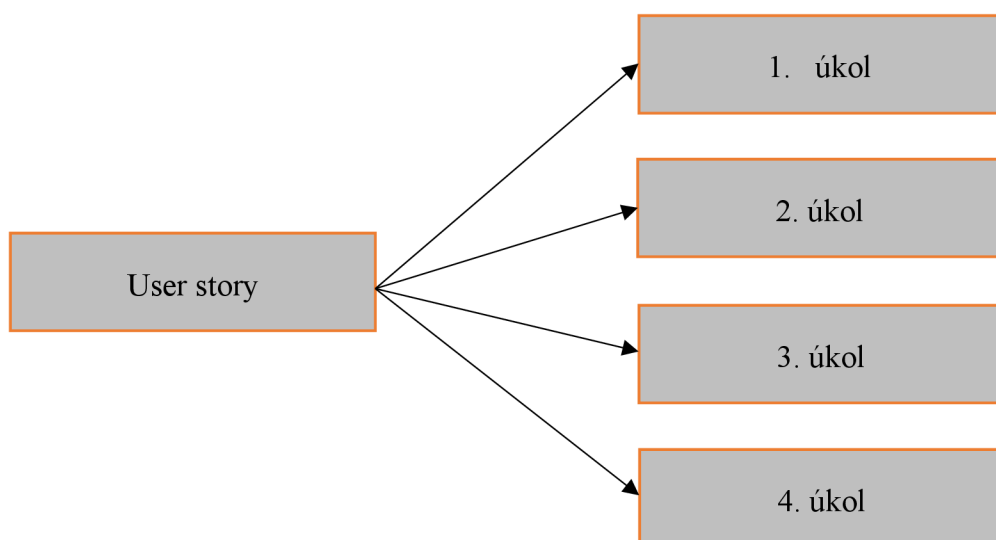
3.3.1 User story

Již od prvních jednání se zákazníkem, kdy je předmětem specifikování vlastností požadovaného produktu, by mělo být v zájmu Product ownera sepisovat požadavky formou user story. Jelikož Product owner je prostředníkem mezi zákazníkem a vývojovým týmem, jeho úkolem je také snažit se spojit technické řešení s uživatelskou představou. Vývojovému týmu by měl požadavky představovat jednoznačně a pochopitelně, aby bylo možné sloučit právě zájmy zákazníka a technické řešení, které přinese vývojový tým. Product owner by měl mít tento proces pod kontrolou, aby vývojový tým dokázal řádně splnit náležitosti user story.

Jednalo by se o efektivní způsob, jak předat vývojovému týmu konkrétní požadavky na funkcionality produktu s co nejvýstižnějším popisem a v některých případech i vizuální podobou. Při tvorbě by vývojový tým také věděl, pro jaké uživatele danou funkcionality zpracovává, a hlavně co od ní uživatel očekává. Tým by tedy věděl, proč například se nějaká změna nebo funkcionality zavádí.

Takto sepsané user story budou zapsány v nástroji Jira definovanou strukturou. Podoba user story bude udávat formu zápisu úloh do daného projektu v Jira a stanou se tak součástí produktového backlogu. V případě zjištění, že je user story příliš obsáhlé, je nutné, aby Product owner user story dekomponoval na menší, splnitelnější řešení. I po rozpadu je nutné dodržet veškerou formu user story.

Jakmile jsou user story sepsána, představí je Product owner Scrum masterovi a následuje diskuse o složitosti a důležitosti jednotlivých položek. Aby bylo možné na user story začít pracovat, je nutný proces rozpadu na úkoly, ze kterých se user story skládá. O technické náročnosti je vedena diskuse mezi Product ownerem, Scrum master i vývojovým týmem. Tímto způsobem získáme nejpřesnější odhad, jak které úkoly bude náročné zpracovat. Může nastat situace, kdy mohou vzniknout hierarchie dalších nezbytných úkolů ke splnění. To vše by mělo být pečlivě projednáno na jednom z projektových meetingu. Jakmile je odsouhlasena finální podoba, jsou úkoly zadány do nástroje Jira.



Obrázek 10 Rozpad user story na úkoly (Zdroj: vlastní zpracování)

3.3.2 Prioritizace

S ohledem na zákazníka popisované firmy se v projektech budou mít určité požadavky tendenci opakovat a jejich implementace bude pro vývojáře zcela jasná. Jedná se o nároky na bezpečnost a postup vývoje softwaru, které by měl produkt splňovat předepsanými vojenskými standardy a normami. Zmíněné požadavky jsou zcela jasně nezbytností pro vyvíjený software, proto budou mít vždy nejvyšší prioritu.

Správa produktového backlogu je v rukou Product ownera. Během zápisu nových user story do backlogu je nutné uvést ujasněnou důležitost dané story či úkolu. Mezi další role Product ownera tak patří prioritizace položek backlogu. Jestliže zná nejlépe požadavky zákazníka, je schopen určit, co je pro zákazníka nejvíce a nejméně důležité.

Rozhodování o prioritách však musí probíhat v rámci diskuse se Scrum masterem a vývojovým týmem nad složitostí technického řešení položek. Nejprve by se měl Product owner poradit se Scrum master o důležitosti sepsaných user story. Scrum master je člen týmu dostatečně odborně znalý technických záležitostí a dokáže přibližně odhadnout náročnost daného user story. Výsledkem této debaty mezi Scrum masterem, který představuje zkušeného a technicky znalého člena, a Product ownerem, jenž reprezentuje pohled a hodnoty zákazníka a též odborně zkušeného zaměstnance, je určení priorit všech vytvořených user stories.

3.3.3 Definition od Done

Všechna zpracovaná user story si musí projít všemi fázemi v nástroji Jira. Právo na rozhodování, kdy se položka může považovat za dostatečně dokončenou a připravenou na přesunutí do další fáze, má pouze Product owner. Pouze ten nejlépe ví, co zákazník skutečně požaduje, a může rozhodnout, zda je user story dle požadavků.

Aby proces vývoje produktu bylo možné úspěšně dokončit, musí existovat kritéria, která určují, kdy je hotovo. Tato kritéria jsou určena pro každou položku v backlogu. Rozhodnutí pro ohodnocení dokončení položek určuje Product owner, který tímto odpovídá za požadovanou kvalitu. V rámci zavedené Kanban tabule v nástroji Jira se dá určit, že fáze, u kterých má Product owner největší vliv, jsou:

- To do – při vytváření user story do backlogu
- Pending – čekání na zahájení testování

- In testing – fáze testování, zda položka odpovídá požadované kvalitě
- Done – definitivní potvrzení, že položka je vyhovující

3.3.4 Praktická ukázka user story

Ve firmě je pro potřeby projektového řízení zaveden nástroj Jira. Zde je tedy veden i backlog projektu. Zadávat nové user story má právo pouze Product owner. K zápisu user story bude využita funkce epic. Vytvořený epic bude představovat user story a k té se poté budou vztahovat úkoly, ze kterých je složena pomocí epic odkazu.

Základní forma user story, které by mělo přinášet odpovědi na následující otázky:

- Kdo? – Základní otázka, která nám říká, kdo je uživatelem, pro koho to děláme
- Co? – Co budeme dělat, jaký je účel naší práce
- Proč? – Co uživatel čeká, co chce, aby bylo dosaženo, co pro něj řešení přinese

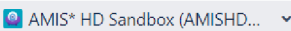
Během zápisu by měl Product owner pamatovat a dodržet klíčové vlastnosti každého user story:

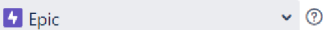
- Jednoznačný popis
- Co nejmenší, ale obsahující všechny informace
- Jednoduché
- Popisuje jednu novou funkcionalitu
- Popsat přínos, její hodnotu
- Vnést pohled uživatele

Vytvořit úlohu

Nastavit pole


Všechna pole označená hvězdičkou (*) jsou povinná.

Projekt* 

Typ úlohy* 










Epic Name*
Provide a short name to identify this epic.

Souhrn*

Termín 

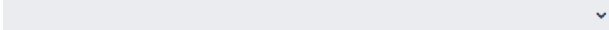
Zákazník Břeclav
 Česká Lípa
 Dvůr Králové
 FN Brno
 Havlíčkův Brod

Popis

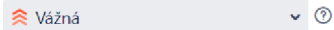
Styl  **B** *I* U A        


Uživatelné chtějí možnost zvolení potřebného nástroje k práci na zázkesru. Chtěli permanentně vizuální řešení, kde uvidí jaké nástroje mohou volit.

Pomocí tohoto řešení, bude jejich práce urychlena a zvýšen uživatelský komfort při práci nad zázkesrem, protože postačí pouze jedno kliknutí pro změnu nástroje.


Řešení ve verzi 

Začněte psát pro získání seznamu odpovídajících, nebo stiskněte dolů pro výběr.

Priorita 

Štítky 

Začněte psát k nalezení a vytvoření štítku nebo stiskněte dolů a vyberte navrhovaný štítek.

Příloha  Přetáhnout soubor pro přiložení, nebo procházet počítač

Vytvořit další

Obrázek 11 Vytvoření user story (Zdroj: vlastní zpracování)

Přínosnost návrhu

- Zamezení nedostatečného množství informací o zadání
- Zefektivnění práce vývojářů
- Definování podmínek, kdy user story naplnilo očekávání
- Reprezentace zájmů zákazníka v roli Product ownera
- Product owner připraví pro vývojáře optimální zadání
- Výstupem je kvalitněji zpracovaný produkt

3.4 Evidence chyb softwaru

Mezi zaměstnanci často dochází k situacím, kdy nejsou dostatečně popsány nově evidované chyby. Kvůli nedostatečnému popisu není schopen programátor reprodukovat chybu na vlastní stanici, neví, v čem chyba spočívá nebo co uživatel od funkce očekává. Chybný úsudek uživatele o funkcionalitě taktéž způsobuje zbytečné plýtvání času a energií. Uživatel předpokládá určité výstupy nebo chování, ale dostaví se jiné, rozdílné od jeho očekávání. Následuje úsudek o chybném fungování systému, způsobené pouze neznalostí softwaru. Tyto důvody mají za následek nemožný začátek prací na opravě, aniž by programátor kontaktoval ohlašovatele chyby pro získání detailnějších informací.

Prvním krokem ohlašovatele chyby by mělo být ověření správné funkcionality a výsledků, které jsou očekávány. Tím je ověřeno, že je chyba legitimní a skutečně se vyskytuje. Cílem je zamezit zbytečnému zápisu chyby, které se nevyskytuje.

Pokud k nežádoucímu chování softwaru opravdu dochází, je třeba podrobný, a přesto výstižný popis ohlašované chyby. Samozřejmostí a základem je správné přiřazení, v jaké aplikaci nebo systému k chybě dochází a výstižný krátký popis. Tento proces je výrazně usnadněn zavedeným nástrojem Jira, kde se nacházejí všechny potřebné aplikace a části systému. Ohlašovatel pouze vybere z nabídky správnou položku a zadá popis. Ten by měl pár slovy zřetelně popisovat, k čemu dochází a při jaké činnosti.

Cílem obsahu popisu by mělo být několik důležitých bodů pro případnou rekonstrukci situace, při které k chybě došlo. Programátor, který hlášení o chybě obdrží, potřebuje vědět několik nezbytných informací. Ve správně provedeném popisu chyby je třeba uvést aplikace, ve které k chybě došlo včetně verze aplikace a operačního systému. Aby mohlo dojít k rekonstrukci situace, je třeba několika větami výstižně sdělit, při jaké situaci a za jakých okolností k chybě došlo, například jaká data byla zpracovávána a podobně. Vhodnou pomůckou jsou i snímky obrazovky chybových hlášení nebo neočekávaného chování aplikace, tak aby vše bylo vizuálně zobrazeno.

K úspěšnému vyřešení chyby můžou přispět i očekávání o chování a výstupech ze strany uživatele. Tím získá programátor bližší představu o nápravě chyby.

Posledním důležitým bodem je označení priority, jakou má oprava chyby důležitost. Může se jednat o situaci pod časovým tlakem, kdy je v brzké budoucnosti očekávána náprava chyby v nové verzi softwaru a lze přiložit i požadované datum dokončení. V takovém případě jde o vysokou prioritu opravy.

Vytvořit úlohu Nastavit pole

Projekt* DEI_ALL_23 (DEIALL23)

Typ úlohy* Chyba

Souhrn* Při importu souborů formátu .nvg aplikace spadne

Zadavatel* Kudrna František

Komponenty Scénář_2021-2023

Popis

Aplikace ZXY při importování všech souborů formátu NVG spadne. Problém se objevuje vždy v aktuální verzi aplikace 1.5. a je nutné aplikaci vždy znovu spustit.

Chyba byla objevena ve verzi operačního systému Windows 11.

V příloze jsou přiloženy snímky obrazovky s chybovou hláškou, po které následuje vypnutí aplikace. Součástí přílohy jsou loq soubory s dalšími informacemi o chybě.

Vytvořit další **Vytvořit** Zrušit

Obrázek 12 Zázpis nově vzniklé chyby do Jira (zdroj: vlastní zpracování)

Přínosy návrhu

- Úspora času v řešení chyb způsobené neznalostí softwaru
- Kvalitnější zadání pro opravu chyby programátorem
- Možnost reprodukce chyby dle popisu
- Programátor se nemusí věnovat získávání detailnějších informací

3.5 Schůzky projektového týmu

Zavedení pravidelných schůzek v rámci projektového by se mělo konat jednou do týdne. Pro tyto účely bylo nadále ponecháno pondělí v 9:00 ráno. Pondělí je začátkem nového pracovního týdne, a to znamená vhodnou příležitost pro diskusi k projektu. Zvolená hodina je z praktických důvodů optimálním časem, kdy jsou přítomni i zaměstnanci, kteří jezdí do práce z větší dálky a také ti, co to mají blíže. Zároveň mají přítomní dostatek času na práci do obědové přestávky, které obvykle začíná v 11:00.

Aby se maximalizovala efektivnost projektové schůzky, bude pro tyto účely zvoleno jiné než obvyklé pracovní místo. V prostorách firmy je k dispozici několik zasedacích místností. Oddělené pracovní prostředí přispívá k většímu soustředění, kdy se členové budou mít tendenci soustředit pouze na téma schůzky a trocha pohybu po prostorách bude také přínosem. Klíčová je však vybavenost zasedacích místností. Disponují projektorem s tabulí nebo velkoplošnou televizí. Schůzka bude probíhat nad Kanban tabulí, kde bude k vidění produktový backlog s řešenými úkoly. Případně lze využít promítací techniku pro jakékoliv další potřebné účely.

Hlavním moderátorem schůzky bude Scrum master, který povede diskusi a v případě odklánění se od tématu bude na to upozorněno. Scrum master by se měl postarat o stručný zápis z porady. Pro zápis z porady je již vytvořena v projektové dokumentaci šablona, která se pouze upraví pro dle potřeb. Pro ukládání zápisů bude vytvořen prostor na sdíleném disku, aby v případě potřeb každý měl přístup k informacím diskutovaným na setkání. V dokumentu o zápisu by se měly objevit základní informace:

- Datum a místo konání schůzky
- Seznam přítomných členů
- Stručný popis průběhu a závěrů z porady
- Úkoly do dalšího období
- Prostor pro případné návrhy na zlepšení

Potřeba koordinace schůzky je hlavně kvůli potenciálnímu riziku zacházení do větších technických detailů. Od rozebírání hlubšího pohledu na technické řešení jsou zřízeny porady programátorů nadcházející po projektové schůzce, ve kterých je na tyto věci prostor. Dalším důvodem potřeby Scrum mastera na schůzce je dohlížení nad harmonogramem a pravidly schůzky. Ty jsou plánované z důvodu řešení obsahu projektu. Za nepřítomnosti moderátora bývá časem plýtváno na rozhovory osobních záležitostí členů týmu, tomuto by se tímto návrhem zamezilo.

Pravidla projektových schůzek a jejich pravidelnosti byla konzultována s vývojáři, co by z jejich pohledu nejvíce ocenili. Navrhovaný týdenní interval a obsah se jevil zcela vyhovující. V případě, že se nastavený interval bude z dlouhodobého hlediska jevit jako nevyhovující, bude navržena optimalizace. Zde se nabízí aplikovat daily meeting po vzoru SCRUM, který byl ale po konzultaci shledán za současného stavu jako

nevyhovující. Příčinou je vytíženost členů projektového týmu i v jiných projektech. Vývojáři často dynamicky rotují mezi projekty dle situace, ve které se aktuálně nachází a dle urgentních požadavků ze stran zákazníků, které je nutno zpracovat. Nekonzistentní práce na jednom projektu by způsobovala riziko, že na daily meetingu by o vybraném projektu nebylo co říct. Dlouhodobým cílem je organizovat práci tak, aby nedocházelo k přeskokování z projektu do projektu. Doporučením je zpracovat návrh na řešení tohoto problému, kterým se bude vedení firmy v budoucnu zabývat a uvažovat o zavedení změn v této oblasti.

Přítomný bude i Product owner, který bude k dispozici k zodpovězení možných dotazů na user stories. Jako hlavní člen týmu, který komunikuje se zákazníkem a stakeholdery, bude mít možnost sdělit týmu aktuální stav projektu, například novinky, plány, poznatky zákazníka a jakékoliv další relevantní informace o projektu, které budou vhodné sdělit týmu. Hlavní prioritou schůzky je však debata nad položkami produktového backlogu, proto pokud bude chtít Product owner sdělit týmu zmíněné informace, bude to předmětem závěru schůzky.

Online komunikace je ve společnosti nastavena formou MS Teams. Za nepřítomnosti některého z členů týmu bude schůzka vysílaná online, aby byla účast vždy maximální. Toto řešení je vhodné i pro současný trend práce z domova. Na tento postup jsou zaměstnanci již zvyklí, a tedy nepředstavovalo by to žádné náklady navíc. Upozornění na schůzky bude rozesíláno vždy týden dopředu pomocí MS Outlook. Součástí je také sdílený kalendář, díky kterému všichni zaměstnanci mají přehled o plánovaných schůzkách. Součástí pozvánky je termín schůzky, místo konání, které obsahuje název zasedací místnosti a odkaz na připojení online schůzky MS Teams.

Přínosy návrhu

- Soustředěnost a informovanost členů týmu o dění v projektu
- Motivace členů týmu být více produktivní, plnit user story
- Pevně stanovená pravidla schůzky pod dohledem moderátora
- Posílení týmových stavů díky pravidelnému kontaktu

3.6 Porada programátorů

Aktuálně je toto setkání nastaveno na každé pondělí v 9:00. Z důvodu navrhované pravidelné projektové schůzky je součástí změn i posunutí porady programátorů. Jednalo by se pouze o posunutí z 9:00 na 9:30 a vybraný den pondělí by zůstal zachován. Projektová schůzka má předpokládanou dobu trvání 15 minut, pro časovou rezervu je zvolena 15minutová přestávka a 9:30 by se konala porada programátorů a v roli moderátora bude Scrum master.

Téma programátorské porady se bude odlišovat od projektové schůzky v hloubce pohledu na technické řešení produktu a složení účastníků. Členy schůzky budou všichni vývojáři společnosti, ostatně jako doposud. Koordinace průběhu porady, její program, zápis a pozvánky bude mít na starosti Scrum master. Jeho kompetence budou totožné jako v popisované projektové schůzce.

Důvodem ponechání této porady je informovanost ostatních programátorů a diskuse nad technickým řešením úkolů. Dalším důležitým přínosem dobré informovanosti programátorů o práci ostatních je do určité míry komplexnost vyvíjených softwarových produktů. Jak již bylo popsáno v kapitole 3.4., firma nabízí jeden velký systém produktů. Jednotlivé projekty, jež probíhají ve firmě, jsou ve většině případů rozšířeními tohoto rozsáhlého systému. Díky existující provázanosti je ku prospěchu, aby jednotliví programátoři měli přehled o současně probíhající práci či úpravách tohoto softwarového systému.

Součástí obsahu sezení je případná diskuse nad dotazy a návrhem technického řešení, které je příliš odborné na to, aby se řešilo v rámci projektových porad. Proto v případě nutnosti se porady může účastnit i Product owner či jiný člen týmu. Řešení konkrétních projektových záležitostí je zařazeno na konec programu po ukončení monologu každého z vývojářů.

Diskuse bude probíhat na téma aktuálně probíhajících úkolů a zmíněny budou též úkoly minulé i budoucí. Každý z účastníků informuje ostatní programátory, na čem aktuálně pracuje, jaký úkol dokončil a kdy odhaduje začátek nového úkolu.

Ostatní pravidla bude porada sdílet s pravidly uvedenými v předchozí podkapitole schůzky projektového týmu, společně s místem konání v některé ze zasedacích místností a online vysílání schůzky.

Přínosy návrhu

- Informovanost programátorů o řešených úkolech
- Možné podílení se na návrhu technického řešení
- Zavedení zápisu z porady
- Zavedení moderátora a pravidel porady
- Prostor pro řešení odborných technických záležitostí projektu

3.7 Školení

Pro účely kvalitního vzdělání zaměstnanců v agilních metodách řízení projektů vývoje softwaru a metody Scrum bude provedena série školení. Jedná se o 3 termíny školení.

Pro úvod do agilního řízení bude určeno téma Scrum a agilní metodiky pro praxi, kde se zaměstnanci vzdělají v oblasti úvodu do agilního světa a základech týkající se komunikace, postupů a nástrojů.

Druhým školením bude aplikace metody Scrum v organizaci. Tímto získají zaměstnanci znalosti, jak řešit problémy, řídit projekt s několika týmy a jak s produktem pracovat.

Posledním školením získají zaměstnanci hlubší znalosti o agilních metodách. Obsahem školení bude poznání detailnějších kompetencí Product ownera, poznání a řízení týmu a spolupráce. Závěrem bude praktický trénink aplikování agilního přístupu na konkrétním příkladu.

Přínosy návrhu

- Představení agilních metod v profesionálním podání
- Získání hodnotných informací od odborníků na Scrum
- Možnost seberozvoje pro zaměstnance
- Uplatnění znalostí v praktickém tréninku

3.8 Ekonomické zhodnocení

Tato podkapitola se zabývá finančními náklady spjatými s návrhy na zlepšení popsané v rámci této kapitoly.

Finanční náklady vznikají u návrhů spojených se zavedením nových rolí Scrum Mastera a Product ownera. Jelikož bylo rozhodnuto, že nebudou otevřeny další pracovní pozice, bude třeba současné zaměstnance navrhované do těchto pozic adekvátně motivovat. Po domluvě s jednotlivými zaměstnanci bude přistoupeno k individuálnímu způsobu motivace v podobě zvýšení mzdy či firemních benefitů. Konkrétní způsob a výše navrhované odměny bude předmětem osobní domluvy mezi zaměstnancem a zaměstnavatelem. Ostatní návrhy nevyžadují další přímé finanční náklady kromě času, který si zaměstnanci budou muset na plánované schůzky a detailnější popis úkolů do JIRA vyhradit.

Pro získání více znalostí o agilních metodách a Scrumu proběhne série školení pro kandidáty do role Scrum mastera a Product ownera. Tento návrh s sebou nese náklady za agenturu poskytující školení a náklady na cestu do prostor školící agentury. Vzhledem k pozitivním zkušenostem a vhodné lokalitě byla vybrána společnost ictPRO se sídlem v Brně.

Náklady na zaměstnance se skládají z příspěvku na stravu, který činí 129 Kč na den pro jednu osobu. Jelikož se jedná o služební cestu, je třeba řešit dopravu. Pro cestu bude zvolen firemní vůz. Cesta na místo školení je dlouhá 24 km. Vzdálenost byla vypočítána z prostor firmy do místa školení. Náklady na cestu pro jedno školení činí 60 Kč. Způsob výpočtu cestovních nákladů udává tabulka níže.

Tabulka 1 Cestovní náklady (Zdroj: vlastní zpracování)

Cestovní náklady	Jedna cesta	Cesty celkem
Ujeto km	24	72
Průměrná spotřeba v litrech na 100 km dle TP	6	6
Cena za 1L PHM	41,50 Kč	41,50 Kč
Celkem	60,00 Kč	180,00 Kč

Pro současné účely jsou náklady kalkulovány pro 2 zaměstnance. V tabulce níže se nachází kalkulace celkových nákladů spojených se školením.

Tabulka 2 Celkové náklady na školení (Zdroj: vlastní zpracování)

Náklad	Cena za osobu/den/cesta	Cena celkem
Scrum 1	7 200,00 Kč	14 400,00 Kč
Scrum 2	10 200,00 Kč	20 400,00 Kč
Scrum 3	8 200,00 Kč	16 400,00 Kč
Stravné	129,00 Kč	258,00 Kč
Cestovní náklady	60,00 Kč	120,00 Kč
Celkem	25 789,00 Kč	51 578,00 Kč

3.9 Přínosy navrhovaného řešení

Návrhy popsané v rámci této části přinášejí do prostředí firmy prvky agilního způsobu řízení vývoje softwaru. Jde především o prvky metodiky Scrum ve spolupráci s prvky Kanban. Z analýzy podniku je zřejmá řada již zavedených procesů a nástrojů vyplývajících z charakteristik agilního přístupu řízení projektů. Po dokončení analýzy byly identifikovány jisté nedostatky a mezery, které byly základem pro zpracování návrhů ke zlepšení. Důležitou podmínkou pro zpracování návrhu bylo dodržet dlouhodobé cíle a plány pro řízení projektů ve spolupráci se zvýšenou efektivitou a užitečností přínosů návrhů.

V rámci shrnutí byly ze zpracovaných návrhů vybrány nejdůležitější přínosy:

- Zavedením role Scrum master dojde k výraznému zlepšení týmové spolupráce
- Dojde k efektivnějšímu řízení projektových i programátorských porad
- Zavedením role Product ownera dojde k výraznému navýšení kvality produktu kvůli nastaveným procesům týkajícím se implementace požadavků
- Častější práce se zákazníkem díky Product ownerovi povede k rychlejší reakci na změny a zpětné vazbě odvedené práce
- Dojde k jednoznačně přehlednosti úkolů díky zavedení formy zápisu úkolů a user story do produktového backlogu
- Evidence chyb softwaru získá jasný vzorec pro zápis, aby programátor neztrácel čas získáváním informací a především věděl, jak chybu najít a opravit
- Definování odpovědností zavedených rolí a větší přehlednost o prováděné práci povede ke snížení stresu a časového tlaku a tím nastane pozitivní posun atmosféry ve firmě

Závěr

Předmětem bakalářské práce je implementace vybraných prvků agilního projektového řízení u vybrané firmy zabývající se vývojem softwarových řešení pro Armádu ČR. Díky analýze současného stavu projektového řízení byl popsán aktuální stav a procesy, jež ve firmě fungují. Z analýzy vyplynulo několik oblastí týkajících se návrhu na zlepšení. Pomocí teoretických podkladů a výstupů analýzy byl pro vybrané prvky agilních metodik vypracován návrh řešení.

První část práce se věnuje teoretickým podkladům. Úvod se zaměřuje na základy projektového řízení. Postupně jsou detailněji charakterizovány agilní metodiky řízení projektů se závěrečným zaměřením na konkrétního zástupce. Nejvíce pozornosti je věnováno představení prvků a týmových rolí z metodiky Scrum. Svůj účel v teoretické části tvoří i kapitola věnována metodě Kanban, na kterou se později v analytické a návrhové části navazuje. Zpracovaný teoretický podklad je pro firmu přínosem v inspiraci zejména dále do budoucna a představuje pro firmu možnosti pro další vývoj.

Podrobnější analýze projektového řízení a procesů je věnována druhá kapitola. Do této části jsou veškeré získané informace a zkušenosti důkladně zapracovány. Tím došlo k podrobnému zachycení aktuálního stavu, jakým způsobem a pomocí jakých nástrojů je realizováno projektové řízení vývoje softwaru ve firmě. Pomocí zpracované analýzy byly identifikovány prostory ke zlepšení, kterými se dále zabývá návrhová část.

V závěrečné kapitole jsou navržena a popsána řešení, která jsou dle teoretického podkladu navržena k implementaci do firemních procesů. S ohledem na zákazníka a zájmy vedení společnosti nebylo navrženo zavedení jedné z agilních metodik v celém rozsahu, ale pouze vybraných prvků agilního řízení, které budou dostatečným pozitivním přínosem. Pro tuto kapitolu byly zpracovány vybrané prvky z agilního řízení, které vedou k zefektivnění projektového řízení. U představených návrhů je shrnutí, co daný návrh řeší a přinese. Závěrem je zpracována kalkulace nákladů plynoucí z navrhovaných řešení.

Hlavního cíle této bakalářské práce bylo dosaženo splněním stanovených dílčích cílů. Prozatím došlo k naplnění především rychleji splnitelných návrhů. Jedná se navrhované změny programátorské rady, které se týkají koordinace meetingu a návrhy vztahující se k zápisu nových úkolů do nástroje Jira splňujících navrhovaná pravidla pro správný zápis položek. Náročnější změny budou považovány za budoucí cíle, kterých má být dosaženo.

Seznam literatury

1. SVOZILOVÁ, Alena. *Projektový management*. Praha: Grada Publishing, 2006, 356 s. ISBN 80-247-1501-5.
2. DOLEŽAL, Jan. *Projektový management podle IPMA*. Praha: Grada Publishing, 2009. ISBN 978-80-247-2848-3.
3. DOLEŽAL, Jan. *Agilní přístupy vývoje produktu a řízení projektu*. Praha: Grada Publishing, 2022. ISBN 978-80-271-6606-0.
4. BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů*. Praha: Grada Publishing, 2005. ISBN 80-247-1075-7.
5. DOUČEK, Petr. *Řízení projektů informačních systémů*. Praha: Professional publishing, 2004. ISBN 80-86419-71-1.
6. SHTUB, AVRAHAM. Work Breakdown Structure. In: SALVENDY, Gavriel. *Handbook of Industrial Engineering: Technology and Operations Management* [online]. 3rd. Canada: John Wiley, 2001, s. 1263–1280 [cit. 2023-01-08]. ISBN0-471-33057-4. Dostupné z: [http://www.las.inpe.br/~perondi/19.07.2010/material_de_apoio_19-07-2010\(1\).pdf](http://www.las.inpe.br/~perondi/19.07.2010/material_de_apoio_19-07-2010(1).pdf)
7. DORA, Sujit Kumar a Pushkar PUSHKAR. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) ANALYTICAL COMPARISON AND SURVEY ON TRADITIONAL AND AGILE METHODOLOGY. *Abhinav journal* [online]. 2013, 10 [cit. 2023-01-15]. ISSN 2277-1174. Dostupné z: https://www.researchgate.net/profile/Pushkar-Dubey2/publication/319716548_SOFTWARE_DEVELOPMENT_LIFE_CYCLE_ANALYTICAL_COMPARISON_AND_SURVEY_ON_TRADITIONAL_AND_AGILE_METHODODOLOGY/links/59bac6eba6fdcca8e55ddc62/SOFTWARE-DEVELOPMENT-LIFE-CYCLE-SDLC-ANALYTICAL-COMPARISON-AND-SURVEY-ON-TRADITIONAL-AND-AGILE-METHODOLOGY.pdf
8. SCHWABER, Ken a Jeff SUTHERLAND. *Průvodce Scrum: Oficiální průvodce pro Scrum: Pravidla hry* [online]. 2020 [cit. 2023-02-01]. Dostupné z: <https://Scrumguides.org/docs/Scrumguide/v2020/2020-Scrum-Guide-Czech.pdf>

9. DEEMER, Pete. The Scrum Primer: A Lightweight Guide to the Theory and Practice of Scrum. *InfoQ* [online]. 2012, 2012(2.0), 1-20 [cit. 2023-02-01]. Dostupné z: https://Scrumprimer.org/Scrumprimer20_small.pdf
10. Visual Studio 2022, 2023. *Microsoft.com* [online]. [cit. 2023-04-02]. Dostupné z: <https://visualstudio.microsoft.com/cs/vs/>
11. Introduction to Hyper-V on Windows 10, 2022. *Microsoft.com* [online]. [cit. 2023-04-02]. Dostupné z: <https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/>
12. FILION, Luc, Nicolas DAVIOT, Jean-Philippe LE BEL a Marc GAGNON, 2017. *Using Atlassian Tools for Efficient Requirements Management: An industrial case study* [online]. Annual IEEE International Systems Conference (SysCon) [cit. 2023-04-02]. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7934769>
13. BUŘITA, Ladislav, 2009. Multilateral Interoperability Programme. *RADIOENGINEERING* [online]. 2009, (18.4), 627-630 [cit. 2023-04-02]. Dostupné z: https://www.radioeng.cz/fulltexts/2009/09_04_627_630.pdf
14. ŠOCHOVÁ, Zuzana a Eduard KUNCE, 2014. *Agilní metody řízení projektů*. Brno: Computer press. ISBN 978-80-251-4194-6.

Seznam Obrázků

Obrázek 1 WBS	14
Obrázek 2 Trojimperativ projektu	15
Obrázek 3 Model SDLC	21
Obrázek 4 Vodopádový model	23
Obrázek 5 Model lean	26
Obrázek 6 Kanban board	27
Obrázek 7 Product backlog	33
Obrázek 8 Organizační struktura	40
Obrázek 9 Kanban tabulka JIRA	45
Obrázek 10 Rozpad user story na úkoly	59
Obrázek 11 Vytvoření user story	62
Obrázek 12 Zápis nově vzniklé chyby do Jira	64

Seznam Tabulek

Tabulka 1 Cestovní náklady	69
Tabulka 2 Celkové náklady na školení	70