



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

**EFEKTIVNÍ VYUŽITÍ SCRUM METODIKY VE VÝVOJOVÉM
TÝMU**

EFFICIENT APPLICATION OF SCRUM METHODOLOGY IN A DEVELOPMENT TEAM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Radek Svoboda

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lenka Smolíková, Ph.D.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav informatiky
Student: **Bc. Radek Svoboda**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **Ing. Lenka Smolíková, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu 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 diplomovou práci s názvem:

Efektivní využití SCRUM metodiky ve vývojovém týmu

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
Přílohy

Cíle, kterých má být dosaženo:

Cílem diplomové práce je zhodnocení využití SCRUM metodiky a návrh změn za účelem jejího efektivnějšího využití ve vývojovém týmu.

Základní literární prameny:

DAVIS, C. W. H. Agile metrics in action: how to measure and improve team performance. Shelter Island, NY: Manning, 2015. ISBN 978-1-61729-248-4.

LACEY, M. The scrum field guide: practical advice for your first year. Upper Saddle River, NJ: Addison-Wesley, c2012. ISBN 978-0-321-55415-4.

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

SUTHERLAND, J. V. Scrum: the art of doing twice the work in half the time. New York: Crown Business, 2014. ISBN 978-0-385-34645-0.

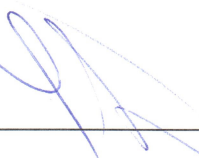
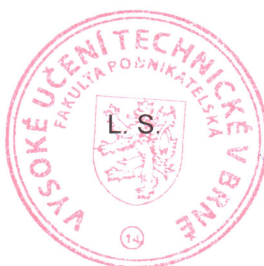
ŠOCHOVÁ, Z. a E. KUNCE. Agilní metody řízení projektů. Brno: Computer Press, 2014. ISBN 978-80-251-4194-6.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 28. 2. 2017



doc. RNDr. Bedřich Půža, CSc.
ředitel



doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Diplomová práce se zabývá efektivnějším využitím Scrum metodiky ve vývojovém týmu v konkrétní společnosti. K dosažení vyšší efektivity využívá práce filosofii SW-CMM, kterou aplikuje v agilním prostředí. Teoretická východiska jsou využita ke komparaci s aktuálním stavem sledované společnosti popsané analytické části. Na základě odhalených nedostatků jsou vzneseny návrhy řešení. Návrhy jsou podrobeny analýze rizik s následným opatřením. Náklady, které jsou vynaloženy na implementaci návrhů a opatření jsou vyhodnoceny v ekonomickém zhodnocení, které zároveň obsahuje přínos těchto změn.

Abstract

The master thesis focuses on more efficient application of Scrum methodology in a development team within specific company. The thesis takes an advantage from the SW-CMM philosophy applied in agile environment in order to achieve better effectivity. Theoretical base is utilized for a comparison with current state in the company described in analytical part. Proposals for a solution of current situation are based on drawback detection. Proposals are subject for risk analysis followed by precaution. Costs of proposals implementation with included precautions are evaluated in economic evaluation part. This part also contains benefits of proposals.

Klíčová slova

Agilní metodika, metriky, proces vývoje software, Scrum, SW-CMM

Keywords

Agile methodology, metrics, Scrum, software development process, SW-CMM

Bibliografická citace

SVOBODA, R. *Efektivní využití SCRUM metodiky ve vývojovém týmu*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 106s. Vedoucí diplomové práce Ing. Lenka Smolíková, Ph.D..

Čestné prohlášení

Prohlašuji, že předložená diplomová 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 24. května 2017

.....

podpis studenta

Poděkování

Rád bych tímto poděkoval vedoucí diplomové práce Ing. Lence Smolíkové, Ph.D. za její ochotu, důvěru, cenné rady a připomínky při zpracovávání diplomové práce. Dále bych chtěl poděkovat společnosti Avast Software s.r.o. a kolegům za poskytnuté zázemí pro tvorbu a uplatnění diplomové.

Obsah

ÚVOD.....	11
1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ.....	12
2 TEORETICKÁ VÝCHODISKA PRÁCE.....	13
2.1 Proces vývoje software.....	13
2.1.1 Model vyspělosti procesu.....	13
2.2 Metodiky vývoje software.....	15
2.2.1 Rigorózní metodiky.....	15
2.2.2 Agilní metodiky.....	18
2.3 Scrum.....	20
2.3.1 Role.....	20
2.3.2 Projektový tým.....	23
2.3.3 Zákazník.....	23
2.3.4 Artefakty.....	23
2.3.5 Schůzky.....	26
2.3.6 Odhady.....	30
2.3.7 Měření.....	31
2.3.8 Data pro měření.....	36
3 ANALÝZA SOUČASNÉHO STAVU.....	38
3.1 Představení společnosti.....	38
3.1.1 Scrum tým.....	39
3.2 Analýza 7S.....	39
3.2.1 Strategie.....	39
3.2.2 Struktura.....	40
3.2.3 Systémy.....	42
3.2.4 Styl řízení.....	48
3.2.5 Spolupracovníci.....	49
3.2.6 Schopnosti.....	50
3.2.7 Sdílené hodnoty.....	50
3.3 Využití Scrum metodiky.....	51

3.3.1	Členové a role Scrum týmu	51
3.3.2	Projektový tým.....	52
3.3.3	Zákazník.....	54
3.3.4	Artefakty	54
3.3.5	Schůzky.....	59
3.3.6	Odhady.....	64
3.3.7	Měření	64
3.4	Vypělost procesu vývoje software	66
3.4.1	EPC diagram vývojového procesu.....	66
3.4.2	RACI matice	67
3.4.3	Sprint.....	68
3.5	SWOT	69
4	NÁVRH ŘEŠENÍ A PŘÍNOS NÁVRHŮ ŘEŠENÍ.....	71
4.1	Role.....	71
4.2	Sprint.....	74
4.2.1	Délka sprintu.....	74
4.3	Schůzky.....	79
4.3.1	Zahajovací schůzka (Kick-off meeting)	79
4.3.2	Plánovací schůzka (Planning meeting).....	80
4.3.3	Posouzení sprintu (Sprint review).....	80
4.3.4	Konečná schůzka (Final meeting).....	81
4.4	Měření	81
4.4.1	Spokojenost (Happiness – H)	82
4.4.2	Míra loajality zákazníků (Net Promotion Score – NPS).....	83
4.4.3	Rychlost (Velocity – V).....	84
4.4.4	Pracovní kapacita (Work Capacity – WC)	85
4.4.5	Faktor zaměření (Focus Factor – FF)	87
4.4.6	Přesnost odhadu náročnosti (Estimation Accuracy - EA)	88
4.5	Identifikace hrozeb a scénářů	90
4.5.1	Změny nebudou přijaty	90
4.5.2	Monitoring dostupný pro tým	90
4.5.3	Využití metrik ke stanovování cílů	90

4.5.4	Vysoká nepřesnost měření	90
4.5.5	Selhání vývojového týmu	90
4.5.6	Chybná interpretace metrik	91
4.6	Kvantifikace rizik	91
4.7	Snižování rizik	93
4.7.1	Mapa rizik	94
4.8	Ekonomické zhodnocení	95
4.9	Přínosy návrhů	99
ZÁVĚR		100
SEZNAM POUŽITÉ LITERATURY		102
SEZNAM OBRÁZKŮ		104
SEZNAM TABULEK		104
SEZNAM GRAFŮ		105
SEZNAM PŘÍLOH		106

ÚVOD

Samotný hardware z funkčního hlediska počítačů, jak je dnes vnímáme, nedává smysl. Software, neboli programové vybavení je tím, díky čemu veškerý hardware získává duši, ožívá, komunikuje jak v rámci svých komponent, tak s uživatelem. Hardware i software počítačů separátně nejsou v konečném důsledku příliš užitečné, jejich synergií dosáhneme toho, že počítač vykoná úkol, který uživatel vyžaduje. Nejlépe, když je úkol vykonán okamžitě, s minimem vstupů a s maximálně užitečným výstupem.

V dnešním světě je tak z tohoto hlediska vývoj kvalitního software, ve smysluplném časovém rámci, nutný. Vývoj software však nelze srovnávat s výrobou. Při vývoji software neexistuje žádný plán či návod, jak dosáhnout výsledku, který uživatel vyžaduje. Na začátku vývoje software je pouhá myšlenka. Uživatel ví, co vyžaduje, neví však, jak se k výsledku dopracovat. Vývoj software tak není ani uměním, které může být zcela bez hranic co se týče využití metodiky i konečného výsledku. Při vytváření software je totiž výsledek známý již v počátku.

Pouze malá část společností dnes nevyužívá informační technologie. Velké množství společností je na využití informačních technologií závislé. Efektivní vývoj software tak často rozhoduje o ekonomickém úspěchu či neúspěchu mnoha firem.

Pokud pro vývoj software neexistuje jednoznačný předpis, jak lze posoudit efektivnost vývojového procesu? Je software, který funguje v řadě společností opravdu vyvíjen efektivně?

V mnoha firemních kulturách se dnes můžeme setkat tzv. s agilním vývojem software. Agilní vývoj software má být přesně takový, jak si vývoj v ideálním světě představujeme, tedy minimálně: rychlý, bez chyb, za přijatelných nákladů a se správnou funkcí.

Tato diplomová práce se zabývá tím, jak pomocí dnes moderního agilního vývoje, vytvářet software efektivněji. Využívá k tomu pomyslnou nekonečnou smyčku, ve které dochází k neustálému ladění chyb, které jsou při vývoji software zapříčiněny lidským faktorem.

1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Hlavním cílem diplomové práce je zhodnocení využití SCRUM metodiky a návrh změn za účelem jejího efektivnějšího využití ve vývojovém týmu. Vývojový tým spadá do společnosti Avast Software s.r.o. a pracuje na vývoji proprietárního IS.

Práce předkládá návrhy pro efektivní využívání metodiky také za pomoci dosažení dílčích cílů:

- zvýšení úrovně z hlediska SW-CMM,
- zhodnocení zavedených procesů a postupů s následným doporučením,
- potlačení slabých stránek, využití příležitostí a vyhnutí se hrozbám.

Za účelem dosažení těchto cílů je využito teoretických východisek, která popisují problematiku SW-CMM a metodiky Scrum.

Analytická část práce vyhodnocuje současný stav ve sledovaném týmu. V rámci této části je zpracována analýza 7S, zavedení metodiky Scrum a analýza z pohledu zařazení do SW-CMM, které umožňují identifikaci silných stránek, slabých stránek, hrozeb a příležitostí, které shrnuje závěrečná tabulka.

Vlastní návrhy řešení vycházejí z komparace teoretické a analytické části práce. Tato část práce předkládá návrhy, které jsou koncipovány v souladu s eliminací hrozeb, posílení silných stránek, využití příležitostí a potlačení slabých stránek sledovaného týmu. K tomuto účelu slouží vhodně navržená doporučení, pro posun na co nejvyšší úroveň z hlediska SW-CMM a také návrhy pro optimalizaci využití Scrum metodiky ve sledovaném vývojovém týmu.

U navržených řešení je provedeno ohodnocení rizik včetně návrhu následných opatření k zajištění bezproblémové implementace.

Návrhy na změny jsou popsány a vyhodnoceny také z ekonomického hlediska, aby bylo možné zohlednit náklady na implementaci v rámci plánování rozpočtu ve sledované společnosti. Ekonomické zhodnocení se zabývá také přínosy, které vyplývají z implementace navržených změn.

2 TEORETICKÁ VÝCHODISKA PRÁCE

Teoretická část práce popisuje nezbytná teoretická východiska, která jsou dále využívána v analytické části, pro zhodnocení současného stavu společnosti, resp. v návrhu řešení, který také čerpá z této části práce. Práce se zaměřuje výhradně na agilní metodiku Scrum. Na ostatní metodiky nebo jejich součásti se práce z důvodu rozsahu a vymezení čitelného rámce soustředí minoritně.

2.1 Proces vývoje software

Tato kapitola popisuje proces vývoje software z hlediska rozkladu jednotlivých klíčových slov. Ačkoli se může zdát tento pojem na první pohled zcela zřejmý, jeho interpretace je nezbytná pro pochopení kontextu této práce.

Vývoj software je specifický následujícím:

- má daný cíl specifikovaný zákazníkem
- nejedná se tedy o tvůrčí činnost dle fantazie vývojářů
- zároveň však nejde o výrobu (unikátní postup, neexistuje návod).

Samotný proces je poté definován jako posloupnost aktivit (uspořádaných po částech) vedoucí od počátečního stavu k předem definovanému cíli (1, str. 9-11).

2.1.1 Model vyspělosti procesu

Model vyspělosti procesu neboli Capability Maturity Model (SW-CMM) je metodika, která se využívá k vývoji a kultivaci procesů ve společnosti. SW-CMM pojednává o pěti úrovních vyspělosti procesu z hlediska vývoje software, které popisují následující podkapitoly. Úrovně vyspělosti procesu ilustruje Příloha č. 1: Stupně vyspělosti procesu..

2.1.1.1 Úroveň 1 – úvodní (initial)

Nultá úroveň vyspělosti procesního vývoje je podmíněna neexistencí procesů. První, úvodní úroveň společnost dosahuje samotnou existencí procesů. Pro dosažení této úrovně kromě samotné existence procesů není nutné nic vykonat. Zároveň na této úrovni neexistují definice procesů, vývoj probíhá intuitivně a problémy jsou řešeny ad hoc. Bez definice procesu je obtížné plánovat i vyhodnocovat. Tato úroveň vyspělosti procesu často nevede ke kýženým výsledkům z hlediska vynaložených nákladů, času i kvality (výjimkou mohou být ty nejjednodušší aplikace). Na této úrovni není definováno

také řízení lidských zdrojů a interní i externí komunikace se zákazníkem, která je omezena na vzniklé problémy a nemá jednotnou podobu. Komunikace tak může probíhat více kanály s více lidmi separátně, což může negativně působit na již zmíněné ukazatele (1, str. 12).

Pro podnik a jeho existenci je nezbytně nutné zvýšit úroveň vyspělosti.

2.1.1.2 Úroveň 2 – opakovatelná (repeatable)

To, že se vývojový tým nebo organizace vyskytuje na druhé úrovni, nemusí být jeho přímým záměrem. Může nastat k využití tzv. best practices, které jsou empiricky na základě historie vývojového týmu/organizace. Na této úrovni jsou tedy části procesu stále řešeny ad hoc a části na základě opakovanými osvědčenými postupy z minulosti. Na této úrovni není zajištěn ucelený popis procesu ani jeho částí, vývojový tým/organizace tedy stále postrádá systematický přístup (1, str. 13).

V případě obměny nebo velké fluktuaci ve vývojovém týmu/organizaci dojde velmi pravděpodobně k poklesu na první úroveň.

2.1.1.3 Úroveň 3 – definovaná (defined)

O třetí úrovni hovoříme jako o úrovni systematického vývoje software zejména z důvodu existence systematického a metodického popisu celého procesu vývoje software, na rozdíl druhé úrovně. Tento popis je jednou z podmínek kvality, která ale není sama o sobě postačující. Ačkoli je třetí úroveň uprostřed SW-CMM, z hlediska vývoje kvalitního software se jedná o první dostačující úroveň (1, str. 13-14).

2.1.1.4 Úroveň 4 – měřitelná (measured)

Systematickým a metodickým popisem celého procesu bohužel nezískáme přehled o jeho efektivitě. Na čtvrté úrovni jsou stanovovány metriky, které musí být relevantně určeny na základě charakteru projektu, nebo použité obecné ukazatele. Pro všechny ukazatele musí nicméně platit to, aby byly smysluplné a přinášely hodnotu. Měřením a vyhodnocováním je možné dojít k závěru, zda proces splňuje představy vývojového týmu či organizace a také, zda je sledovaný proces správný (1, str. 14-15).

2.1.1.5 Úroveň 5 – optimalizovaná (optimizing)

Pátá, nejvyšší úroveň SW-CMM představuje stav, kdy existuje systematicky a metodicky popsany proces se stanovenými klíčovými ukazateli a na základě jejich vyhodnocování

je v případě odchylek implementován opravný mechanismus. Úspěšným zvládnutím této úrovně se vývojový tým/organizace postupně poučuje z chyb a předchází dalším chybám. Iterativně tak dochází k ladění procesu, jehož efektivita se přibližuje k potencionálnímu maximu (1, str. 16-17).

2.2 Metodiky vývoje software

O sadě procesů, které nám umožňují komplexní zhodnocení a vylepšení, tedy 4. a 5. úroveň SW-CMM, můžeme hovořit jako o metodice vývoje software, o které pojednává následující kapitola.

Při vývoji software je nutné metodiku znát, nicméně není nezbytně nutné se jí striktně držet. To zejména z toho důvodu, že vývoj není totéž, co výroba. Ve výrobě se výrobní proces striktně dodržuje. V případě vývoje je zde určitá „volnost“ a neexistuje předpis, jak konkrétně dojít k výsledku, který zákazník požaduje (2, str. 17)

Metodika ve vývoji software představuje průvodce. Stanovuje, co dělat, nestanovuje jak (1, str. 17).

2.2.1 Rigorózní metodiky

Rigorózní neboli tradiční metodiky vznikly dříve než agilní metodiky. Pro tradiční metodiky je specifické:

- přísnější dodržování procesů a co nejmenší míra vágnosti,
- přísnější termíny,
- striktní popisy požadavků,
- striktně dané role
- maximalizace specializace, téměř neexistují přesahy (horší zastupitelnost a komplikovanější komunikace),
- vysoké nároky na dokumentaci (oddalování doručení funkcionality či její části).

Tradiční metodiky je vhodné využít zejména u rozsáhlejších projektů. Při kooperaci více týmů, integraci více systémů, koexistence více technologií. Vhodné jsou taky pro projekty, kdy je jasně daná specifikace vyvíjeného software s vymezeným rozpočtem ve stanoveném časovém rámci. Obecně lze tradiční metodiky využít v takových projektech,

jejichž charakter vyžaduje co nejvyšší řád a předvídatelnost procesů. Jasně dané role, zodpovědnost a časový harmonogram (1, str. 23-24).

2.2.1.1 Vodopádový model

Vodopádový model neboli Waterfall model vznikl v roce 1979, představil jej Winston Royce a je jednou z nejznámějších tradičních metodik. Model se skládá z fází, které zobrazuje Příloha č. 2: Vodopádový proces. Aby bylo možné vodopádový proces řídit a spravovat musí tyto fáze následovat po sobě dle daného předpisu.

Winston Royce také identifikoval problém modelu. V případě, že byla v rámci testování identifikována chyba, předcházelo její opravě mnoho schvalování, které často nemělo pozitivní vliv na časový harmonogram projektů (modrá a červená). Identifikovaný problém je nejčastější překážkou v implementaci modelu. Vzhledem k tomu, že původně byl model zamýšlen pouze sekvenčně, naznačované šipky napovídají, že byl upraven v tom smyslu, že bylo možné vrátit se v procesu o krok zpět (3, str. 328-330).

Na základě výše uvedeného není vodopádový model v dnešním vývoji software příliš populární. Je ale možné ho použít například u velmi krátkých projektů, kde je v případě odhalené chyby celý byrokratický proces rychle odbaven.

Tabulka č. 1: Výhody a nevýhody vodopádového modelu (1, str. 26)

Výhody	Nevýhody
Jednoduchost	Dlouhý časový interval mezi zadáním a výsledkem
Snadná orientace ve fázích projektu	Problematické opravy při nalezení chyb
Možnost přesného plánování	Zákazník není zapojen v procesu
Jednoznačné zadání s minimem změn	Linearita (fáze následující za sebou)

2.2.1.2 Iterativní metodiky

Vstupem pro existenci iterativních metodik jsou nevýhody vodopádového modelu popsané v Tabulka č. 1: Výhody a nevýhody vodopádového modelu. Mezi zadáním a výsledkem je tedy menší časový rozdíl, opravy nalezených chyb jsou řešeny rychleji a pravidelně v každé iteraci. Zákazník je průběžně zapojován do procesu tak, aby

upřesňoval svoje požadavky, případně korigoval vývojový tým správným směrem (4, str. 101-103).

Iterativní vývoj je založen na opakování jednotlivých fází metodiky v cyklech, které jsou popsány následovně:

1. specifikace požadavků (system, software, business requirements),
2. analýza (analysis),
3. návrh (program design),
4. kódování, resp. deployment (coding),
5. testování (testing).

Pokud v rámci testování není dosaženo požadovaného cíle, začíná další iterace opět v prvním bodě až k poslednímu tak dlouho, dokud není zajištěna požadovaná přesnost dle specifikace. První iterace bývají zpravidla hrubé, což umožňuje relativně rychlé nasazení, resp. rychlou zpětnou vazbu za relativně nízkých nákladů. V dalších iteracích pak případně dochází k upřesňování a ladění detailů (4, str. 102).

V případě dosažení cíle daného požadavky, je provedeno vydání (release) dané funkcionality v takové podobě, v jaké bylo specifikováno.

Společná specifika agilních metodik jsou:

- spustitelný kód po každé iteraci (zadavatel má možnost seznámit se s výsledkem a dříve tak vznášet požadavky na změny tzv. vznést change request),
- testování v každé iteraci (které jsou v následující iteraci opraveny),
- změna je součástí procesu (pokud je zadání ze strany vývojového týmu chybně interpretováno, je po zpětné vazbě v další iteraci prováděno zlepšení, aby bylo dosaženo cílovým požadavkům) (1, str. 28).

Mezi iterativní metodiky se řadí například:

- RUP (Rational Unified Process),
- Unified Software Development Process (UP),
- Spirálový model,
- OPEN.

Tyto metodiky řadíme do iterativních z toho důvodu, že se v rámci jejich použití opakuje specifický proces s podobnými (výše popsány) prvky. Bližší popis těchto metodik přesahuje rozsah teoretické části této práce (2, str. 34-39).

2.2.2 Agilní metodiky

Z důvodu dynamicky se rozvíjejícího trhu, rychlé informovanosti a také stále vyšší konkurenci. Rigorózní metodiky v tomto prostředí nejsou dostatečně efektivní a z toho důvodu se prosazují metodiky, které reagují na změny požadavků rychleji (například agilní metodiky). Metodiky s počátkem v druhé polovině 90. let prosazují myšlenku co možná nejrychlejšího vývoje funkcionality s následným předložením zákazníkovi pro získání zpětné vazby, ze které vyplynou případné další modifikace.

Agilních metodik existuje více, všechny mají ctít shodné hodnoty. V únoru 2001 vznikl „Manifest agilního vývoje software“ podpisem představitelů jednotlivých metodik, kteří zároveň vytvořili „Alianci pro agilní vývoj software“ (5).

2.2.2.1 Hodnoty agilních metodik

Na základě schůzky sedmnácti odborníků z oblasti vývoje software došlo ke zjištění, že jejich snahy zprostit vývojový proces přílišné byrokracie a nevhodných procesů jsou více či méně společné, sestavili „Manifest agilního vývoje software“ (5).

Manifest staví na následujících hodnotách:

- umožnit změnu je efektivnější, než se jí snažit bránit,
- je třeba být připraven reagovat na nepředvídatelný sled události, protože ty nepochybně nastanou.

Autoři na základě těchto hodnot upřednostňují:

- | | | |
|-------------------------------------|------|------------------------|
| • individualita a komunikace | před | procesy a nástroji, |
| • funkční software | před | obsáhlou dokumentací, |
| • spolupráce se zákazníkem | před | sjednáváním kontraktu, |
| • reakce na změnu | před | plněním plánu. |

S dodatkem, že položky, nacházející se na pravé straně mají také svou hodnotu, nicméně více hodnotné jsou položky vlevo (1, str. 31), (6).

Na základě deklarovaných hodnot je definováno 12 hlavních principů agilních metodik (6).

1. **Nejvyšší prioritou je uspokojení zákazníka brzkými a kontinuálními dodávkami hodnotným software.** Zákazníci jsou spokojenější, pokud obdrží funkční software na pravidelné bázi namísto dlouhých časových úseků.
2. **Změny v požadavcích jsou vítané i v pozdní fázi vývoje. Zákazník má možnost získat konkurenční výhodu.** Schopnost vyhnout se zpoždění v případě, když dojde ke změně požadavků.
3. **Dodávky funkčního software frekventovaně od několika týdnů po několik měsíců s preferencí kratších časových úseků.**
4. **Zadavatelé a vývojáři pracují na projektech společně na denní bázi.** Při spolupráci a pohledů více zainteresovaných osob vznikají lepší rozhodnutí.
5. **Podpora, důvěra a motivace pro zapojené jedince.** Motivované týmy odvádějí práci lépe.
6. **Nejúčinnější a nejefektivnější metoda sdělování informací ve vývojovém týmu je přímá konverzace.** Komunikace je úspěšnější, pokud je vývojový tým centralizován.
7. **Funkční software je primárním měřítkem pokroku.** Doručování funkčního software zadavateli je primární faktor úspěchu a měřitelného pokroku.
8. **Agilním procesem k podpoře a konzistenci vývojového tempa.** Ustanovení opakovatelné a udržitelné rychlosti, ve které je tým schopen dodávat funkční software opakovaně s každým vydáním.
9. **Pozornost kladená na technický detail a návrh zlepšuje agilitu.** Správné dovednosti a dobrý návrh utvrzuje tým ve schopnosti udržení tempa, konstantního vylepšování produktu a udržitelných změn.
10. **Jednoduchost.** Vyvinout právě tolik, kolik je v současnosti nutné k dokončení.
11. **Nejlepší architektury, požadavky a návrhy pochází ze samoorganizujících se týmů.** Zkušenosti a motivování členové týmů, kteří dělají rozhodnutí, přebírají odpovědnost, sdílejí myšlenky, komunikují přímo s ostatními členy týmu.
12. **V pravidelných intervalech tým uvažuje, jak se stát efektivnějším. Poté ladí a přizpůsobuje jejich proces.** Osobní rozvoj, rozvoj procesů, rozvoj ve specializaci a technických znalostech pomáhá týmu k efektivnější práci.

Agilní metodiky, které vychází z výše uvedených hodnot a principů jsou:

- TDD (Test Driven Development),
- FDD (Feature Driven Development),
- XP (Extreme Programming),
- Crystal,
- ASD (Adaptive Software Development),
- DSDM (Dynamic Software Development Method),
- Lean Development,
- SCRUM Development Process.

Jednotlivé metodiky nejsou z důvodu rozsahu práce blíže specifikovány. Práce se detailně zaměřuje pouze na SCRUM Development Process z důvodu zavedení této metodiky ve sledované společnosti, resp. ve sledovaném projektovém týmu (2, str. 43-62)

2.3 Scrum

Scrum předpokládá, že na konci každé iterace je zákazníkovi dodán funkční software nebo jeho část. Metodika využívá různé nástroje a praktiky, jejichž upřesnění obsahují návazné podkapitoly. Metodika také předpokládá, že některé procesy jsou empirické a nelze je popsat.

2.3.1 Role

Tato kapitola popisuje role, se kterými se v metodice Scrum můžeme setkat. Obecně lze tvrdit, že v agilních metodikách je méně rolí, než v tradičních metodikách. Agilní metodiky tak mají vyšší zastupitelnost členů týmů. Tradiční naopak nižší zastupitelnost ale také jasně definované odpovědnosti. V metodice Scrum je zpravidla odpovědný celý tým (7, str. 72).

2.3.1.1 Pigs & Chickens

Nejprve je nutné zmínit, že rozdělení Pigs & Chickens nemá žádnou hanlivou konotaci. Pigs & Chickens představují dva hlavní segmenty, se kterými se v metodice Scrum setkáváme (1, str. 55)

Pigs

- osoby, přímo zahrnuté v projektu,
- součást projektového týmu,
- odpovědní za výsledek vývoje,
- PO, SM, STM.

Chickens

- osoby, kterých se projekt týká nepřímo,
- nejsou odpovědní za výsledek vývoje,
- koncový uživatel, člen vedení organizace, konzultanti.

Popis níže je věnován rolím, které jsou z pohledu diplomové práce nejdůležitější a spadají přímo do projektového týmu (1, str. 56-57).

2.3.1.2 Product Owner

Product Owner (dále také PO) je typicky klíčový stakeholder projektu. Product Owner zastupuje v projektovém týmu zákazníka (kde zastupuje jeho zájmy) a je zodpovědný za definice vize týmu. Definuje směr týmu a také oblasti, kterým se má vývojový tým zcela vyhnout.

Povinností Product Ownera je také tvorba a prioritizace backlogu. Jednotlivé požadavky jsou zadány zákazníkem, kterého představuje Product Owner, který na základě těchto požadavků vytvoří User Stories, které následně dostanou prioritu (ta se může v průběhu času měnit) definující, kdy se tyto User Stories mají dostat ke zpracování.

PO může být interním či externím zaměstnancem. PO musí respektovat technickou část týmu a měl by týmu dovolit pracovat samostatně a nezávisle, má ovšem pravomoc ke zrušení sprintu, vyžaduje-li to situace (8, str. 33-34), (9, str. 315).

2.3.1.3 Scrum Master

Scrum Master (dále také SM) je role, metodikou vyžadovaná. Tato role spočívá především v zajištění podpory scrum týmu a vytvoření prostředí, ve kterém se bude tým schopen maximálně koncentrovat na odbavování user stories. Podpora ze strany SM v týmu tedy není technického rázu. Součástí pracovní náplně Scrum Mastera také může být sběr a vyhodnocování metrik.

Činností Scrum Mastera lze najít poměrně velké množství. Pro potřeby této práce není nutný striktní popis všech činností, ale rámcový popis toho, kdo Scrum Master v projektovém týmu je (8, str. 31-33), (9, str. 316).

2.3.1.4 Scrum Team Member

Člen týmu v metodice Scrum, nemá blíže definovanou roli, jako je tomu například u tradičních metodik. Scrum Team Member nezastupuje jednu roli jako například: programátor, tester, kodér. V metodice Scrum je od člena týmu (vyjma Scrum Mastera a Product Ownera) vyžadována všestrannost. Členové týmu se můžou blíže specializovat v oblastech, které preferují. Členové týmu si však musí zachovat rozhled také o dalších oblastech, aby byla mimo jiné možná co nejvyšší zastupitelnost.

Každý člen týmu si vybírá práci libovolně, ze stanovených user stories pro daný sprint. Tuto práci vykonává také samostatně a zpracování je libovolné. Klíčové je držet se doručení požadované funkcionality zákazníkovi a technologií, se kterými tým či firma pracuje. Za předpokladu kooperace jednotlivých členů týmu, si každý vybere práci takového charakteru, aby bylo dosaženo co nejlepšího výsledku sprintu.

Každý člen týmu má právo vznášet věcné dotazy a připomínky, v metodice Scrum je to na rozdíl od tradičních metodik více žádoucí. Toto připomínkování by se mělo týkat hlavně vývojového týmu samotného, nikoli interních procesů zákazníka (1, str. 76-78).

2.3.1.5 Činnost dle roli

V obecné rovině můžeme k jednotlivým rolím přiřadit činnosti, kterými se v rámci metodiky Scrum zabývají a s jakou prioritou. V praxi se ovšem vyskytují také dělené role, nebo jsou povinnosti přiřazeny dle zájmů společnosti, případně jednotlivců.

Tabulka č. 2: Činnost rolí (7, str. 73)

Činnost/povinnost	Scrum Master	Product Owner	Team
Komunikace se zákazníkem	-	primární	sekundární
Správa požadavků	-	primární	sekundární
Správa rozpočtu	-	primární	-
Motivace týmu	primární	-	-
Projektová dokumentace	-	-	primární
Oprava chyb a odstranění překážek	primární	-	primární

2.3.2 Projektový tým

Jak již bylo nastíněno v kapitole 2.3.1.4, jednotliví členové týmu by měli mít všeobecný přehled o technologiích a postupech, které jsou využívány v rámci vývoje v daném projektu. Každý člen týmu pochopitelně nemůže ovládat všechny odborné znalosti, měl by ale mít dostatečný přehled na to, aby porozuměl celému kontextu a přebíral práci od druhého člena týmu.

Skladba projektového týmu je také důležitá z hlediska seniority členů týmu. V projektovém týmu by měli být obsaženi jak nováčci, tak zkušení vývojáři. Složení ovlivňuje cenu projektového týmu, problematiku plánování a také samotné vykonávání jednotlivých úkolů (8, str. 35-37).

2.3.3 Zákazník

Product Owner, kterého popisuje kapitola 2.3.1.2 je často se zákazníkem zaměňován (a o to častěji v případech, kdy je zákazník interního charakteru). Product Owner je ovšem jen zástupcem (reprezentantem) zákazníka, který je hlavním komunikačním kanálem mezi firmou či jedincem na druhé straně.

Zákazník je odpovědný za vznášení požadavků, jaké funkcionality se mají doručit, nebo obecněji, za definici vize a cílů celého projektu. Zákazník také validuje již vyvinutý software, kdy ideálně dává zpětnou vazbu vývojovému týmu, zda je doručená funkcionality dle jeho představ či nikoli (1, str. 80-82).

Jednou z nejdůležitějších oblastí vztahu zákazníka a vývojového týmu je určení délky sprintu o které pojednává kapitola 2.3.4.2.

2.3.4 Artefakty

Tato kapitola pojednává o základních pojmech a artefaktech využívaných v metodice Scrum. Cílem kapitoly není popsat veškeré pojmy a artefakty metodiky, ale rámcové seznámení nezbytné v kontextu diplomové práce.

2.3.4.1 User story

V rámci vývoje funkcionalit pro konečné uživatele je nutné předat požadavky zákazníka vývojovému týmu jednoznačnou formou pro vývojový tým a dostatečně jednoduchou formou ze strany zákazníka. Obě strany si musí vzájemně porozumět, což může být v praxi často obtížné. K čím menšímu porozumění dochází, tím více je generováno chyb,

resp. doručováno méně hodnoty. K zamezení chybovosti využívá metodika Scrum roli Product Owner kterou popisuje kapitola 2.3.1.3. Zastupující osoba tak upřesňuje v průběhu sprintu zadání do takové míry, aby plnila představy zákazníka. Výhodou tohoto přístupu je, že zadávající není nucen do detailu vypracovávat dokumentaci k dané User story a nadměrně tak zatěžovat celý vývojový proces (8, str. 48-50).

User stories by měly mít jednotnou strukturu definující roli, která vyžaduje změnu, cíl této změny a také užitek, tedy k čemu bude daná funkcionality reálně sloužit.

Některá specifika user story:

- stručný popis jednotlivých funkcionalit, které budou po dokončení implementovány,
- umožňují rozložení úkolů v projektu do požadované úrovně detailu,
- bezúdržbové (nebo jen minimálně) celky,
- bližší spojení člena týmu se zákazníkem.

Nevýhody využití user stories:

- vágnost,
- problematika popisu grafického rozhraní.

Nebezpečím user stories může být jejich velikost. V praxi je velmi obtížné user stories ze strany zákazníka zadávat takovým způsobem, aby představovaly z pohledu Scrum týmu jednotlivé funkční celky. User stories jsou tak často dále děleny na více oddělených úkolů, k čemuž zpravidla dochází na schůzce Upřesňovací schůzka (Backlog refinement meeting kterou popisuje kapitola 2.3.5.2 (8, str. 50-53).

2.3.4.2 Sprint

Sprint v metodice Scrum představuje neustále se opakující časový cyklus. Cílem sprintu je v rámci jedné iterace dodat zákazníkovi funkční celek, který bude zákazník schopen otestovat a prohlásit za validní, případně vznést připomínky.

Sprint se od tradičních metodik, co by iterační jednotka, liší tím, že není vnitřně strukturován. Není zde tedy určeno, kdy dochází k vývoji, testování, analýze apod. Tyto fáze jsou realizovány na základě aktuální potřeby, případně priorit (8, str. 43-45).

Obsahem Sprintu (jednoho cyklu) je:

- Sprint Planning Meeting,
- Daily Meetings,
- Retrospective Meeting,
- Review Meeting.

Agendu a detailní popis jednotlivých schůzek popisuje kapitola 2.3.5.

Délka sprintu

Při tomto rozhodování je nutné zohledňovat prostředí, ve kterém se projekt realizuje, jaké je prostředí, ve kterém se pohybuje zákazník a řadu dalších faktorů, které popisuje kapitola 4.2.1 (7, str. 84).

Charakteristika kratších sprintů:

- častější zpětná vazba od zákazníka,
- rychlejší zpracování často měnících se požadavků,
- vhodné pro ladění týmu (měření a retrospektivní schůzky),
- vhodné pro projekty, kde je nutné rychle vidět inkrementální funkcionality.

Charakteristika delších sprintů:

- menší režie a lepší soustředěnost na vykonávanou práci,
- v projektu nepřichází velké množství požadavků na změny,
- vhodné pro týmy, které spolupracují delší dobu.

Tyto charakteristiky jsou (jak již bylo uvedeno výše) závislé zejména na konkrétním zákazníkovi a projektu (1, str. 96).

Sprint management

Sprint je během jeho průběhu nutno řídit. K tomu slouží kromě denní schůzky (která je spíše informační) jak softwarové, tak konvenční nástroje (Scrum board), které jsou v praxi často kombinovány.

Úkolem těchto nástrojů je podávat informace o tom, v které fázi (analýza, vývoj, testování, code review apod.) se daná User Story nachází a který člen týmu aktuálně s User Story pracuje.

V ideálním případě, by se měl být schopen tým řídit pomocí těchto nástrojů sám. Přílišné zasahování do přidělování úkolů narušuje svobodné rozhodování členů týmu a není příliš v souladu se Scrum metodikou (1, str. 92-97).

2.3.4.3 Backlog

S pojmem Backlog se v metodice Scrum setkáváme, pokud hovoříme o seznamu User Stories, které jsou připraveny pro odbavení. Pokud hovoříme o seznamu, který má být odbavený za celý projekt, pak používáme výraz Product Backlog. Pokud hovoříme o seznamu User Stories, pro odbavení v rámci sprintu, používáme výraz Sprint Backlog (8, str. 45-48).

Zásadnější rozdíl mezi pojmy Sprint Backlog a Product Backlog je ten, že Product Backlog je kdykoli během projektu libovolně upravovatelný. Mohou se do něj přidávat nové User Stories, může docházet k přecenění a jiným změnám. Sprint Backlog by měl být po naplánování sprintu co možná nejvíce neměnný, aby bylo možné plánovat a měřit efektivně. V praxi se však modifikaci, přidávání nebo odebírání User Stories nelze vyhnout, jedná se o „daň“ Scrum Metodiky (7, str. 359-360).

Backlog ať už projektový nebo sprintový je v praxi často spravován pomocí specializovaných SW nástrojů. Například ve stylu tabulky, která obsahuje informace o jméně User Story, prioritě, stavu a story points (1, str. 97-99).

2.3.5 Schůzky

Metodika Scrum definuje čtyři základní schůzky (Planning, Daily, Review, Retrospective), o kterých dále pojednává tato kapitola. Dále kapitola popisuje další schůzky, se kterými se můžeme setkat v rámci celého projektu.

2.3.5.1 Zahajovací schůzka (Kick-off meeting)

Kick-off Meeting neboli zahajovací schůzka představuje důležitou část z hlediska projektu. Nicméně v samotných sprintech se nekoná.

Hlavní agendou této schůzky je:

- sestavení týmu a vzájemné seznámení,
- představení projektu a vize,
- vytvoření předběžných User Stories do Product Backlogu,
- stanovení základních atributů projektu,
 - komunikačních kanálů,
 - schůzky a místa konání,
 - délka sprintu,
 - termíny.

Tyto atributy se mohou v průběhu projektu měnit. Je ovšem nutné z nějakého stavu vycházet.

Scrum tým je nutné sestavit v souladu s požadavky konkrétního projektu. Pro Scrum tým je také určen Product Owner, kterého většinou jmenuje zákazník, jako svého zástupce. Po sestavení týmu včetně Scrum Mastera a Product Ownera dochází k vzájemnému představení.

Po vzájemném představení jak členů, tak projektu dochází k prezentaci vize celého projektu a následné vytvoření předběžného Product Backlogu (součástí je připomínkování, technická proveditelnost, diskuze). Product Backlog neobsahuje standardní User Stories, ale pouze jejich hrubou podobu (nižší granularita). Někdy se s touto formou User Stories můžeme setkat jako s Epics (1, str. 100-103).

2.3.5.2 Upřesňovací schůzka (Backlog refinement meeting)

Backlog Refinement Meeting se koná po zahajovací schůzce, nebo častěji, v závislosti na potřebách Scrum týmu. Některé týmy jej organizují v každé iteraci před plánovací schůzkou.

Hlavním cílem této schůzky je připravení User Stories do takové formy, aby jimi mohl být zásoben následný Sprint Backlog. Hrubě vytvořené User Stories (Epics) na zahajovací schůzce jsou převedeny do dostatečně detailní a jasné (jak pro zákazníka, tak Scrum tým) podoby, tak aby bylo možné výsledné User Stories po jejich uhlazení implementovat v rámci budoucích sprintů (jsou zde uvedena akceptační kritéria). User Stories jsou

následně převedeny do finálního Product Backogu, kde jim PO přidělí prioritu (1, str. 104-108).

V rámci této schůzky je nutné vyhodnotit také náročnost jednotlivých User Stories, aby bylo možné s tímto výstupem kalkulovat také v plánovací schůzce.

V případě, že jsou v budoucnu vzneseny požadavky na změny, je nutné reagovat zavedením těchto změn také v Product Backlogu (8, str. 83).

2.3.5.3 Plánovací schůzka (Planning meeting)

Plánovací schůzka obvykle probíhá v místnosti, kde je pouze daný projektový tým s příslušným PO (nebo zákazníkem). Délka plánovací schůzky není striktně daná, nejvíce zde záleží na preferencích konkrétního týmu. Odborná literatura o této schůzce pojednává jako o schůzce, jejíž agendou je:

- backlog review,
- objasnění (klarifikace) user stories,
- obecné diskuze o implementaci a možných problémech, které mohou nastat,
- definice cílů sprintu,
- vytvoření sprint backlogu,
- vytvoření požadavků do budoucna.

Plánovací schůzka je nedílnou součástí každé iterace. Standardně je prováděna jednou za iteraci. V případě nutnosti je možné tuto schůzku zorganizovat vícekrát. Například v případě, že nastane velká změna cíle sprintu, dána vysokou prioritou požadavků na změnu (1, str. 108-109), (7, str. 361).

2.3.5.4 Denní schůzka (Daily Scrum meeting)

Daily Scrum Meeting, nazývaný také jako Stand-up Meeting, je specifickou schůzkou metodiky Scrum. Probíhá na denní bázi po celou dobu sprintu. V rámci této schůzky se sejde celý Scrum tým včetně Product Ownera, který zde zaujímá spíše roli pozorovatele a kontrolora, zda jde projekt správným směrem.

Hlavní cíle schůzky:

- informovanost celého týmu,
- včasná identifikace a řešení problémů,
- plán na daný den sprintu.

Z počátku schůzky Scrum Master stručně shrne aktuální stav sprintu. Poté se jednotliví členové týmu vyjadřují ke třem základním bodům schůzky:

- pracovní náplň předcházejícího dne a výsledek,
- plán na aktuální den,
- výskyt problémů a nestandardních situací, o kterých by měl zbytek týmu vědět, nebo být nápomocen.

Standup není report pro management ani Scrum Mastera. V této schůzce jde především o vzájemnou komunikaci členů týmu (4, str. 128-129).

2.3.5.5 Posouzení sprintu (Sprint review)

Sprint Review Meeting se uskutečňuje na konci samotného sprintu. V případě, že dojde ke zrušení sprintu ze strany Product Ownera, ke schůzce nedojde. Na této schůzce je mimo PO, SM a Scrum týmu přítomen také zákazník. Délka schůzky je podobná jako délka plánovací schůzky. Je ovšem nutné zvážit délku sprintu a konkrétní vyvíjené funkcionality, které délku schůzky mimo jiné ovlivňují.

Zákazník na schůzce přichází do kontaktu s vyvinutou funkcionalitou v rámci daného sprintu. Tato funkcionalita je mu prezentována (demo) a zákazník přitom vznáší případné dotazy, na které odpovídají členové týmu (1, str. 117-118).

V případě, že je zákazník nespokojen, vznesе požadavek na změny. Na základě rozsahu změn dojde k přidání User Stories do Product Backlogu, případně k Backlog Refinement Meetingu (pokud je nutno User Stories dopřesnit), o kterém pojednává kapitola 2.3.5.2 (7, str. 363).

2.3.5.6 Retrospektivní schůzka (Sprint retrospective)

Sprint Retrospective Meeting neboli retrospektiva následuje po Sprint Review na úplném konci sprintu.

Tato schůzka nabízí Scrum týmu příležitost identifikovat oblasti, ve kterých se Scrum tým chce v budoucnosti zlepšit, resp. kterým se chce vyvarovat. Schůzka je povinná pro Scrum tým i Scrum Mastera.

Hlavní agenda schůzky:

- co jsme udělali v tomto sprintu dobře,
- co bychom mohli v následujícím sprintu zlepšit.

Mimo vyhledávání špatných stránek, je z hlediska motivace vhodné, aby se Scrum tým také pochválil za pozitivní výsledky. Scrum Master v rámci moderování schůzky oblasti pro zlepšení monitoruje, zapisuje a následně společně se Scrum týmem těmto oblastem uděluje priority.

Na rozdíl od pozitivních bodů, jsou ty negativní v agendě na následující retrospektivní schůzce (po další iteraci). Body pro zlepšení jsou vyhodnoceny z toho hlediska, zda se tým opravdu zlepšuje v oblastech, které si sám vytyčil (7, str. 363), (8, str. 70).

2.3.5.7 Konečná schůzka (Final meeting)

Konečná schůzka je formálním a také oficiálním ukončením projektu. V rámci schůzky dochází k předání dokumentace, uzavření dohody o správě příslušných systémů a odsouhlasení akceptačních protokolů. Product Owner zhodnocuje průběh projektu, přebírá za zákazníka projekt (konečná akceptace) a zajišťuje technické stránky údržby systému. Scrum Master je moderátorem schůzky, formálně zakončuje projekt za vývojářský tým a předává za tým projekt. Všichni zúčastnění se mohou vyjadřovat k hodnocení projektu (1, str. 119).

2.3.6 Odhady

Odhadování doby trvání, náročnosti či nákladů je z hlediska trojimperativu každého projektu stěžejní. Následující kapitola pojednává o odhadech pracnosti User stories, resp. o jejich nákladnosti.

Některé techniky vhodné k odhadování náročnosti User Stories:

- odhad na základě historických dat,
- znalecký posudek,
- metoda shora-dolů,
- metoda zdola-nahoru,
- planning poker.

Bez ohledu na to, jaká technika je k odhadu využita, zadavatel požaduje vždy nejpřesnější možnou hodnotu nákladů. Na základě hodnot, které agilní přístup vyznává to však není možné (User stories, které popisuje kapitola 2.3.4.1, jsou záměrně definovány s jistou mírou vágnosti) (7, str. 301).

V rámci této práce bude využito techniky planning poker, jelikož se jedná o techniku, která využívá výhod několika výše uvedených technik, zahrnuje celý tým a je preferována množstvím odborné literatury (8, str. 61).

2.3.7 Měření

Ve srovnání s tradičními metodikami je měření v metodice SCRUM zjednodušené. Zaměřuje se především na to, aby bylo snadno proveditelné, snadno implementovatelné a pochopitelné jak pro zákazníka, tak pro vývojáře.

Metodika SCRUM si neklade za cíl využívat méně přesné měření, metodika takového měření pouze akceptuje (filosofie metodiky a její základní hodnoty). Zaměřování se na co nejpřesnější měření, jehož výsledky bychom v praxi ani nevyužili, je nežádoucí (1, str. 120-121).

Pro efektivní využívání metodiky Scrum je nutné na tým nahlížet jak z vnějšího (business) tak vnitřního prostředí. V souvislosti se zlepšujícím se stavem implementace Scrum metodiky si v rámci této práce klademe následující otázky (11, str. 4-5):

- Jak spokojený je Scrum Team a Stakeholder?
- Zvyšujeme stále vyspělost našeho týmu?
- Odpovídá úsilí vývojářů obchodním výsledkům?
- Zvyšujeme množství dodané práce (kvalita v souladu s normami)?

O metrikách, jejichž úkolem je tyto otázky zodpovědět pojednávají podkapitoly.

2.3.7.1 Spokojenost (Happiness – H)

Míra spokojenosti (štěstí) je nedílnou součástí nejen metodiky Scrum. Pokud však o štěstí hovoříme v rámci metodiky Scrum, tak je to ukazatel nezbytný pro udržitelnost samotné metodiky jak v projektu, tak v končeném důsledku i ve společnosti. Spokojenost Scrum týmu znamená spokojenější, výkonnější a motivovanější zaměstnance, kteří jsou schopni dosahovat hodnotnějších výstupů (12, str. 280-282).

Kvantifikace

Měření probíhá na retrospektivní schůzce. V rámci této schůzky jsou zúčastněnými zodpovězeny například tyto otázky (stupnice 1-5, umožňuje-li to charakter otázky):

- jak se cítíte ve své roli (ve společnosti),
- jak vnímáte společnost jako celek,
- proč se takto cítíte (viz. předchozí otázky),
- definujte jednu skutečnost, která by vás učinila v následujícím sprintu šťastnějším.

Následně si tým na základě předchozích odpovědí vytyčí úkol s největší prioritou, na jehož zlepšení bude následující Sprint pracovat (vč. akceptačních testů). K ověření úspěšnosti zlepšení v dané oblasti dojde na následující retrospektivní schůzce.

Metrika očekává, že jednotliví členové týmu jsou maximálně otevření a nezatajují informace. Tým je tak maximálně informovaný a transparentní (12, str. 289-293).

2.3.7.2 Míra loajality zákazníků (Net promotion score – NPS)

Podobně jako na spokojenost členů Scrum týmu, nahlížíme také na spokojenost zákazníků či stakeholders. NPS z pohledu zákazníka vyjadřuje, s jakou pravděpodobností by daný zákazník doporučil produkt či službu (v tomto případě vývoj software).

NPS využívá pro hodnocení škálu 1 – 10 (kterou si samozřejmě můžou jednotlivé Scrum týmy přizpůsobit, dle vlastních požadavků). V praxi jsou často označovány hodnoty vyšší než 8, jako uspokojivé. Vždy však záleží na zasazení metriky do kontextu (11, str. 6-7).

2.3.7.3 Rychlost (Velocity – V)

Velocity nalezne své uplatnění zejména při plánování v rámci Sprint planning, nebo také odhadování doby trvání projektu. Velocity pomáhá odhadnout, jak rychle bude dokončen daný objem práce, resp. jak dlouho bude trvat určitý projekt.

Metrika je často kalkulována na základě historických dat z 3-5 předešlých sprintů. Pokud bylo například v rámci posledního sprintu (14 – denní cyklus) dodáno a zároveň schváleno zákazníkem 10 user stories, kde byla každá user story oceněna 3 body, pak je Velocity scrum týmu rovna 30 (8, str. 63), (13).

Využití metriky:

- možnost vytvořit přesný plán pro budoucí vydávání (vytváří Product Owner),
- možnost odhadovat množství práce provedené za určité období,
- schopnost vytvářet kvalifikované odhady, které jsou založeny na reálných možnostech společnosti (10, str. 40).

Výpočet: $V = \sum OE$

- OE – Původní odhad všech odevzdaných úkolů (Original Estimates of All Approved Cards).

2.3.7.4 Pracovní kapacita (Work capacity – WC)

Pracovní kapacita je součtem veškeré práce reportované během daného sprintu, bez ohledu na to, zda byla user story dokončena, či nikoli. Work capacity by měla být rovna hodnotě Velocity, nebo vyšší. $Velocity > Work\ capacity$ indikuje nadhodnocené náročnosti user story (11, str. 8), (14).

Využití metriky:

- měření množství práce, kterou tým dokáže vykonat v daném Sprintu (bez ohledu na původní odhad),
- možnost kvantifikovat celkové schopnosti (množství odvedené práce) týmu,
- sledování odchylek od reálných odhadů s následným dotazováním a připomínkováním.

Výpočet: $WC = \sum WR$

- WR – Reportovaná práce v rámci user story (Work Reported during sprint).

2.3.7.5 Faktor zaměření (Focus factor – FF)

Focus factor je ukazatelem, který předpovídá množství funkčního produktu, které bude doručeno zákazníkovi v následující iteraci.

Pokud má například vývojový tým 7 členů, kteří jsou produktivní 6 MDs v jedné iteraci a zároveň Velocity toho týmu je rovna 31, aktuální focus factor je roven 0,74.

$$FF = \frac{31}{7 * 6} = 0,74$$

Pokud některý člen týmu onemocní, lze focus factor použít na předpověď množství nových funkcionalit, které bude vyjádřené objemem práce v následující iteraci.

$$Forecast = FF * 6 * 6 = 22 [story points]$$

Využití metriky:

- měření množství funkčního produktu, který je možné uvést do produkčního prostředí,
- optimalizace množství plánované práce.

Výpočet: $FF = \frac{V}{WC}$

- V – Velocity (rychlost),
- WC – Work Capacity (pracovní kapacita).

Odborná literatura shledává hodnotu FF pod 0,5 jako indikaci toho, že vývojový tým nebyl soustředěn na svoji práci. Naopak hodnoty vyšší než 0,8 indikují, že vývojový tým jedná ve prospěch ukazatele a snaží se vypadat perfektní (11, str. 8), (13).

2.3.7.6 Přesnost odhadu náročnosti (Estimation Accuracy - EA)

Ukazatel nachází využití v oblasti vyhodnocení kvality odhadu náročnosti sledovaného týmu. S rostoucí vyspělostí týmu se tento ukazatel blíží hodnotě 1. Tato metrika byla zařazena jako reakce na současný stav týmu v průběhu vypracování této práce (na základě praktické části, resp. kapitoly 4.4), (11, str. 8).

Výpočet: $EA = \frac{DSWC}{V}$

- V – velocity, resp. původní odhad náročnosti dokončených user stories,
- DSWC – reálná náročnost dokončených user stories, která je dána množstvím reportované práce (Done Stories Work Capacity).

2.3.7.7 Burndown chart

Burndown diagram je spojnicovým grafem, kde nezávislou proměnnou představuje čas, resp. den ve sprintu, závislá proměnná poté množství zbývajících práce ve sprintu.

V rámci burndown diagramu nepracujeme s pokročilými ukazateli, rozptylem, směrodatnou odchylkou, regresní analýzou či podobnými nástroji. Vzhledem k nepřesnostem v ohodnocení pracnosti jednotlivých User Stories by využívání těchto nástrojů nepřinášelo vzhledem k pracnosti požadované výsledky (8, str. 65).

Vzhledem k tomu, že ideální průběh sprintu lze charakterizovat přímkou Ideal Progress (Graf č. 1: Burndown chart, příklad), může Scrum Master sledovat odchylky, na které může vhodně reagovat, aby v poslední den sprintu byla odbavena veškerá zbývajících práce (7, str. 361).

Porovnávání výsledků z jednotlivých sprintů je nutné pro dosažení 5. úrovně v modelu SW-CMM, jehož podstatou jsou korekce procesů na základě měření efektivity.

Měření zbývajících objemu práce

Měření zbývajících objemu spočívá ve snaze odhadnout časovou náročnost jednotlivých úkolů v souladu s kapitolou 2.3.6, kdy můžeme využít interního hodnocení, nebo hodnocení využívající reálné ukazatele jako jsou například mandays (člověkodny), (1, str. 126-128).

- Interní hodnocení využívá bezrozměrné jednotky, které si konkrétní Scrum tým může přizpůsobit dle specifických požadavků.

Tabulka č. 3: Interní hodnocení objemu práce, výhody a nevýhody (1, str. 128)

Výhody	Nevýhody
• jednoduchost • hodnocení dle potřeb organizace	• zkreslený údaj • nekompatibilita s hodnocením ostatních subjektů • absence vazby na reálné ukazatele jako jsou náklady, čas

- Mandays (člověkodny) – s využitím tohoto přístupu, kdy je využito techniky planning poker lze profitovat v tom smyslu, že využíváme reálnou jednotku (čas), kterou používáme také na stanovení délky sprintu. Jednotky na ose Y v tomto případě představují mandays (dále také MD/MDs). Přesné ohodnocení pracovních celků je největším problémem tohoto přístupu. S využitím tohoto přístupu lze sestavit Sprint Backlog tak, aby byla jeho pracnost srovnatelná bez ohledu na počet user stories. Tato práce bude na základě doporučení odborné literatury preferovat tento přístup (1, str. 128-129).

Tabulka č. 4: Hodnocení objemu práce s využitím MDs (1, str. 129)

Výhody	Nevýhody
• snadné plánování Sprint Backlogu • jednoduché na interpretaci a pochopitelnost • pracujeme v jednotkách, které běžně používáme v projektu i sprintu	• zkreslený údaj • obtížnost odhadu časové náročnosti user story

2.3.8 Data pro měření

Pro úspěšné zavedení metrik uvedených v kapitole 2.3.7 je nutné shromažďovat nezbytná data. Aby byla zároveň zachována filosofie Scrum metodiky, je nutné vyhnout se přílišné byrokracii a administrativní zátěži Scrum týmu.

V rámci shromažďování vstupních dat pro výše uvedené metriky si klademe následující otázky:

- jaká data je nutné pro měření získat,
- v jakých časových intervalech data shromažďovat,
- zdroj dat.

V rámci zachování co největšího detailu a relevance je nezbytně nutné získávat data od samotných členů Scrum týmu. V souladu s hodnotami Scrum metodiky (a nízké administrativní zátěže) jsou data získávána na denní bázi (v rámci denní schůzky).

Nezbytné informace pro vyhodnocování jednotlivých ukazatelů jsou:

- odhadovaný čas, který zbývá k dokončení user story
- čas věnovaný konkrétní user story (předchozí den).

(10, str. 129-130)

Tyto informace získává Scrum master během denní schůzky, při vzájemné diskusi členů Scrum týmu. Data, která jsou potřebná pro vyhodnocení NPS jsou získávána na Sprint Review schůzce (11, str. 6).

3 ANALÝZA SOUČASNÉHO STAVU

Analytická část se zaměřuje na představení společnosti Avast Software s.r.o. a konkrétního Scrum týmu, který bude detailněji analyzován. V této části práce bude také zhodnocen stav společnosti, resp. týmu z hledisek, které popisuje teoretická část. Na základě analytické části práce budou navržena řešení, jejichž aplikací bude zvýšena efektivita aplikované metodiky Scrum.

3.1 Představení společnosti

Avast Software s.r.o. je společnost, zabývající se bezpečností více než 400 milionů uživatelů v internetovém prostředí. Její historie sahá do roku 1988, kdy čeští výzkumníci Eduard Kučera a Pavel Baudiš pokořili ViennaVirus. Společnost pod stejnojmennou značkou nabízí více typů bezpečnostního software (verze zdarma i prémiové) pro konvenční i mobilní platformy.



Obrázek č. 1: Avast Software s.r.o. logo

Společnost Avast Software s.r.o. nabízí také produkty dříve konkurenční společnosti AVG Technologies (původně GRISOFT, s.r.o.), kterou v roce 2016 odkoupila. Zákazníci společnosti Avast Software s.r.o. si tak mohou vybrat z širokého portfolia kvalitních produktů pro koncové zákazníky i pro firmy (15).

V současné době společnost z centrální pobočky v Praze řídí pobočky nacházející se v:

- Kanadě,
- Velké Británii,
- Nizozemí,
- Izraeli,
- Spojených státech amerických,
- Německu,
- Srbsku,
- Norsku.

3.1.1 Scrum tým

Diplomová práce je zaměřena na konkrétní vývojový tým, jehož původem je společnost AVG Technologies. Tento tým se zaměřuje především na M&A (mergers and acquisitions) projekty a provádění změn pro interní zákazníky v rámci vývoje proprietárního informačního systému (e-shop back-end). Metodiku Scrum tento tým (s drobnými modifikacemi ve složení týmu) využívá přibližně 2 roky.

3.2 Analýza 7S

Níže vypracovaná analýza je využita pro zhodnocení kritických faktorů společnosti Avast, resp. exAVG části společnosti, které rozhodují o jejím úspěchu či neúspěchu na trhu bezpečnostního software.

3.2.1 Strategie

Strategie společnosti Avast Software s.r.o. vychází z modelu one-to-many. Vývoj jedné aplikace je finančně stejně nákladný pro jednoho koncového uživatele, jako pro více koncových uživatelů (s výjimkou provozních a servisních nákladů).

Společnost Avast se zaměřuje především na spotřebitelský trh a malé až střední podniky. Její hlavní strategií je akviziční model založený na freemium produktech, kdy mohou uživatelé získat základní verzi produktů zdarma. Monetizace je založena na placených rozšířených funkcích u produktů pro domácnosti, nebo na samotných prémiových produktech pro malé a střední firmy. Dalším monetizačním kanálem je reklama v aplikacích.

Dalším důležitým prvkem ve strategii společnosti je schopnost analyzovat velké množství dat (big data) a vytvořit tak silnější společnost z pohledu inovací (jeden z přínosů akvizice AVG).

Společnost Avast se tak snaží o maximální diferenciaci a dodání nových funkcí, resp. vyšší bezpečnosti svým klientům.

3.2.2 Struktura

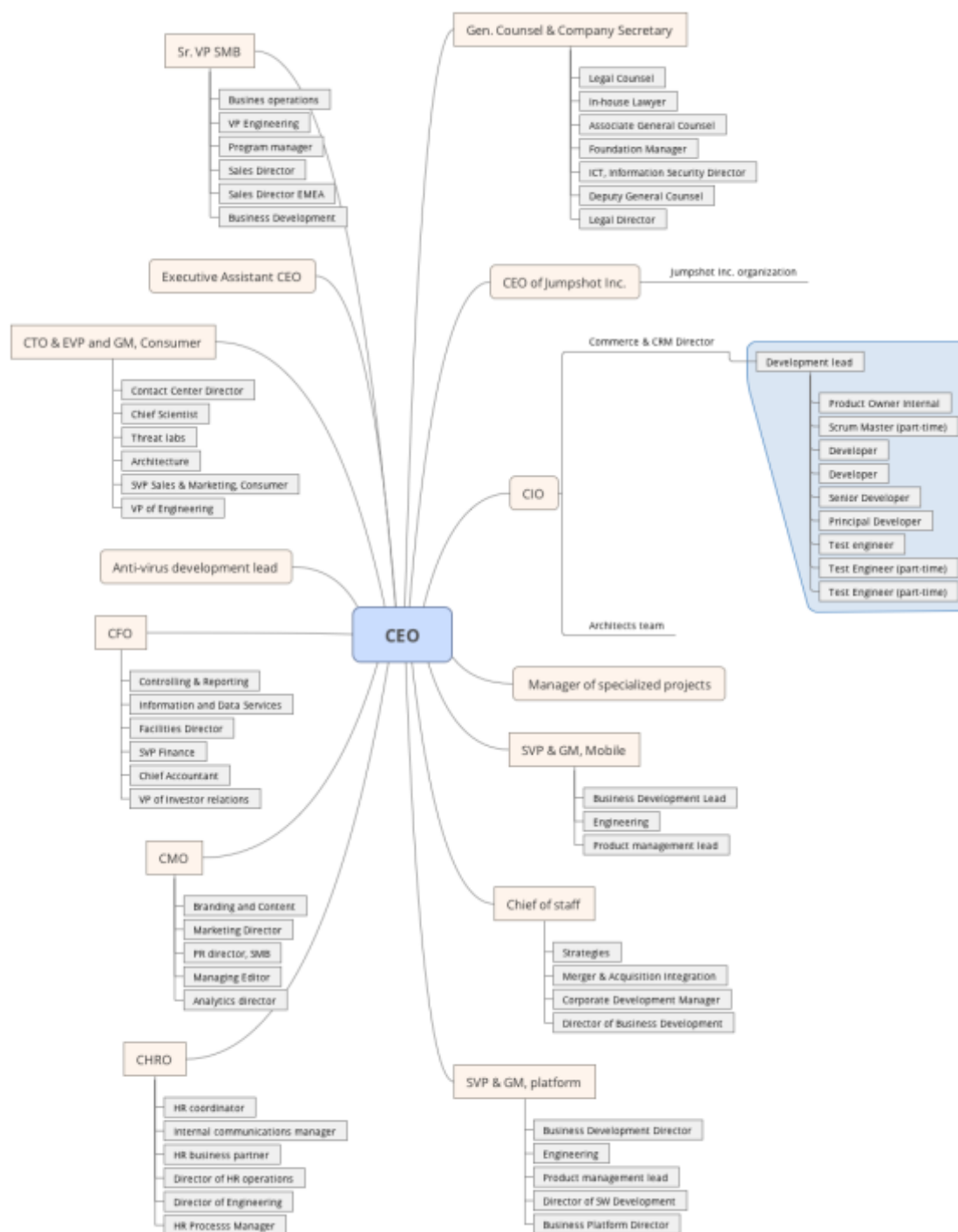
Společnost má v době vypracování této práce 2021 zaměstnanců (Avast + exAVG) jejichž počet se v rámci konsolidace po akvizici snižuje. Původně se společnost Avast software s.r.o. i AVG Technologies s.r.o. vyznačovali plochou, resp. širokou organizační strukturou. Kombinací těchto struktur došlo k razantnímu rozšíření tzv. boardu.

Jednotliví spolupracovníci se sdružují na základě podobnosti úkolů, dovedností nebo aktivit. Soustředění odborníků z jedné oblasti umožňuje delegování velmi specifických úkolů.

Zavádění nových produktů, služeb či technologií vyžaduje kooperaci různých týmů z různých oblastí. Tato skutečnost brání vzájemnému pochopení, vzájemné komunikaci a dosahování cílů. Úspěchy či neúspěchy zavádění nových produktů, služeb či technologií jsou tak výstupem činností mnoha specializovaných útvarů. Je obtížné zjistit jaký byl podíl každé z těchto složek na zdárném či nezdárném dokončení projektu.

Z výše uvedeného vyplývá, že se jedná o funkcionální strukturu s centralizovanou strukturou rozhodování. Obrázek č. 2: Zjednodušená organizační struktura (vlastní zpracování) zachycuje zjednodušený rozpad organizační struktury z vrcholové úrovně, na úroveň sledovaného týmu.

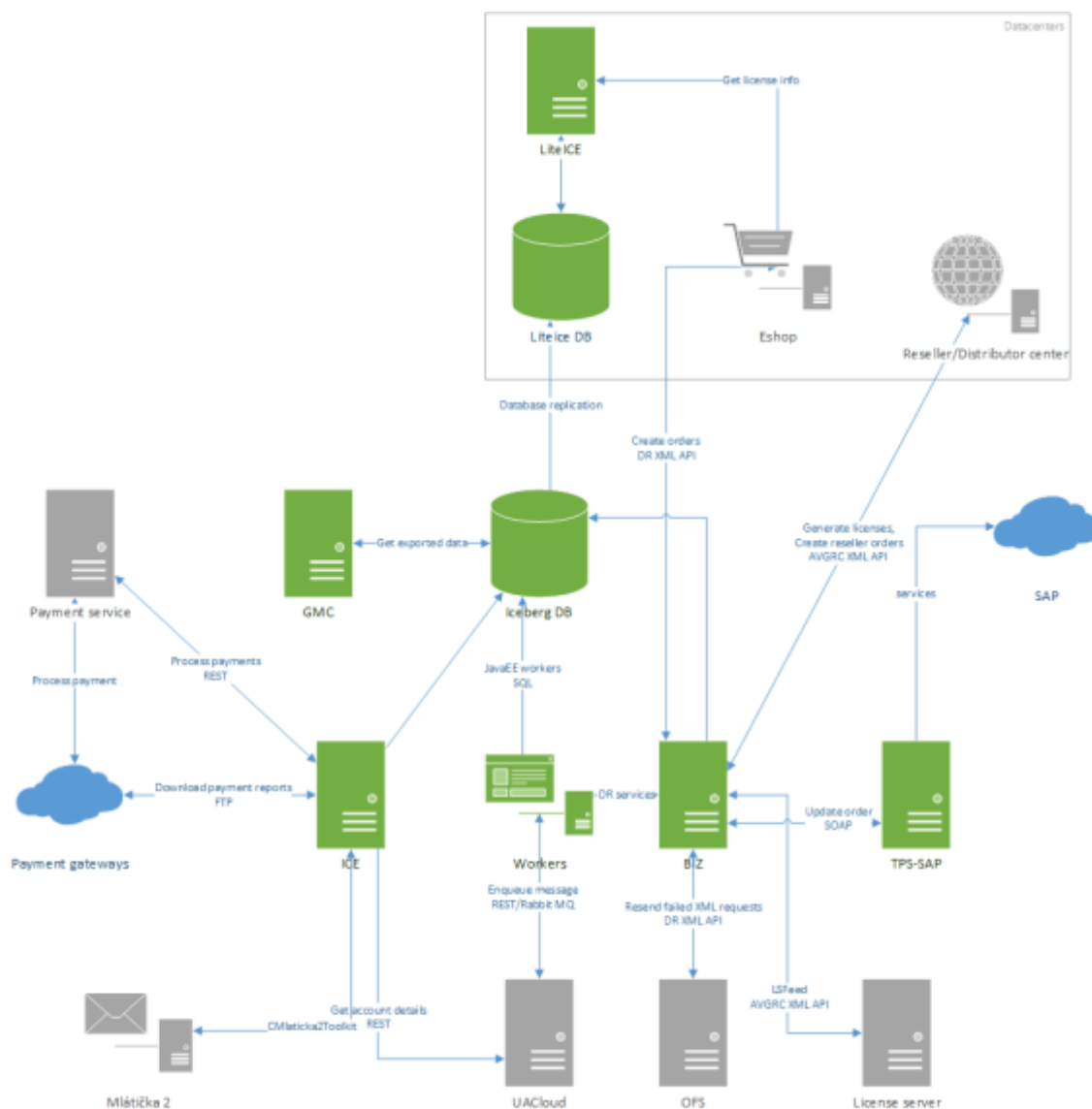
Organizační struktura je z důvodu velkého rozsahu zjednodušena na úroveň, která reflektuje potřeby této práce.



Obrázek č. 2: Zjednodušená organizační struktura (vlastní zpracování)

3.2.3 Systémy

Analýza systémů, které jsou ve společnosti Avast software s.r.o. využívány, je zaměřena pouze na systémy, které jsou významné pro sledovaný projektový tým. Zaměření na veškeré systémy, metodiky a postupy je nad rozsah této práce. Obrázek č. 3: Systémová architektura (převzato z IS Avast software s.r.o.) zachycuje tyto významné systémy (zelená).



Obrázek č. 3: Systémová architektura (převzato z IS Avast software s.r.o.)

3.2.3.1 Databázový server Iceberg

Iceberg DB představuje hlavní databázový server využívající Microsoft SQL. Tato databáze je hlavním úložištěm pro všechna transakční data (licence, objednávky, údaje o zákaznících a další informace).

Iceberg neobsahuje pouze data, ale také velké množství logiky, která je obsažená ve spouštích (triggers), procedurách (procedures), funkcích (funcitons) a v opakovaně spouštěných akcích (jobs).

Databázový server je replikovaný do dalších dvou databázových serverů, které jsou ve Velké Británii a Spojených státech amerických. Tyto databáze jsou využívány pro čtecí operace a pro snižování zátěže na samotný Iceberg. Tyto databáze jsou zachyceny v systémové architektuře jako Liteice DB. Veškeré databáze jsou rozloženy na více strojů z důvodu redundance, resp. bezpečnosti.

3.2.3.2 ICE

ICE je proprietární řešení CRM s dalšími rozšiřujícími funkcemi. Poskytuje GUI i pro další systémy, se kterými komunikuje. ICE je tak hlavním informačním systémem z hlediska zákaznické podpory. Tento systém obsluhuje přibližně 550 interních zákazníků (přibližně 172 000 interakcí denně) a je dostupný pouze z vnitřní sítě.

Klíčové služby ICE:

- správa zákaznických, partnerských a distributorských profilů,
- správa licencí a subskripcí,
- objednávkový proces včetně automatických fakturací,
- správa účetních dokumentů,
- refundace,
- e-mailing,
- správa ceníků,
- reporty,
- a mnohé další.

ICE využívá databázový server Iceberg DB. Iceberg DB komunikuje s množstvím dalších systémů. K samotnému zápisu, modifikace i čtení tak nedochází pouze prostřednictvím ICE.

Obrázek č. 4: ICE, detail zákazníka (převzato z IS společnosti)

3.2.3.3 Lite ICE

Jak již název napovídá, Lite ICE je zjednodušenou verzí systému ICE, která slouží převážně k rychlému poskytování dat externím systémům. Lite ICE poskytuje data o licencích pro e-shop, kde jsou rychlé reakce systému při komunikaci se zákazníkem nezbytné. Lite ICE podporuje pouze čtení, ostatní operace jsou z bezpečnostních důvodů zakázány. Lite ICE využívá jako databázi Lite ICE DB.

3.2.3.4 BIZ

BIZ a ICE jsou důležitou dvojicí systémů, kterou však zákazník (interní) vnímá pouze jako ICE, a to z důvodu, že BIZ obsahuje výpočetní logiku, jejíž výsledky jsou interním uživatelům systému zobrazovány především prostřednictvím ICE.

Tento systém obsahuje také logiku pro automatické či podmíněné zpracování dat jako je odesílání dat do systémů třetích stran, e-mailing, licenční operace, automatická tvorba objednávek, generování voucherů a další. BIZ obsluhuje databázi Iceberg DB.

BIZ obsahuje dvě rozhraní pro komunikaci s okolními systémy, systém pro komunikaci využívá XML API. BIZ poskytuje rozhraní (DR) interním systémům, které s tímto systémem komunikují například při objednávkovém procesu (založení objednávky, účetních dokumentů, e-mailing, vystavení/prodloužení/povýšení licence). Pro externí komunikaci s partnery a distributory využívá rozhraní AVGRC. Vzhledem k tomu, že se jedná o externí službu, dbá společnost Avast software s.r.o. na zabezpečení (využití DMZ).

3.2.3.5 GMC

GMC je systém, který slouží k rozesílání marketingových e-mailů specifickým uživatelským skupinám (personalizace). Z Iceberg DB jsou na denní bázi exportována do tohoto systému aktuální data.

3.2.3.6 Workers

Aplikační server pro JavaEE aplikace. Jednou z nejdůležitějších aplikací, kterou aplikační server obsahuje je Subscription billing worker. Subscription billing worker zpracovává požadavky v rámci opakujících se plateb (subskripce). Tato aplikace zpracovává tisíce platebních operací na denní bázi, což představuje značnou část tržeb.

Další aplikací je Sefaz worker, který zpracovává požadavky pro evidenci tržeb v rámci Brazílie pro tamní finanční instituci (Sefaz).

Aplikační server obsahuje také další aplikace. Popis těchto aplikací však není klíčový z hlediska dosažení cílů, které si tato práce klade.

3.2.3.7 TPS-SAP

TPS je komunikačním rozhraním pro komunikaci mezi BIZ a SAP. TPS umožňuje frontování jednotlivých požadavků, a tak předchází zahlcení SAP. TPS také obsahuje funkci, která umožňuje zaslat chybně zpracovaný požadavek opětovně (automatický obousměrný proces).

3.2.3.8 Další systémy

E-shop

Internetový obchod (eshop.avg.com), kde koncoví zákazníci nakupují produkty AVG. Internetový obchod generuje požadavky, které zasílá systému BIZ. BIZ ukládá data o nákupech a na jejich základě generuje pro zákazníky příslušné licence, dokumenty, e-maily a další.

Reseller/Distributor center

Systém zaměřený na partnery a distributory společnosti Avast, resp. části společnosti AVG. Systém slouží partnerům pro kompletní management objednávek, licencí, zákazníků a dokumentů.

RC/DC komunikuje s BIZ prostřednictvím AVGRC (XML API), prostřednictvím kterého provádí příslušné operace (čtení, zápis, modifikaci).

JIRA

JIRA je softwarový nástroj od společnosti Atlassian. Využívá se především k řízení projektů a ke sledování reportovaných incidentů (issue tracking). V rámci společnosti Avast software s.r.o. je využíván právě k výše uvedeným účelům. Společně s pluginem JIRA Agile je také využíván v rámci metodiky Scrum či kanban. Pomocí JIRA sledovaný tým vytváří project/sprint backlogy, stories a ohodnocení pomocí story points. JIRA Agile obsahuje dostatečně komplexní sadu nástrojů a neomezuje tak sledovaný tým v implementaci metodiky.

Další rozšíření pro JIRA, které společnost Avast využívá jsou:

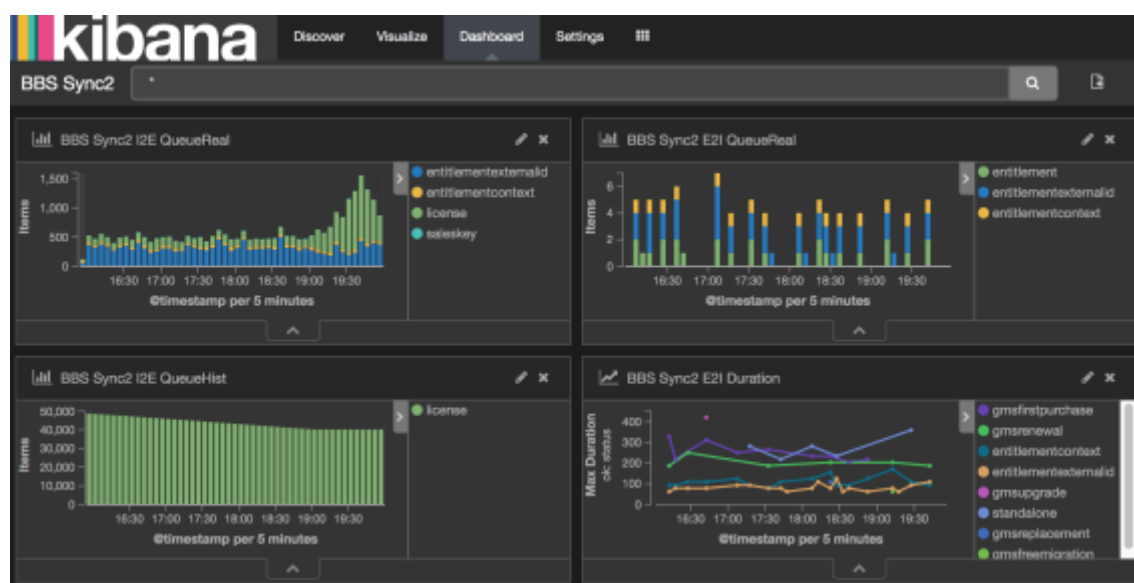
- Bamboo – nástroj pro průběžnou integraci, building, auto testing,
- Bitbucket – služba pro podporu vývoje a kolaboraci týmu při využití Git,
- Confluence – znalostní báze, resp. sdílené úložiště veškerých dokumentů a informací pro zápis např: výstupů ze schůzek, popisů projektů či dokumentace,
- Service Desk – nástroj pro (issue tracking) reportování chyb, který je využíván zejména u servisních týmů, umožňuje neomezenému počtu externích uživatelů (klientů) zadávat své požadavky.

Náhled do prostředí této služby poskytuje například Obrázek č. 7: Scrum board.

Kibana

Jedná se o vizualizační nástroj pro Elasticsearch databázi, který je zároveň vhodný pro správu logovaných záznamů. Vývojový tým tento nástroj používá na vizualizaci stavu systému tzn. sledování chybových hlášení, odbavování front, doby trvání jednotlivých procesů, autobillingu a dalších metrik, které si tým sám definuje.

Náhled, který zobrazuje Obrázek č. 5: Kibana, vizualizace stavu procesů je permanentně zobrazen v kanceláři, ve které se nachází sledovaný projektový tým. V případě nestandardní situace tak přispívá k rychlému zásahu a minimalizaci dopadů zejména v rizikových obdobích po uvedení nových kódů do produkčního prostředí.



Obrázek č. 5: Kibana, vizualizace stavu procesů

Slack

Slack je využíván jako jednotný komunikační instant messaging kanál napříč celou organizací (společně s e-mailem). Umožňuje individuální i skupinovou konverzaci, která je využívána u jednotlivých projektových týmu, nebo orientovaná na celé projekty. Velice využívána je také integrace na cloudové služby.

Git

Systém správy verzí souborů, který je využíván převážně vývojáři. Využívání systému je jednotné napříč celou organizací. Systém detekuje změny v konkrétních repozitářích a umožňuje přechod mezi jednotlivými verzemi. Podporuje také větvení a slučování větví.

Google suite

Komplexní balík služeb od společnosti Google, který umožňuje zaměstnancům společnosti Avast přistupovat k příslušným službám odkudkoli a z libovolného zařízení, bez nutnosti využití VPN. Přístup k uvedeným službám je podmíněný pouze přístupem k uživatelskému účtu Gmail.

Balík obsahuje následující služby:

- e-mail,
- kalendář,
- adresář,
- úložiště souborů,
- kancelářský balík aplikací,
- videohovory a chat.

3.2.4 Styl řízení

Vedoucí týmů (manažeri) a členové týmu k sobě chovají vzájemný respekt. Vzhledem k charakteru metodiky Scrum mají členové týmu přehled o činnosti a cílech celé skupiny. Tým ať už v rámci Scrum schůzek či plánování pohotovostí rozhoduje společně. Manažer vytváří kvalitní prostředí pro práci projektového týmu, snaží se eliminovat negativní skutečnosti ovlivňující tým, resp. vytváří kvalitní prostředí pro výkon práce.

Vedoucí týmu podporuje iniciativu týmu a vítá nové návrhy. Vedoucí týmu hovoří s členy týmu o veškerých problémech, které by mohli mít negativní dopad na jedince či na tým.

Na základě výše uvedené charakteristiky lze vyhodnotit styl řízení jako demokratický. Z charakteru metodiky Scrum však můžeme v projektových týmech sledovat také některé prvky Laissez-faire (16, str. 20).

3.2.5 Spolupracovníci

Mezi spolupracovníky společnosti Avast funguje otevřená a přímá komunikace, které je podpořena lehce neformální komunikací (tykání) napříč pozicemi a postavením v organizační struktuře.

Společnost Avast software s.r.o. vybírá zaměstnance se specializací na následující obory:

- vývoj,
- marketing,
- obchod,
- produktoví specialisté,
- zákaznická podpora,
- administrativa,
- engineering,
- management.

Inovativní přístup společnosti Avast software s.r.o. pochází od světových špiček ve svém oboru. Řada inženýrů, vědců a matematiků absolvovala vzdělání těch nejrespektovanějších institucí, jako jsou Stanford, MIT, Oxford i NASA.

Motivačními faktory, který fungují jako vnitřní motivace jsou:

- možnost tvořit vlastní projekty a účastnit se výběrových řízení k jejich reálné implementaci,
- výměnné pracovní pobyty,
- hackatons.

Motivačním faktorem z hlediska zázemí jsou rozsáhlé relax zóny, posilovna, minigolf, ping-pong, kulečník, fotbálek a mnoho dalších volnočasových aktivit, kde mohou zaměstnanci relaxovat. Zcela unikátní je jídelna na pobočce v Praze, kde mají všichni zaměstnanci jídlo i pití zdarma, ke všemu od těch nejlepších kuchařů v ČR.

Ostatní pobočky tímto benefitem nedisponují. I zde však mají veškeré nápoje a potraviny z automatů zdarma vč. kávy, čajů a slazených nápojů v kuchyňkách.

Mezi oběma českými pobočkami pravidelně jezdí soukromá přímá linka, která zastavuje v bezprostřední blízkosti kanceláří, této dopravy může využít každý zaměstnanec.

Dalším benefitem je parkování v podzemních garážích, stravenky, multisport karta, příspěvek na pojištění, sickdays a pět týdnů dovolené.

Zaměstnanci mají možnost zúčastnit se jazykových kurzů a školení, která si sami vybírají v rámci přiděleného rozpočtu.

Podobný, nebo mírně odlišný standard (v závislosti na místních zvyklostech) jako na pobočkách v České republice lze pozorovat také na ostatních pobočkách společnosti Avast.

3.2.6 Schopnosti

Ve společnosti je mnoho rolí, jejich definice přesahuje rozsah této práce. Obecně se jedná o role manažerské (na různých úrovních hirearchie) a role exekutivní. Na základě rozdílného zařazení v organizační struktuře vznikají individuální nároky na dovednosti zaměstnance.

Obecné dovednosti každého zaměstnance spočívají v:

- připravenost na změnu,
- schopnost přizpůsobit se dynamickému prostředí,
- kontinuální schopnost inovace, vzdělávání a využití nových technologií.

3.2.7 Sdílené hodnoty

Podniková kultura je ve sledované společnosti neformální a velice uvolněná. Podnik dbá na vzdělání a osobní růst svých zaměstnanců. Vizí společnosti je chránit soukromí lidí, nikoli chránit jen jejich zařízení.

Společenská odpovědnost

Avast software s.r.o. daruje 2,5% ze svého provozního zisku do svého nadačního fondu. Prostředky jsou využity na rozvojové projekty v ČR nebo také v zahraničních destinacích, kde se nachází pobočky společnosti.

Hlavními programy nadačního fondu jsou:

- spolu až do konce – program se zaměřuje na dožití,
- spolu do života – zaměřuje se na podporu při zdravotním znevýhodnění,
- spolu se zaměstnanci – program pro podporu vybraných organizací,

- spolu s důvěrou – podpora neziskových organizací a jejich projektů.

Fond společnosti Avast je druhým největším fondem v České republice z hlediska objemu rozdělených nadačních příspěvků.

Otevřenost a transparentnost

V rámci vyhodnocování jednotlivých čtvrtletí probíhají společné All hands meetingy, doprovázené občerstvením, kde mohou zaměstnanci v uvolněné atmosféře vznést dotazy směrem k CEO společnosti. Jedná se tedy o velice otevřenou a přímou komunikaci napříč celou organizací, která představuje významnou sdílenou hodnotu uvnitř společnosti.

Dalšími sdílenými hodnotami vyjma komunikace a otevřenosti jsou hodnoty jako uvolněná atmosféra (minimum stresu), čemuž v některých týmech napomáhá agilní přístup. Zaměstnanci jsou orientovaní prozákaznický a při vývoji aplikací zpravidla předpokládají budoucí business požadavky, proto je často sami připomínají a napomáhají tak maximální kvalitě výsledného produktu.

3.3 Využití Scrum metodiky

Tato kapitola se zaměřuje na míru a způsob využití specifík metodiky. V rámci této části práce jsou v souladu s teoretickou částí hodnoceny:

- role,
- projektový tým,
- zákazník,
- artefakty,
- meetingy,
- odhady,
- měření.

3.3.1 Členové a role Scrum týmu

Tabulka níže zachycuje aktuální stav Scrum týmu z hlediska přiřazení rolí, jak je uvedeno v kapitole 2.3.1. Počet členů v týmu: 8, z nichž 7 je zaměstnáno na plný úvazek a jeden z členů zastupující roli Test engineer na poloviční úvazek.

Tabulka č. 5: Členové Scrum týmu a jejich role (vlastní zpracování)

Role	Počet členů v týmu
Product Owner (interní)	1
Scrum Master	0,5
Developer	2
Senior developer	1
Principal developer	1
Test engineer	2 (1 + 0,5 + 0,5)

3.3.2 Projektový tým

Product Owner

V projektovém týmu je Product Owner (interní zaměstnanec), který zastupuje zákazníky, o kterých pojednává kapitola 3.3.3.

Product Owner se stará o tvorbu a prioritizaci backlogu více projektových týmů, jeho alokovaná kapacita sledovaného týmu v rámci této práce je přibližně třetina z jeho celkové kapacity.

Product Owner v rámci definování požadavků zákazníka využívá unifikovaný formát user stories, kterým se blíže věnuje kapitola 3.3.4.1.

Scrum Master

Role Scrum Master je v projektovém týmu zastoupena jedním členem týmu, který této roli věnuje polovinu z celé své pracovní kapacity. Druhá polovina pracovní kapacity je alokována roli Test Engineer.

Činnost, kterou Scrum Master vykonává je v souladu s rámcovým popisem, na který se zaměřila kapitola 2.3.1.3. Tato činnost se skládá z:

- podpory týmu k dosažení jeho cílů,
- eliminace, eskalace a řešení problémů,
- motivace týmu,
- dohledu nad správnou aplikací Scrum metodiky,
- moderování a koordinace meetingů,
- Scrum evangelizace,
- zajišťování prostředků pro nově příchozí zaměstnance,
- komunikace mezi členy týmu (například na SW úrovni),
- komunikace s okolím (SM chrání tým před vnějšími vlivy).

Scrum Team Member

Role Scrum Team Member se ve sledovaném týmu tak, jak ji definuje kapitola 2.3.1.4 nenachází. Namísto Scrum Team Member bez bližší definice obsahuje tým více rolí:

- developer – SW vývoj PHP, Java, C#, CSS, XML,
- senior developer – SW vývojář s přesahem do návrhu aplikací,
- principal developer – návrh aplikací, začlenění v rámci infrastruktury, výběr technologií, aktivně nevyvíjí,
- test engineer – zajištění kvality SW, především manuální (automatizované s minoritním podílem) testování.

Absence role Scrum Team Member ve formě, jakou metodika Scrum definuje způsobuje problémy v zastupitelnosti jednotlivých členů a rozvoji členů týmů z hlediska specializace (například velký rozdíl mezi specializacemi test engineer a senior developer, tyto dvě role si budou nové know-how předávat poměrně komplikovaně).

3.3.3 Zákazník

Zákazník, jehož zástupcem je Product Owner představuje pro sledovaný tým především:

- obchodní oddělení – M&A projekty,
- oddělení zákaznické péče – změny a rozšíření pro stávající partnery,
- právní oddělení – M&A projekty (zejména u aktivit mimo EU a USA),
- finance – SAP, TPS komunikace,
- BI – poskytování potřebných dat pro podporu rozhodování.

Zájmy výše uvedených zainteresovaných stran prosazuje konkrétní Product Owner, který odpovídá za tvorbu a prioritizaci backlogu v souladu s požadavky zastupovaných zákazníků. Dále také formuluje user stories, které představují parafrázování původních požadavků zákazníků do formy, která je dostatečně srozumitelná a pochopitelná pro vývojový tým.

3.3.4 Artefakty

Tato kapitola se zaměřuje na analýzu artefaktů ve sledovaném týmu. Zohledňuje především jaké artefakty sledovaný tým využívá, resp. nevyužívá a jak k nim přistupuje.

3.3.4.1 User story

V rámci předávání zákaznických požadavků vytváří (zpravidla) Product Owner jednotlivé user stories, prostřednictvím kterých poskytuje informace vývojovému týmu.

Společnost Avast software s.r.o. využívá pro projektové řízení systém JIRA, který umožňuje vytváření, přiřazování, sledování neboli obecně management (nejen) user stories.

Product Owner prostřednictvím JIRA vytváří user stories do předdefinované šablony, která je unifikovaná pro celý projekt. Tato šablona definuje kontaktní osoby, vyžadovanou změnu, cíl této změny, technické informace a také výstup, resp. akceptační kritéria (výsledek implementace funkcionality v konkrétní user story).

Description

Objective: Sync of activated entitlements between ICE and EMS

Justifications: Activated AC via the legacy ICE flow wont be posted to new entitlement system which will have an impact on data consistency, accessing entitlement data using SC console or MyAccount.

Acceptance criteria:

1. Sync tool supports activated ICE sales keys (ICE->EMS)
2. Data migration script (ICE->EMS) is compatible with sync ICE->EMS
3. Deployment plan is created

Key contact:

- Petr Černý
- Martin Pop
- Michal Sedlář
- David Klíma

Technical info:

1. Implement sync of ICE activated SalesKey to EMS entitlement link with type ACTIVATION (it's between activation code entitlement and following active entitlement as id_licence FK in SalesKey table)
2. Check data migration script from <https://jira.cz.avg.com/browse/MA-3053>
3. Create complete deployment plan for Activating entitlement:
 - a. Data migration script
 - b. Sync tool ICE<=>EMS
 - c. EMS activation codes support

Obrázek č. 6: Příklad user story pro projektový tým (převzato z IS Avast software s.r.o.)

V systému JIRA lze také přidávat k user stories přílohy libovolného formátu, což eliminuje problematiku popisu grafického rozhraní.

User story v rámci vývoje prochází fázemi, které zachycuje Obrázek č. 7: Scrum board. Tyto fáze jsou:

- To Do: user story, čekající na vývoj,
- In progress: proces vývoje konkrétní funkcionality, definované v user story,
- Review: kontrola vyvinutého řešení jiným členem týmu,
- Test: testování funkcionality (manuální, automatické, zátěžové, regresní),
- Done: funkcionality je připravena k integraci, resp. vydání do produkčního prostředí.

Zpracování user stories pro sledovaný vývojový tým jsou dostatečně komplexní a pochopitelné. User stories, které by se musely dále dělit na menší celky představují minoritní podíl, což je pravděpodobně dáno senioritou zástupce role Product Owner.

3.3.4.2 Sprint

Jednotlivé iterace trvají 14 kalendářních dní, resp. 10 pracovních dní. Závěr sprintu představuje vydání nových funkcionalit ve spolupráci s release managementem.

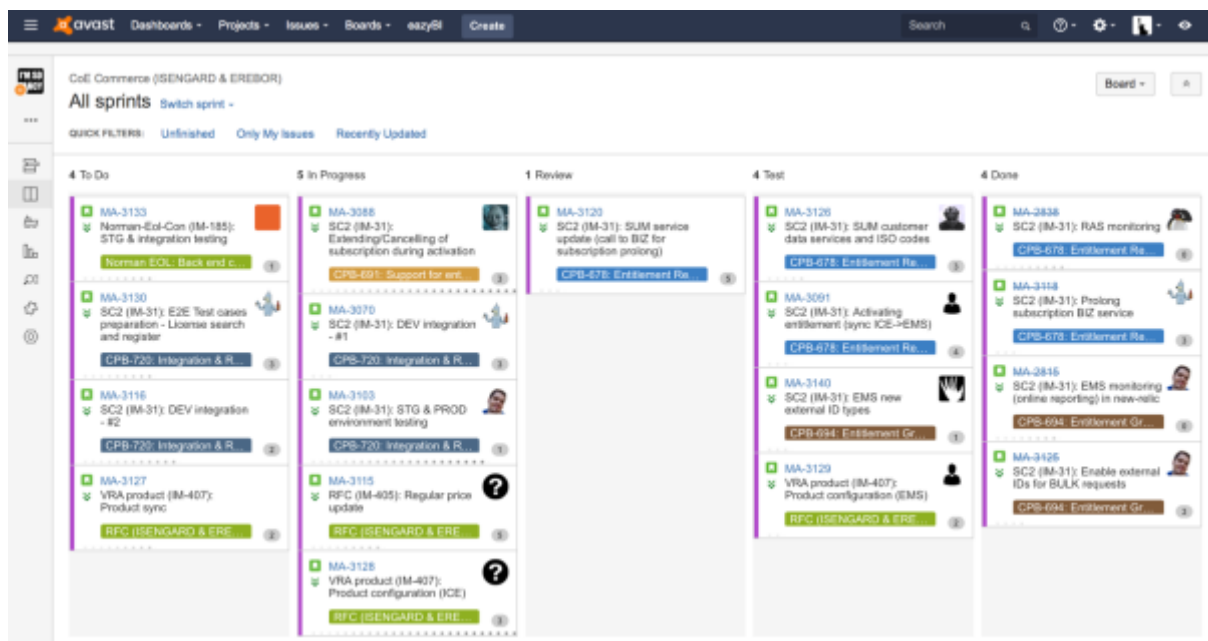
Níže uvedená tabulka představuje průběh sprintu:

Tabulka č. 6: Průběh sprintu (vlastní zpracování)

Den sprintu n	Den v týdnu	Schůzky	Náplň
1	čtvrtek	stand-up	• vývoj (DEV), • QA & bugfix (STG sprint n-1)
2	pátek	stand-up	• vývoj (DEV), • QA & bugfix (STG sprint n-1)
3	pondělí	• stand-up, • backlog refinement (sprint n+1)	• vývoj (DEV), • QA & bugfix (STG sprint n-1), • freeze (12:00 PM)
4	úterý	stand-up	• vývoj (DEV), • regular release (sprint n-1)
5	středa	stand-up	vývoj (DEV)
6	čtvrtek	stand-up	vývoj (DEV)
7	pátek	stand-up	vývoj (DEV)
8	pondělí	stand-up	vývoj (DEV)
9	úterý	stand-up	vývoj (DEV)
10	středa	• stand-up, • review (sprint n), • planning (sprint n+1), • retrospective (sprint n), • scope meeting	• vývoj (DEV), • začátek STG integrace

Pravidelné denní schůzky začínají zpravidla v ranních hodinách. Několik dní z aktuální iterace je logicky ovlivněno předchozí, resp. následující iterací.

Scrum tým využívá JIRA Scrum board pro tracking a management user stories, pohled na Scrum board poskytuje Obrázek č. 7: Scrum board.



Obrázek č. 7: Scrum board (převzato z IS Avast software s.r.o.)

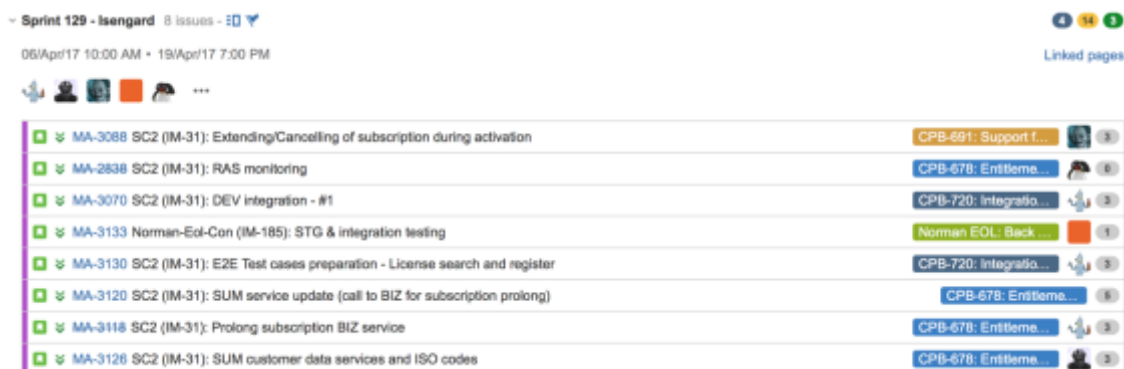
Jednotlivé fáze vývoje user stories jsou popsány v kapitole 3.3.4.1.

3.3.4.3 Backlog

V rámci této kapitoly je provedena analýza artefaktů Sprint Backlog a Project Backlog ve sledovaném projektovém týmu, resp. sledované společnosti Avast software s.r.o.

Sprint Backlog

- veškeré user stories, které obsahuje sprint backlog, mají shodnou prioritu,
- výstup z plánovací schůzky,
- seznam úkolů, které mají být splněny za sprint,
- tým je odpovědný za správu,
- User Stories jsou přiřazeny mezi jednotlivé členy týmu,
- po dokončení User Story je označená jako hotová (kapitola 3.3.4.1).



Obrázek č. 8: Sprint backlog (převzato z IS Avast software s.r.o.)

Product Backlog

- veškeré user stories, které obsahuje project backlog, mají shodnou prioritu,
- libovolně modifikovatelný během celého projektu,
- prioritizace na základě business hodnoty a rizika (nebo jiné smysluplné metriky pro projekt),
- všechny User Stories, nutné k implementaci k dosažení cíle projektu,
- správcem je Product owner,
- tým společně s PO odhaduje náročnost jednotlivých User Stories,
- nejvíce prioritní stories by měly být jasné a zároveň co nejrychleji dodané.

MA-3122	GMS-CB (IM-459): GMS activation codes billing (closing balance)	RFC (ISENGARD & ...)	4
MA-3068	SC2 (IM-31): BAU support (project documentation for KANBAN team)	CPB-720: Integratio...	3
MA-3139	SC2 (IM-31): EMS monitoring (online reporting) in new-relic - PROD phase	CPB-694: Entiteme...	1
MA-3117	Norman-EOL-Con (IM-185): BAU support (project documentation for KANBAN team)	RFC (ISENGARD & ...)	1
MA-3106	MNTR (IM-463): Configure alerts for thresholds	RFC (ISENGARD & ...)	3
MA-3107	MNTR (IM-463): Configure insight chart for GC fulfillment requests in time	RFC (ISENGARD & ...)	1
MA-3110	MNTR (IM-463): Monitor transaction metrics on Tomcats per URL	RFC (ISENGARD & ...)	3
MA-3112	MNTR (IM-463): Monitor auto-renewal notifications	RFC (ISENGARD & ...)	2
MA-3109	MNTR (IM-463): Monitor cancel/refunds	RFC (ISENGARD & ...)	2
MA-3111	MNTR (IM-463): Prepare common transaction & DB metrics for wildfly platform	RFC (ISENGARD & ...)	1
MA-3113	SC2 (IM-31): (Technical dept) Safer asynchronous integration EMS -> Alpha	CPB-678: B2B licen...	1
MA-3123	SC2 (IM-31): (Technical dept) SUMS salesforce update	CPB-678: Entiteme...	2
MA-3131	SC2 (IM-31): EZE Test cases preparation - Ordering	CPB-720: Integratio...	3
MA-3132	SC2 (IM-31): EZE Test cases preparation - Ordering with license operations	CPB-720: Integratio...	3
MA-3137	VPIN-CS (IM-448): Support for activation codes	RFC (ISENGARD & ...)	5
MA-3124	SC2 (IM-31): (Technical dept) Alpha license model	CPB-678: B2B licen...	5
MA-3114	RFC (IM-429): Invoice VAT/TAX info update - Australia	RFC (ISENGARD & ...)	3
MA-2594	GC1 (IM-125): BAU support (project documentation for KANBAN)	Regular Initiatives (L...	3

Obrázek č. 9: Project backlog (pře vzato z IS Avast software s.r.o.)

3.3.5 Schůzky

V rámci této kapitoly jsou analyzovány jednotlivé schůzky, které probíhají v rámci sledovaného vývojového týmu. Zhodnocení průběhu spočívá v dodržování agendy (cíl schůzky), zúčastněných stran a délky samotné schůzky (efektivita).

3.3.5.1 Zahajovací schůzka (Kick-off meeting)

V rámci schůzky k zahájení projektu dochází zpravidla k odhadu náročnosti rozsáhlejších user stories (epics), které jsou v rámci Backlog refinement schůzky nadále děleny na menší celky. Toto ocenění neprovádí tým společně ale pouze představitelé interně zavedené role Principal developer.

Během schůzky nedochází k sestavování týmu. Projektový tým představuje skupina spolupracovníků, kteří jsou spolu spjati již po delší dobu (například z jiných projektů). Nedochází tedy k sestavování projektového týmu na míru, ale k pouhému přiřazování projektů stávajícím týmům.

Tým, kterému je projekt přidělen má již zavedené komunikační kanály, časy schůzek, místa konání schůzek i délku sprintu. K řešení těchto oblastí na kick-off schůze zpravidla nedochází.

3.3.5.2 Upřesňovací schůzka (Backlog refinement meeting)

Backlog refinement se ve sledovaném týmu koná během každé iterace. Tuto skutečnost demonstruje Tabulka č. 6: Průběh sprintu. Čas, který je vyhrazený na schůzku je jedna hodina. Tímto přístupem se projektový tým a product owner snaží předcházet ztrátám pozornosti a také přinášet postřehy z průběhu projektu do aktuálně řešených user stories.

V rámci iterativního přístupu nicméně dochází k upřesnění a ocenění pouze určité (prioritní) části product backlogu. V případě, že by došlo k náhlé změně priorit, je nutné backlog refinement provést opakovaně.

Agendou této schůzky je definování dostatečně podrobných user stories (jak z pohledu zákazníka, tak i projektového týmu), tedy klarifikace. Po připomínkování a vyjasnění zákaznických požadavků probíhá odhad náročnosti jednotlivých user stories, který je prováděn celým projektovým týmem. Konkrétní průběh odhadování náročnosti user stories popisuje kapitola 3.3.6.

Na této schůzce také dochází k backlog review a také obecných problémech o implementaci, které mohou nastat.

3.3.5.3 Plánovací schůzka (Planning meeting)

Plánovací schůzky probíhají v posledním dni předcházejícího sprintu. Zúčastněn je celý projektový tým i product owner. Hlavním cílem této schůzky je vytvoření a prioritizace sprint backlogu, pro následující sprint.

Cíl sprintu není explicitně definován. Projektový tým má za cíl doručit všechny naplánované user stories dle jejich priorit.

3.3.5.4 Denní schůzka (Daily scrum)

Daily Scrum, neboli stand-up je jedinou schůzkou, jejíž místo konání je jiné, než v zasedací místnosti. Tato schůzka se koná na denní bázi přímo v kanceláři, kde je přítomen projektový tým a product owner.

Projektový tým a product owner se během schůzky zaměřují na jednotlivé user stories (pořadí dle priorit). Zohledňováno je především:

- fáze, ve které se user story nachází,
- možné problémy,
- požadavky na dodefinování případných nejasností,
- odhad množství práce k dokončení,
- přiřazování nových úkolů.

V rámci této schůzky si jednotliví členové projektového týmu vzájemně kladou a zodpovídají další otázky, které jsou v souladu s kapitolou 2.3.5.4. Schůzka probíhá v ranních hodinách a trvá přibližně dvacet minut.

Ke sledování user stories a jejich stavu z hlediska procesu vývoje (na vývojovém prostředí) využívá projektový tým a product owner tzv. scrum board, který je zachycen na obrázku



Obrázek č. 10: Scrum tabule (vlastní zpracování)

User story v rámci tabule představuje post-it lístek (žlutá barva). Prioritu určuje jejich seřazení v záhlaví (nejvyšší prioritá vlevo, nejnižší vpravo). V případě, že některý z členů projektové týmu na konkrétní user story pracuje, zaznamená do záhlaví svoje iniciály.

Značky růžové barvy představují pokrok v manuálním testování konkrétní user story, oranžové značky v automatickém testování.

Levý sloupec scrum boardu zachycuje přibližný pokrok v doručení user story, společně s následujícími fázemi po samotném vývoji (review, testing).

Jakmile se ocitne user story ve fázi „done“ je připravena k integraci na prostředí vyšší instance (staging) resp. je připravena k doručení zákazníkovi, které probíhá v pravidelných intervalech v souladu s kapitolou 3.3.4.2.

3.3.5.5 Posouzení sprintu (Sprint review)

Sprint review se koná poslední den iterace, doba trvání je závislá na charakteru a množství dodávaných user stories, zpravidla však trvá jednu hodinu.

Ve společnosti Avast software s.r.o. je přístupováno ke sprint review schůzkám, jako ke sdílení know-how mezi ostatními projektovými týmy. Sprint review sledovaného týmu spočívá ve vzájemné prezentaci nově vytvořených funkcionalit se dvěma dalšími týmy, které pracují na příbuzných systémech.

Podstatou této schůzky tak není prezentace nových funkcionalit (demo) projektovým týmem pro product ownera a především zákazníka, který může výsledek připomínkovat, resp. vznášet požadavky na změny. Prezentace funkcionalit zákazníkovi jsou prováděny na speciálně vyhrazené schůzce (Demo meeting), která se koná dle uvážení zákazníka, resp. product ownera.

3.3.5.6 Retrospektivní schůzka (Sprint retrospective)

Retrospektivní schůzka je stejně jako sprint review závěrečnou ceremonií na samotném konci sprintu. Sledovaný tým na retrospektivní schůzce vyhodnocuje průběh současného sprintu. V rámci vyhodnocení současného sprintu jsou vytyčeny body, které zachycují úspěchy a neúspěchy ve sprintu. Na základě neúspěchů sestavuje projektový tým plán, jak do budoucna dané situaci předejít.

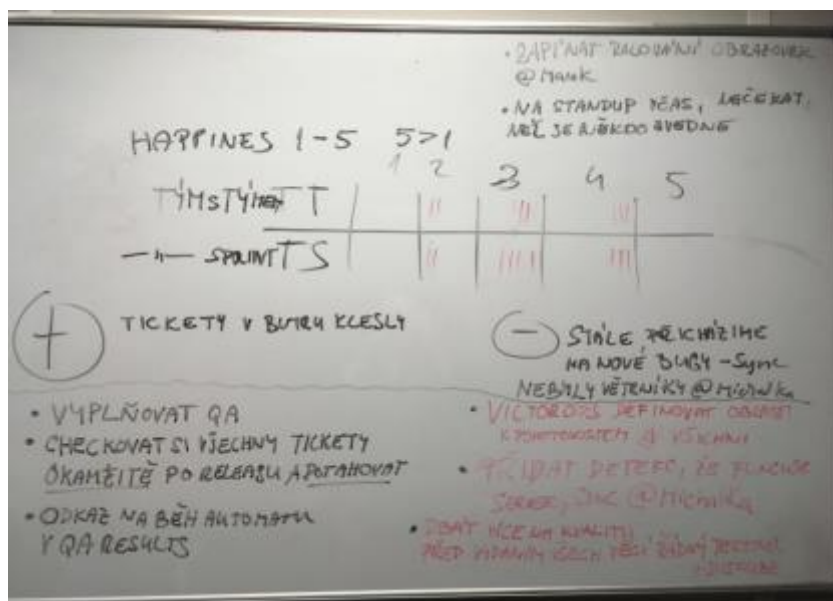
V případě, že z akčních bodů vznikne konkrétní úkol, je k němu přiřazen člen týmu, který odpovídá za jeho výsledek. Akční body jsou dále vyhodnocovány na další retrospektivní schůzce.

Sprint retrospective moderuje scrum master. Ve sledovaném týmu se scrum master snaží často měnit formát schůzky, aby byla pro tým zábavná a uvolňující (gamifikace). Scrum master také schůzku dokumentuje, aby bylo možné sledovat případné změny, či stále se opakující chyby v čase.

Na základě zaznamenaných historických záznamů retrospektivních schůzek byly vyselektovány nejčastější problémy, se kterými se projektový tým střetává.

Těmito problémy jsou:

- absence zpětné vazby,
- nestabilní prostředí (staging),
- nízká odpovědnost za vyvinuté funkcionality při jejich zavádění do produkčního prostředí,
- nízké povědomí o nových systémech napříč spolupracujícími týmy,
- tým má pocit, že není dostatečně oceněn a není vidět jeho přínos, kvůli vývoji back-end služeb, které nejsou vizuálně atraktivní jako front-end.



Obrázek č. 11: Výsup z retrospektivní schůzky (vlastní zpracování)

3.3.5.7 Konečná schůzka (Final meeting)

Konečná schůzka se ve sledovaném týmu koná pouze ve výjimečných případech, nikoli na konci každého projektu. Tato konstelace neumožňuje konečnou formální akceptaci projektu, zhodnocení projektu (Product Owner, zákazník) a zajištění údržby systémů.

Častá absence schůzky v rámci projektů také způsobuje následnou absenci dokumentace, v důsledku čehož nemá operativní tým, který se stará o údržbu systémů v produkčním prostředí, dostatečné know-how pro řešení problémů a deleguje požadavky na původní vývojové týmy.

3.3.6 Odhady

V rámci sledovaného týmu dochází k odhadování náročnosti user stories především na backlog refinement schůzkách pomocí metody planning poker. V případě náhlého přijímání nové user story v průběhu sprintu, probíhá ohodnocení ad-hoc, případně na stand-up schůzce.

Planning poker (někdy také Scrum poker) je gamifikovaná technika, která se realizuje v rámci Scrum týmu, kde každý člen týmu disponuje sadou číselně ohodnocených karet (sledovaný tým pracuje s ohodnocením dle Fibonnaciho posloupnosti). Při hodnocení konkrétní user story každý člen týmu položí kartu, která odpovídá jeho představě o náročnosti úkolu tak, aby číslo nebylo viditelné. Jakmile každý člen týmu provede volbu, jsou na následnou výzvu karty sejmuty.

V případě odlišností hodnocení náročnosti dané user story jsou jednotlivá hodnocení diskutována, dokud se Scrum tým neshodne na konečné hodnotě.

Konečná hodnota může být pouze relativní, nebo může představovat konkrétní hodnotu (například čas, $2 = 2$ MDs). V případě sledovaného týmu se jedná o konkrétní hodnotu 1 bod = 1 MD. Sledovaný projektový tým pracuje s 1 MD, který odpovídá 6,5h lidské práce (zbylá část kapacity je věnována schůzkám a Scrum ceremoniím), (17, str. 56-57).

3.3.7 Měření

I přes to, že se v metodice Scrum setkáváme se značně jednoduššími metrikami, než jak je známe z tradičních metodik, sledovaný tým s metrikami příliš nepracuje. Sledovaný tým si pravděpodobně neklade otázky uvedené v kapitole 2.3.7.

3.3.7.1 Burndown chart

Nástroj, který sledovaný tým využívá je burndown chart. Účelem tohoto diagramu je v co nejjednodušší formě poskytovat informaci o aktuálním stavu sprintu. Scrum Master pracuje s diagramem na denní bázi, kdy zaznamenává množství práce, která byla

vykonána za poslední časový úsek (v tomto případě den). Tyto informace Scrum Master získává na stand-up schůzce od svých spolupracovníků.

Zbývající objem práce je rozdělen na následující části:

- vývoj,
- manuální testování,
- automatizované testování.

Samozřejmostí je vyznačení ideálního odbavování práce. Burndown chart dále poskytuje možnost výpočtu trendu a zobrazuje plánovaný objem práce (pro případy přidávání, resp. odebírání user stories ze sprintu).



Graf č. 1: Burndown chart, příklad (převzato z IS Avast software)

3.4 Vypělost procesu vývoje software

Za předpokladu existence procesů ve firmách můžeme zhodnocovat jejich kvalitu. K měření kvality procesu vývoje software lze využívat více možností, pro účely práce byla k tomuto účelu zvolena filosofie (využití kompletní metodiky je nad rozsah této práce), kterou využívá model hodnocení vypělosti procesu SW-CMM (Capability Maturity Model).

V rámci této práce je SW-CMM využit na úrovni sledovaného vývojového týmu, nikoli však na úrovni celé organizace, či její části.

Ve sledované společnosti dochází k systematickému vývoji software na základě existujícího metodického popisu, který je také zachycen pomocí EPC diagramu a RACI matice, které se blíže věnuje kapitola 3.4.2. Důležitou roli v tomto procesu zastupuje release management a v souvislosti s ním také role vývojáře (integrátora), který zodpovídá za vydávání nových funkcionalit do produkčního prostředí v rámci jednotlivých sprintů.

Dalším důležitým prvkem z hlediska procesů je jasně definovaný sprint, o kterém pojednává kapitola 3.3.4.2.

Jmenování rolí ve sledovaném týmu tak, jak je uvedeno v kapitole 3.3.1 není z pohledu metodiky Scrum standardní. Z pohledu SW-CMM však poskytuje řadu výhod, zejména z hlediska odpovědnosti za příslušené části procesu vývoje software.

3.4.1 EPC diagram vývojového procesu

Aktuální stav procesu vývoje software z hlediska sledovaného týmu mapuje EPC diagram, který je z důvodu delšího rozsahu uveden v Příloha č. 3: EPC diagram vývojového procesu.

3.4.2 RACI matice

RACI matice, která náleží EPC diagramu vývojového procesu. RACI představuje:

- R – role odpovědná za vykonání dané činnosti (responsible),
- A – role, která nese za vykonanou činnost odpovědnost (accountable),
- C – role, se kterou je vedena obousměrná komunikace (consulted),
- I – role, se kterou je vedena jednosměrná komunikace (informed).

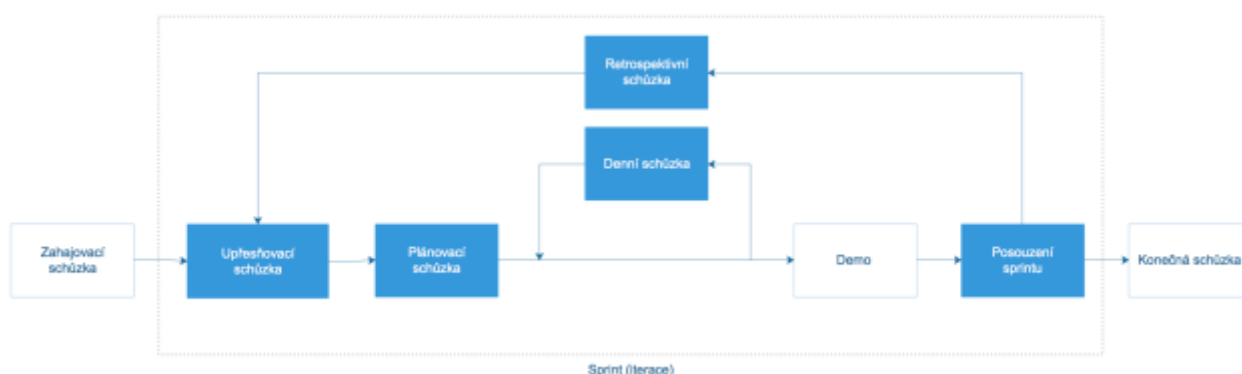
Tabulka č. 7: RACI matice - vývojový proces (vlastní zpracování)

Aktivita	Tester	Developer	Release Managem	Product Owner	Scrum Team	Senior Developer
Přiřazení user story		R			I	
Stažení a sloučení současného produkčního kódu		R				
Vytvoření nové větve		R				
Šetření problému		R				
Vývojový proces		R				
Revize kódu		C				R
Testování – DEV	R				C	
Je user story stále prioritní?	R			C	I	
Vytvoření požadavku na opravu	R	I				
Vytvoření položky pro vydání	R	I				
Příprava vydávacích skriptů	R		C			
Integrace		R	I			
Vydání do produkčního prostředí			R		I	
Testování	R			I		
Vývoj a šetření problému		R				

3.4.3 Sprint

Níže uvedený diagram zachycuje aktuální stav sledovaného týmu z hlediska průběhu jednoho či více sprintů v projektu. Další popis průběhu sprintu je obsažen v kapitole 3.3.4.2.

- prvky bez výplně jsou vykonávány v závislosti na požadavcích zadavatele a charakteru projektu,
- prvky s výplní jsou rutinní součástí každé iterace.



Obrázek č. 12: Diagram průběhu projektu a sprintu (vlastní zpracování)

Dalšími prvky, které ve sledovaném týmu podporují metodický proces vývoje jsou:

- JIRA – softwarový nástroj především k řízení projektů a sledování požadavků,
- plán integrátora – postup pro integraci v závěru vývoje,
- definitions of done (DoD) – definice dodané user story, využívaná v rámci testování.

Na základě výše uvedeného (z hlediska zařazení do konkrétní úrovně v SW-CMM) sledovaný tým spadá do 3. úrovně (defined). Splňuje dle charakteru SW-CMM také 1. a 2. úroveň vyspělosti procesu vývoje software.

Sledovaný tým využívá také některé prvky (Burndown chart, Kibana, Grafana, New relic), které nalezneme ve 4. úrovni SW-CMM. Využití těchto nástrojů však není standardizované a jejich využití je zcela závislé na roli scrum master.

3.5 SWOT

Tato analýza využívá principů SWOT, ve smyslu zhodnocení slabých a silných stránek a také příležitostí a hrozeb. Tato analýza se zaměřuje na sledovaný projektový tým, nikoli na celou společnost Avast software s.r.o.

Analýza sumarizuje veškeré stěžejní skutečnosti, které byly zjištěny na základě analýzy současného stavu v průběhu vypracování této práce (závěrečné stručné shrnutí).

Z Tabulka č. 8: SWOT analýza projektového týmu vyplývá, že realizace změn by měla být zaměřena především na posílení slabých stránek, díky čemuž dojde k eliminaci řady hrozeb a využití většiny příležitostí. V konečném důsledku pak i posílení stávajících silných stránek, či vytvoření nových.

Tabulka č. 8: SWOT analýza projektového týmu (vlastní zpracování)

Silné stránky	Slabé stránky
1. široké využití monitoringu, 2. zavedená metodika Scrum, 3. DoD (QA), 4. integrační příručka, 5. manuální i automatické testování, 6. znalostní báze 7. dokumentace systémů.	1. nízká spolehlivost automatických testů (nestabilní STG prostředí) 2. nízká odpovědnost vývojářů, 3. nedostatečné sdílení know-how se spolupracujícími týmy, 4. nízká motivace (tým nepociťuje dostatečné docenění, absence zpětné vazby), 5. špatné vnímání projektového týmu v rámci společnosti Avast, 6. pouze základní úroveň vyspělosti vývojového procesu, 7. zanedbávání klíčových projektových schůzek, 8. sprinty nemají jasně definovaný cíl, 9. neměnná délka sprintu, 10. nízká zastupitelnost členů v rámci projektového týmu.
Příležitosti	Hrozby
1. konfigurace výkonnějších strojů pro STG prostředí, 2. vyšší kvalita vyvinutého software (růst tržeb), 3. vzdělanější projektový tým, 4. motivovaný a výkonnější tým, 5. prosazení týmu v rámci společnosti Avast, 6. zvýšení vyspělosti vývojového procesu 7. zvýšení povědomí o projektech, 8. flexibilní délka sprintu dle charakteru projektu a zákazníka, 9. zvýšení zastupitelnosti v rámci projektového týmu.	1. nízká kvalita vyvinutého software (pokles tržeb), 2. vysoká fluktuace zaměstnanců, 3. selhání metodiky Scrum, 4. nepřívětivé prostředí pro zaměstnance, 5. snížení úrovně vyspělosti vývojového procesu 6. neporozumění kontextu v rámci konkrétního projektu, 7. nedodržení termínů projektů z důvodu absence, resp. nízké zastupitelnosti.

4 NÁVRH ŘEŠENÍ A PŘÍNOS NÁVRHŮ ŘEŠENÍ

Teoretická část práce poskytla nezbytné know-how k následné analýze současného stavu, ve kterém se společnost Avast software s.r.o. nachází. Na základě komparace těchto dvou pramenů bylo možné sestavit vlastní návrhy řešení, o nichž pojednává tato kapitola.

Některá z navrhovaných řešení (měření) byla přijata již během vypracování této práce, jelikož jsou vhodným ukazatelem výsledného pokroku z hlediska úrovně vyspělosti vývoje software. Po následné implementaci zbylých návrhů řešení bude možné porovnat situaci před a po implementaci.

4.1 Role

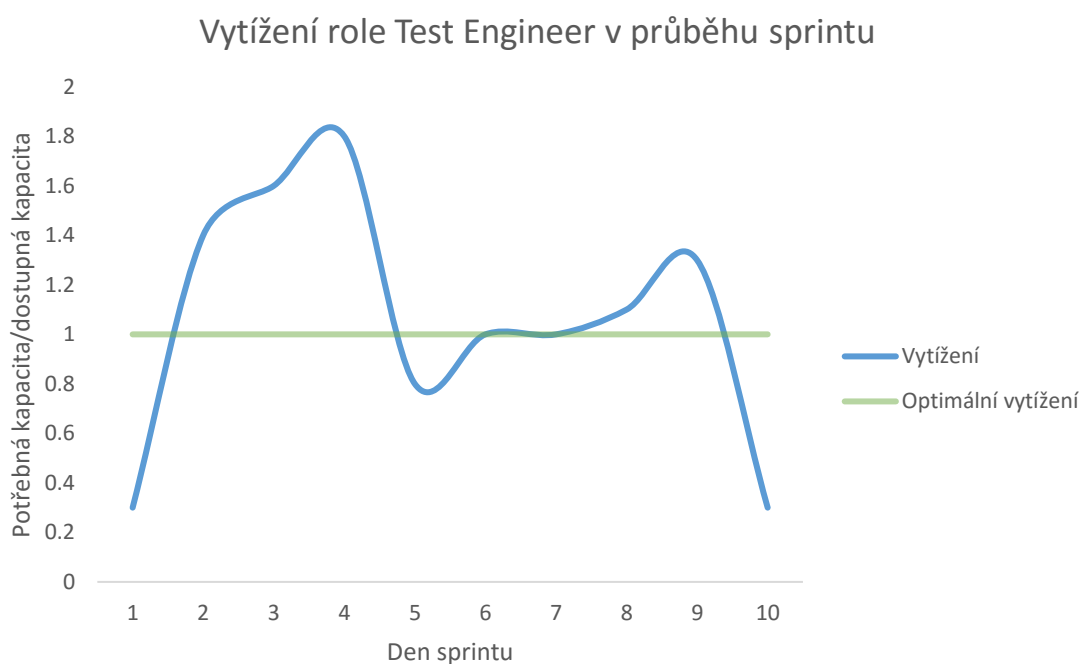
Aktuální stav rolí, který zachycuje kapitola 3.3.1 metodika Scrum v analytické části této práce, nedefinuje. Dělení rolí dle kapitoly 3.3.1 může vést ke špatné zastupitelnosti a neoptimálního využití kapacit projektového týmu.

Tabulka č. 9: Změna rolí projektového týmu (vlastní zpracování)

Role ve sledovaném týmu	Role metodiky Scrum	Výsledné sjednocení	Blížší specializace
Product Owner	Product Owner	Product Owner	-
Scrum Master	Scrum Master	Scrum Master	-
Developer	Team Member	Team Member	Development
Senior Developer	-		System Design
Principal Developer	-		Architect
Test Engineer	-		Testing, QA

Například z kapitoly 3.3.4.2 lze konstatovat, že role Test Engineer je nejvíce vytížená v 1. – 4. dni sprintu z důvodu testování STG a produkčního prostředí. Poslední den ve sprintu je pak kapacita role Test Engineer využita minimálně, a to z toho důvodu, že nové

funkcionality ještě nejsou dostupné pro testování a Test Engineer vzhledem ke své specializaci není schopen odbavit žádnou z vývojových user stories.

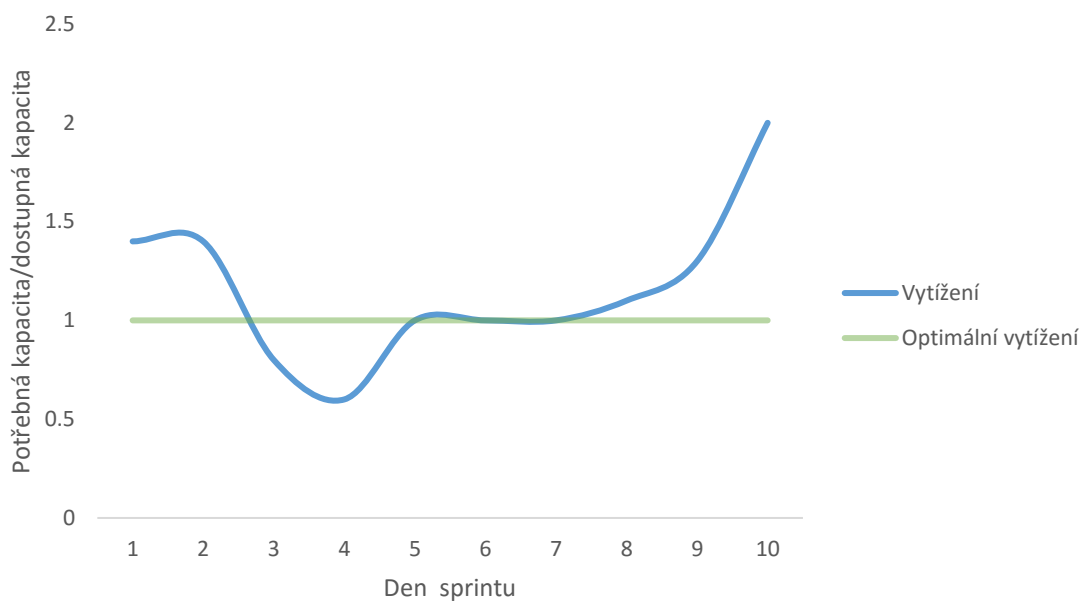


Graf č. 2: Vytížení role Test Engineer v průběhu sprintu (vlastní zpracování)

V důsledku přetížení role Test Engineer může docházet doručení méně kvalitního software (vyšší chybovost).

Podobný jev je možné pozorovat také u vývojářů v projektovém týmu, pro které je stěžejní vyvinout novou funkcionalitu před integrací. V důsledku nutnosti vývoje a navíc také počáteční fáze integrace. V 10. dni sprintu tak dochází k největšímu vytížení, které je možné pozorovat v Graf č. 3: Vytížení role Developer v průběhu sprintu.

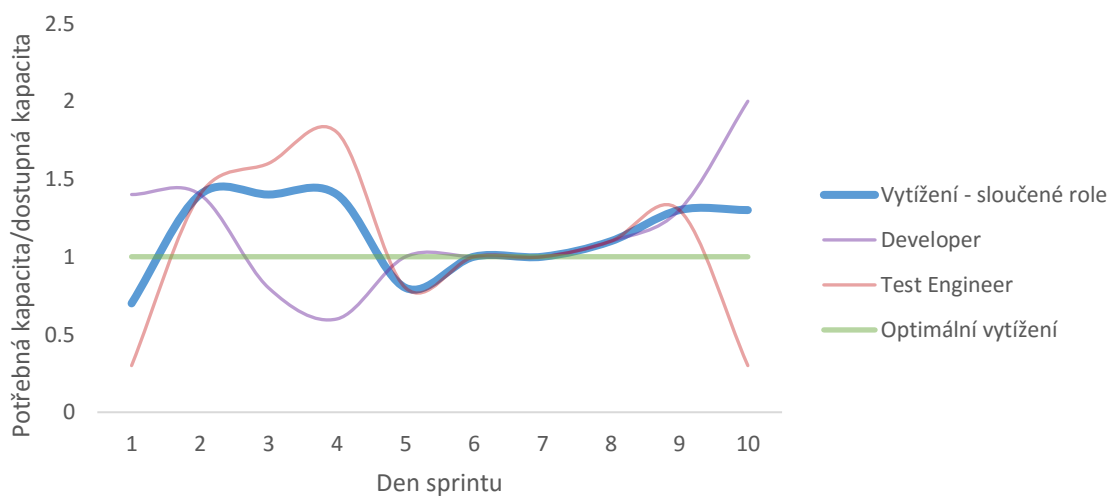
Využití role Developer v průběhu sprintu



Graf č. 3: Vytížení role Developer v průběhu sprintu (vlastní zpracování)

V důsledku přetížení vývojářů může docházet doručení méně kvalitního software (vyšší chybovost) a také k ohrožení samotné integrace.

Možné vytížení role Team Member v průběhu sprintu



Graf č. 4: Možné vytížení role Team Member v průběhu sprintu (vlastní zpracování)

Možným řešením je využití nadbytečné kapacity rolí ve smyslu zvýšení vzájemné zastupitelnosti. V případě, že role Developer je využita pouze na 50 %, poskytne zbytek svojí kapacity a zastoupí tak roli Test Engineer. Předpokladem pro toto řešení je sestavení projektového týmu, jehož členové jsou komplexněji zaměřeni, resp. jejich definice odpovídá kapitolám 2.3.1.4. a 2.3.2.

4.2 Sprint

Sledovaný tým přistupuje ke sprintům (co se týče délky a obsahu jednotlivých iterací) ze základního hlediska poměrně standardně. Při pohledu na problematiku v dlouhodobém horizontu bylo zjištěno, že projektový tým žádným způsobem nepřizpůsobuje délku sprintu charakteristice projektu a zákazníka. To může způsobit problémy v komunikaci se zákazníkem, zpoždění projektu či dokonce selhání projektového týmu v rámci metodiky Scrum.

4.2.1 Délka sprintu

Klíčové je zvolit délku sprintu tak, aby nevytvářela tlak na zákazníka. Scrum tým by v případě stresování zákazníka nemusel dostat dostatečně kvalitní zpětnou vazbu (problém alokace času zákazníka). Prostředí zákazníka je v praxi často opomíjené, proto je žádoucí, aby vývojový tým o Scrum metodice zákazníka dostatečně poučil, na rozdíl od doporučování konkrétní délky sprintu. Délka sprintu by dle některých odborníků neměla přesahovat 6 týdnů, nejdůležitější je však synergie zákazníka, vývojového týmu a charakteru projektu. (93 myslín, 84 – 84, Scrum field guide)

4.2.1.1 Stanovení délky sprintu

Pro stanovení délky sprintu můžeme využít například skórovací metodu, kterou popsal Mitch Lacey, v publikaci *The Scrum Field Guide: Practical Advice for Your First Year*.

Skórovací metoda spočívá v zodpovězení řady otázek, které pokládá Tabulka č. 10: Skórovací metoda k určení délky sprintu, otázky.

Tabulka č. 10: Skórovací metoda k určení délky sprintu, otázky (7, str. 84)

1. DOBA TRVÁNÍ PROJEKTU	1	Jaká je doba trvání projektu?	A. 0 – 3 měsíce B. 3 – 6 měsíců C. 6 – 9 měsíců D. 9 – 12 měsíců E. 12 a více měsíců
	2	Jaká je dostupnost zákazníka/stakeholder(s) pro zpětnou vazbu a vedení projektu?	A. vysoká dostupnost B. nízká dostupnost C. není dostupný
2, 3, 4, 5 ZÁKAZNÍK	3	Jak je zákazník/stakeholder(s) obeznámen s metodikou Scrum?	A. dobře obeznámen B. průměrně obeznámen C. není obeznámen
	4	Hodnocení kulturních bariér zákazníka a týmu.	A. žádné bariéry B. průměrné bariéry C. mnoho bariér
	5	Zhodnocení množství právních předpisů.	A. žádné omezující předpisy B. průměrné množství předpisů C. mnoho předpisů
6, 7, 8 TÝM	6	Posouzení zkušeností týmu s metodikou Scrum.	A. Několik let zkušeností, více jak 10 projektů. B. Průměrné zkušenosti 4 – 9 projektů. C. Žádné nebo velmi málo zkušeností.
	7	Technické dovednosti týmu.	A. Excelentní B. Průměrné C. Špatné
	8	Jak dobře je tým schopen dekomponovat úkoly?	A. Excelentně B. Průměrně C. Špatně

Po zodpovězení výše uvedených otázek přichází na řadu jejich vyhodnocení. K vyhodnocení slouží Tabulka č. 11: Skórovací metoda k určení délky sprintu, klíč. S níže uvedeným klíčem lze vyhodnotit délku sprintu následujícím způsobem:

- provedeme spárování čísla otázky a volby,
- v příslušném řádku nalezneme různé délky sprintu (1 – 4 týdny),
 - hodnota 1 představuje **chybnou délku sprintu**,
 - hodnota 3 představuje **možnou délku sprintu**,
 - hodnota 5 představuje **ideální délku sprintu**,
- klasifikaci je nutno provést vždy pro všechny varianty délky sprintu,
- výsledná délka sprintu je ve výsledku určena aritmetickým průměrem výstupních hodnot, pro danou délku sprintu.

Jako každý klasifikační nástroj je nutno i k této skórovací metodě přistupovat s jistou mírou vágnosti. V případě, že si zákazník výslovně přeje konkrétní délku sprintu a projektový tým se tomuto požadavku dokáže přizpůsobit, není nutné tuto metodu využívat (případně ji lze využít pro validaci), (7, str. 84-85).

Tabulka č. 11: Skórovací metoda k určení délky sprintu, klíč (7, str. 85)

	otázka č.	volba	Délka sprintu			
			1 týden	2 týdny	3 týdny	4 týdny
1. DOBA TRVÁNÍ PROJEKTU	1	A	5	5	1	1
		B	5	5	5	1
		C	3	5	5	3
		D	3	5	5	3
		E	1	5	5	3
2, 3, 4, 5 ZÁKAZNÍK	2	A	5	5	3	1
		B	3	5	5	5
		C	1	1	1	1
	3	A	5	5	5	5
		B	1	5	5	5
		C	1	1	3	3
	4	A	5	5	5	5
		B	1	5	5	5
		C	1	1	1	1
	5	A	5	3	3	1
		B	3	5	5	3
		C	1	3	3	5
6,7,8 TÝM	6	A	5	5	3	1
		B	3	5	5	3
		C	1	3	1	5
	7	A	5	5	3	1
		B	3	3	3	3
		C	1	3	5	5
	8	A	5	5	1	1
		B	3	5	5	3
		C	1	1	3	5

Příklad výsledné délky sprintu na základě využití skórovací metody s využitím Tabulka č. 10: Skórovací metoda k určení délky sprintu, otázky a Tabulka č. 11: Skórovací metoda k určení délky sprintu, klíč představuje Tabulka č. 12: Výstup skórovací metody pro určení délky sprintu.

Tabulka č. 12: Výstup skórovací metody pro určení délky sprintu (7, str. 86)

Otázka č.	1 týden	2 týdny	3 týdny	4 týdny
1	3	5	5	3
2	3	5	5	5
3	1	5	5	5
4	1	3	3	3
5	3	5	5	3
6	3	5	5	3
7	5	5	3	1
8	5	5	1	1
Průměr	3	4,75	4	3

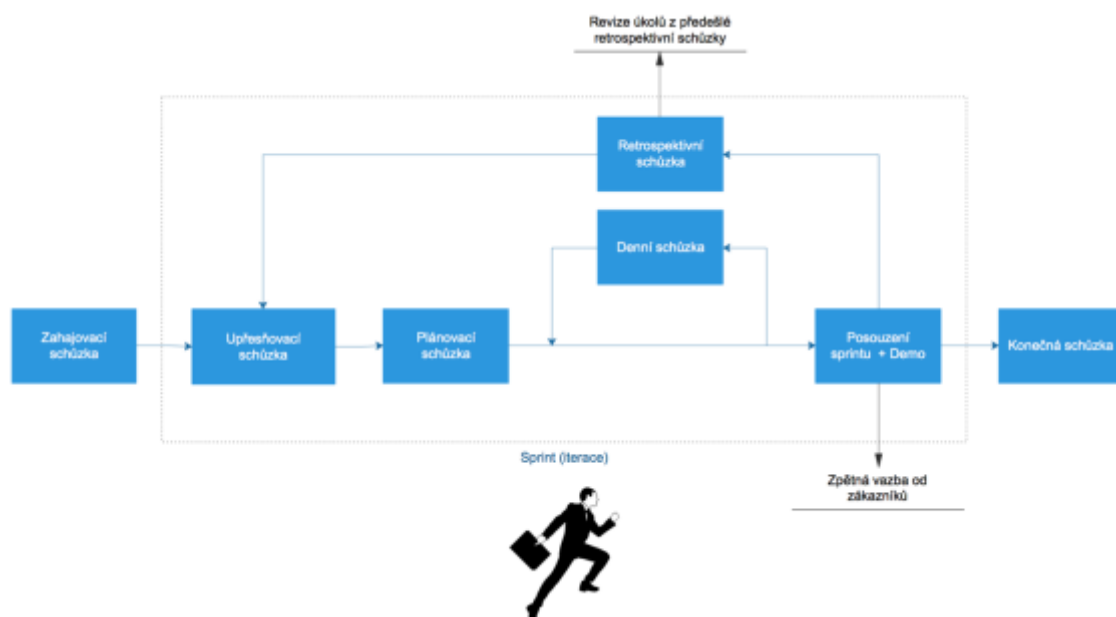
Z výše uvedeného postupu vyplývá, že nejlepší možnou délkou sprintu jsou 2 týdny. Tři týdenní iterace jsou také poměrně reálně proveditelné. Pro týdenní a čtyřtýdenní délku sprintu je proveditelnost pouze průměrná.

4.2.1.2 Korekce průběhu sprintu

Doporučením pro sledovaný tým je pravidelné uskutečňování zahajovací i konečné schůzky, přínosy těchto doporučení jsou uvedeny v kapitolách 4.3.1, resp. 4.3.4.

Dalším doporučením je pravidelně prezentovat nově vyvinutou funkcionalitu (demo) v rámci schůzky posouzení sprintu. Tento přístup je v souladu s teoretickou částí práce a má značný vliv na využití příležitostí a potlačení slabých stránek uvedených v kapitole 3.5.

Detailní popis doporučení, která jsou zobrazena na Obrázek č. 13: Korekce průběhu sprintu, obsahují příslušné kapitoly, které se věnují uvedeným schůzkám.



Obrázek č. 13: Korekce průběhu sprintu (vlastní zpracování)

4.3 Schůzky

Tato kapitola obsahuje doporučení pro projektový tým, zaměřená na optimalizaci konaných schůzek. Důraz je kladen především na zvýšení informovanosti v týmu, poskytnutí zpětné vazby, zvýšení motivace a další problematické oblasti, které jsou blíže definované v analytické části práce, resp. shrnuty ve SWOT analýze v kapitole 3.5.

4.3.1 Zahajovací schůzka (Kick-off meeting)

Vzhledem k tomu, že ocenění náročnosti rozsáhlejších user stories (epics) dochází pouze na základě ohodnocení člena týmu, který zastává roli Principal developer, může být tento údaj značně zkreslený.

V souvislosti s absencí ostatních členů také nedochází k připomínkování projektu a také diskuzi o proveditelnosti projektu.

Přirazováním projektu konkrétním vývojovým týmům může být problematické z hlediska využití různých technologií (v rámci projektu), nedostatečné velocity týmu, nevhodného složení týmu, aj.

Doporučení vyplývající z výše uvedeného, která jsou v souladu s kapitolou 2.3.5.1 jsou:

- oceňování rozsáhlejších user stories celým týmem (s tím související připomínkování a diskuze o proveditelnosti),
- sestavování projektového týmu na míru (nebo validace schopností členů týmu),
- plánování schůzek (místa i času) dle charakteru projektu,
- vyhodnocování délky sprintu pro každý projekt zvlášť.

4.3.2 Plánovací schůzka (Planning meeting)

Současný stav (kapitola 3.3.5.3) zhodnocuje aktuální způsob konání plánovací schůzky. Na jeho základě můžeme konstatovat, že v rámci této (ani žádné jiné schůzky) není explicitně definován hlavní cíl sprintu.

Definice cíle sprintu je důležitá zejména v situacích, kdy se sprint nevyvíjí dle představ zákazníka. Scrum Master je na základě dat z Burndown diagramu (kapitola 2.3.7.7 a 3.3.7.1) nucen zasáhnout a vést tým ke splnění hlavního cíle sprintu, na úkor user stories, které nejsou s hlavním cílem spjaté.

Za předpokladu kvalitního stanovení cílů lze také do jisté míry eliminovat hrozby, odstranit slabé stránky či využít příležitosti, které jsou uvedeny v kapitole 2.3.5.1:

- vyšší motivace pracovníků,
- dodržování stanovených termínů,
- vyšší kvalita výsledného software,
- lepší prosazení ve společnosti Avast.

4.3.3 Posouzení sprintu (Sprint review)

U sledovaného týmu v rámci schůzky Sprint Review nedochází k prezentaci dosažených výsledků zákazníkovi a zástupci roli Product Owner.

Z výše uvedeného vyplývá, že zákazník nemůže výsledek práce v této fázi vývoje nijak připomínkovat nebo vznášet požadavky na změny v případě nespokojenosti s dodaným řešením či neočekávané změně vnějších okolností.

K tomuto procesu v současnosti dochází až ve fázi, kdy je daná funkcionality vydána do produkčního prostředí příslušných systémů.

Doporučením pro projektový tým je předvádět výsledky práce v rámci každé iterace tak, aby bylo možné předejít incidentům, které nastanou v produkčním prostředí po zavedení nového kódu. Doporučená změna je v souladu s kapitolou 2.3.5.5.

Tým od zákazníka obdrží připomínky, požadavky na změny neboli obecně především zpětnou vazbu, díky které dokáže předejít problémům většího rozsahu v kratším časovém horizontu.

Mimo výše uvedeného má návrh změn za cíl eliminovat hrozby, odstranit slabé stránky či využít příležitosti, které jsou uvedeny v kapitole 2.3.5.1:

- odstranit špatné vnímání projektového týmu společnosti Avast (podpoření prezentacemi pro zákazníky),
- prosadit tým v rámci společnosti Avast,
- docílit vyšší kvality výsledného software,
- docílit zvyšování povědomí o projektech.

4.3.4 Konečná schůzka (Final meeting)

Doporučením pro sledovaný tým je pravidelné konání konečné schůzky, resp. formálního ukončování projektů. Agendou této schůzky je akceptace projektu zákazníkem, zhodnocení projektu, zajištění údržby systémů v dodávce a také nezbytné předání dokumentace vč. předání pro operativní tým.

Předáním dokumentace pro operativní tým lze předcházet následnému zatěžování členů sledovaného týmu v případech, kdy se se v produkčním prostředí vyskytne incident, o němž nemá operativní tým dostatečné know-how.

4.4 Měření

Tato kapitola obsahuje shrnutí návrhů pro měření v rámci projektové týmu. V souladu s hodnotami metodiky Scrum (kapitola 2.2.2.1) jsou využity adekvátní metriky, jejichž odchylky jsou akceptovány (cílem metodiky není použít maximálně přesné měření, jehož výsledky bychom v praxi nemohli využít).

Měření, vyhodnocení výstupu a následné přizpůsobení procesu je nezbytně nutné z hlediska dosažení vyšší vyspělosti procesu vývoje software (SW-CMM úrovně 4 a 5), což je zároveň cíl této práce.

Výstupy z měření, představují také přidanou hodnotu pro management společnosti. V případě nestandardního vývoje dojde k indikaci potencionálního problému, díky čemuž proběhne následná investigace a případná podpora týmu v řešení problému.

Předpokladem pro informovanost o aktuálním a historickém stavu metrik, je zobrazení metrik v rámci monitoringu v kanceláři.

Pomocí metrik uvedených v této kapitole je také vhodné z toho důvodu, aby bylo možné porovnat výsledky sledovaného týmu před a po zavedení změn, které jsou uvedeny v kapitolách 4.1, 4.2 a 4.3, díky čemuž lze zhodnotit přínos těchto změn.

Návrh na zavedení měření byl přijat již v průběhu vypracování této práce, což umožňuje vyhodnotit vliv zavedení ostatních změn i v budoucnu.

Data, která jsou uvedená v této kapitole a podkapitolách, bylo možné získat na základě akceptaci doporučených změn v rámci vypracování této práce. Data jsou získávány v souladu s kapitolou 2.3.8.

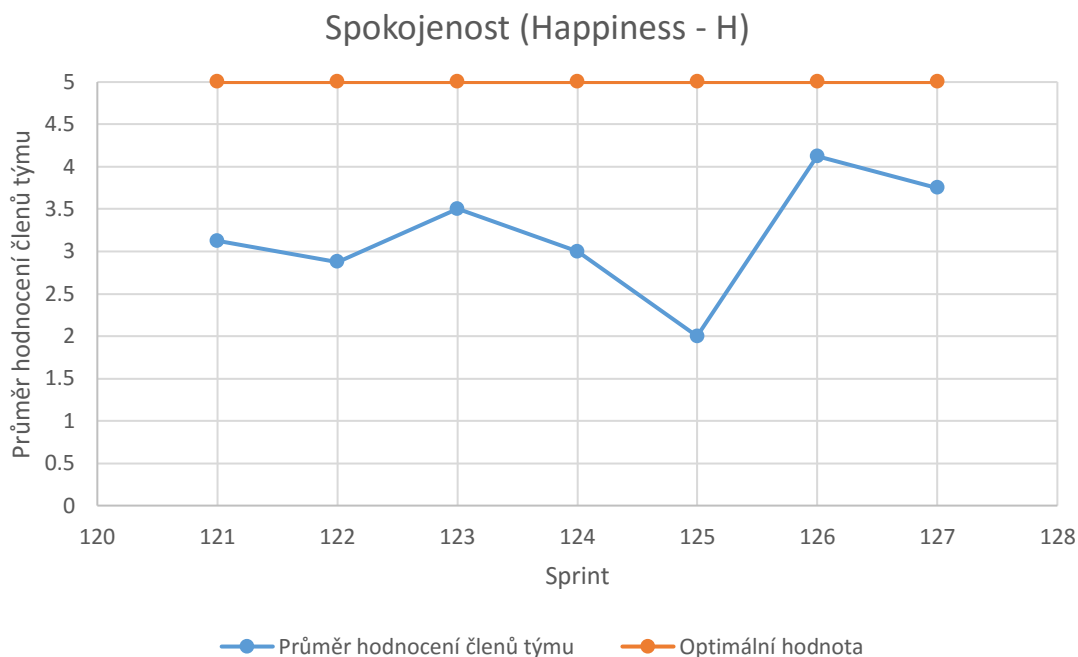
4.4.1 Spokojenost (Happiness – H)

Data nezbytná pro měření jsou získána na retrospektivní schůzce od jednotlivých členů týmu. Při hodnocení je velmi důležité (nikoli však povinné) uvést také komentář, který může být vstupem pro zlepšení slabých stránek týmu, či posílení silných stránek.

Při hodnocení si členové týmu mohou klást otázky, které uvádí kapitola 2.3.7.1, nebo otázky, které si tým určí za účelem dosažení výstižného hodnocení.

Tabulka č. 13: Spokojenost - data (vlastní zpracování)

Sprint	Průměr hodnocení členů týmu	Člen týmu 1	Člen týmu 2	Člen týmu 3	Člen týmu 4	Člen týmu 5	Člen týmu 6	Člen týmu 7	Scrum master
121	3,125	3	2	3	3	4	4	3	3
122	2,875	2	2	3	4	4	3	2	3
123	3,5	4	3	4	4	3	3	3	4
124	3	3	3	3	2	2	4	3	4
125	2	1	2	1	3	3	3	1	2
126	4,125	4	4	4	5	4	4	3	5
127	3,75	3	4	4	4	3	4	4	4



Graf č. 5: Spokojenost (Happiness - H), (vlastní zpracování)

Výstup z retrospektivní schůzky Sprint 125 (důsledek nízké spokojenosti):

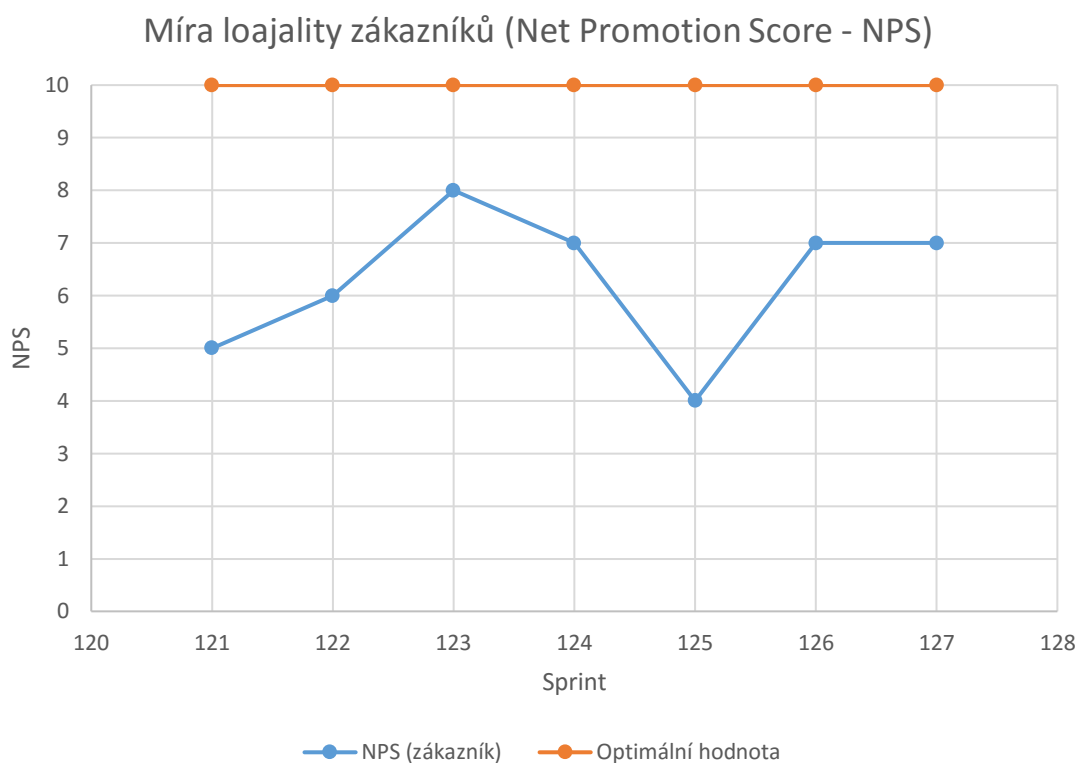
- nedostatečná rychlost STG prostředí pro automatické testování,
- pozdní upřesnění zadání ze strany zákazníka,
- problematická aktualizace Wildfly v rámci klíčového technického vydání.

4.4.2 Míra loajality zákazníků (Net Promotion Score – NPS)

Data o spokojenosti zákazníků jsou získávána po uvedení dané funkcionality do produkčního prostředí, nebo v rámci schůzky, kde je funkcionality prezentována (Demo meeting). Spokojenost zákazníků, resp. s jakou pravděpodobností by zákazník doporučil sledovaný projektový tým ke spolupráci, má podobný trend jako vlastní spokojenost týmu.

Tabulka č. 14: NPS (vlastní zpracování)

Sprint	NPS
121	5
122	6
123	8
124	7
125	4
126	7
127	7



Graf č. 6: Míra loajality zákazníků (Net Promotion Score - NPS), (vlastní zpracování)

Problematické oblasti, vytyčené v rámci retrospektivní schůzky (sprint 125), jsou uvedeny v kapitole 4.4.1.

4.4.3 Rychlost (Velocity – V)

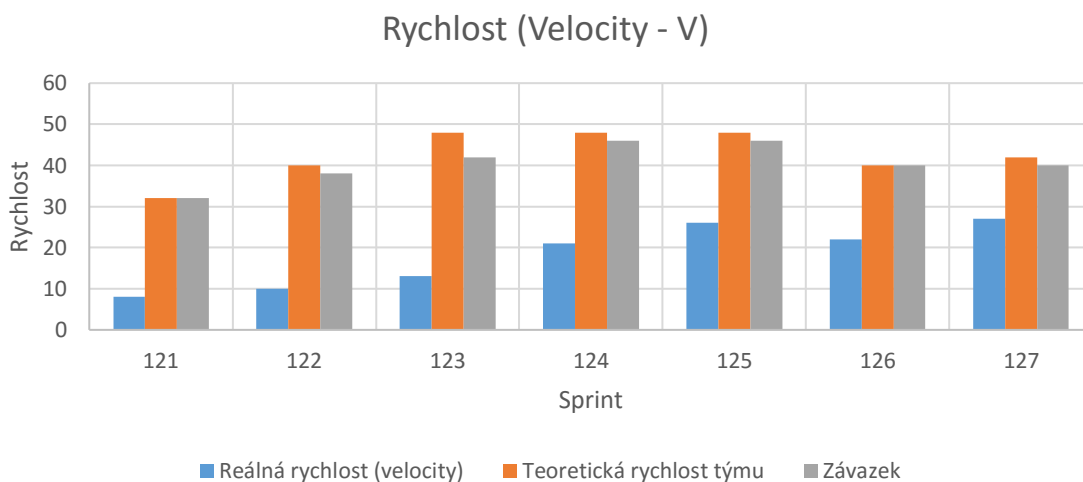
Na základě uvedených informací v kapitolách 3.3.1 a 3.3.4.2 vyplývá, že teoretická rychlost sledovaného týmu lze stanovit na základě délky sprintu (10 pracovních dní) počtu zastupitelů rolí, které se přímo podílejí na vývoji (celkem 6).

Zároveň je však nutné v teoretické rychlosti zohlednit také vlivy, které tuto hodnotu snižují (například schůzky a Scrum ceremonie). Sledovaný tým dlouhodobě pracuje s přístupem, který předpokládá 8 pracovních dní, namísto 10 po odečtení času, který členové týmu nemohou věnovat vývoji.

Teoretická rychlost týmu tak představuje hodnotu přibližně 48 MDs. Závazek představuje součet hodnot původních odhadů user stories, které byly v rámci plánovací schůzky zařazeny do sprint backlogu.

Tabulka č. 15: Rychlost (Velocity - V), (vlastní zpracování)

Sprint	Reálná rychlost (velocity) [MDs]	Teoretická rychlost týmu [MDs]	Závazek
121	8	32	32
122	10	40	38
123	13	48	42
124	21	48	46
125	26	48	46
126	22	40	40
127	27	42	40



Graf č. 7: Rychlost (Velocity - V), (vlastní zpracování)

4.4.4 Pracovní kapacita (Work Capacity – WC)

Na základě zaznamenaných dat lze konstatovat, že poměr pracovní a teoretické kapacity je vyšší, než poměr reálné a teoretické rychlosti. Členové týmu tedy reportují více

provedené práce, než je následně vydáno do produkčního prostředí ve smyslu práce provedené na příslušných funkcionalitách.

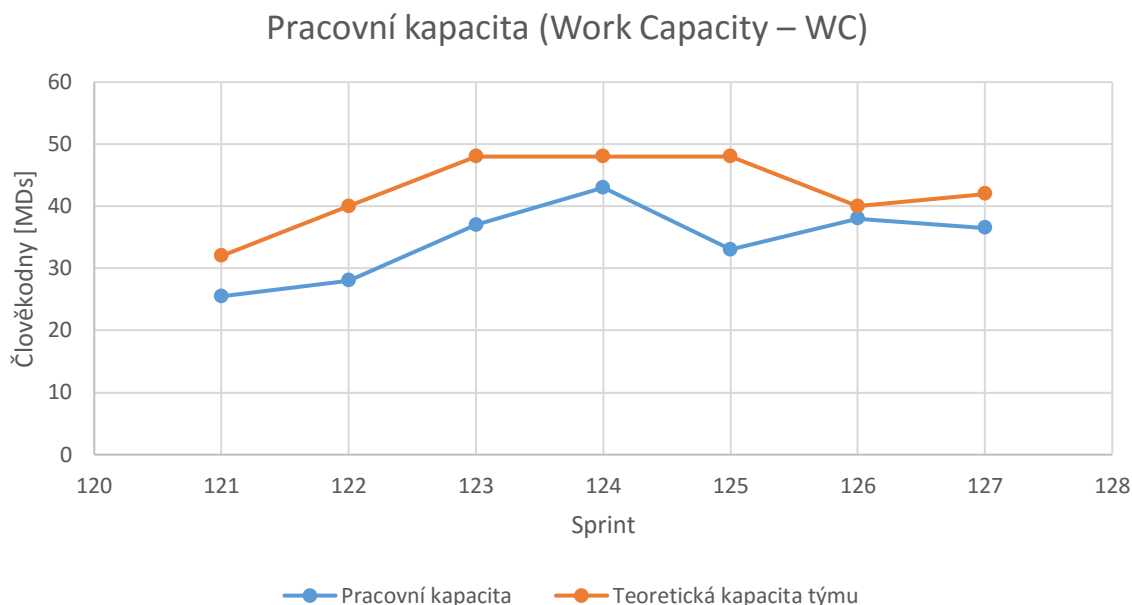
Možné příčiny:

- náročnost user stories je nesprávně odhadnuta,
- sprint backlog obsahuje některé user stories, které výrazně převyšují náročnost ostatních.

K zjištění příčiny byla použita v průběhu vypracování této práce další metrika, která umožňuje sledovat přesnost odhadu náročnosti user stories, tuto metriku popisuje kapitola 2.3.7.6, výsledky popisuje kapitola 4.4.6.

Tabulka č. 16: Pracovní kapacita (Work Capacity - WC), (vlastní zpracování)

Sprint	Pracovní kapacita	Teoretická kapacita týmu	Pracovní kapacity / Teoretická kapacita
121	25,5	32	80%
122	28	40	70%
123	37	48	77%
124	43	48	90%
125	33	48	69%
126	38	40	95%
127	36,5	42	87%



Graf č. 8: Pracovní kapacita (Work Capacity - WC), (vlastní zpracování)

4.4.5 Faktor zaměření (Focus Factor – FF)

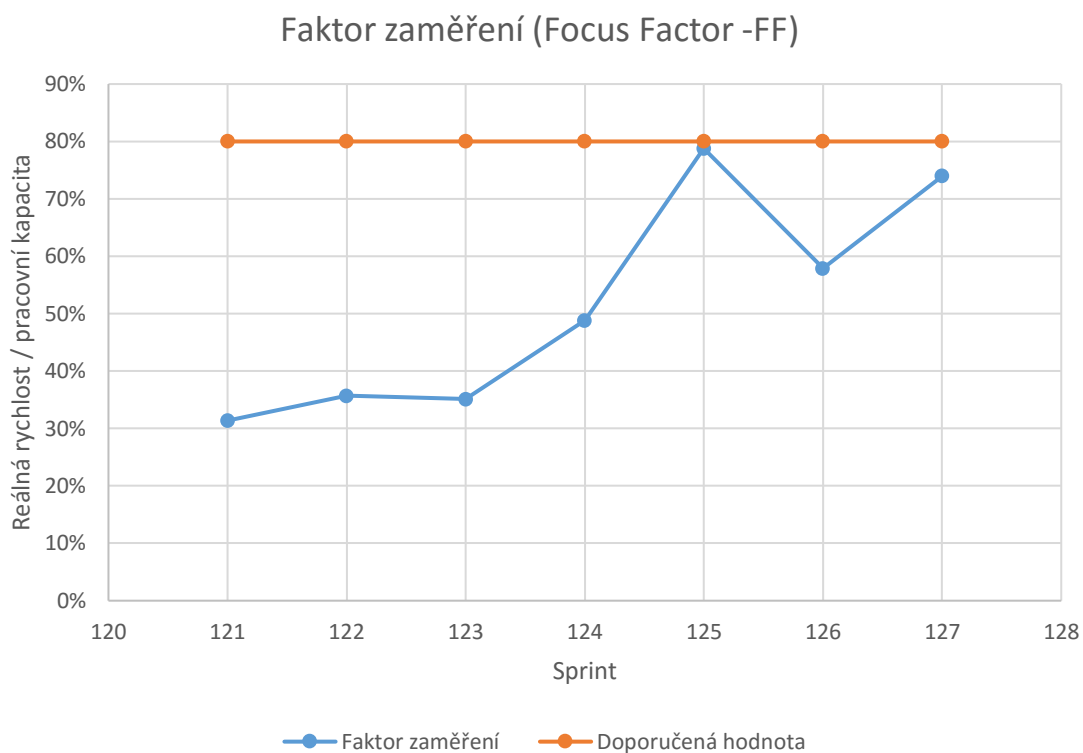
V níže uvedeném grafu lze pozorovat, že tým v některých bodech dosáhl doporučené hodnoty. V případě dlouhodobého udržování faktoru soustředění co nejblíže doporučené hodnotě, můžeme hovořit (dle teoretické části) o zdravém Scrum týmu.

Hodnoty nižší než 50 % představují situaci, ve které Scrum tým není schopen odbavovat požadavky zadavatelů na základě působení negativních vlivů. Tento ukazatel slouží zastupiteli roli Scrum Master k tomu, aby provedl investigaci a následně sjednal nápravu.

V případě absence člena týmu v následující iteraci můžeme předpokládat dodané množství nových funkcionalit na základě kapitoly 2.3.7.5.

Tabulka č. 17: Faktor zaměření (Focus Factor – FF), (vlastní zpracování)

Sprint	Reálná rychlost (velocity)	Pracovní kapacita	Faktor zaměření
121	8	25,5	31%
122	10	28	36%
123	13	37	35%
124	21	43	49%
125	26	33	79%
126	22	38	58%
127	27	36,5	74%



Graf č. 9: Faktor zaměření (Focus Factor – FF), (vlastní zpracování)

4.4.6 Přesnost odhadu náročnosti (Estimation Accuracy - EA)

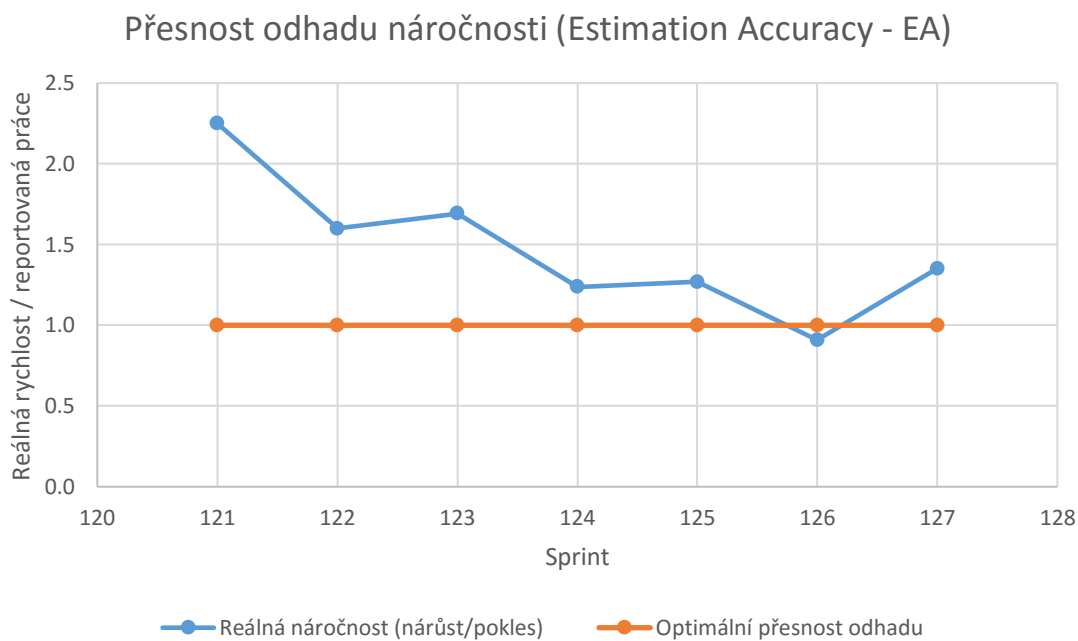
Na základě kapitoly 4.4.4 byla provedena investigace, která porovnává reálnou náročnost s původním odhadem všech dokončených user stories (reálná rychlost).

S pomocí této metriky bylo zjištěno, že tým se v některých případech odchyluje v odhadu náročnosti práce o více, než 100 %.

Doporučení pro vývojový tým je (kromě vyhodnocení příčin) zanechat tuto metriku ke sledování stavu budoucího stavu. Jelikož se jedná o slabinu vývojového týmu, je nutné ji monitorovat.

Tabulka č. 18: Přesnost odhadu náročnosti (Estimation Accuracy - EA), (vlastní zpracování)

Sprint	Reálná rychlost (velocity)	Reportovaná práce	Reálná náročnost (nárůst/pokles)
121	8	18	2,3
122	10	16	1,6
123	13	22	1,7
124	21	26	1,2
125	26	33	1,3
126	22	20	0,9
127	27	36,5	1,4



Graf č. 10: Přesnost odhadu náročnosti (Estimation Accuracy - EA), (vlastní zpracování)

4.5 Identifikace hrozeb a scénářů

Při implementaci změn, které jsou navrženy v této práci, mohou nastat některá z rizik uvedených v této kapitole. Úvodní popis rizik doplňuje zjednodušený popis zpracovaný ve formě tabulky Tabulka č. 19: Identifikace hrozeb a scénářů, která představuje nutný předpoklad pro navazující kapitoly.

4.5.1 Změny nebudou přijaty

Jedním ze základních předpokladů provedení změn je pochopení podstaty a přínosu sledovaným vývojovým týmem. V případě, že podstata a přínos nebudou pro tým dostatečně srozumitelné a hodnotné, může dojít k tomu, že navrhované změny nebudou ve sledovaném týmu přijaty, což zapříčiní, že sledovaný tým nezvýší úroveň vyspělosti vývoje software na 4., resp. 5. úroveň z pohledu SW-CMM.

4.5.2 Monitoring dostupný pro tým

Riziko je způsobeno snahou o maximální informovanost v rámci týmu. Tento návrh na změnu je vznesen v kapitole 4.4. V případě zachování tohoto návrhu vzniká riziko, že cílem sledovaného týmu bude jednat ve prospěch samotných metrik, nikoli však k doručování maximální hodnoty zákazníkovi dle hodnot Scrum metodiky.

4.5.3 Využití metrik ke stanovování cílů

V případě využití metrik ke stanovování cílů může dojít k jejich následnému podvržení, na základě předávání nepravdivých informací, zatajování problematických oblastí a slabin vývojového týmu za tím účelem, aby bylo dosaženo co nejlepšího, resp. cílového hodnocení, dle stanovených metrik.

4.5.4 Vysoká nepřesnost měření

Při měření a následném vyhodnocování je důležitá maximální možná přesnost, která je ovšem v souladu s kapitolou 2.3.7 a hodnotami samotné Scrum metodiky. V případě významných odchylek od skutečnosti budou následné výstupy definovaných metrik irelevantní.

4.5.5 Selhání vývojového týmu

V případě, že vývojový tým nebude kontinuálně dodávat nové funkcionality a bude nutné zrušení jednoho či více sprintů, dojde k jeho selhání. V případě selhání vývojového týmu tak dojde k nutnosti sestavit nový vývojový tým.

4.5.6 Chybná interpretace metrik

Ve snaze zvýšit úroveň úspěšnosti vývoje software dochází v případě odchylek od požadovaných hodnot ke snaze o optimalizaci, resp. zajištění opravného mechanismu za účelem dosažení lepších výsledků v budoucnu. V případě chybné interpretaci uvedených metrik však nemusí dojít ke zlepšení výsledků (v důsledku neadekvátně řešených problémů, které nastaly).

Tabulka č. 19: Identifikace hrozeb a scénářů, (vlastní zpracování)

ID	Hrozba	Scénář
1	Změny nebudou přijaty	Sledovaný tým nezvýší úroveň úspěšnosti vývoje software
2	Monitoring dostupný pro tým	Maximalizace jednání ve prospěch metrik
3	Využití metrik ke stanovování cílů	Podvržení měření
4	Vysoká nepřesnost měření	Irelevantní výstupy metrik
5	Selhání vývojového týmu	Nutnost sestavení nového týmu
6	Chybná interpretace metrik	Neadekvátní řešení problémů
7	Odmítání změn (demo) zákazníkem	Přínos týmu v rámci společnosti Avast nebude reflektován

4.6 Kvantifikace rizik

V rámci kvantifikaci rizik je kalkulováno s celkovými náklady na člověkodenní ve společnosti Avast software s.r.o., které činí 2800,- Kč. Celkový dopad představuje náklady na vypracování příslušné části práce včetně času, který je nutný pro implementaci a nákladů na nápravu škod. Kvantifikace rizika (stejně tak opatření) byla provedena na základě expertního odhadu (brainstorming) podpořenou o historická data z jiných projektů. Výše odhadovaných dopadů mohou být odchýleny od reálných dopadů.

Tabulka č. 20: Kvantifikace rizik (vlastní zpracování)

ID*	P-st	Dopad [Kč]	Hodnota rizika [Kč]	Ohodnocení rizika
1	0,3	60 000	18 000	5
2	0,8	30 000	24 000	12
3	0,6	20 000	12 000	6
4	0,2	30 000	6 000	3
5	0,05	230 000	11 500	0
6	0,3	60 000	18 000	10
7	0,3	9 000	2 700	2

* ID odpovídá hrozby v Tabulka č. 19: Identifikace hrozeb a scénářů, (vlastní zpracování)

Výpočet: $\text{Hodnota rizika} = \text{pravděpodobnost} / (\text{sazba za člověkodenní} * \text{časová náročnost})$

Tabulka č. 21: Ohodnocení rizik (vlastní zpracování)

Hodnota rizika	Pravděpodobnost	Dopad [Kč]
<0,1>	0 – 20%	< 10 000
(1,2>	20 – 40%	10 000 – 20 000
(2,3>	40 – 60%	20 001 – 30 000
(3,4>	60 – 80%	30 001 – 40 000
(4,5>	80 – 100%	> 40 000

Závažnost rizika je dána toleranční mezí, která je akceptovatelná. Tuto toleranční mez představuje Tabulka č. 22: Závažnost rizika (vlastní zpracování).

Tabulka č. 22: Závažnost rizika (vlastní zpracování)

Hodnota rizika	Závažnost rizika
< 5	Běžné
5 – 10	Důležité
> 10	Kritické

4.7 Snižování rizik

Pro snížení ohodnocených rizik jsou využita níže uvedená opatření. Každému opatření náleží slovní komentář, který pojednává o realizovatelnosti a účinnosti. Náklady na opatření a zodpovědná osoba zachycuje Tabulka č. 23: Snižování rizika (vlastní zpracování).

Snižování rizik je realizováno pouze pro důležitá a kritická rizika dle klasifikační stupnice v Tabulka č. 22: Závažnost rizika (vlastní zpracování).

Scrum & SW-CMM školení

- Předpokladem pro přijetí změn sledovaným týmem je edukační blok, ve kterém budou prezentovány relativně vysoké přínosy, za cenu malých změn.
- Účinnost: Po realizaci opatření je předpokládané snížení pravděpodobnosti na 0,2 a dopadu na 30 000,- Kč

Zamezení monitoringu

- Monitoring, dostupný v rámci společných přehledů pro tým na sdílené obrazovce v kanceláři, nebude obsahovat informace o uvedených měřeních v rámci této práce. Tyto metriky budou dostupné pouze pro zástupce role Scrum Master.
- Účinnost: V případě, že dříve navrhovaná změna nebude implementovaná, nebudou na opatření vynaloženy žádné dodatečné náklady (0,- Kč) s maximálním možným výsledkem (pravděpodobnost 0 %).

Zamezení stanovení cílů dle metrik

- Metriky nebudou využívány při vytváření cílů sprintů. Vytváření cílů sprintu bude na uvedených metrikách zcela nezávislé.
- Účinnost: V případě striktního dodržení pravidla vycházejícího z opatření bude dopad rizika snížen přibližně o 0,33 za předpokladu nulových nákladů.

Brainstorming opravných mechanismů

- Zavádění opravných mechanismů, jejichž podnětem jsou metriky uvedené v této práci, bude vždy konzultováno s managementem a zástupci rolí Scrum Master z ostatních týmů.

- Účinnost: Pravděpodobnost bude snížena na přibližně 0,1 s 60 000,- Kč dopadem. Náklady na opatření představuje čas, který je věnován schůzkám řešícím výše uvedenou problematiku.

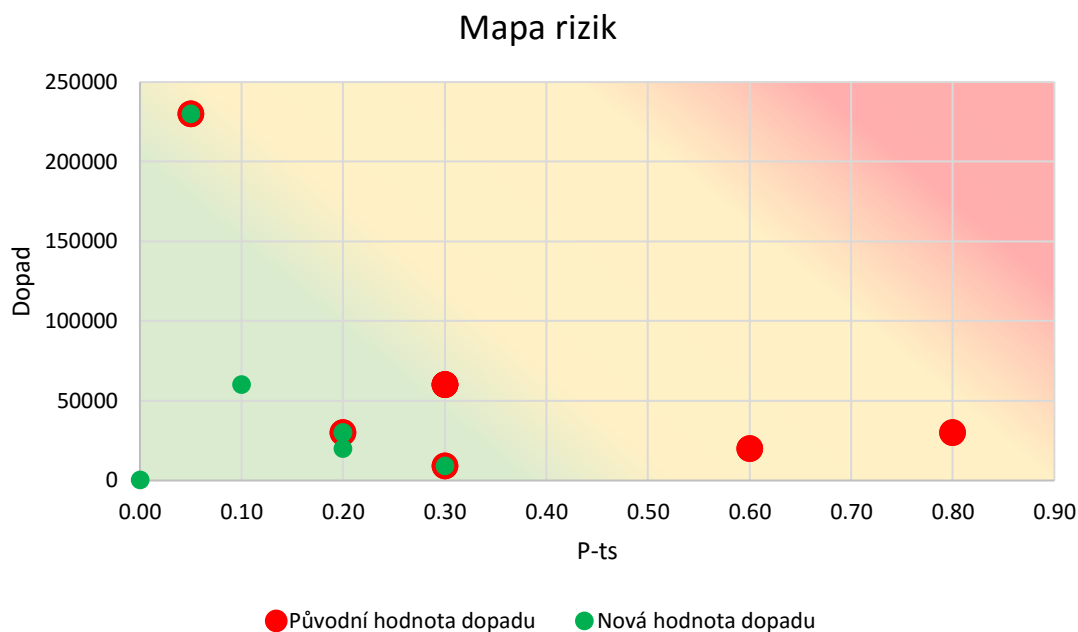
Tabulka č. 23: Snižování rizika (vlastní zpracování)

ID hrozby	Návrh na opatření	Nová P-st	Nový dopad	Nová hodnota rizika	Náklady na opatření	Zodpovědnost pro zajištění
1	Scrum & SW-CMM školení	0,2	30 000	6 000	2 800	Scrum Master
2	Zamezení monitoringu	0	0	0	0	Scrum Master
3	Zamezení stanovení cílů dle metrik	0,2	20 000	4 000	0	Product Owner
4	<i>Riziko přijmáme</i>	-	-	-	-	-
5	<i>Riziko přijmáme</i>	-	-	-	-	Scrum Master
6	Brainstorming opravných mechanismů	0,1	60 000	6 000	2 800	Scrum Master
7	<i>Riziko přijmáme</i>	-	-	-	-	-

* ID hrozby označení hrozeb v Tabulka č. 19: Identifikace hrozeb a scénářů, (vlastní zpracování)

4.7.1 Mapa rizik

Níže uvedená mapa rizik zachycuje původní hodnoty dopadu a nové hodnoty dopadu po aplikaci opatření, která jsou uvedena v kapitolách 4.5, 4.6 a 4.7. Na základě této vizualizace je možné jednoduše komparovat výsledky výše uvedených opatření.



Graf č. 11: Mapa rizik, (vlastní zpracování)

4.8 Ekonomické zhodnocení

Ekonomické zhodnocení této práce je založeno na nákladech, které vyplívají z implementace uvedených doporučení, resp. na nákladech na opatření. Vzhledem k tomu, že výnosy plynoucí z přínosu implementace, jsou dlouhodobého charakteru (nejedná se o jednorázové změny v projektu, ale ve všech budoucích projektech), je přínos kalkulován na základě nutné změny hospodářský výsledek (dále jen HV) vzhledem k současnému stavu, resp. definování budoucího očekávaného stavu.

Hodnoty odhadů jsou založeny na odhadu sledovaného týmu společně s náklady, které bylo možné vyčíslit (s pomocí finančního oddělení společnosti).

Tabulka č. 24: HV sledovaného týmu, (vlastní zpracování)

Hospodářské období	2012	2013	2014	2015	2016
HV sledovaného týmu z min. let (tis. Kč)	6 297	6 724	7 475	7 072	7 378
Růst/pokles HV	-	7%	11%	-5%	4%

Průměrné tempo růstu HV činí **4 %**, z čehož vyplývá očekávaný HV **7 673** (tis. Kč) pro rok 2017.

Meziroční přírůstek očekávaného HV pro rok 2017, který je zvýšen o náklady spojené s implementací doporučení v této práci zachycuje Tabulka č. 25: Předpoklad budoucího období. Celkové tempo růstu, které předpokládá úspěšné zavedení změn je nejméně **5,26 %** (kompenzace dodatečných nákladů). Rozdíl 1,26 % je poměrně nevýznamný. Tento rozdíl může ovlivnit také řada dalších faktorů. Vzhledem k relativně nízkým nákladům, které jsou na projekt vynaloženy, nejsou očekávány negativní dopady ve větším měřítku.

Tabulka č. 25: Předpoklad budoucího období, (vlastní zpracování)

Očekávaný HV 2017 (tis. Kč)	7 673
Náklady na zavedení změn – jednorázové (Kč)	60000
Náklady na opatření – jednorázové (Kč)	2800
Náklady na opatření – roční (Kč)	33600
Hranice úspěšnosti přijatých změn (HV), (tis. Kč)	7 770
Meziroční přírůstek	1,26%
Meziroční přírůstek + průměrné tempo růstu	5,26%

Ve stavu před implementací náklady na člověkodenní výrazně převyšují náklady firmy na člověkodenní (Tabulka č. 26: Náklady na člověkodenní doručené práce - současný stav) ve sledovaném týmu, resp. sledovaném oddělení.

Pro vyšší robustnost předpokladu nebyla v kalkulaci využita průměrná rychlost sledovaného týmu, nýbrž doposud nejvyšší.

Tabulka č. 26: Náklady na člověkodenní doručené práce - současný stav, (vlastní zpracování)

Současný stav	
Náklady na člověkodenní (Kč)	2800
Roční náklady na sledovaný tým (tis. Kč)	5 250
Nejvyšší dosažená rychlost týmem (velocity)	27
Množství doručené práce (člověkodny)	675
Náklady na člověkodenní doručené práce (Kč)	7 778

V případě zavedení změn, doporučených v této práci může nastat více scénářů, které vyhodnocují níže uvedené tabulky.

Pesimistickým scénářem je zvýšení rychlosti z 27 MDs na 28 MDs. V takovém případě dosáhne sledovaný tým roční úspory nákladu na člověkodenní doručené práce pouze 69 444,- Kč. Náklady na zavedení změn a opatření nicméně činí 96 400,- Kč, z čehož vyplývá, že pesimistický scénář není z pohledu finanční úspěšnosti zavedení změn přípustný.

Tabulka č. 27: Náklady na člověkodenní doručené práce - pesimistická varianta, (vlastní zpracování)

Pesimistická varianta	
Náklady na člověkodenní (Kč)	2800
Roční náklady na sledovaný tým (tis. Kč)	5 250
Nejvyšší dosažená rychlost týmem (velocity)	28
Množství doručené práce (člověkodny)	700
Náklady na člověkodenní doručené práce (Kč)	7 500
Roční úspora (Kč)	69 444
Náklady na zavedení změn a opatření (Kč)	96 400
Výsledek varianty (Kč)	- 29 956

V případě optimistické varianty může sledovaný tým těžit z celkové úspory nákladů na člověkodenní doručené práce ve výši 348 044,- Kč.

Tabulka č. 28: Náklady na člověkodenní doručení práce - optimistická varianta, (vlastní zpracování)

Optimistická varianta	
Náklady na člověkodenní (Kč)	2800
Roční náklady na sledovaný tým (tis. Kč)	5 250
Nejvyšší dosažená rychlost týmem (velocity)	35
Množství doručení práce (člověkodny)	875
Náklady na člověkodenní doručení práce (Kč)	6 000
Roční úspora (Kč)	444 444
Náklady na zavedení změn a opatření (Kč)	96 400
Výsledek varianty (Kč)	348 044

Za předpokladu zvýšení velocity (rychlost) o přibližně 10 % dojde k roční úspoře na člověkodenní doručení práce ve výši 250 896,- Kč, což zapříčiní celkovou úsporu 154 496,- Kč.

Tabulka č. 29: Náklady na člověkodenní doručení práce - realistická varianta, (vlastní zpracování)

Realistická varianta (~10% nárůst rychlosti)	
Náklady na člověkodenní (Kč)	2800
Roční náklady na sledovaný tým (tis. Kč)	5 250
Nejvyšší dosažená rychlost týmem (velocity)	31
Množství doručení práce (člověkodny)	775
Náklady na člověkodenní doručení práce (Kč)	6 774
Roční úspora (Kč)	250 896
Náklady na zavedení změn a opatření (Kč)	96 400
Výsledek varianty (Kč)	154 496

Prahová hodnota rychlosti (velocity), kdy roční úspora převyšuje náklady na zavedení změn a opatření činí 29 MDs. Jedná se tedy o 7,4% (2,6%, resp. 26% rezerva realistické varianty) nárůst od maximální hodnoty, která byla dosažena v rámci vypracování této práce.

4.9 Přínosy návrhů

Přínosy vychází z analytické, resp. z návrhové části práce. V rámci analytické části bylo vyhodnoceno velké množství, slabých stránek, příležitostí a hrozeb, jejichž shrnutí představuje kapitola 3.5. Navrhované změny mají potlačit slabé stránky, eliminovat a hrozby a využít příležitosti.

Přínosy navrhovaných korekcí implementace metodiky Scrum, zavedení metriky a využívání filosofie SW-CMM:

- vyšší odpovědnost vývojářů (využití RACI matice a EPC diagramu vývojového procesu),
- vyšší sdílení know-how mezi týmu a také směrem k zákazníkovi (zavedení pravidelných Demo prezentací v rámci revize sprintu),
- vyšší motivace sledovaného týmu (odstranění absence zpětné vazby styku se zákazníkem na konečné schůzce a Demo prezentace v rámci sprint review),
- lepší vnímání přínosu sledovaného týmu v rámci společnosti Avast (Demo prezentace),
- přizpůsobení délky sprintu na míru konkrétnímu zákazníkovi, členům týmu a projektu (konkrétní metoda dle kapitoly 4.2.1),
- definování cíle sprintu (změna agendy příslušné schůzky),
- vyšší zastupitelnost členů v projektovém týmu (zjednodušení přístupu k rolím, nižší zátěž na jednotlivé role v průběhu sprintu),
- indikace problémových oblastí sledovaného týmu s možností korekcí a vyhodnocování přínosu navržených změn (zavedení metrik),
- neustále zlepšování procesu vývoje software s využitím SW-CMM.

Z výše uvedených (dílčích) přínosu vyplývá, že hlavním přínosem této práce je zvýšení efektivity sledovaného týmu a zajištění, že tato efektivita nebude snížena v souladu s filosofií 5. úrovně modelu SW-CMM, což povede k nižším nákladům na člověkoděn (z hlediska úspěšně realizovaného zadání). Zvýšení efektivity vede ke spokojenosti zákazníka (na základě vyšší spolehlivosti týmu), projektového týmu (nižší tlak ze strany zákazníků a nadřízených), managementu (nižší pravděpodobnost krizových situací) i konečným zákazníkům využívající produkt společnosti Avast software s.r.o (spolehlivost).

ZÁVĚR

Součástí hlavního cíle diplomové práce, zhodnocení využití Scrum metodiky a návrh změn za účelem jejího efektivnějšího využití ve vývojovém týmu, byly také podpůrné dílčí cíle, díky kterým bylo možné hlavní cíl realizovat.

Teoretická část práce komplexně pojednává o problematice agilní metodiky Scrum. Pro ucelenější kontext se tato část práce okrajově zaměřuje na jiné agilní metodiky i rigorózní metodiky. V této části lze nalézt teoretická východiska, která se zaměřují na základní hodnoty agilních metodik, zaměření na metodiku Scrum s artefakty, rolemi a schůzkami, které pod tuto metodiku spadají. Teorie pojednává také o modelu vyspělosti procesu vývoje software (SW-CMM), se kterým se častěji můžeme setkat u rigorózních metodik. Práce se tímto modelem striktně neřídí, ale inspiruje se.

Analýza současného stavu zachycuje aktuální stav ve sledované společnosti. Analýza 7S pojednává o hodnocení kritických faktorů, z hlediska sledovaného vývojového týmu i celé společnosti. Tato analýza je spolu se zhodnocením využití metodiky Scrum, které je založeno na teoretických předpokladech, podnětem pro následný návrh změn za účelem efektivnějšího využití metodiky. Tato část práce analyzuje sledovaný tým také z hlediska zařazení v SW-CMM. Završení představuje shrnutí klíčových faktů s využitím principů SWOT analýzy.

Vlastní návrhy řešení pojednávají o doporučeních, která vyplývají z teoretické a analytické části práce. Tato doporučení jsou v souladu s dosažením dílčích, resp. hlavního cíle této práce. Na základě výstupů z analytické části práce jsou zde navrženy postupy řešení, které se zaměřují na odstranění chyb, kterých se sledovaný tým ve svém počínání z hlediska aplikace metodiky Scrum dopouští. Pro dosažení nevyšší úrovně z hlediska SW-CMM je navržena (a v průběhu práce také přijata) implementace metrik, na základě kterých je možné aplikovat korekce implementace metodiky. Tato filosofie vede k tomu, že sledovaný tým průběžně zvyšuje rychlost (velocity) a využívá tak metodiku Scrum efektivněji.

Návrhy řešení vyhodnocují také rizika implementace navržených řešení. Na základě analýzy rizik je od některých původních návrhů upuštěno, či jsou upraveny ve smyslu zavedení opatření.

Ekonomické zhodnocení vyhodnocuje náklady na zavedení doporučených řešení a opatření. Tato část práce předkládá minimální změnu ve výkonnosti týmu, kterou je nutné dosáhnout pro ziskovost implementace. Aplikace navržených řešení představuje minoritní náklady i rizika, z hlediska obratu sledovaného vývojového týmu. Na základě tohoto vyhodnocení lze konstatovat, že se jedná o projekt, jehož potencionální přínos je vysoký a náklady na realizaci jsou relativně nízké.

SEZNAM POUŽITÉ LITERATURY

- (1) MYSLÍN, J. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.
- (2) BUCHALCEVOVÁ, A. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada, 2005. ISBN 80-247-1075-7.
- (3) ROYCE W. Proceedings: 9th International Conference on Software Engineering, 30 April 1987, Monterey, California, USA. Washington, D.C.: IEEE Computer Society Press.
- (4) PROCHÁZKA, J. a C. KLIMEŠ. *Provozujte IT jinak: agilní a štihlý provoz, podpora a údržba informačních systémů a IT služeb*. Praha: Grada, 2011. ISBN 978-80-247-4137-6.
- (5) *Manifest Agilního vývoje software* [online]. c2001 [cit. 2017-02-11]. Dostupné z: <http://agilemanifesto.org/>
- (6) *Comprehensive Guide to the Agile Manifesto: The Twelve Agile Manifesto Principles*. Smartsheet [online]. Bellevue, WA 98004-4357 USA: Smartsheet.com, 2012 [cit. 2017-01-13]. Dostupné z: <https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>
- (7) LACEY, M. *The scrum field guide: practical advice for your first year*. Upper Saddle River, NJ: Addison-Wesley, c2012. ISBN 978-0-321-55415-4.
- (8) ŠOCHOVÁ, Z. a E. KUNCE. *Agilní metody řízení projektů*. Brno: Computer Press, 2014. ISBN 978-80-251-4194-6.
- (9) DOLEŽAL, J. *Projektový management: komplexně, prakticky a podle světových standardů*. Praha: Grada Publishing, 2016. ISBN 978-80-247-5620-2.
- (10) DAVIS, C. W. H. *Agile metrics in action: how to measure and improve team performance*. Shelter Island, NY: Manning, 2015. ISBN 978-1-61729-248-4
- (11) Agile Metrics: Let the Numbers tell the Tale [online]. , 1-14 [cit. 2017-05-13]. Dostupné z: <https://www.scribd.com/document/335152033/whitepaper-agile-metrics-1-pdf>
- (12) SUTHERLAND, J. V. *Scrum: the art of doing twice the work in half the time*. New York: Crown Business, 2014. ISBN 978-0-385-34645-0.

- (13) *Agile Forecasting with Focus Factor. Scrum Alliance* [online]. Westminster, CO 80021 USA, 2014 [cit. 2017-04-13]. Dostupné z: <https://www.scrumalliance.org/community/articles/2014/july/agile-forecasting-with-focus-factor><https://www.slideshare.net/sradics/agile-metrics-50873631>
- (14) *Agile metrics* [online]. Mnichov: ZOOPLUS AG, 2015 [cit. 2017-03-20]. Dostupné z: <https://www.slideshare.net/sradics/agile-metrics-50873631>
- (15) *O společnosti Avast.* [online]. Praha: Avast Software, 2016 [cit. 2017-03-20]. Dostupné z: <https://www.avast.com/cs-cz/about>
- (16) RAIS, K. a R. DOSKOČIL. *Risk management: studijní text pro kombinovanou formu studia.* Brno: Akademické nakladatelství CERM, 2007. Učební texty vysokých škol. ISBN 978-80-214-3510-0.
- (17) COHN, M. *Agile estimating and planning.* Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, c2006. ISBN 978-0-13-147941-8.
- (18) *CMM – model procesní vyspělosti organizace. Procesní řízení ITIL/ITSM* [online]. Praha: O2 IT Services, 2016 [cit. 2017-04-20]. Dostupné z: <http://www.ital.cz/index.php?id=956>

SEZNAM OBRÁZKŮ

Obrázek č. 1: Avast Software s.r.o. logo	38
Obrázek č. 2: Zjednodušená organizační struktura (vlastní zpracování)	41
Obrázek č. 3: Systémová architektura (převzato z IS Avast software s.r.o.).....	42
Obrázek č. 4: ICE, detail zákazníka (převzato z IS společnosti)	44
Obrázek č. 5: Kibana, vizualizace stavu procesů.....	47
Obrázek č. 6: Příklad user story pro projektový tým (převzato z IS Avast software)	55
Obrázek č. 7: Scrum board (převzato z IS Avast software s.r.o.).....	57
Obrázek č. 8: Sprint backlog (převzato z IS Avast software s.r.o.)	58
Obrázek č. 9: Project backlog (převzato z IS Avast software s.r.o.)	59
Obrázek č. 10: Scrum tabule (vlastní zpracování)	61
Obrázek č. 11: Výstup z retrospektivní schůzky (vlastní zpracování).....	63
Obrázek č. 12: Diagram průběhu projektu a sprintu (vlastní zpracování)	68
Obrázek č. 13: Korekce průběhu sprintu	79

SEZNAM TABULEK

Tabulka č. 1: Výhody a nevýhody vodopádového modelu (1, str. 26).....	16
Tabulka č. 2: Činnost rolí (7, str. 73)	22
Tabulka č. 3: Interní hodnocení objemu práce, výhody a nevýhody (1, str. 128)	35
Tabulka č. 4: Hodnocení objemu práce s využitím MDs (1, str. 129).....	36
Tabulka č. 5: Členové Scrum týmu a jejich role (vlastní zpracování).....	52
Tabulka č. 6: Průběh sprintu (vlastní zpracování)	56
Tabulka č. 7: RACI matice - vývojový proces	67
Tabulka č. 8: SWOT analýza projektového týmu (vlastní zpracování).....	70
Tabulka č. 9: Změna rolí projektového týmu (vlastní zpracování).....	71
Tabulka č. 10: Skórovací metoda k určení délky sprintu, otázky (7, str. 84)	75
Tabulka č. 11: Skórovací metoda k určení délky sprintu, klíč (7, str. 85).....	77
Tabulka č. 12: Výstup skórovací metody pro určení délky sprintu (7, str. 86)	78
Tabulka č. 13: Spokojenost - data (vlastní zpracování)	82
Tabulka č. 14: NPS (vlastní zpracování)	84
Tabulka č. 15: Rychlost (Velocity - V), (vlastní zpracování).....	85

Tabulka č. 16: Pracovní kapacita (Work Capacity - WC), (vlastní zpracování)	86
Tabulka č. 17: Faktor zaměření (Focus Factor – FF), (vlastní zpracování).....	88
Tabulka č. 18: Přesnost odhadu náročnosti (Estimation Accuracy - EA)	89
Tabulka č. 19: Identifikace hrozeb a scénářů, (vlastní zpracování).....	91
Tabulka č. 20: Kvantifikace rizik (vlastní zpracování).....	92
Tabulka č. 21: Ohodnocení rizik (vlastní zpracování).....	92
Tabulka č. 22: Závažnost rizika (vlastní zpracování)	92
Tabulka č. 23: Snižování rizika (vlastní zpracování)	94
Tabulka č. 24: HV sledovaného týmu, (vlastní zpracování).....	95
Tabulka č. 25: Předpoklad budoucího období, (vlastní zpracování)	96
Tabulka č. 26: Náklady na člověkodenní doručené práce - současný stav	96
Tabulka č. 27: Náklady na člověkodenní doručené práce - pesimistická varianta	97
Tabulka č. 28: Náklady na člověkodenní doručené práce - optimistická varianta	98
Tabulka č. 29: Náklady na člověkodenní doručené práce - realistická varianta.....	98

SEZNAM GRAFŮ

Graf č. 1: Burndown chart, příklad	65
Graf č. 2: Vytížení role Test Engineer v průběhu sprintu (vlastní zpracování).....	72
Graf č. 3: Vytížení role Developer v průběhu sprintu (vlastní zpracování)	73
Graf č. 4: Možné vytížení role Team Member v průběhu sprintu	73
Graf č. 5: Spokojenost (Happiness - H), (vlastní zpracování)	83
Graf č. 6: Míra loajality zákazníků (Net Promotion Score - NPS).....	84
Graf č. 7: Rychlost (Velocity - V), (vlastní zpracování).....	85
Graf č. 8: Pracovní kapacita (Wokr Capacity - WC), (vlastní zpracování)	87
Graf č. 9: Faktor zaměření (Focus Factor – FF), (vlastní zpracování)	88
Graf č. 10: Přesnost odhadu náročnosti (Estimation Accuracy - EA)	89
Graf č. 11: Mapa rizik, (vlastní zpracování)	95

SEZNAM PŘÍLOH

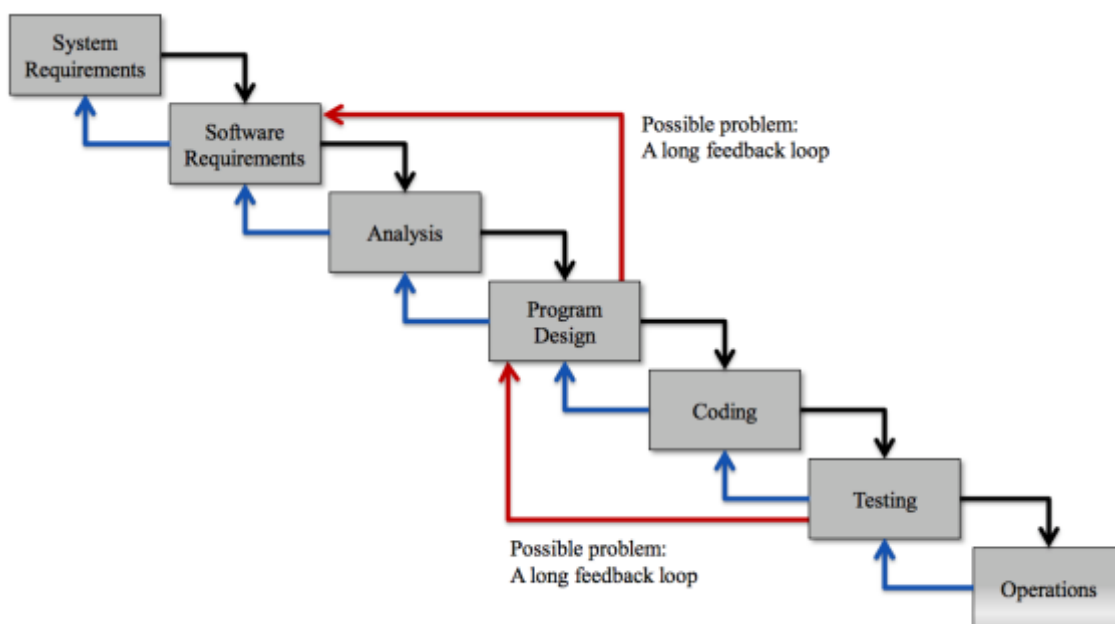
Příloha č. 1: Stupně vyspělosti procesu	I
Příloha č. 2: Vodopádový proces (3)	I
Příloha č. 3: EPC diagram vývojového procesu	II

PŘÍLOHY

Příloha č. 1: Stupně vyspělosti procesu (18)



Příloha č. 2: Vodopádový proces (3)



Příloha č. 3: EPC diagram vývojového procesu (vlastní zpracování)

