

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	Zimní 2017/2018
Autoři	Bc. Zuzana Čecháková, cecz00 Bc. Matej Ďurica, xdurm25 Bc. Daniel Mäsiar, masd00
Téma	Six Ways Agile Can Turn Static
Datum odevzdání	17. 12. 2017

Abstrakt:

V současné době je velmi rozšířené využití agilních metodik na vývoj software. Při použití těchto metodik mohou nastat různé problémy. Tato práce je proto zaměřená na představení šesti oblastí dle článku “*Six Ways Agile Can Turn Static*”, ve kterých mohou nastat v agilních metodikách problémy.

Klíčová slova: metodiky, vývoj software, agilní metody, projekt, spolupráce

Obsah

1	Úvod.....	3
	1.1 Cíl semestrální práce	3
	1.2 Postup dosažení cíle	3
2	Metodiky pro vývoj software	4
	2.1 Rigorózní metodiky	4
	2.2 Agilní metodiky	5
3	Šest oblastí, ve kterých mohou nastat v agilní metodice problémy.....	7
	3.1 Architektura	7
	3.1.1 Problém špatně zvolené míry flexibility architektury	7
	3.2 Nástroje pro spolupráci mezi IT a businesssem	8
	3.2.1 Problém neexistence nástrojů pro spolupráci mezi IT a businesssem v agilních metodikách	8
	3.3 Lidské zdroje	9
	3.3.1 Problém “osamělých vlků”	9
	3.3.2 Problém neflexibility členů týmu	9
	3.4 Komunikace	10
	3.4.1 Problém nedostatečně řízené komunikace na velkých projektech ...	10
	3.4.2 Problém nedostatečně řízené komunikace v geograficky rozptýlených týmech	11
	3.5 Typ projektu	12
	3.5.1 Problém použití agilní metodiky pro nevhodný typ projektu.....	13
	3.6 Spolupráce	13
	3.6.1 Problém nemožnosti každodenní spolupráce	13
4	Závěr.....	14
5	Seznam literatury	15
6	Seznam obrázků	16

1 Úvod

V dnešní době je snahou společností držet tempo s inovacemi, dosahovat okamžitých výsledků a klást důraz na kvalitu. To je i důvod, proč čím dál tím více IT týmů využívá agilní metodiky pro vývoj software.

Agilní vývoj velice rychle poskytuje počáteční obchodní hodnotu za pomoci průběžného plánování a zpětné vazby. Výsledkem tohoto iterativního plánovacího a zpětnovazebního cyklu jsou týmy schopné nepřetržitě uspokojovat požadované obchodní potřeby a snadno se přizpůsobit měnícím se požadavkům v průběhu celého procesu. Výhodou je i měřitelnost a hodnocení aktuálního stavu projektu, který je viditelný ve všech jeho fázích všem zainteresovaným stranám. Na konci procesu existuje tedy systém, který lépe odpovídá požadavkům zákazníka.

Ačkoli jsou dnes agilní metodiky vývoje prosazovány a jejich použití zní optimálně, realita může být často odlišná. Existuje mnoho situací, kdy agilní metodiky nemusí vždy přinášet očekávané výsledky.

1.1 Cíl semestrální práce

Cílem semestrální práce je analyzovat šest oblastí, ve kterých se agilní metodika může potenciálně stát statickou.

1.2 Postup dosažení cíle

V semestrální práci je nejdříve vysvětleno, co je to metodika, k čemu se používá a následně jsou popsány dva typy metodik – rigorózní a agilní. Hlavní část semestrální práce se zabývá jen agilními metodikami, a to konkrétně analýzou a popisem problémů v šesti oblastech, kdy může dojít k přeměně agilní metodiky na statickou.

Primárním zdrojem informací, ze kterého je vycházeno při dané analýze, je článek *Six Ways Agile Can Turn Static* od autorky Ronit Eliav. Pro hlubší analýzu a popis jednotlivých oblastí je dále čerpáno i z jiných zdrojů a v určitých případech jsou i dohledány prokázání některých tvrzení v proběhlých výzkumech v praxi.

2 Metodiky pro vývoj software

Metodik pro vývoj software je velké množství a nejsou jednotně popsány. Je obtížné je srovnávat a vyhledat vhodnou metodiku. Mnohé metodiky se zaměřují jen na určité aspekty vývoje nebo na určité fáze životního cyklu. Je mnoho důvodů, proč existuje takové množství metodik. Různé technologie vyžadují různé techniky, organizace se liší firemní kulturou, každý tým a taky každý jedinec je jedinečný, a také se projekty liší svou velikostí a důležitostí. (Buchalcevová, 2009)

Metodiky lze rozdělit do dvou skupin – rigorózní a agilní metodiky. V následujících podkapitolách jsou blíže popsány.

2.1 Rigorózní metodiky

Do rigorózních metodik spadají metodiky, které využívají vodopádový model životního cyklu software. Vycházejí z předpokladu, že všechny požadavky je možné specifikovat předem, a jakýmkoliv změnám se snaží zabránit. Jde o náročné a těžké metodiky, které jsou podrobné, velmi formální a obsahují velké množství meziproductů. Rigorózní metodiky jsou postaveny na nedůvěře, snaží se vykázat lidi do role zaměnitelné součástky. (Buchalcevová, 2009)

Roycův původní vodopádový model obsahuje sedm fází v následujícím pořadí (ROYCE, 1970):

- Specifikace požadavků
- Návrh
- Implementace
- Integrace
- Testování a ladění (validace)
- Instalace
- Údržba

Vodopádový model je mnohými považován za nevhodný pro praxi. Kritici jsou především přesvědčeni, že u jakéhokoli netriviálního projektu je nemožné dovést jednu fázi životního cyklu softwarového produktu k dokonalosti předtím, než se přejde k fázi následující. Klientům nemusí být například úplně jasné, jaké jsou jejich požadavky, dokud neuvidí fungující prototyp, ke kterému se pak mohou vyjádřit.

Myšlenkou vodopádového modelu může být „dvakrát měř, jednou řež“. Odpůrci vodopádového modelu namítají, že tato myšlenka selhává, když problém, kterým se zabýváme, je neustále měněn kvůli měnícím se požadavkům.

Hlavně z důvodu nutnosti specifikovat požadavky předem, a také z důvodu averze ke změnám, jsou rigorózní metodiky pro mnohé současné projekty nevyhovující. Taky se zde ztrácí cíl vývoje a to vytvoření fungujícího software odpovídajícího potřebám uživatelů.

Příkladem rigorózních metodik jsou OPEN, RUP (Rational Unified Process) nebo EUP (Enterprise Unified Process). (Buchalceková, 2009)

2.2 Agilní metodiky

Agilní metodiky jsou lehké metodiky, které nepopisují procesy ale principy, obsahují málo dokumentace a dávají přednost osobní komunikaci. Jsou zaměřeny na ty činnosti, které vytvářejí hodnotu a eliminují činnosti, které hodnotu nepřinášejí. Přesouvají zodpovědnost za definování požadavků na zákazníka. (Buchalceková, 2009)

Společnými principy agilních metodik jsou iterativní vývoj s velmi krátkými iteracemi, důraz na spolupráci a komunikaci, tolerance ke změnám a automatizované testování. Jsou formovány na důvěře a respektu, využívají individualit a silných stránek lidí. (Buchalceková, 2009)

Podle manifestu pro agilní vývoj software dáváme přednost (Cunningham, 2001):

- individualitám a komunikaci před procesy a nástroji
- provozuschopnému software před obsažnou dokumentací
- spolupráci se zákazníkem před sjednáváním kontraktu
- reakci na změnu před plněním plánu.

10 hlavních principů agilních metodik:

- Nejvyšší prioritou je včas a kontinuálně dodávat software, který zákazníkům přináší hodnotu.
- Změnu požadavků je možné provést i v pozdějších fázích vývoje, protože tím může zákazník získat konkurenční výhodu.
- Uživatelé a vývojáři spolupracují denně na projektu.
- Motivovaní jedinci, kteří mají vytvořeny podmínky pro práci a mají podporu vedení, jsou klíčovým faktorem úspěchu projektu.
- Nejefektivnějším způsobem přenosu informací v rámci vývojového týmu je osobní komunikace.
- Primární mírou úspěchu je fungující software.
- Agilní procesy předpokládají „zdravý“ vývoj.
- Perfektní technické řešení i návrh.
- Zásadním požadavkem je jednoduchost řešení, tj. umění maximalizovat množství neudělané práce.
- Nejlepší architektury, požadavky a návrhy vznikají ze samoorganizujících se týmů.

Mezi původní agilní metodiky patří ASD (Adaptive Software Development), DSDM (Dynamic Systems Development Method), FDD (Feature-Driven Development), XP (Extreme Programming), SCRUM, Lean Development, Crystal metodiky a další. (Buchalceková, 2009)

3 Šest oblastí, ve kterých mohou nastat v agilní metodice problémy

V následujících kapitolách jsou popsány oblasti, ve kterých mohou nastat při použití agilní metodiky problémy. Tyto problémy pak mají za následek přeměnu agilní metodiky na statickou. Ne vždy totiž v praxi mohou agilní metodiky přinášet očekávané výsledky.

3.1 Architektura

Ve dvanácti principech, které jsou uvedeny v českém vydání Agilního Manifestu lze nalézt princip v následujícím znění:

“Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.”
(Cunningham, 2001)

Zatímco Agilní Manifest uvádí principy, nikoliv konkrétní postupy či doporučení, je možné si pod technickou výjimečností a dobrým designem představit ledacos. V rámci této části práce je však věnována pozornost především dobrému designu systémové architektury, čili pomyslné páteře technické stránky projektu. (Eliav, 2017)

V agilních projektech se lze nejčastěji setkat s pojmem adaptivní architektura. Adaptivní architektura spočívá v průběžném modifikování a upravování architektury v takovém rozsahu, aby vždy s minimální pracností bylo docíleno stavu, kdy architektura podporuje funkcionality, které jsou v rámci agilního vývoje již dostatečně specifikované. To nechává prostor pro přidání dalších, dříve nepožadovaných funkcionalit, neboť v případě neočekávaných požadavků lze v další fázi plánování architektury s novými skutečnostmi počítat. Tento aspekt adaptivní architektury je však zároveň hrozbou, která v extrémních případech může vést k zastavení nebo výraznému prodražení celého projektu. (Eliav, 2017)

3.1.1 Problém špatně zvolené míry flexibility architektury

Problémem v této oblasti je, s jakým časovým horizontem bude při plánování architektury kalkulováno. Pokud plánování vychází pouze z funkcionalit, které budou pouze bezprostředně implementovány, v budoucnu jistě nastane situace, kdy tato krátkozrakost zapříčiní, že na

nové požadavky architektura již nebude připravena a dojde k pomyslnému “vrstvení” nového kódu na kód stávající, místo pouhého elegantního začlenění do obecněji připravené architektury. (Eliav, 2017)

V opačném případě, kdy je architektura navrhována dlouhodobě dopředu, nastává úbytek flexibility a oné agility v agilním projektu, neboť v případě nečekaných požadavků je velice složité architekturu upravovat. V obou dvou krajních případech tedy tým vytvoří architekturu, která je v konečné fázi neflexibilní a obtížně udržovatelná. (Eliav, 2017) Cílem je tedy správně odhadnout, jak moc napřed je potřeba architekturu plánovat tak, aby byla zachována konzistence a zároveň flexibilita a v mnoha případech je velmi obtížné tuto rovnováhu nalézt.

3.2 Nástroje pro spolupráci mezi IT a businesssem

Další oblast, která může mít negativní dopad na proces vývoje pomocí agilních metodik, je existence, respektive absence, nástrojů pro takzvaný Application Lifecycle Management (ALM), které se používají při spolupráci mezi IT zaměstnanci (vývojáři, testeři, administrátoři aj.) a business uživateli (vlastníci produktů, prodejci, manažeři a další). ALM může mít podobu jednoduché wiki stránky, kde se veškeré změny zaznamenávají ručně, po komplexní systém, které automaticky rozřazují změny v aplikaci a monitorují celý její průběh od počátku do konce.

3.2.1 Problém neexistence nástrojů pro spolupráci mezi IT a businesssem v agilních metodikách

Trh s ALM je však z velké části naplněn monolitickými, on-premise, řešeními, které jsou často navrženy pro použití ve vodopádovém přístupu vývoje a postrádají větší podporu pro využití ve spojení s agilními metodikami. (Eliav, 2017)

Zatímco odvětví vývoje software plně využívá široké škály agilních nástrojů pro spolupráci jako je například Jenkins, VersionOne či RallyDev, tyto nástroje nebyly navrženy pro použití v rozsáhlých IT organizacích. Jsou to nástroje, které mají sloužit právě softwarovým inženýrům, nikoliv však business uživatelům, pro které tyto nástroje nejsou vhodné kvůli přemíře funkcionalit a technické složitosti. (Eliav, 2017)

3.3 Lidské zdroje

Agilní metodiky jsou založeny na spolupráci zákazníků a vývojářů. Jak ve svém článku uvádí i Ronit Eliav: „*collaboration is a key success factor*“ (Eliav, 2017). Úspěšné fungování agilní metodiky velice záleží právě na lidech v týmu. Je důležité, aby byli ztotožněni s hlavními principy agilní metodiky, byli schopni změnit své návyky a stále rozvíjet své sociální i jiné dovednosti.

3.3.1 Problém “osamělých vlků”

Problém však může nastat u jedinců, kteří jsou zvyklí pracovat autonomně a domnívají se, že požadovaná spolupráce může zpomalit jejich výkon. Tato skutečnost může způsobit odmítání některých členů týmu přizpůsobit jejich pracovní styl. Myšlenka sdíleného učení, párového programování nebo společného rozhodování je nepřipustná, zvláště pro vývojáře, kteří vykonávají aktivity samostatně nebo spolupracují s homogenní skupinou analytiků a designerů (Nerur, 2005). Těmto lidem se často říká “osamělí vlci”. Jsou to jedinci, kteří moc nekomunikují, nejsou zvyklí s někým spolupracovat a nejraději si samostatně dělají svoji práci.

Myšlenka o vývojářích či jiných členech týmu jako o “osamělých vlcích” je v agilním přístupu nepoužitelná a porušení této zásady může v týmu, používající agilní metodiku, způsobit velké problémy. (Eliav, 2017) Agilní metodika by se poté přeměnila ve statickou.

3.3.2 Problém neflexibility členů týmu

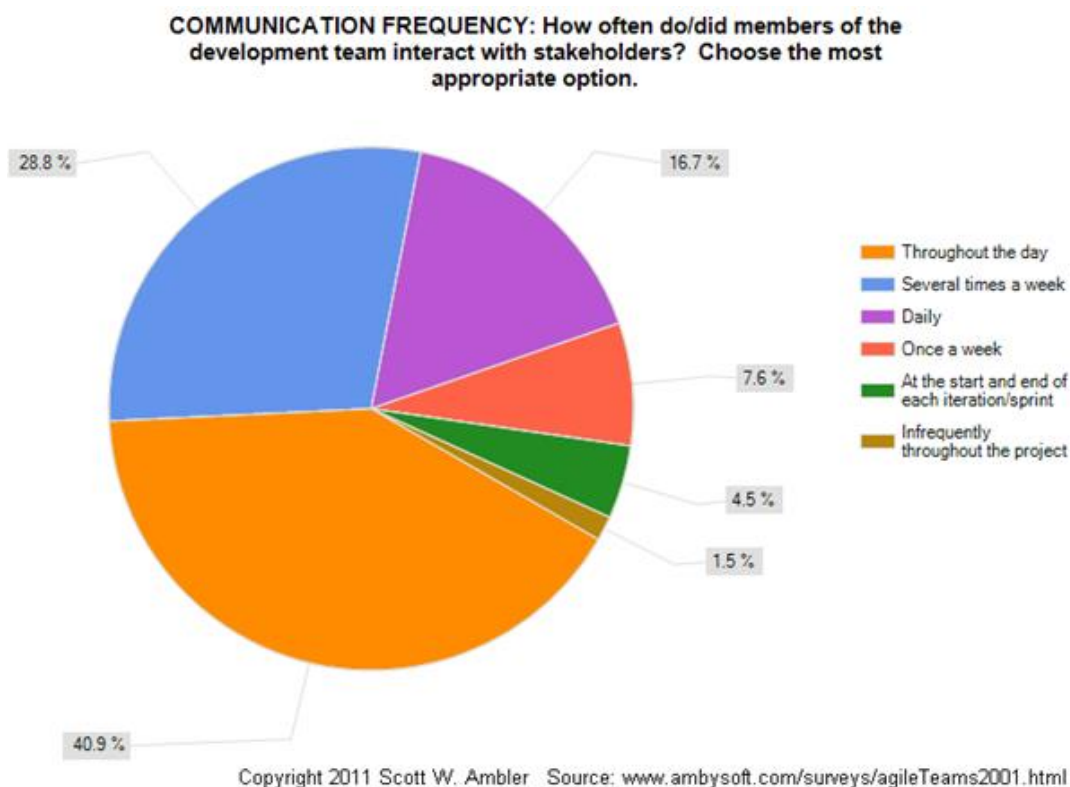
Dalším faktorem pro úspěšné fungování agilního týmu je v dnešní době i flexibilita jeho členů. (Eliav, 2017) Společnost Gartner v roce 2016 vydala zprávu, ve které je mnoho zajímavých poznatků v oblasti testování aplikací. Jedná se o shrnutí současných trendů a předpovědí budoucího vývoje tohoto oboru. Jednou z předpovědí je, že do roku 2020 bude 60% testerů potřebovat schopnosti nejen v oblasti testování, ale i v oblastech vývoje aplikací, business procesů a v daném odvětví. (Eliav, 2017)

3.4 Komunikace

Dříve bylo použití agilních metodik spjaté pouze s malými a středními podniky. V dnešní době se však tyto lehké metodiky více a více stávají populárnější a používanější i ve velkých organizacích. Ve velkých organizacích mívají i projekty větší rozsah než v malých firmách, a to má za následek nutnost použití dalších koordinačních opatření. Projekty ve velkých firmách jsou propojeny s mnoha organizačními jednotkami jako jsou lidské zdroje, marketing a prodej, řízení produktů atd. Často jsou také členové týmu v různých geografických lokalitách. Proto je na problém rozsáhlosti projektu a distribuovanosti členů týmu nutno reagovat – je potřeba klást důraz na efektivní komunikaci. (Eliav, 2017)

3.4.1 Problém nedostatečně řízené komunikace na velkých projektech

V agilních metodikách je častá nejen komunikace mezi různými týmy, ale i mezi všemi zainteresovanými osobami projektu. Čím větší rozsah projektu, tím i více zainteresovaných osob. Konzultant Scott W. Ambler provedl v roce 2011 dotazníkové šetření zabývající se agilními metodikami, kterého se zúčastnilo 82 respondentů. Jeho cílem bylo zjistit základní informace o velikosti a geografickém rozšíření daného agilního týmu, a dále frekvenci interakce jednotlivých členů týmu se zainteresovanými osobami. (Ambler, 2011) Výsledek frekvence interakce znázorňuje následující graf.



Obr. 1 – Frekvence interakce se zainteresovanými osobami v agilním týmu (Ambler, 2011)

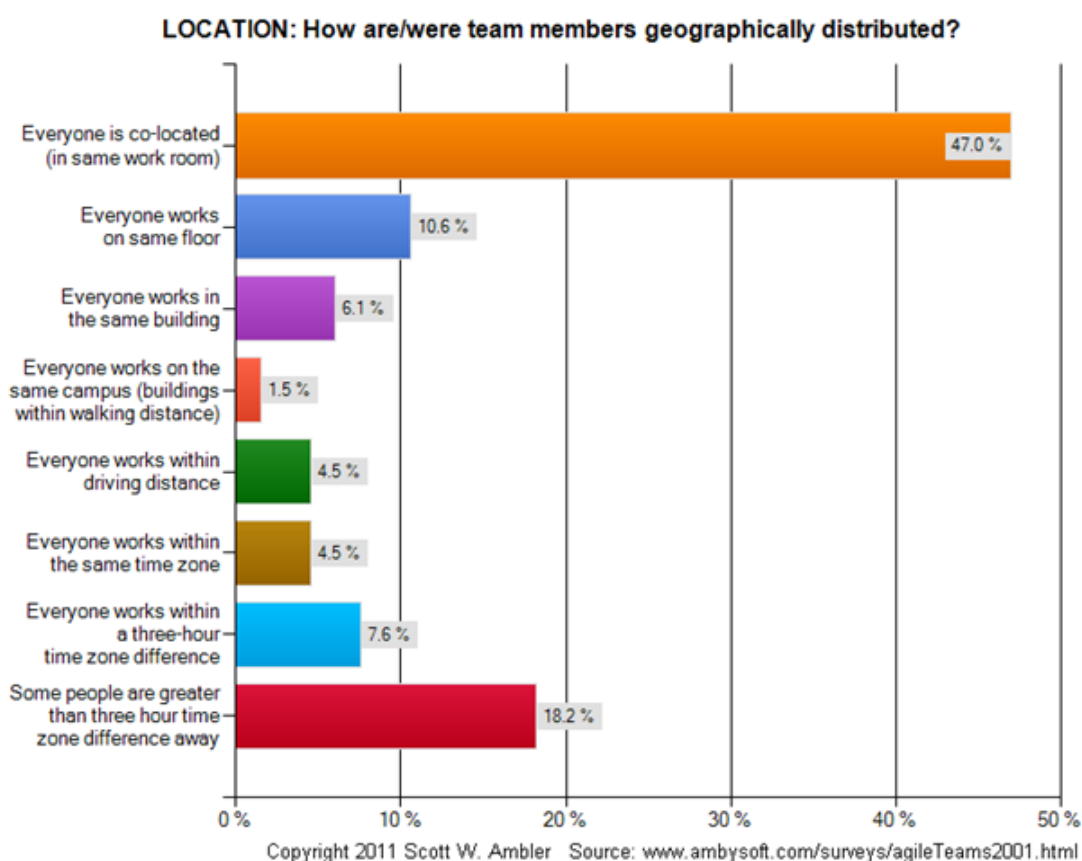
Z grafu je viditelné, že téměř 58 % dotazovaných uvedlo, že interakce se zainteresovanými osobami v daném týmu probíhá minimálně jednou každý den. Dalších 28 % uvedlo, že k tomu dochází několikrát týdně. Z toho je patrné, že komunikace se zainteresovanými osobami je v agilních metodikách velmi důležitá.

Agilní metodiky jsou založené na blízkých pracovních vztazích a zpětné vazbě od uživatelů. Na rozdíl od rigorózních metodik jsou postavené na osobní komunikaci. Tím, jak roste velikost projektu, tak i komunikační toky navyšují svůj počet, a to dokonce exponenciálně. (Eliav, 2017) Je tedy potřeba komunikaci vhodně řídit, jinak by mohlo dojít k problémům na projektu.

3.4.2 Problém nedostatečně řízené komunikace v geograficky rozptýlených týmech

Podniky se také rozšiřují do nových geografických oblastí nebo se spojují s jinými organizacemi a projekty, jsou rozptýleny v několika týmech na různých místech. I toto se může stát potenciálním problémem.

Nutnost reagovat i na tyto skutečnosti ukazují i následující výsledky výzkumu pana Amblera, ve kterých poměrně značné procento respondentů uvedlo, že jsou členové týmu geograficky vzdáleni.



Obr. 2 – Geografické rozmístění členů týmu (Ambler, 2011)

V případě, že není adekvátně zareagováno na geograficky rozptýlené týmy, tak může dojít ke zpoždění projektu či snížení kvality. (Eliav, 2017)

3.5 Typ projektu

Projekty se liší velikostí týmu, zaměřením, rozsahem, přesností definovaných požadavků a dalšími znaky. Všechny tyto faktory se musí vzít v potaz při výběru vhodné metodiky.

Rigorózní metodiky je vhodné používat tam, kde je dopředu možné definovat požadavky a lze očekávat, že budou neměnné. Jsou doporučovány pro standardní a velké projekty. (Buchalceková, 2009)

Naopak agilní metodiky lze využít u projektů, kde jsou předem známy jen hrubé požadavky. Často je jejich použití doporučováno pro výzkumné projekty, time-to-market a menší týmy. (Buchalceková, 2009)

3.5.1 Problém použití agilní metodiky pro nevhodný typ projektu

Problém nastává v momentě, kdy se pro projekt zvolí agilní metodika, zatímco by se pro daný projekt více hodila metodika rigorózní. Může se jednat o situaci, kdy jsou požadavky naprosto přesně definované – například projekt, jehož cílem je implementace legislativních změn. V tomto případě použití agilní metodiky postrádá smysl.

3.6 Spolupráce

V agilních metodikách je velmi důležitá spolupráce. Ta by měla probíhat na denní bázi a je založená na osobní komunikaci. Spolupráce je realizována jak mezi členy týmu tak i se zákazníkem.

3.6.1 Problém nemožnosti každodenní spolupráce

Realita většiny projektů dnes spočívá v tom, že různí členové týmu nejsou v těsné blízkosti nebo nejsou k dispozici pro komunikaci tváří v tvář. V takových případech jsou pak členové týmu odkázáni na využívání komunikačních nástrojů, které však nemusejí být vždy dostupné a mohou být náchylné k chybám či nedorozuměním. (Eliav, 2017)

4 Závěr

Cílem semestrální práce bylo analyzovat šest oblastí, ve kterých mohou nastat problémy, kterými se agilní metodika může potenciálně stát statickou. Tento cíl byl zcela naplněn a práce je zpracována takovou formou, která umožňuje čtenáři snadno získat náhled do jednotlivých oblastí. Zpracování práce nedoprovázely komplikace zásadního charakteru, neboť téma agilních metodik je v současné době aktuální a lze nalézt dostatek relevantní a otevřené literatury.

V této semestrální práci jsou zpočátku definovány dva typy metodik – rigorózní a agilní. Stěžejní část práce pak rozebírá dohromady šest rozdílných oblastí, kde se lze setkat v agilní metodice s problémy, kterými může dojít k přeměně agilní metodiky na statickou. Tyto problémy jsou blíže v každé části popsány.

Při rozhodování o použití agilních metodik je nezbytné se na těchto šest oblastí zaměřit a přijmout opatření pro eliminaci problémů, které v nich mohou nastat.

5 Seznam literatury

AMBLER, Scott. *Agile Teams Mini-Survey Results: April/May 2011*. 2011. Dostupné z URL <http://www.ambysoft.com/surveys/agileTeams2011.html>.

BUCHALCEVOVÁ, Alena. *Metodiky budování informačních systémů*. 2009. Praha: Oeconomica. ISBN 978-80-245-1540-3.

CUNNINGHAM, Ward. *Manifest Agilního vývoje software*. 2001. Dostupné z URL <http://agilemanifesto.org/iso/cs/principles.html>.

ELIAV, Ronit. *Six Ways Agile Can Turn Static*. 2017. Dostupné z URL <https://www.infoq.com/articles/ways-agile-static>.

NERUR, Sridhar. *Challenges of migrating to agile methodologies*. 2005. Communications of the ACM – Adaptive complex enterprises. Volume 48 Issue 5. Pages 72-78.

ROYCE, Winston W. *Managing the Development of Large Software System*. 1970. IEEE WESCON, Aug. 1970, pages 1-9.

6 Seznam obrázků

Obr. 1 – Frekvence interakce se zainteresovanými osobami v agilním týmu (Ambler, 2011)...11

Obr. 2 – Geografické rozmístění členů týmu (Ambler, 2011).....12