

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY



Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS

Semestr

LS 2016

Autoři

Lucie Pokorná, xpokl18

Markéta Zukalová, xzukm00

Téma

Four Must-Have Rules for Scaling Enterprise
Agile

Datum odevzdání

15. 5. 2016

Abstrakt

Každý podnik se před vlastním vývojem software rozhoduje, jakým způsobem k celému procesu přistoupí – zda-li agilně, či rigorózně. Oba přístupy mají své výhody a svá omezení, a společnosti tak musí pečlivě zvážit jejich charakteristiky, aby byl vývoj efektivní a maximálně přínosný pro zákazníka. Tradičně jsou agilní metodiky určeny pro menší týmy pracovníků, protože jsou schopny reagovat pružněji a nemají vždy přesně stanovenou hierarchii. Co dělat ale v případě, kdy se velký podnik rozhodne, že chce využít přínosů agilních metodik a přistoupit agilně k jeho vývoji? Existují nějaké zásady, kterých se musí držet, aby implementace metodiky byla úspěšná? Na tyto otázky hledáme odpovědi v naší semestrální práci. Výsledkem jsou tak 4 hlavní zásady, které by měly jakémukoli podniku dopomoci k efektivní adopci agilní metodiky Scrum pro vývoj projektu.

Klíčová slova

Agilní metodiky, implementace, pravidla, Scrum, velké podniky, vývoj software.

Obsah

Úvod.....	1
1 Agilní versus rigorózní metodiky vývoje SW	2
1.1 Rigorózní metodiky	2
1.2 Agilní metodiky.....	2
1.3 Srovnání agilních a rigorózních metodik	3
1.4 Příklady metodik agilního vývoje	4
1.5 Metodiky vývoje SW ve velkých firmách	5
2 Doporučená příprava k přijetí agilních metodik ve velkých korporacích.....	6
3 Základní pravidla při podnikovém využití agilních metodik	9
3.1 Porozumění konceptu	9
3.2 Odvykání od starých návyků	10
3.3 Respektování omezení agilních metodik	10
3.4 Lidé na prvním místě.....	11
4 Příklady využití agilního vývoje u velkých projektů	13
Závěr	15
Seznam použitých zdrojů	16
Seznam objektů	17

Úvod

Dnešní svět se nese v duchu rychlých technologických změn a vývoje veškerého IT software a aplikací, a tradiční rigorózní metodiky vývoje se tak dostávají do pozadí za agilní metodiky, které společně umožňují vytvořit řešení velmi rychle a pružně jej přizpůsobovat měnícím se požadavkům. Agilní metodiky se začaly rozvíjet od poloviny 90. let a jejich podstata tkví právě v co nejrychlejší vývoji softwaru, těsné komunikaci a spolupráci se zákazníkem a do popředí posouvají hotové produkty před rozsáhlou dokumentací a přesně dodrženým postupem zpracování produktu (Buchalceová, 2005).

Tradičně jsou agilní metodiky dedikovány menším a pružnějším týmům, které nekladou zvýšený důraz na skupinovou či podnikovou hierarchii; jinými slovy, rozdělení rolí závisí na daném projektu a dispozici pracovníků, schvalovací procesy nejsou pevně stanoveny (cílem je co nejrychlejší vývojový proces). V případě, že se velký podnik rozhodne implementovat některou agilní metodiku, může tak narazit na značné problémy, vyplývající z podstaty agilních metodik. Je zřejmé, že s velikostí podniku roste i jeho organizační hierarchie, která, v případě, že není zaměstnanci dodržována, může vést k celkovému chaosu a neefektivitě vývojového týmu. Z tohoto důvodu je nezbytné, aby před vlastní implementací podnik dodržel několik základních pravidel, která mu pomohou bezproblémově přejít na agilní vývoj.

Tato práce se věnuje právě těmto zásadním pravidlům, která mohou sloužit velkým společnostem jako východisko pro adoptování agilních metodik. Cílem této práce je stanovení minimálně čtyř zásad, kterými by se měly velké podniky řídit, a to na základě článku „Four Must-Have Rules for Scaling Enterprise Agile“ (Kot, 2015). Využijeme k tomu metodu indukce, tedy na základě několika konkrétních faktů určíme základní a obecně použitelná pravidla, která budou aplikovatelná na jakýkoli typ velkého podniku při implementaci agilních metodik.

1 Agilní versus rigorózní metodiky vývoje SW

Na úvod práce krátce charakterizujeme jednotlivé přístupy k vývoji software. Jak již bylo řečeno výše, každý přístup má své výhody a svá omezení, a podnik tak musí prvně stanovit, zda-li jeho výběr odpovídá celému projektu.

1.1 Rigorózní metodiky

Rigorózní metodiky vycházejí z předpokladu, že veškeré požadavky na výsledný produkt je možné definovat předem a snaží se tak minimalizovat celkový počet změn v rámci projektu. Mezi minimální prvky těchto metodik lze zařadit role, procesy, činnosti a produkty. Řídí se tím, že veškeré procesy a mezivýstupy je nezbytné pečlivě dokumentovat, proto jsou tyto metodiky často spojovány s rozsáhlou dokumentací (Buchalceová, 2009). Vychází z myšlenky, že zákazník přesně neví, co chce, a tedy projektový tým je tou osobou, která to musí zákazníkovi představit a vytvořit. Standardně jsou využívány velkými podniky s rozsáhlou organizační strukturou a při řízení velkých, nákladných a časově dlouhotrvajících projektů (Chlapek, 2011).

1.2 Agilní metodiky

Oproti rigorózním metodikám se agilní metodiky zaměřují na co nejrychlejší vývoj produktu, upřednostňují dodání fungujícího výstupu před rozsáhlou dokumentací, zaměřují se na velmi těsnou spolupráci se zákazníkem, respektují, že pro zákazníka není vždy možné definovat veškeré požadavky na produkt před vlastním zahájením vývoje a snaží se tak na veškeré změny reagovat pružně, a ne je minimalizovat. Mezi typické prvky agilních metodik tak patří pouze principy a praktiky (Buchalceová, 2005).

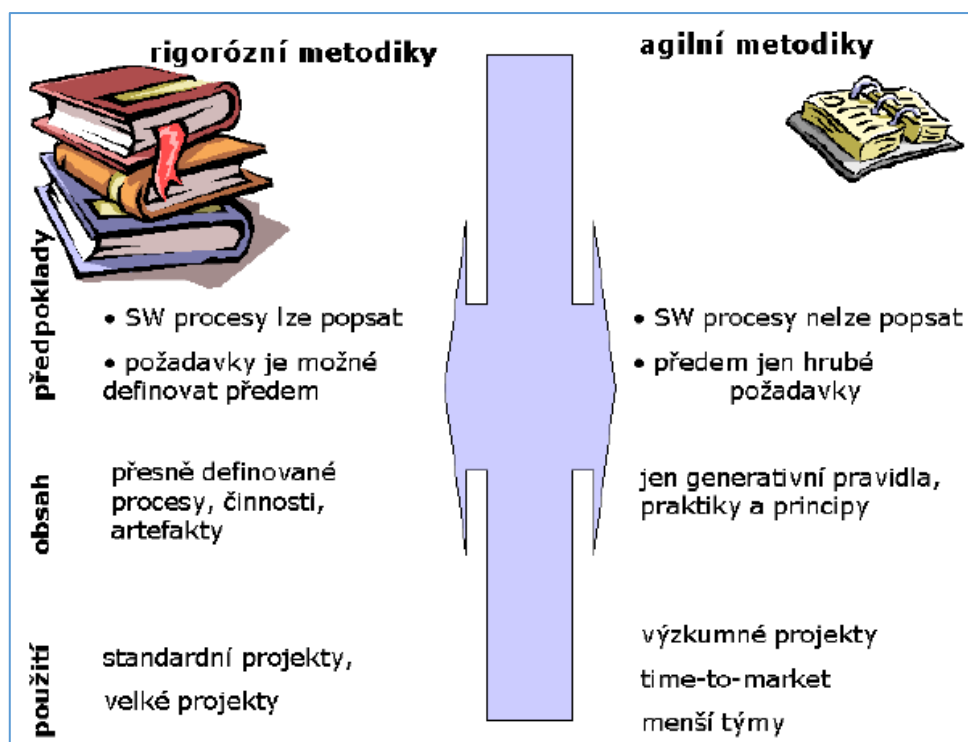
Principy agilních metodik jsou formulovány v Manifestu pro agilní vývoj softwaru, který byl vytvořen v roce 2011 v reakci na nevyhovující současný stav rigorózních metodik. Základní principy Manifestu byly později rozpracovány do 12 základních principů (Buchalceová, 2005):

1. Naši nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného softwaru.
2. Víťáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
3. Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.
4. Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.
5. Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.
6. Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.

7. Hlavním měřítkem pokroku je fungující software.
8. Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.
9. Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.
10. Jednoduchost--umění maximalizovat množství nevykonané práce--je klíčová.
11. Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.
12. Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.

1.3 Srovnání agilních a rigorózních metodik

Adopce vybrané metodiky vychází z rozdílných předpokladů pro proces vývoje softwaru. Schematické



Obrázek 1 Srovnání metodik. Zdroj (Buchalceková, 2005, s. 70)

Obrázek 2 Srovnání metodik. Zdroj (Buchalceková, 2005, s. 70)

znázornění této skutečnosti je uvedeno na obrázku níže (Buchalceková, 2005):

Tyto dva přístupy je možné srovnávat dále na základě zvolených hledisek, vybrané body jsou uvedeny v tabulce níže (Buchalceková, 2009):

Hledisko	Rigorózní metodiky	Agilní metodiky

Náplň metodiky	Procesy, zaměřují se na explicitní znalost a pohlíží na lidi jako na sekundární faktor	Praktiky, zaměřují se na tacit znalosti, chápou lidi jako klíčové faktory úspěchu
Předvídatelnost	Důraz na anticipaci, předpokládá předvídatelnost budoucnosti	Nepředvídatelnost budoucnosti, důraz na adaptaci na změny
Změny		
Definovatelnost procesu vývoje SW	Vývoj SW je definovatelný proces a je možné ho opakovat	Vývoj SW je empirický proces, nemůže být konzistentně opakován
Dokumentace	Rozsáhlá dokumentace	Podstatná není dokumentace, ale pochopení
Způsob vývoje	Spíše vodopádový, případně iterativní a přírůstkový	Přírůstkový vývoj s krátkými iteracemi

Tabulka 1 Srovnání agilních a rigorózních metodik. Zdroj: (Buchalcevoová, 2009, s. 63)

1.4 Příklady metodik agilního vývoje

Agilní metodiky jsou tedy často využívány v případě, kdy projekt splňuje následující kritéria:

- Je krátký (z časového hlediska),
- Je charakteristický nízkou mírou organizační struktury,
- Je možná osobní komunikace mezi členy týmu,
- Často je také využíván při vypracování různých bakalářských a diplomových prací.

Mezi nejčastěji používané agilní metodiky patří (Myslín, 2016):

1. Extrémní programování

Tato metodika dovádí do extrému mnohé principy známé z jiných metodik, odtud její název. Spočívá ve velmi krátkých iteracích, přičemž některé mohou trvat pouhý den; testovací proces probíhá neustále a zákazník se v podstatě stává součástí vývojového týmu. Extrémní programování stojí na čtyřech základních hodnotách – komunikace (komunikovat musí všichni členové týmu otevřeně), jednoduchost (co není nezbytně nutné, neprovádí se), zpětná vazba (neustálé testování) a odvaha.

2. Lean development

Jedná se spíše o souhrn pravidel, jejichž používání by mělo zefektivnit vývojový proces. Tato pravidla zní: eliminovat zbytečné, zdůraznit proces učení, rozhodovat se tak rychle, jak je možné, posílit odpovědnost týmu a vidět systém jako celek. Součástí metodiky jsou také principy a nástroje, které tato pravidla umožní zrealizovat.

3. Scrum

Podrobnosti využití této metodiky jsou uvedeny dále v semestrální práci. Základem této metodiky jsou tzv. Sprints (nebo iterace), což jsou základní jednotkou vývoje. Sprint je časově omezená snaha, je tedy omezen na specifický čas. Tým pracovníků se skládá ze Scrum mastera, Product ownera a vývojového týmu. Veškeré požadavky jsou uchovávány v tzv. backlogu, a je na nich průběžně pracováno právě skrze jednotlivé Sprints.

4. LeSS

Large Scale Scrum (LeSS) je agilní metodika, která je určena pro velké týmy, její jednotlivé procesy v podstatě odpovídají procesům ve Scrumu, nicméně na větší bázi. Aby velký podnik mohl praktikovat LeSS či Scrum jako takový, musí i přesto dodržet některé zásady, kterým se věnujeme dále v textu.

1.5 Metodiky vývoje SW ve velkých firmách

Ačkoliv firmy nejsou příliš ochotné sdělovat informace o svých interních procesech, i přesto je ale současný trh do určité míry zmapován. V následujících odstavcích budeme vycházet z oficiálního reportu "State of Agile", který každý rok na základě dotazníkového šetření zpracovává společnost Version One Ltd. (poslední report je dostupný z roku 2013 na odkaze <https://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>)

Průzkum probíhal mezi velkými podniky (100 až 4000 zaměstnanců) a většinu dotazovaných tvořily firmy z USA (66 %) a Evropy (20 %). Ačkoliv téměř 90 % dotázaných má zkušenost s agilním přístupem, jen polovina je používá k řízení svých projektů. Zajímavé je ale zjištění, že v případě dobré zkušenosti s tímto přístupem, je agilní vývoj prosazován na většině projektů.

V současnosti tedy agilní a rigorózní metodiky zaujímají na trhu stejnou část, nicméně je zřejmé, že agilní metodiky jsou na vzestupu. Jak studie poukazuje, mezi nejčastěji používanými patří Scrum (55 %), následovaný kombinací Scrumu a XP (11 %) a první příčky uzavírá Kanban (5 %). Na opačném konci je DSDM, Agile Modeling či AgileUp (1 %).

2 Doporučená příprava k přijetí agilních metodik ve velkých korporacích

Za předpokladu, že se vedení projektů velké korporace rozhodne následovat jakoukoli z agilních metodik, je třeba tento proces podstoupit v několika na sebe navazujících a nezbytných krocích. Postup lze popsat poměrně obecně a většina velkých korporací ho může aplikovat i ve svém unikátním prostředí.

Dle článku na webových stránkách StrategieProFirmy (2015) mají agilní metodiky – tedy konkrétně metodika SCRUM výhody i nevýhody, které je třeba důsledně zvážit při zvažování o přechodu na agilní metodiku ve velkém podniku.

SCRUM má své výhody:

- Byznys nemusí dopředu detailně definovat zadání a může doladovat výsledný produkt spolu s IT
- Akceptační testy bývají rychlejší
- Rychlejší time-to-market – nasazení na produkci po částech
- Každodenní přehled o stavu vývoje

SCRUM má také své nevýhody:

- Je to poměrně velká kulturní změna pro IT i byznys (jiné role, jiný styl práce)
- Může být náročnější na plánování a využití zdrojů a tedy dražší
- Pokud není nasazení na produkci po každém sprintu, ztrácí se výhoda rychlejšího time-to-market
- Nesnadné spojení klasické projektové metodologie a SCRUM metodiky, zejména v projektech, kde se integruje více systémů

Ve stejném článku je pak uvedeno, že právě metodika SCRUM byla původně používána zejména při vývoji webových aplikací u většinou malých firem, protože byla pro tyto účely výjimečně vhodná. Nasazení metodiky SCRUM umožnilo zejména rychlejší – často v článku zmiňovaný termín – „time-to-market“, tzn. nasazování po jednotlivých funkčních celcích. Dnes se začíná SCRUM využívat i ve velkých podnicích, kde většině případů *„funguje robustní projektové řízení svázané s poměrně složitým plánováním lidských a finančních zdrojů.“*

Poslední část převzatá z článku na webových stránkách StrategieProFirmy se týká obecného ale také optimálního postupu při nasazování agilní metodiky, ne nutně pouze metodiky SCRUM.

- V prvním kroku nazvaném „Problémy a cíle“ je třeba klást si otázky typu: Jaké současné problémy s IT projekty jsou často řešeny? Nebo také: Jaké jsou měřitelné cíle, kterých je třeba dosáhnout? K tomuto bodu je uvedeno, že „popsání současného stavu a očekávání byznys a IT stakeholderů je velmi důležité pro nalezení optimálního řešení. Pokud např. stávající problémy jsou příliš drahé IT oddělení nebo nefungující spolupráce mezi IT a byznysem, je potřeba detailně analyzovat příčiny.“
- Dále je třeba zvážit všechny alternativy řešení. Jedná se o samotné rozhodnutí jestli SCRUM (nebo jiná agilní metodika) je skutečně vhodnou variantou. „Nasazení SCRUMu není všelék, je potřeba zvážit všechna možná řešení na základě předchozí analýzy. Vylepšení současné projektové metodologie, např. pomocí design testování, nebo sladění IT a byznys projektové metodologie může být lepším řešením než zavedení SCRUMu.“
- Ustanovení projektového týmu s potřebnými dovednostmi je nezbytným krokem číslo tři. „Pokud SCRUM je to pravé řešení, ideální implementace je skrze projekt. Zavedení SCRUMu ovlivní projektovou metodologii. Proto je pro takový transformační projekt důležité vybrat vhodné lidské zdroje, které znají současnou projektovou metodologii a také SCRUM metodiku.“
- Nejen pro dodržení pravidel postupu nasazení zrovna agilní metodiky bude velice užitečné provést predikční analýzu a popis žádaného budoucího stavu s ohledem na procesy, role a systémy. Takový dokument obsahující souhrn zpracovaných a utříděných informací o ideálním budoucím stavu projektu, potažmo projektů v celém podniku, bude možné využít i při rozhodování o projektové strategii. „Je potřeba analyzovat dopady na všechny zainteresované strany, protože SCRUM zavádí procesy a role, které nejsou v běžných waterfall metodikách. Je také potřeba promyslet dopad na projekty, kde nebude implementován SCRUM. Nesmí se opomenout ani dopad na cíle (KPI) liniových i projektových pozic, jejich plánování a hodnocení.“
- Plán přechodu ze současného do budoucího stavu je logickým následujícím krokem. Pečlivé naplánování zavedení agilní metodiky a všechny související změny organizace procesů a procesního workflow je potřeba komunikovat všem zainteresovaným stranám. Pro nové projektové role, které vznikly v důsledku implementace metodiky a nejsou tak ještě zastoupeny žádnými zaměstnanci, je potřeba najít odborníky s vhodnými dovednostmi, případně na takovou pozici vyškolit stávající interní zaměstnance nebo najmout a vyškolit jiné již externě spolupracující osoby.
- Při realizaci přechodu je třeba brát ohled na finance, lidi, rizika a většinou všechny běžící projekty (i ty, kterých by se přechod zdánlivě nemusel týkat). „Je vhodné zvolit pilotní projekt na pilotním systému, kde se vyzkouší nové procesy jak uvnitř SCRUM metodiky, tak v

metodologii projektového portfolia. Po pilotním projektu je potřeba transparentně analyzovat a komunikovat výsledky projektu a doladit dotčené procesy.“

3 Základní pravidla při podnikovém využití agilních metodik

Každý ze dvanácti základních bodů definující agilní metodiku dle agilního manifestu (Agile Manifesto, 2001) je třeba interpretovat s ohledem na rozdílnost velkých a malých společností a zejména na jejich rozdílný přístup k požadovanému způsobu práce na projektech. Dle stanoveného hlavního zdroje (Kot, 2015) pro seminární práci jsou doporučena čtyři hlavní pravidla pro úspěšnou implementaci agilních metodik – konkrétně nejčastěji zmiňované metodiky Scrum – ve velkých korporacích.

3.1 Porozumění konceptu

Neporozumění agilním metodikám zapříčiňuje, že nespočet organizací se agilním metodikám zcela vyhýbá nebo ještě hůř, zavede agilní metodiky nesprávným způsobem. Častým mýtem o agilních metodikách je přesvědčení, že Scrum a další agilní metodiky nejsou schopny poskytnout přesné naplánování celého projektu tak, aby skutečně podporovalo organizační rozhodnutí. Pravdou však je, že implementace agilní metodiky se svým dostatečně včasným plánováním a podrobnou analýzou, podporuje organizační plánování způsobem, který je účinnější než většina ostatních metodik vývoje.

S lehce spravovatelnými a prioritizovanými jednotlivými komponentami spadajícími pod stanovený projektový plán usnadňují agilní praktiky dodržovat termíny s fixními zdroji, takže náklady na projekt jsou předvídatelnější. Za druhé, práce na aplikačních komponentách založená na jejich prioritě spíše než čekání s jejich vývojem až na plánovanou projektovou fázi eliminuje riziko, že bude existovat množství aplikačních komponent, které bude třeba ještě změnit, rozšířit nebo úplně zrušit.

Velkopodnikové pojetí agilních metodik se skládá ze všech částí, které jsou dostupné u klasického pojetí agilních metodik, ale přidává ještě několik dalších elementů, které se ukázaly užitečnými při propojování základních metodických stavebních bloků ve velkých týmech, v rozsáhlém prostředí. Vezměme si například zapojení zákazníků ve větším měřítku - v každém projektovém týmu je od zákazníka požadován různý soubor dovedností, aby komunikace mohla probíhat efektivně. Může být tedy postupem času požadováno vytvoření samostatné funkční jednotky odpovědné za správu zapojení zákazníků.

Dalším příkladem je integrace výstupů produkovaných všemi týmy pracujícími na jednom produktu do jedné potenciálně nasaditelné iterace výsledného produktu. To může být dosaženo několika následujícími způsoby.

- Koordinace toho, jak je v rámci celého podniku definován "hotová část" s cílem definovat kritéria přijatelnosti. Jedná se o míru, s jakou jsme ochotni produkt považovat za finální i s určitým počtem vlastností a vad, které lze nalézt ve výstupech dokončených každým jednotlivým týmem;

- Provádět setkání – tzv. „scrumy scrumů“ za účelem pomoci vyřešit problémy týmů a závislosti jejich prací. Z každého scrum týmu by měl být určen zástupce, aby se účastnil na společném přezkoumání sprintů, které jsou určeny ke zvýšení celkového přehledu o celkovém pokroku.

3.2 Odvykání od starých návyků

Dokonce i jednoznačné pochopení agilních praktik nemusí nutně odradit organizaci od pokusu změnit základní pravidla Scrumu tak, aby byla nějak kompatibilní s dlouhodobě zavedenými vodopádovými návyky spojené s vývojem všech dosavadních produktů. Podniky se nechtějí vzdávat relativně dobře fungujících nebo alespoň fungujících způsobů, jakým řídí své projekty. To je dvojnásobně nebezpečné, protože hrozí, že skončí s tím nejhorším z obou přístupů a bez jediné skutečné výhody.

Hbitost a flexibilita je získána pomocí souvislé analýzy požadavků, návrhu, vývoje, integraci a testování. Navíc hranice mezi všemi těmito aktivitami jsou rozmazané, takže jakýkoli pokus o zavedení tzv. „milestonů“ - milníků završujících stanovené fáze projektů bude nevyhnutelně neúčinná. Přesto se některé organizace spokojí s analýzami a návrhy provedenými na počátku projektu a nadále již neprováděnými, ačkoli by podle agilních metodik měly být tyto fáze opakovány s každou další iterací. Jediné, co provádějí v iteracích je samotný vývoj, ale to určitě nestačí, chtějí-li metodiku opravdu následovat a těžit z jejích výhod. Takto se znovu částečně zavádí vodopádový model vývoje a úsilí na změnu na jinou metodiku je mařeno. Autor zdrojového článku byl sám zapojen do projektu, jehož klient klient se spoléhal na to, že Scrum zvýší hodnotu softwarového produktu, které jeho společnost vyvíjí. Bohužel, společnost se zdráhala pustit formálního schvalovacího postupu pro zákaznické požadavky, který byli projektoví manažeři zvyklí používat několik let. Ačkoli skutečně bylo dosaženo lepší výkonnosti týmu, v důsledku nepřijetí agilních metodik a jejich doporučení v plném rozsahu došlo k výraznému opoždění dodání výstupů. Toto zpoždění se stalo velmi častým jevem u každé dodávky po každém sprintu. Až poté, co firma připustila kýženou flexibilitu nejen při vývoji, ale i úvodních fázích projektu, se jí podařilo zajistit dodávku kvalitních a včasných přírůstků produktu na konci každého sprintu.

Dalším častým zanedbáním je dělba práce na projektu dle uživatelských rolí a nikoli jednotlivých funkčních celků. K tomu dochází při rozdělování požadavků na velké a nové aplikační části produktu, které je třeba provést mezi několika týmy. Lepší metodou by bylo myslet na každou část jako na nezávislou komponentu větší funkčnosti produktu, která je implementována napříč všemi vrstvami aplikační architektury - uživatelským rozhraním, obchodní logikou, datovou vrstvou a databázemi.

3.3 Respektování omezení agilních metodik

Přechod na agilní metodiky rozhodně pomáhá podnikům potenciálně zlepšit jejich efektivitu už na nejnižším hierarchickém stupni. K tomu dochází posílením kvality používaného i vyvíjeného softwaru a posílením vztahu a komunikace se zákazníky. Agilní metodiky také napomáhají organizovat práci, ale to neznamená, že každý člen týmu se náhle stane tím nejlepším odborníkem v příslušných

přirazených oblastech. Agilní metodika a její správná implementace vyžaduje trpělivost vzhledem k úsilí nezbytnému při investování do intenzity týmové spolupráce; multifunkční tým musí běžet vyzkoušet práci na několika pilotních sprintech před tím, než začne pracovat na reálných projektech požadovaným způsobem. Kromě toho budou všichni členové týmu potřebovat naučit zcela nové procesní dovednosti, což bude velice pravděpodobně na začátku tým výrazně zpomalovat.

Stanovení „agilních týmů“ – týmů, pracujících za doporučení agilních metodik, a správných procesů je jen část z nutných změn. Mnoho dalších ještě musí být provedeno tak, aby bylo možné koordinovat a stabilizovat veškeré podnikové operace. Jeden způsob, jak čelit této výzvě je spolupracovat s týmy, které již SCRUM a další agilní metodiky úspěšně implementovali a vyvíjejí tak své produkty. Jako příklad může sloužit zkušenost autora s klientem, jehož organizace se nacházela v procesu konverze na agilním přístup během realizace projektu. Měli stanoveny agilní týmy v rámci celého podniku a těšili se získávání stabilní rychlosti vývoje a předvídatelnými dobami vydávání produktů. Když se projektové procesy synchronizovaly, bylo zjištěno, že společnost měla problémy s řízením projektového backlogu a s plánováním dat vydání. Dobrou zprávou v takové situaci bylo, že samotné zjišťování problému byla ta nejsložitější část. Po pouhých dvou sprintech, se podařilo posílit tým, zlepšit plánování a dát zúčastněným stranám lepší představu o tom, jak mohou vyvíjet své produkty.

Není možné se na agilní metodiky vývoje spoléhat jako na spásu. Manažeři a projektoví vedoucí nastaví celkové projektové prostředí a budou hrát klíčovou úlohu při formování, rozšiřování a zlepšování členů týmu prostřednictvím vyrovnávání tradičních metod řízení systémů a dalších inovativních metod řízení „sebeorganizovaných“ systémů. Pokud jde ovšem o dohled nad samospravujícími multifunkčními týmy, tradiční svět projektových manažerů se obrátil vzhůru nohama. Oni jsou nyní vystaveni tlaku učít se novou rozpočtovou politiku projektu, zaměstnávání, tvoření harmonogramu projektu a koordinaci asistence a podpory z jiných oddělení. Ve scrum týmu sám rozhoduje o rozsahu závazků, které se tým rozhodne splnit po každém sprintu. Jediná práce zbývající pro manažery je zajišťování transparentnosti procesů s cílem zajistit, aby úkoly nebyly plněny s nadměrným či nedostatečným nasazením svých určených vykonavatelů.

3.4 Lidé na prvním místě

Scrum není metoda vývoje, ale spíše rámec, který má za úkol pomoci organizovat práci a zvýšit efektivitu týmu. Scrum týmy jsou organizovány tak, aby umožňovaly členům urychlení procesů a dosažení jejich největší produktivity. Existuje mnoho faktorů, které je třeba vzájemně kombinovat, aby se tak mohlo skutečně stát.

- Základní organizace práce

Sto procent dedikace konkrétnímu týmu, nepřetržitý a ničím nerušený vývoj během sprintu a trvale udržitelný růst tempa bez jakýchkoli nucených přesčasů může zabránit ztrátám v zaměstnanecké efektivitě práce, kvalitě a celkové produktivitě.

- Dostupnost nástrojů

Síla Scrumu pochází ze spolupráce mezi členy týmu, což platí i pro spolupráci na dálku a spolupráci na projektu ve velkém měřítku. Vzniká tak potřeba takových nástrojů, které jsou vhodné pro práci mezi týmy a například také podporují sledování změn v souborech v reálném čase. Existuje také mnoho řešení určených pro využití mezi členy z různých zemí a kontinentů a také produktové manažery a týmy dohromady – mimo jiné například aplikace pro instant messaging, wiki servery a vzdálená úložiště dokumentů a další softwarové možnosti, jak sdílet obrazovky a organizovat videokonference.

- Firemní vzdělávání

Holé nadšení a odhodlání k transformaci nestačí k tomu, aby vývojové týmy a manažeři mohli efektivně pracovat v agilním prostředí vývoje. Aby transformace byla možná, je třeba stanovit vysoké standardy implementovatelné napříč celou organizací. Tyto standardy by měly být rozvíjeny pevnou politikou a firemním vzděláváním by mělo poskytovat počáteční a průběžnou odbornou přípravu pro takový rozvoj.

- Technická dokonalost

Zaměření se na dosažení optimální rychlosti práce v týmu a rychlou dodávku požadované funkčnosti nesmí ani v nejmenším kompromitovat kvalitu dodávaného kódu. Je třeba se ujistit, že všichni členové Scrum týmu ovládají, nebo přinejmenším jsou dobře seznámeni s takovými technickými postupy a nástroji, jako je test-driven development (TDD), statická analýza kódu, kontinuální integrace a automatizované testování.

Manifest agilního softwarového vývoje začíná prohlášením o lepších způsobech, jak vyvíjet software s upřednostňovanými jednotlivci a interakcemi před procesy a nástroji. V tomto ohledu je interakce napříč vývojovými týmy a mezi členy týmu aspekt, na který je nezbytné brát ohled. Ale to platí také pro posunutí interakce se zákazníky na zcela novou úroveň. Intenzivnější zapojení zákazníka vyžaduje, aby se projektový tým dobře sladil se zákaznickým postojem a aby plně respektoval to, co zákazník skutečně požaduje. V případě, že zákazník vykazuje známky frustrace z vysoké míry požadované interakce, tým může například změnit styl komunikace na diskusi formou pravidelných interview v zájmu zachování vysoké intenzity komunikace a potřebného budoucího zapojení zákazníků.

4 Příklady využití agilního vývoje u velkých projektů

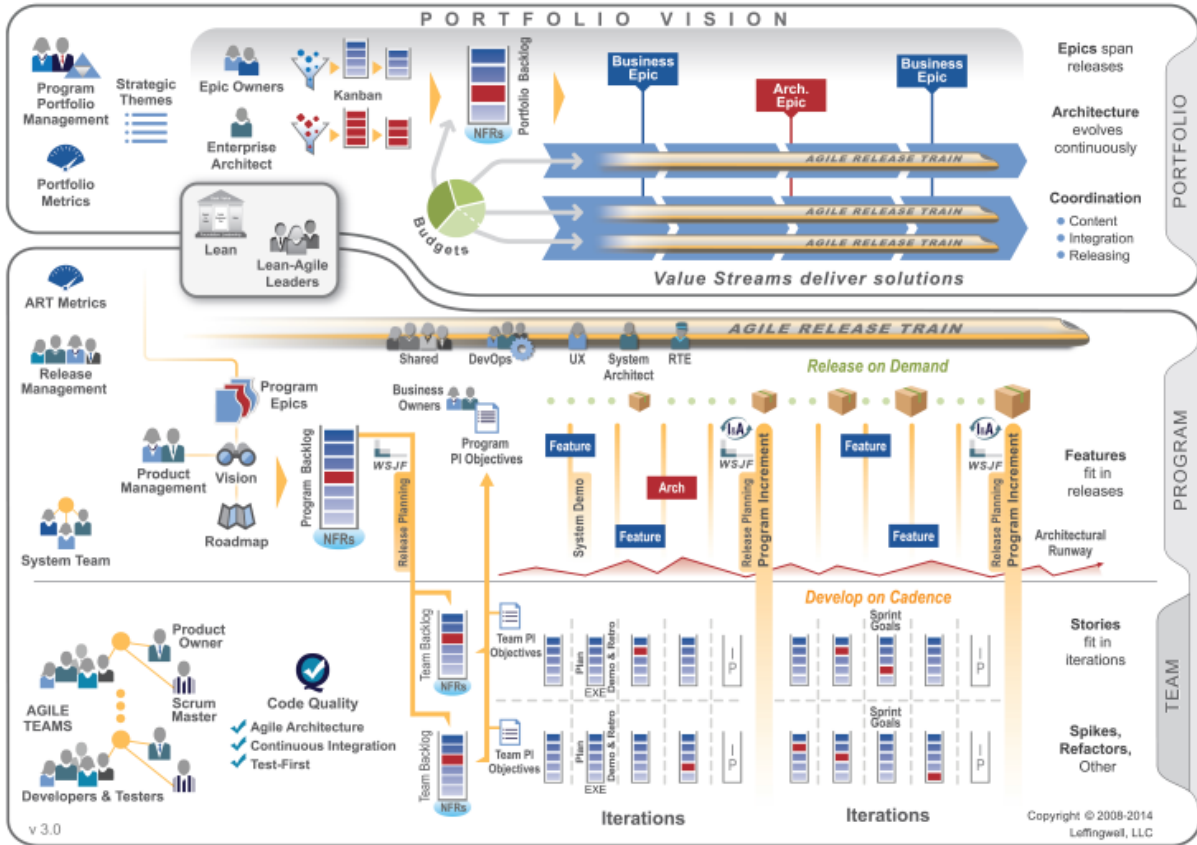
S velkým projektem vedeným pod hlavičkou velkého podniku se pojí mnoho skupin. Čím je větší podnik a projekt, tím více je takových zainteresovaných stran. Dame Dave (Dave, 2014) ve svém článku uvádí, že při implementaci agilních metodik je třeba do procesů s ní spojených zahrnout všechny smluvní strany - dodavatele, obchodní partnery, zákazníky, externě spolupracující firmy – tedy ty strany, které bude mít výsledek projektu jakýkoli dopad. Ačkoli je obtížnost implementace s velikostí podniku skoro přímo úměrná, některé z velkých společností uspěly a dokázaly využít výhod agilního vývoje, ať už díky jen částečné nebo postupné a opatrné implementaci nové metodiky nebo velkými strukturálními změnami a zásadními interními investicemi.

Například společnost IBM se podle (Frye, 2012) s agilními metodikami povedlo dosáhnout velkých úspěchů, když se stala jednou z těch nejrychlejších velkých korporací pokoušejících se adoptovat nové způsoby vývoje a řízení. V roce 2012 tvrdil viceprezident společnosti, že klíčovými faktory úspěchu jsou procesy, nástroje a ukázněnost. Také ovšem přiznal, že je samozřejmě potřeba definované agilní principy jemně modifikovat – přizpůsobit velké společnosti -a není možné doporučení vyplývající plnit bez výhrad a bez kompromisů.

Dobrým příkladem uvedeným ve zdroji je společnost HP. Tam také nedochází ke stoprocentnímu osvojení metodik, a tam, kde se o to pokouší, tam je třeba přistoupit k tak zvanému „škálování“ agilního přístupu – tedy právě přizpůsobování a vybírání konkrétních doporučení, které se nejlépe hodí pro velký projekt a mnoho týmů s mnoha členy. Některé z procesů jsou však řízeny agilním způsobem bez modifikací, například u procesu uvádění produktů na trh – jde v zásadě o menší projekty, kde je členům týmu umožněna úzká spolupráce.

Mezi další společnosti, které se k agilnímu přístupu hrdě hlásí a přispívají k jeho rozvoji a osvojování dalšími společnostmi patří například Microsoft. Dochází k velké podpoře i prostřednictvím aplikací této společnosti, jako je vývojové prostředí Visual Studio, které je uzpůsobené odlišnému workflow. Obrázek ilustruje aplikaci podporovaný framework s názvem „Scaled Agile Framework“.

Scaled Agile Framework



Obrázek 3 Scaled Agile Framework. Zdroj: <https://msdn.microsoft.com/en-us/library/vs/alm/work/scale/scaled-agile-framework>

Závěr

Agilní přístup slibuje mnohá zlepšení ať už v efektivitě, plynulosti workflow, rychlosti dodávání uspokojivých výsledků nebo v samotné vysoké kvalitě výsledného produktu, není proto divu, že se i velké korporace začaly pokoušet si doporučené agilní postupy a techniky osvojit bez větších zásahů do hlavních principů této metodiky (Agile Manifesto, 2001). Začíná být zřejmé, že možnosti implementace principů agilních metod vývoje ve velkých podnicích, nejsou tak omezené, jak by se na první pohled mohlo zdát.

Nejen dle zdrojového článku (Kot, 2015), ale i ostatních (Frye, 2012) v případě, že není možné následovat všechna metodická doporučení, vznikají ovšem hybridní metodiky vývoje softwaru, nebo jen jejich částečné nasazení, ale v takových podmínkách již není a nemohou být zajištěny – z hlediska agilního přístupu – optimální postupy. Nejen postupy, ale ani procesy svou strukturou, délkou trvání a celkovou organizací nemusí odpovídat metodickým požadavkům, čímž se nejen porušují základní principy, ale dochází i k odklonu od původní myšlenky. Pak ani výsledky nemusí odpovídat těm požadovaným a očekávaným, avšak takové následky pak nelze klást za vinu metodice samotné.

S agilními metodikami ve velkých podnicích je třeba pracovat s obezřetností a nevyhnutelné překážky v procesu implementace překonávat s rozvahou – nalézat nikoli nabízející se kompromisy, ale optimální řešení plné implementace principu v celé šíři podnikového projektu. Proto ne každý velký podnik implementuje agilní metodiky s úspěchem, k dosažení kýžených výsledků je obvykle třeba přechodem na jiný a zpočátku zdánlivě nekompatibilní způsob vývoje softwaru pověřit takové odborníky, kteří mají zkušenosti právě s touto oblastí a byli součástí minimálně několika úspěšných projektů, u nichž bylo prokazatelně dosaženo úspěchů díky kompletně a optimálně adoptované metodice – a to po celou dobu trvání projektu, od úvodní analýzy až po dodání produktu.

Lze se tak domnívat, že s dostatečnou finanční investicí, velmi dobře zpracovanou přípravou a zapojením celého projektového týmu (a často i dalších projektových týmů, u kterých k přechodu na agilní metodiku ani docházet nemusí) je pak ve většině velkých podnicích možné přechod na agilní metodiku realizovat tak, aby změna skutečně přinesla četné výhody, které tato metodika nabízí. Cíl práce se tak podařilo naplnit – pokud chce podnik postupovat agilně, je nezbytné, aby dodržel tato základní kritéria.

Seznam použitých zdrojů

- BUCHALCEVOVÁ, Alena. 2005. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada. Management v informační společnosti. ISBN 80-247-1075-7.
- BUCHALCEVOVÁ, Alena. 2009. *Metodiky budování informačních systémů*. Praha: Oeconomica. ISBN 978-80-245-1540-3.
- DAME, Dave. Agile in the Enterprise – How to Act Locally and Develop Globally. In: Product Lifecycle Report [online]. 2014 [cit. 2016-05-10]. Dostupné z: <http://blogs.ptc.com/2014/10/30/agile-in-the-enterprise-how-to-act-locally-and-develop-globally/>
- FRYE, Colleen. Implementing Agile in very large enterprises. In: SearchSoftwareQuality [online]. 2012 [cit. 2016-05-10]. Dostupné z: <http://searchsoftwarequality.techtarget.com/feature/Implementing-Agile-in-very-large-enterprises>
- CHLAPEK, Dušan, Václav ŘEPA a Iva STANOVSKÁ. 2011. *Analýza a návrh informačních systémů*. Praha: Oeconomica. ISBN 978-80-245-1782-7.
- Jak implementovat SCRUM ve firmě se zavedeným projektovým řízením. In: StrategieProFirmy [online]. 2015 [cit. 2016-05-09]. Dostupné z: <http://www.strategieprofirmy.cz/clanky/jak-implementovat-scrum-ve-firme-se-zavedenym-projektovym-rozenim.htm>
- KOT, Ivan. Four Must-Have Rules for Scaling Enterprise Agile. In: InfoQueue [online]. 2015 [cit. 2016-05-09]. Dostupné z: <http://www.infoq.com/articles/rules-scaling-agile>
- MYSLÍN, Josef. 2016. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press. ISBN 978-80-251-4650-7.
- Principles behind the Agile Manifesto. Agile Manifesto [online]. [cit. 2016-05-12]. Dostupné z: <http://agilemanifesto.org/principles.html>

Seznam objektů

Obrázek 1 Srovnání metodik.	3
Obrázek 1 Srovnání metodik.	3
Obrázek 2 Scaled Agile Framework.....	14
Tabulka 1 Srovnání agilních a rigorózních metodik.	4