

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	ZS 2017/18
Autoři	Marek Neufus (neum00), Anatolii Zhiliaev (zhia01), Martin Kadeřábek (kadm01)
Téma	Agile Scaling in Action
Datum odevzdání	7.12.2017

Abstrakt

Práce se věnuje tématu agilního škálování, přičemž čerpá z článku, který popisuje postup agilního škálování šesti týmů z nejmenované velké firmy. Pozornost se věnuje zvoleným postupům a problémům, které v průběhu škálování vznikaly. Práce nejprve představuje podkladový článek, ze kterého čerpá. Následně připomíná téma agilního způsobu řízení týmu v obecné rovině. V dalších kapitolách se pozornost věnuje jednotlivým metodám vedoucím ke škálování agilních týmů a jejich případným rizikům. Na závěr je vyhodnocen výsledek škálování ze zdrojového článku.

Klíčová slova

Agile, agilní frameworky, agilní škálování, tým, komunikace, Scrum of scrum

Obsah

Obsah	1
1 Úvod	2
1.1 Cíl práce.....	2
1.2 Proč je vhodné se tématu věnovat	2
1.3 Přístup zvolený k dosažení cíle	3
2 Představení podkladového textu	4
3 Úvod do Agile	5
3.1 Co je to Agile	5
3.2 Agilní frameworky (metodiky).....	5
3.2.1 Extrémní programování (XP).....	6
3.2.2 Lean	6
3.2.3 Kanban.....	6
3.2.4 SFAe	6
3.2.5 Scrum.....	7
4 Možné rozdíly práce týmů se stejnou agilní metodikou	8
4.1 Způsoby spolupráce těchto týmů.....	9
5 Prostředky pro spolupráci agilních týmů	10
5.1 Product council forum.....	10
5.2 Scrum of Scrums	10
5.3 Lean coffee session.....	11
6 Softwarová podpora týmů	12
6.1 Ticketovací systém	12
6.2 Software pro správu agilních projektů.....	13
7 Vyhodnocení a přínos integrace agilních týmů	14
7.1 Monitoring projektu	14
7.2 Proč to nešlo podle představ?.....	15
8 Závěr	17
Použité informační zdroje	18

1 Úvod

Přestože byly agilní metodiky původně určeny pro malé a střední projekty, tak se postupem času, díky jejich popularitě a výsledkům, začaly využívat i ve velkých projektech. Pro tyto situace však nebyly agilní metodiky určeny, proto se využívá jejich škálování, které umožní jejich využití i v prostředí velkých týmů, které pracují na komplexním projektu.

Postup řešení takové výzvy, tedy využít agilního škálování ve velké společnosti, která chce plně využít výhod agilních metodik, se věnuje autor podkladového článku k této práci. Podkladovému článku se dále věnuje kapitola 2.

Kapitola 3 věnuje pozornost pojmu Agile, přičemž se snaží poskytnout úvod do problematiky agilního vývoje softwaru a charakterizovat některé agilní metodiky.

Další kapitoly se zaměřují na problematiku řešenou v podkladovém článku. Tedy ty oblasti a problémy, kterým autor ve svém článku věnoval větší pozornost.

Kapitola 7 vyhodnocuje výsledky a přínosy, kterých zkoumaná společnost při škálování svých vývojových týmů dosáhla.

1.1 Cíl práce

- Identifikace problémů a rizik při spolupráci agilně řízených týmů.

1.2 Proč je vhodné se tématu věnovat

Téma této semestrální práce má hluboký přesah do praxe firem, kde jsou k řízení a vývoji projektů využívány agilní metodiky. V případě větších projektů, bývá běžná spolupráce více týmů na jednom projektu. A jelikož agilní metodiky představují pouze určitý rámec, týmy si jej obvykle adaptují po svém tak, aby vyhovovaly přesně jejich potřebám. Na základě toho se od sebe různé týmy (i v rámci jedné firmy) odlišují v řadě vlastností jako jsou například kultura uvnitř týmu, způsob spolupráce uvnitř týmu, způsob motivací a odměn v týmu, způsob řízení týmů a jiné. Jakmile mají týmy začít spolupracovat na společném projektu, vyplývají všechny tyto odlišnosti na povrch a je nezbytné je začít řešit. V opačném případě hrozí vysoké riziko, že projekt nebude dokončen úspěšně.

1.3 Přístup zvolený k dosažení cíle

K dosažení stanoveného cíle bylo nejprve přistoupeno tak, že byl čtenář seznámen s danou problematikou obecně a na základě obecného seznámení bylo umožněno jednoznačně vymezit konkrétní předmět této práce. Práce se následně věnovala už samotnému jádru problematiky, která byla naznačena v článku Agile Scaling in Action od Yousefa Awada. Jádrem problematiky byly především komplikace, které se během vývoje projektu vyskytly při spolupráci různých, agilně řízených týmů. Komplikace vyplývající z podkladového článku byly komparovány s informacemi z jiných veřejně dostupných zdrojů. K takto popsaným komplikacím bylo následně vždy nalezeno možné východisko.

2 Představení podkladového textu

Podkladový článek, který je hlavním zdrojem inspirace pro tuto práci, se nazývá „Agile scaling in action“, v překladu agilní škálování v akci. Autor tohoto článku, Yousef Awad, je generálním ředitelem ve společnosti Integrant. Současně má za sebou úspěšnou kariérou v programování, administraci databází a řízení projektů. Firma Integrant zaměstnává více než 130 zaměstnanců a poskytuje svým zákazníkům týmy pro vývoj a testování softwaru na platformě .NET.

Hlavní inspirací pro tento článek byla účast společnosti Integrant na projektu v prostředí velké společnosti z oblasti E-commerce, jejíž roční příjmy se pohybují ve výši 500 milionů dolarů. V tomto projektu bylo třeba využít agilní škálování, právě kvůli rozsáhlému prostředí, k přizpůsobení agilního vývoje, který probíhal pomocí agilní metodiky scrum. Agilní škálování neprobíhalo podle nějakého předepsaného postupu, nebo frameworku, ale spíše podle dostupných možností, best practices a přijetím toho, co pro konkrétní situaci fungovalo.

V dané společnosti bylo šest agilních týmů, které pracovali nezávisle na ostatních na podpoře stejného projektu. Zmíněná nezávislost mezi jednotlivými týmy znamená, že mezi sebou neměli žádné jasné propojení komunikace a spolupráce. Každý tým byl sám o sobě dobře agilně řízen, měl velmi dobrou komunikaci, spolupráci a dokázal včas reagovat na změny priorit. Ukázalo se, že náklady spojené s tímto stavem převažují problémy spojené s propojením těchto týmů pomocí agilního škálování. Jelikož jednotlivé týmy nebyli navzájem agilní, tak se ztrácelo mnoho výhod agilního prostředí v nesprávné komunikaci a neoptimalizované vzájemné závislosti. Projekt byl proto neuspokojivý s téměř nulovými hmatatelnými rysy. Pozornost se proto obrátila na agilní škálování týmů, díky čemuž by projekt mohl získat novou hybnou sílu.

Tato společnost se tedy rozhodla svých šest individuálních, navzájem nezávislých týmů propojit do jednoho týmu. S tím, že jejich obecný přístup pro agilní škálování byl jasný – definovat krátké iterace, které přidají hmatatelnou hodnotu do projektu jako celku. Současně se rozhodli oslovit flexibilní členy týmu, kteří jsou ochotni využít své různorodé dovednosti v závislosti na sprintu, dovednostních mezerách ostatních a vztazích v týmu. Záměrně nevyužili konkrétní framework nebo vzor pro agilní škálování. Rozhodli se využít poněkud iterativnější přístup, tedy upravovat tým, schůzky a přístup podle zpětné vazby týmu a analýze dat. Souběžně s tímto škálováním probíhala práce původních oddělených týmu běžným způsobem, pokračovali každodenní stand-up schůzky a stále měli svého scrum mastera. Hlavním cílem škálování bylo dosažení toho, aby jednotlivé týmy pracovali ve shodě s jedním projektem. (Awad, 2017)

3 Úvod do Agile

V dodaném článku popisuje, jak je možné ve firmě zavést používání Agile škálování v podnikání. Také o obtížích, kterým musela při implementaci čelit a jaké změny toto firmě přineslo ve finále. Dříve, než bude provedena analýza, je třeba se věnovat slovu „agile“.

3.1 Co je to Agile

Překlad z angličtiny, „agilní“ znamená „živý, svižný“, ale překládá se častěji jako „flexibilní“ (KARPOV, 2011). V oblasti vývoje softwaru se tento termín objevil na počátku roku 2000, kdy v Utahu vyšla publikace „Manifest flexibilního vývojového softwaru“. Agile je zrychlující metodika pro vytváření projektů. Podstata agilního přístupu je uvedena v "manifestu" (Agile Manifesto authors, 2001), ale může být stručně formulována takto:

- Vývoj je prováděn v krátkých cyklech (iteracích) trvajících 1-4 týdnů;
- Na konci každé iterace dostane zákazník aplikaci (nebo její část), která může být použita v podnikání;
- Vývojový tým spolupracuje se zákazníkem v průběhu celého projektu;
- Změny v projektu jsou vítány a rychle zahrnuty do práce.

V současné době se agilní principy používají v práci desítek tisíc týmů po celém světě.

3.2 Agilní frameworky (metodiky)

Agile je soubor filozofických hodnot, ale je obtížné je používat v životě. Ne každý tým může zítra začít pracovat bez šéfa. Není jasné, jak je možné projekt provádět bez podrobného technického zadání. Ne každý klient bude souhlasit, že půjde každý den do vývojové kanceláře nebo zavolá několikrát denně. Jak a kde začít být „agile“ není jasné. Aplikovat filozofii v praxi je náročné a používají se k tomu určité postupy (rámce). Mezi nejznámější patří SAFe, FDD, Scrum, Lean, Scrumban, Kanban, Continuous Delivery a mnoho dalších.

Framework (také technika) je soubor pravidel, akční šablona pro přenesení filozofie „agile“ do života. Jsou to různé stupně konkrétnosti. Například v Kanbanu existuje šest obecných pravidel a Scrum popisuje role, schůzky a artefakty. Sady pravidel lze rozšířit a přizpůsobit, především je nutné dodržovat postulace a principy „agile“.

Chceme-li získat obecnou představu, zde jsou hlavní rysy a nevýhody několika populárních rámců z knihy "Agile Software Development Methods: Review and Analysis" (ABRAHAMSSON, 2002).

3.2.1 Extrémní programování (XP)

- Klíčové body: Malé týmy, denní setkání, neformální komunikace v týmu, minimální dokumentace
- Jedinečné vlastnosti: neustálé přizpůsobení projektu ke zlepšení jeho účinnosti a přizpůsobení se změnám
- Nevýhody: Je vhodnější pro individuální praktiky než pro globální řízení, neboť v posledně uvedeném případě existuje riziko vzniku nedisciplinovaných týmů

3.2.2 Lean

- Klíčové body: Použití vizualizačních nástrojů, vývoj prostřednictvím testování, krátké iterace
- Jedinečné funkce: Hlavní jsou softwarové funkce, které jsou pro zákazníka cenné, je to pro tým hodně motivační
- Nevýhody: Rozhodování jsou přijímána dlouho, nedostatek disciplíny, jako u malých projektů

3.2.3 Kanban

- Klíčové body: Samoorganizující tým, ztracený význam pojmu „iterace“ – vše zaměřeno na úkoly
- Jedinečné vlastnosti: Není lhůta pro implementaci, ale je tam limit na počet „pracujících v tuto chvíli,“ je to efektivní, pokud není známo, co se může v budoucnu žádat zákazník
- Nevýhody: nedostatek disciplíny, zdlouhavá příroda

3.2.4 SFAe

Klíčovou výhodou pro velké společnosti je mít jednoduchou a transparentní organizační strukturu v tandemu s jasným rozdělením odpovědnostního modelu. Scrum, extrémní programování a další Agilní metody vývoje, tradičně nepřekročují hranice týmové úrovně. Nicméně SFAe poskytuje jednotný pohled na vykonávanou práci z pohledu vedení společnosti, což umožňuje ponořit se do podrobností podle potřeby, analyzovat a identifikovat problémy a riziká. SFAe se skládá ze tří úrovní: tým, program a portfolio.

SFAe tým se může skládat z 8-10 osob napříč funkcemi, což znamená, že má všechny kompetence potřebné pro vývoj softwaru, od shromažďování požadavků až po realizaci. V rámci jednoho programu spolupracuje několik týmů, které společně tvoří program jako

celek. Zvláštní položka rozpočtu organizace odpovídá tomuto projektu. Pro ředitele firmy je to malý projekt, o kterém lze diskutovat samostatně. V rámci portfolia SAFe se rozumí úplný soubor programů, na které je používán celý rozpočet organizace, který je přidelen na vývoj softwaru. Podle SFAe se doporučuje vytvořit jednotný úřad pro správu portfolia, zodpovědný za strategii rozvoje, investic a tvorby rozpočtu. (MALCOLM)

3.2.5 Scrum

Jelikož v článku se píše ze spolupracující týmy pracují podle Scrumu, bude níže detailněji popsána. Je to nejpobulárnější „Agile“ metodika široce používaná společnostmi, jako je GE, Nike, PayPal, Yahoo, SAP, Google. V klasickém Scrumu existují tři základní role():

- Vlastník produktu (PO) je spojení mezi vývojovým týmem a zákazníkem. Cílem PO je maximalizovat hodnotu vyvíjeného produktu a práci týmu. Jedním z hlavních nástrojů PO je Product Backlog. Produkt Backlog obsahuje potřebné pracovní úkoly (například Story, Bug, Task atd.), Seřazené podle pořadí priority (naléhavost).
- Scrum master (SM) je „vedoucí týmu“. Úkolem Scrum Master je pomoci týmu maximalizovat jeho efektivitu tím, že eliminuje překážky, pomáhá, trénuje a motivuje tým, pomáhá PO.
- Vývojový tým se skládá ze specialistů, kteří přímo pracují na produktu.

Základem Scrum je Sprint, během kterého se provádí práce na výrobku. Na konci Sprintu je třeba získat novou pracovní verzi produktu. Sprint je vždy omezen v čase (1-4 týdny) a má stejné trvání po celou dobu životnosti produktu (KNESL, 2016). Před začátkem každého Sprintu se provádí Sprint Planning, během kterého se prioritizují úlohy a tvoří se Backlog Product. Spring Backlog obsahuje úkoly, které mají být provedeny v aktuálním Sprinu. Každý Sprint musí mít cíl, který je motivujícím faktorem a dosažení kterého je možné pomocí splnění úkolu ze Spring Backlogu.

Každý den se provádí Daily Scrum, kde každý člen týmu odpovídá na otázky:

- Co jsem udělal včera?
- Co mám v plánu udělat dnes?
- Jaké překážky mne omezují v práci?

Na konci Sprintu také se provádí Sprint Review a Sprint Retrospective, jejíž úkolem je posoudit týmovou účinnost (produktivitu) v uplynulém Sprintu, a úkolem je také předvídat očekávané výkonnosti (produktivitu) v příštím Sprintu, identifikovat stávající problémy, zhodnotit pravděpodobnost dokončení všech nezbytných prací na výrobku a další.

4 Možné rozdíly práce týmů se stejnou agilní metodikou

Týmový vývoj má v agilních metodikách zcela zásadní význam. Zatímco v tradičním vývoji je předem dáno, jak má vývoj probíhat, a pracovníci, případně týmy podle popsaných procesů postupují, v agilním vývoji nic předem dané není. Tým je specifickým druhem pracovní skupiny. Je definován jako „vnitřně formálně nestrukturovaná malá skupina lidí, kteří v jejím rámci podávají po stanovenou dobu společný výkon“. Rozdíl mezi pracovní skupinou a týmem je takový, že:

- v týmu neexistuje jasně stanovená vnitřní formální organizační struktura, samozřejmě existují výjimky, jako je například role scrum mastera.
- členové týmu společně hledají řešení, společně rozhodují, podávají společný výkon a mají tedy také společnou odpovědnost za výsledky.
- existence týmu je časově omezena. Jedná se o organizaci, která zahrnuje vlastní složitost danou jednotlivými členy a jejich vztahy, ale působí i jako samostatný celek. (Čížek, 2009)

Rozdíly v práci různých týmů se stejnou agilní metodikou určitě existují a to z různých důvodů. Vliv na to může mít různorodost využívaného softwaru pro podporu agilního týmu. Dále práci v týmu ovlivňuje kultura a prostředí, ve kterém jednotlivé týmy pracují, například v nadnárodních společnostech, které mají své agilní vývojové týmy rozmístěny v různých zemích světa. Protože je v agilním přístupu k vývoji softwaru soustředěna hlavní pozornost na spolupráci a komunikaci jednotlivých členů v týmu mezi sebou, tak práci týmu jako celku ovlivňují především osobnosti jednotlivých členů tohoto týmu. Samotní členové týmu však ovlivňují práci týmu i v dalších aspektech, jelikož jim agilní metodiky dávají značnou volnost, aby byl tým jako celek schopen řešit komplexní a těžko strukturovatelné zadání. Posledním, zde zmíněným faktorem, který může mít vliv na práci týmu je zákazník, nebo jeho zástupce. Zákazník může mít svého zástupce přímo v agilním vývojovém týmu, a pokud tomu tak není, tak agilní tým musí vyřešit jinou cestou způsob komunikace se zákazníkem, jako to má například nastavené metodiky scrum, v níž je definována role Product Owner, která představuje hlas zákazníka.

Kultura a prostředí v jakém tým působí je úzce spjata i s dalším vlivovým faktorem, kterým je osobnost každého člena týmu. Tuto osobnost ovlivňuje a utváří právě kultura a prostředí v jakém pracoval a žil, životní i pracovní zkušenosti, motivace pracovního jednání aj.

(Čížek, 2009)

4.1 Způsoby spolupráce těchto týmů

Všechny výše zmíněné vlivy a rozdíly v práci samostatných týmů komplikují nebo ztěžují možnosti jejich vzájemné spolupráce.

Ve zkoumaném článku využili roli agilního kouče ke zlepšení spolupráce mezi jednotlivými týmy. V této roli agilního kouče se angažovala externí osoba, ale obecně lze využít i interní osobu v závislosti na schopnostech a dovednostech. V tomto případě agilní kouč inspiroval firemní agilní kulturu, povzbudil výkon týmů a vedl týmy, vedení a další zainteresované osoby k harmonizaci. Jeho role byla klíčová při změně myšlení členů týmů ze starého k novému.

5 Prostředky pro spolupráci agilních týmů

V textu je uvedeno, že pokud každý tým pracuje na projektu s použitím agilní metody, neznamená to, že celý projekt je také považován za agilní. Práce v rámci týmů je organizována dobře a harmonicky, avšak interakce mezi jednotlivými týmy může být velmi problematická. K tomu, aby celý projekt byl považován za agilní, je třeba, aby každý tým používal agilní metody, nástroje a frameworky. Tak může vzniknout jeden velký agilní projekt. To umožňuje flexibilně škálovat projekt a správně určovat priority, integrovat zdroje různých týmů a přidělovat odpovědnost vůči každému konkrétnímu týmu pro projekt jako celek. K docílení tohoto efektu se podle článku využívá několik zajímavých metod, které jsou popsány níže.

5.1 Product council forum

Dříve měl každý tým svůj určitý počet lidí a úkoly týmu měly přednost. Aby všechny týmy spolupracovaly a usilovaly o stejné cíle (které jsou v současné době relevantní pro daný projekt) bylo rozhodnuto, aby všechny cíle týmů v rámci projektu byly soustředěny v jeden společný backlog. K dosažení tohoto byla shromážděna rada (Product council forum), které se setkalo každý den po dobu jedné hodiny. Hlavním účelem tohoto setkání bylo zjistit a probrat úkoly projektu, stejně jako přerozdělování lidských zdrojů mezi týmy v době, kdy to bylo nutné.

5.2 Scrum of Scrums

Zatímco Product council forum řešilo rozložení úkolů a priorit na úrovni projektu, bylo navíc třeba zajistit efektivitu vzájemného působení. Za tímto účelem byl použit „scrum of scrums“. Scrumu of Scrumu – je to setkání několika Scrum-mastrů, které se koná každé dva dny poté, co proběhly denní schůze (OZKAYA, 2013). To umožňuje několika Scrum-mastrům více společně diskutovat o práci, více se zaměřit na oblasti konvergence a vzájemná integrace, diskutovat o budoucích výzvách, které blokují rozvoj, jakož i případné problémy mezi jednotlivými týmy, anebo i v rámci týmů (LINDERS, 2014). Na Scrum of Scrums jsou kladeny následující otázky:

1. Co udělal každý tým od předchozího setkání?
2. Co udělá každý tým na příští schůzi?
3. Existují nějaké problémy, které narušují nebo zpomalují práci každého týmu?
4. Potřebujete pomoc nějakého jiného týmu, aby udělal nějaký váš úkol?

5.3 Lean coffee session

Při zavedení Product council forum a Scrum of Scrums se našla, dle očekávání, skupina lidí, kteří byli nespokojeni se změnou práce, zvýšení počtu shromáždění, setkání a byrokracií. Znamená to, že ti lidé nebyli plně zapojeni do procesu, a proto nemohli pochopit všechny přínosy inovací. Spíše, než podlehnout jejich názoru bylo rozhodnuto nejen nadále pohybovat se v rámci záměru, ale také zvýšit počet schůzek přidáním „lean coffee relace“ a „agile coffee sessions“. Tato setkání byla vytvořena k řešení problémů na individuální úrovni. Diskuse se může účastnit každý, kdo chce, z jakéhokoliv týmu a jakéhokoli pozice. Na těchto setkáních se diskutuje o problémech, otázkách a námětech. Cílem bylo zdůraznit, že pro projekt jsou důležití především lidé a spokojení zaměstnanci budou vždy dobře pracovat.

6 Softwarová podpora týmů

Softwarové nástroje pro podporu agilních týmů mají důležitou roli v celém agilním vývoji. Pomáhají rozdělovat jednotlivé úkoly a pravomoci, aktivně reportovat, shromažďovat data nebo koordinovat informační toky mezi týmy. Může se jednat pouze o nástroje od jednoduchých ticketovacích systémů až po komplexní software pro správu agilních projektů.

6.1 Ticketovací systém

Ticketovací systém je software pro podporu vývoje, vztahů se zákazníky a kooperace uvnitř týmu. V dnešní době se ustupuje od klasických desktopových klientů ticketovacího systému k webovým rozhraním, která nemusí být instalována na uživatelská zařízení a jejich aktualizace a správa probíhá centrálně na serveru. Výhodou webových ticketovacích rozhraní je také jejich dostupnost odkudkoli kde je internetové připojení. Aplikace typicky pracuje tak, že požadavek od zákazníka je přiřazen konkrétnímu členu týmu a ten tento požadavek řeší, přiřazuje jiným členům, anebo ho komunikuje s jeho autorem. Vyřešené požadavky jsou uchovávány v databázi aplikace pro možnou budoucí kontrolu.

Požadavky mohou být různého typu, ticketovací systém obvykle nechává rozlišení na konfiguraci aplikace. Požadavek může oznamovat chybu, být přáním o rozšíření funkcionality nebo se může jednat například o reklamaci zboží. Ticketovací systémy jsou v tomto ohledu mnohem univerzálnější než rozšířené Bugtrackery.

Aplikací tohoto typu je celá řada z různým rozsahem využití. Následuje výpis čtyř vybraných nástrojů s krátkým popisem.

- Bugzilla - open source webová aplikace pro správu chyb původně vyvinutá a používaná organizací Mozilla.
- MantisBT - open-source řešení, který se již svým návrhem tolik neomezuje pouze na správu chyb a je tak univerzálnější než Bugzilla.
- Microsoft Dynamics - komerční řešení od společnosti Microsoft, také je koncipováno jako webová aplikace.
- Redmine - open source software pro řízení projektů a bug tracking systém.

(Coufal, 2012)

6.2 Software pro správu agilních projektů

Software pro správu agilních projektů se díky stále větší oblíbenosti agilních metodik stává stále konkurenceschopnějším. Pro takový software jsou klíčové tři dále popsané body, které jsou společné pro všechny agilní týmy

- Agilní přehledy a metriky zahrnují sledování a kontrolování času, snadno srozumitelné správy o pokroku pro všechny účastníky, kontrolování kvality a procento dokončení projektu
- Komunikace s lokálními i distribuovanými týmy, to zahrnuje sdílení seznamu úkolů, zpětnou vazbu a přidělování úkolů.
- Vyhodnocení stavu projektu pomáhá s identifikací a odstraňováním překážek projektu, vyhodnocuje výkonnost a finanční oblast projektu.

Na základě nyní zmíněných funkcí lze vyzdvihnout některé nástroje, které je splňují. Jsou to například:

- Active Collab
- Agilo for Scrum
- Pivotal Tracker
- VersionOne (Burger, 2016)
- TFS (Team Foundation Server) je produkt společnosti Microsoft, který svou komplexností možná přesahuje požadavky na software pro správu agilních projektů. Poskytuje totiž celou řadu nástrojů. Například lze uvést nástroje pro správu zdrojových kódů, reporting, správu požadavků nebo řízení projektů a to jak pro agilní metodiky, tak pro rigorózní metodiky.

Společnost ze zkoumaného článku již měla zkušenost právě s výše zmíněným nástrojem TFS, který využívala pro kódování a sledování chyb a zjistila, že je také snadné ho využít pro reportování. Důležité bylo, aby byl tento nástroj používán všemi týmy, na všech úrovních reportování a retrospektivy, které je potřeba. Nelze se spoléhat jen na aplikace typu Excel či Word. Zde se vyskytl problém, protože některé individuální týmy chtěli pracovat právě s nástrojem Excel, na který byli zvyklí.

7 Vyhodnocení a přínos integrace agilních týmů

Obecně platí, že žádný projekt nemůže být úspěšně dotažen do svého zdárného konce, pokud není dostatečně monitorován jeho průběžný stav realizace v čase. Uvědomíme-li si, že každý z pracovních týmů musí v rámci svých rolí plnit určité dílčí úkony, které v přeneseném slova smyslu můžeme také chápat jako projekty (byť malé), pak nám vyplývá jednoznační potřeba, a sice mít nastavené monitorovací a reportovací procesy takovým způsobem, abychom vždy a jednoznačně zjistili aktuální stav každého z dílčích úkonů (a tím i celého projektu) v rámci jednotlivých týmů.

7.1 Monitoring projektu

Účel těchto monitorovacích a reportovacích procesů nespočívá jen v potřebě projektového manažera, ale i v tom, že jednotlivé úkony mohou mít vzájemnou návaznost. Bez dokončení jednoho úkonu, nemůžu začít úkon druhý, bez druhého nemůže začít zase úkon další a tak dále. Toto řetězení, respektive pozdržení některého z počátečních úkonů řetězce, může mít bez dostatečného monitoringu fatální dopad na celý projekt. A proč? Protože monitoring může být zdrojem prvního signálu toho, že v některém z dílčích úkonů projektu hrozí včasné nedodání potřebného výstupu, čímž je ohrožen projekt jako celek. Právě tento signál by měl být dostatečným varováním pro vedoucího projektu, který by měl na daný úkon alokovat více lidských kapacit nebo poskytnout více potřebných zdrojů. (Gudda, 2011)

Běžným argumentem pracovníků jednotlivých pracovních týmů proti monitorovacím procesům je, že vznikne určitá administrativní zátěž, která pracovníkům sníží rychlost práce na svých úkonech. To svým způsobem pravda být může, proto je vždy nutné nalézt určitý kompromis mezi zmíněnou zátěží a přínosů z monitoringu. Nicméně monitoring nespočívá jen v aktivním zadávání dat o aktuálním stavu úkonů do monitorovacích systémů. Sbírat a vyhodnocovat můžeme i data tzv. pasivní, tedy taková, která nevyžadují aktivní potřebu zadávat data do systému ze strany pracovníka jako například čas strávený v práci, počet odeslaných emailů a další. (Gudda, 2011) Avšak nutno si uvědomit, že zde se častokrát dostáváme na tenký led, který pracovníkům nemusí být zcela příjemný.

V případě agilních týmů se snažíme většinu administrativy odbourat, ale aby přehled o aktuálním stavu projektu nebyl ztracen úplně, vstupuje zde role agilního trenéra. Úloha agilního trenéra spočívá nejen v koordinaci celého projektu, motivování kolegů, ale především v tom, že na základě scrumů a stand-up meetingů má o celém projektu dokonalý přehled a dokáže na případná, nově vzniklá kritická místa včasně reagovat.

V projektu, který je popisován ve výchozím článku k této práci byla data shromažďována v různých úrovních a z různých zdrojů. V první řadě proto bylo nezbytně důležité, aby se týmy shodli na společném nastavení monitorovacích a reportovacích procesů, především metodice sběru a vyhodnocování dat a softwarové výbavě. Toto bylo předmětem jednání jednoho z prvních scrum of scrums. Situaci značně ulehčila skutečnost, že všechny z týmu využívaly shodný software - TFS (Team Foundation Server) pro sdílení kódů, sledování jeho vývoje a spolupráci mezi členy týmu. Sjednocení tak proběhlo pouze na úrovni systému pro sledování bugů. Neméně důležitým zdrojem dat byly i zmíněné výstupy ze scrumů a data sesbíraný agilním trenérem v průběhu projektu. Na základě takto sesbíraných dat byly stanoveny následující klíčové metriky:

- Počet dokončených úkonů (= rychlost)
- Kategorizace úkonů (například nedokončené)
- Závislost mezi agilními týmy
- Výhled na dokončení projektu na základě (podle výstupů z dosavadních scrumů)

7.2 Proč to nešlo podle představ?

Že realita má k původní představě daleko se částečně potvrdilo i v tomto případě. Při integraci týmů se vyskytlo několik faktorů, které danou záležitost výraznou měrou komplikovaly. V některých případech dokonce hrozilo, že integraci se nepodaří úspěšně završit a to i napříč tomu, že celá integrace probíhala v souladu s obecně uznávanými a na mnoha projektech ozkoušenými agilními postupy.(Abrahamsson, 2002)

Jedním z faktorů, které danou integraci neočekávaně zkomplikovaly, bylo to, že členové jednotlivých týmů byly postaveni před hotovou věc, tedy před hotové zadání projektu. To samo o sobě nezní jako komplikace, nicméně komplikace spočívá v tom, že každý z týmů má jinou kulturu, jiné postupy a jiné způsoby se se zadání projektu vypořádat. Zde tak došlo k prvotním neshodám a nedorozuměním napříč týmy. Bylo proto důležité nastavit důvěru a respekt mezi týmy navzájem. Jenže jak toho docílit? Byl navržen postup, kdy vedoucí projektu vyslechl obavy všech klíčových členů týmů a tyto obavy konfrontoval na společném sezení zástupců všech týmů, kde po společné domluvě došlo k jejich řešení ve formě určitých ústupků a kompromisů mezi týmy, aby byly zjištěné obavy minimalizovány.

Další z faktorů se přímo dotýká předchozí kapitoly, která pojednávala o monitoringu. A sice, že část pracovníků se nepřímou odmítala podílet na podpoře monitorovacích procesů aktivním zadáváním a sdílením svých informací o jednotlivých úkonech. Později se ukázalo, že právě u pracovníků, kteří tato data odmítali poskytovat, bylo nejvíce těch, kteří měli na dokončení jednotlivých úkonů negativní vliv tak, že je zpozdili, či dokonce zablokovali.

Za účelem spolehlivosti monitorovacích procesů bylo důležité, aby každý z pracovníků využíval smluvené softwarové vybavení. Respektive, aby nevyužíval jiné. Přestože se na tomto zástupci jednotlivých týmů vzájemně domluvili, ne vždy to tak reálně fungovalo. Klíčovým softwarem v tomto případě bylo TFS (Team Foundation Server). Nicméně řada pracovníků jej nebyla ochotna zpočátku využívat, protože jim údajně více vyhovovali zaběhlé a jednoduché nástroje jako je například Excel. Tato skutečnost celou integraci blokovala.

Jako jiný faktor se ukázalo, že zainteresovaných stran projektu je celá řada a ne všechny byly ochotny věnovat projektu takové úsilí, jaké bylo k jeho úspěšnému dokončení vyžadováno. Některé ze stran dokonce považovali meetingy, scrumy, nebo scrum of scrums za ztrátu svého drahocenného času. Domnívali se totiž, že řada z těchto setkání se nemůže dobrat žádného zdárného konce a jen je vyrušují od jejich práce. Jelikož mezi stranami, které se meetingů neúčastnili, byly mnohdy i klíčové osoby, celý projekt výrazně se komplikoval.

Posledním z uvedených faktorů je, že každý z pracovníků má trochu odlišné představy o tom, co vlastně je agilní. Agilní způsob v tomto případě definoval pouze základní funkce a rámce pro jednotlivé role pracovníků. Celková agilita spočívala především v dosažení správné dynamiky práce v týmu a to skrze reakce na potřeby projektu jako jsou přiřazení a získání potřebných zdrojů, zapojením vhodných pracovníků, vyloučením nevhodných pracovníků nebo přenesení zodpovědností. Díky monitorovacím procesům se tak dařilo identifikovat takové pracovníky, kteří byli pro úspěch projektu hrozbou, ti byli následně přeřazeni do jiného projektu. Proces fungoval i opačně a sice pokud byl identifikován v rámci jiného projektu vhodný pracovník pro tento projekt, byl přeřazen právě sem. Tím byla zaručeno, že pracovníků na daném projektu bude vždy dostatek. Hrozbou pro projekt byli identifikováni především pracovníci

- kteří fungují samotářsky a příliš jim nevyhovuje práce v týmu
- kteří nejsou ochotni přizpůsobit se měnícímu se prostředí
- kteří se nesnažili získávat a poskytovat aktuální informace o stavu úkonů

8 Závěr

Celý proces integrace jednotlivých týmů a jejich spolupráce na projektu jako celku bych přirovnal ke skoku do neznáma. Sice existovala určitá představa, jak by spolupráce mohla fungovat, ale realita byla ve výsledku naprosto odlišná. Přestože v podkladovém textu není explicitně zmíněno, zda se výsledný projekt podařilo dokončit včas a v souladu s předem stanovenými požadavky, lze projekt i tak soudit jako úspěšný. Úspěšný minimálně z toho úhlu pohledu, že může být vhodnou inspirací pro další obdobné spolupráce několika agilně řízených týmů na jednom projektu. Inspirací v tomto smyslu rozumíme poučení se z jejich chyb jako byly například nadměrná administrativa, neochota pracovníků, nezájem zainteresovaných stran, ignorování rozdílných týmových kultur a dalších.

Na základě výše uvedeného textu můžeme úvodem vytyčený cíl práce jako *“Identifikace problémů a rizik při spolupráci agilně řízených týmů.”* považovat za splněný. Práce postupovala z obecného popisu agilních metod, přes jejich integraci do prostředí spolupráce agilně řízených týmů skrze softwarového vybavení, ale i techniky jako jsou scrum of scrums nebo lean coffee sessions. Zároveň byla jednoznačně identifikována celá řada problémů a rizik, které se při spolupráci agilně řízených týmů mohou vyskytovat. K jednotlivým problémům a rizikům bylo vždy nabídnuto příslušné řešení. Celý text se v průběhu své práce opíral a odkazoval na poznatky získané z textu *Agile Scaling in Action* od Yousefa Awada.

Použité informační zdroje

- AWAD, Yousef. *Agile Scaling in Action*. [online]. 2017. [cit. 2017-11-15]. Dostupné z: https://www.infoq.com/articles/agile-scaling-action?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=03282017news&utm_content=b;lang=cz
- Agile Manifesto authors*. 2001. [cit. 2017-11-15]. Dostupné z: <https://www.agilealliance.org/agile101/the-agile-manifesto/>
- KARPOV, Dmitry. *Flexibilní metodika pro vyvoj softwaru*. [online]. 2011. [cit. 2017-11-20]. Dostupné z: [http://www.unn.ru/pages/issues/vestnik/99999999_West_2011_3\(2\)/36.pdf](http://www.unn.ru/pages/issues/vestnik/99999999_West_2011_3(2)/36.pdf)
- MALCOLM, Isaacs. *What's new in SAFe 4.0? The value stream level and new roles*. [online]. [cit. 2017-10-22] Dostupné z: <https://techbeacon.com/whats-new-safe-40-value-stream-level-new-roles>
- OZKAYA, Ipek. *Architecting for Large-Scale Agile Development: A Risk-Driven Approach*, [online]. 2013. [cit. 2017-10-22]. Dostupné z: https://resources.sei.cmu.edu/asset_files/Article/2013_101_001_87865.pdf
- KNESL, Jiří. *Co je Scrum?* [online]. 2016. [cit. 2017-10-22]. Dostupné z: <http://www.knesl.com/co-je-scrum>
- ABRAHAMSSON, Pekka. *Agile Software Development Methods: Review and Analysis*. 2002. [online]. [cit. 2017-11-22]. Dostupné z: <https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf>
- LINDERS, Ben. *Using Scrum of Scrums with Agile Teams to Coordinate and Collaborate*. 2014. [online]. [cit. 2017-11-22]. Dostupné z: <https://www.infoq.com/news/2014/03/scrum-of-scrums>
- BURGER, Rachel. *7 of the Top Agile Project Management Software*. [online]. 2016. [cit. 2017-11-15]. Dostupné z: <https://blog.capterra.com/agile-project-management-software/>
- COUFAL, Jakub. *Webové služby pro tiketovací systém*. [online]. 2012. Bakalářská práce: Masarykova Univerzita. [cit. 2017-11-15]. Dostupné z: https://is.muni.cz/th/359409/fi_b/Text_prace.pdf
- ČÍŽEK, Pavel. *Sociální aspekty agilních metodik vývoje softwaru*. [online]. 2009. Diplomová práce: Vysoká škola ekonomická v Praze. [cit. 2017-11-15]. Dostupné z: https://library.vse.cz/F/G56V6HUL7A417AM3TUJEGD269YXA9S3CCMKS7K441EH531FV4T-05286?func=find-b&REQUEST=Sociální+aspekty+agilních+&x=0&y=0&find_code=WRD&ADJACENT=N
- GUDDA, Patrick. *Guide to project monitoring & evaluation*. S.l.: Authorhouse, 2011. ISBN 9781456784782.