Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	ZS 2017/2018
Autoři – jméno, příjmení, xname	Bc. Marie Hrabětová (hram00) Bc. Aneta Blažková (xblaa09) Bc. David Urban (urbd01)
Téma	**Inženýrská kultura a distribuované agilní týmy**
Datum odevzdání	16.12.2017

Abstrakt

Tato práce se zaměřuje na důležitost budování správné inženýrské kultury a její vliv na práci v distribuovaných týmech. Nejprve seznamuje čtenáře s inženýrskou kulturou a nejzásadnějšími pilíři její tvorby. Následuje definování distribuovaných týmů, popis typů distribuovaných týmů, jejich zapojení do agilních týmů vývoje, a častých problémů, které práci v těchto týmech provázejí. V závěru jsou definovány nástroje, které využívají pracovníci pracující v distribuovaných týmech včetně popisu, jak pomáhají problémy těchto týmů řešit.

Klíčová slova

inženýrská kultura, týmová práce, distribuované týmy, distribuované agilní týmy, inženýrství, distribuovaný vývoj software, párové programování

Obsah

1 Úvod 1

2 Inženýrská kultura 2

2.1 Optimalizace času jednotlivých iterací 2

2.2 Vysoká kvalita s revizemi kódu 3

2.3 Sdílené vlastnictví kódu 3

2.4 Respekt ve firemním prostředí 3

2.5 Rozvoj schopností 3

2.6 Nábor zaměstnanců 4

3 Distribuované týmy 5

3.1 Míra distribuce týmů 5

3.2 Distribuované týmy a agile 6

3.3 Problémy distribuovaných týmů 7

3.3.1 Uvažování “my” versus “oni” 7

3.3.2 Držení týmu ve tmě 7

3.3.3 Kultura je záhada 7

3.3.4 Komunikace 8

3.3.5 Černá skříňka 8

4 Techniky a nástroje využívané v distribuovaných týmech 9

4.1 Principy distribuovaného vývoje software 9

4.2 Párové programování 10

4.2.1 Párové programování v distribuovaných týmech 11

4.3 Nástroje využívané v distribuovaných týmech 11

5 Závěr 13

Zdroje 14

1 Úvod

V dnešní době se vývoj software neustále zrychluje a zároveň je svět čím dál tím více propojený, takže už je realitou, že tým pracující na jednom projektu vývoje software je složený z expertů doslova z celého světa. Na zrychlování vývoje software reaguje rozvoj agilních metodik, které oproti tradičním metodikám zažívají v posledních letech raketový nárůst popularity nejen v jednodušších projektech softwarového vývoje jako je například vývoj webových aplikací, ale pronikly i do tvorby klíčových podnikových aplikací.

Práce v distribuovaných týmech se rozvíjí až v posledních několika letech. Umožnil to především rozvoj vysokorychlostního internetu do všech koutů naší planety. Díky tomu už není nemožné, dokonce ani náročné, spolupracovat s kolegy z jiných kontinentů. Stejně jako v jakémkoliv jiném týmu, je důležité i v distribuovaných týmech pěstovat zdravou atmosféru a budovat fungující kulturu v týmu.

Cílem této práce je analyzovat vliv inženýrské kultury na práci týmů, popsat hlavní pilíře pro její tvorbu a současně definovat agilní distribuované týmy a zaměřit se na problémy, které se mohou v těchto týmech vyskytovat. Součástí cíle je popsat nástroje a principy, díky kterým je možné těmto problémům předcházet či je řešit, pokud nastanou.

Práce je rozdělena do tří hlavních částí. První část je věnována inženýrské kultuře a hlavní pilířům její tvorby. Následuje část, která zobrazuje fungování distribuovaných týmů, definování typů distribuovaných týmů a časté problémy, které práci v těchto týmech provázejí. V závěrečné části se práce zaměřuje na principy a nástroje, které mohou některé problémy a úskalí spojené s distribuovanými týmy, především komunikaci v nich, velmi ulehčit.

2 Inženýrská kultura

Každá firma by v dnešní době měla dbát na to, aby měla velmi kvalitní firemní, ale i vysoce produktivní inženýrskou kulturu. To platí bez ohledu na to, zda se v dané firmě pracuje v lokálních, nebo distribuovaných týmech. Podle Conwayova zákona z roku 1967 by organizace budující systémy měly tvořit návrhy, které jsou kopiemi jejich organizačních struktur. Z toho tedy jasně vyplývá, že kvalita softwaru je závislá na organizační struktuře a míře spolupráce mezi jednotlivými členy týmu. Právě proto je firemní a inženýrská kultura tak důležitá. (Franzen, 2017)

Obecně je firemní kultura velmi rozšířený fenomén, který může firmám poskytnout velkou konkurenční výhodu. V poslední době se často za firemní kulturu označují teambuildingové akce (paintball, kino), nebo například obědy zdarma. Spíše by ale mělo jít o pravidla, postupy a hodnoty, které by zaměstnanci dané firmy měli ctít, a na kterých by měli nadřízení lpět, aby určitým způsobem formovali chování svých zaměstnanců. (Salazar, 2015)

Pro inženýrskou kulturu existuje několik zásadních prvků, které byly nalezeny napříč nejlepšími týmy. Pokud firma buduje svou inženýrskou kulturu, měla by se na tyto společné body zaměřit, a ne se za každou cenu lišit od své konkurence.

Pro stanovení následujících stojných pilířů dobré a produktivní inženýrské kultury bylo vycházeno především ze studie Emunda Laua z roku 2012, která zahrnovala více než 500 pohovorů a odpovědí na otázky, co lidé měli rádi a neradi na kultuře ve své firmě, ale také ze zkušeností bývalých CEO společností Touchstone Semiconductor a Stormpath, Bretta Foxe a Alexe Salazara.

2.1 Optimalizace času jednotlivých iterací

Rychlé a krátké iterace totiž zvyšují motivaci, ale také výkonnost jednotlivých pracovníků. Do práce přinášejí značnou část vzrušení a dávají inženýrům a návrhářům možnost se samovolně rozhodovat bez svolení svých nadřízených. Týmoví lídři jsou zde však více než důležití musí totiž tato rozhodnutí a celkovou práci všech členů koordinovat a usměrňovat jejich snahu tak, aby byla účinná. Odhadnutý čas však musí být odpovídající k práci, která má být vykonána. Neměli bychom nutit naše pracovníky pracovat přesčas do pozdních nočních hodin, jen proto aby stíhali naprosto nereálný termín. (Lau, 2012)

2.2 Vysoká kvalita s revizemi kódu

Kvalitní a čistý kód je mnohem méně náchylný k chybám, podstatně rychleji se dále rozvíjí a snadněji se přizpůsobuje změnám. Zařazení revizí kódu pak přispívá právě zmíněné kvalitě. Jednak tím, že vytváří tlak na samotné vývojáře, kteří vědí, že jejich práce bude revidována a jejich špatný výsledek může ovlivnit další spolupracovníky. Dále také samotným kontrolováním, z kterého mohou přijít další podněty na zlepšení. (Lau, 2012) (Fox, 2017) (Franzen, 2017)

2.3 Sdílené vlastnictví kódu

Tím, že kód je sdílený a podílí se na něm více lidí, se můžeme vyhnout hned několika nepříjemnostem. Za prvé se razantně sníží riziko toho, že když jeden člen opustí tým, nebude danému kódu nikdo rozumět. Také se sníží tlak na jednotlivé pracovníky, kteří se poté při rozhodování mohou vzájemně podpořit. Sdílení kódu dále usnadňuje práci na problémech s vysokou prioritou, velmi snadno se díky tomu mohou členové mezi týmy přesouvat a obtížné věci tak vyřešit rychleji. (Lau, 2012)

2.4 Respekt ve firemním prostředí

Lidé by se v týmu, nebo ve firmě měli cítit respektováni. To platí oběma směry napříč organizací či týmem. Nadřízení by měli svým zaměstnancům dát možnost vyjádřit svůj názor, respektovat jejich čas a jejich práci. Při práci na návrhu a budování IS je nutné hodně komunikovat a pokud si zaměstnanci nejsou něčím jisti, měli by mít možnost vznesení dotazu, nebo rozpoutání debaty. Ve firmě pracují velmi inteligentní lidé, kteří často mohou přijít s inovativním řešením, musejí však vědět, že tuto šanci vůbec mají. Stejně tak by ale podřízení měli respektovat své nadřízené, a to se týká především přijatých rozhodnutí. V každém týmu se může stát, že nastane špatné rozhodnutí. Pro tým by z toho ale mělo vzejít poučení do budoucích rozhodovacích procesů, ne kritika spojená s neúctou k vedoucím pozicím. (Salazar, 2015) (Lau, 2012)

2.5 Rozvoj schopností

Do inženýrské kultury je také nutné zahrnout určitý systém vzdělávání a rozvíjení schopností svých pracovníků. To samozřejmě může být obsáhnuto různými školeními. Doporučuje se toto dále rozšířit. Při tvorbě produktů vyhradit čas a dát inženýrům možnost představit svou

práci a postup. Tím získají rozhled o jejich práci i další členové týmu, kteří v ni mohou nacházet další příležitosti, nebo třeba dané postupy aplikovat na své problémy. Důležitým prvkem zdokonalování pracovníků je samozřejmě také konstruktivní a pravidelná zpětná vazba, která pomůže ke zlepšení schopností jednotlivých pracovníků, ale také k rychlejšímu zařazování změn a úprav produktu. (Lau, 2012) (Salazar, 2015) (Franzen, 2017)

2.6 Nábor zaměstnanců

Lauova studie je v tomhle případě velmi striktní. Podle ní, pokud chcete dosáhnout nejlepších výsledků, musíte pracovat jedině s nejlepšími lidmi. Bývalý CEO firmy Touchstone Semiconductor Brett Fox k tomuto tématu dokonce poznamenal, že jedině neúspěšné firmy najímají zaměstnance, kteří jsou pouze dostačující. Nejmírněji v této části hovoří CEO Alex Salazar. Upozorňuje především na určité vlastnosti najímaných zaměstnanců, jako je například inteligence, schopnost dotahovat věci do konce, ale také pokora. Oba jak Fox, tak Salazar pak dávají důraz na jakousi týmovou chemii. Tým musí být otevřený novým nápadům, musí fungovat jako celek. Jen tak se mohou spolupracovníci motivovat, posouvat dál a podávat, co nejkvalitnější výkony. (Lau, 2012) (Salazar, 2015) (Fox, 2017)

Důležité je ale zmínit, že nestačí jen se řídit výše zmíněnými prvky. Ty mohou sloužit jako opěrné body při stavbě celé inženýrské kultury. Na začátku však musí být jasně definovaná firemní či týmová strategie z níž teprve zmíněné kultury (firemní, inženýrská) vycházejí a jsou přizpůsobovány tak, aby byly přijatelné konkrétním týmům. Díky správnému vymezení těchto kultur pak dokáže firma tvořit produkty, které pro ni něco znamenají a přinášejí ji prospěch. (Salazar, 2015)

Distribuované agilní týmy se při práci musí potýkat s celou řadou problémů, a právě tyto body by měly pomoci dané potíže překonat. Už jen proto, že podporují týmovou spolupráci a komunikaci, která je při práci v těchto týmech naprosto stěžejní. (Franzen, 2017)

3 Distribuované týmy

Distribuované týmu jsou dnes již standardem v mnoha organizacích. Díky možnostem informačních a komunikačních technologií již není nezbytné, aby se tým pracující na jednom projektu nacházel na stejném místě, a tak je čím dál tím běžnější situace, kdy se členové týmu nachází na mnohem různých místech, mnohdy i po celém světě.

Přechod k distribuovaným týmům má několik příčin. Jednou z nich je hledání talentů, které začíná v oblasti centrály společnosti, nicméně rychle expanduje do dalších oblastí.

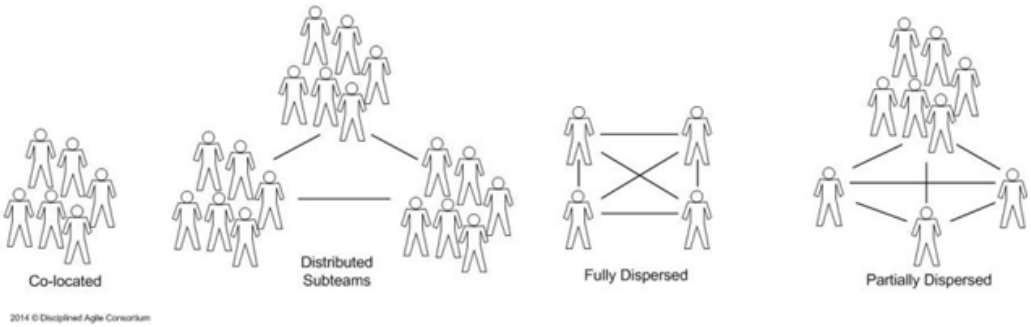
Distribuované tým také často vnikají společně s vytvořením kontraktu mezi společností a jejím dodavatelem. V tomto případě je velmi běžným problémem oddělování práce, kdy například na úrovni centrály společnosti vzniká návrh a dodavatel vytváří pouze kód. Tento postup zřídkakdy vede k úspěchu. (PAHUJA, FRANZEN, 2017)

3.1 Míra distribuce týmů

Mluvíme-li o týmu jako o distribuovaném, nemusí to nezbytně znamenat, že se členové týmu nachází na opačných kontinentech. O distribuci týmu můžeme mluvit již ve chvíli, kdy je tým rozmístěn v oddělených kancelářích, či na jiných patrech kancelářské budovy. Podle toho, jak vzdáleni od sebe jednotlivý členové týmu jsou, rozlišujeme 3 typy distribuce týmů:

- **Co-located** = nacházejí se na stejném místě
- **Near located** = členové týmu se nachází ve vzdálenosti, která je s pomocí automobilu překonatelná v rámci každého dne (je reálné, aby se členové každodenně osobně setkali)
- **Far located =** Není možné, aby se členové týmu každý den osobně setkali (vzdálenost mezi nimi např. vyžaduje leteckou dopravu)

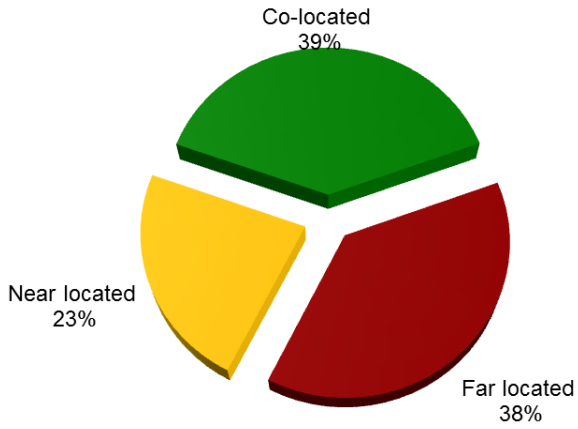
Kromě samotné vzdálenosti může u distribuovaných týmů rozlišovat několik typů distribuce. Základní z nich jsou znázorněny v následujícím obrázku, v realitě ale vnikají jejich nejrůznější kombinace.



Obrázek 1: Typy distribuce týmů (A FOUNDATION FOR BUSINESS AGILITY, 2014)

3.2 Distribuované týmy a agile

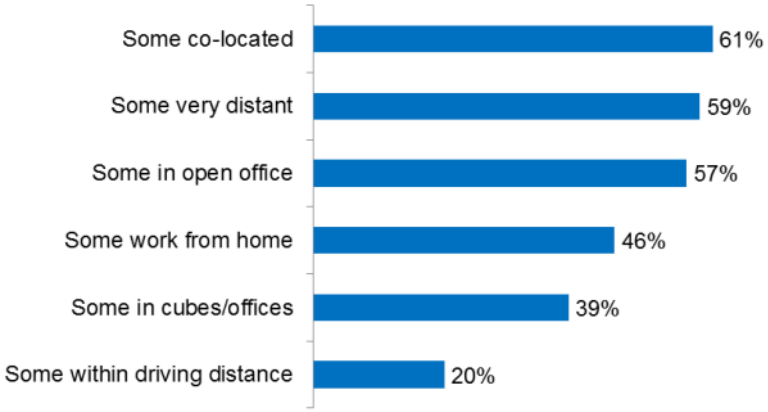
Práce je zaměřena na agilní týmy, proto je na místě otázka, jak jsou na tom s geografickou distribucí agilní týmy. Graf z roku 2014 ukazuje, že již tehdy 61 % agilních týmů bylo geograficky distribuovaných, pouze 39 % týmů pracuje na jednom místě.



Obrázek 2: Agilní týmy a geografická distribuce (A FOUNDATION FOR BUSINESS AGILITY, 2014)

Jak ale vyplývá z předchozí podkapitoly, realita je mnohem složitější. Týmy se nedají rozlišit pouze na distribuované a nedistribuované, takové rozdělení totiž vůbec nevypovídá o tom, o jaký tým se jedná. Je distribuovaný proto, že má jednoho člena pracujícího z domova nebo je to tým rozptýlený po celém světě? O mnoho přesnější informace přináší detailnější informace přináší průzkum z roku 2017, který neříká, zda je tým co-located, near located nebo far located, ale zkoumá, zda v týmu je někdo velmi vzdálený, alespoň část týmu pracuje na stejném místě atd. Tento průzkum také uvádí, že 29 % agilních týmů je near located, což je sice o 6 % více než v roce 2014,

ale stále je to poměrně malé číslo a znamená, že značná část agilních týmů má far located členy.



Obrázek 3: Jak jsou geograficky distribuováni členové agilních týmů (A FOUNDATION FOR BUSINESS AGILITY, 2017)

3.3 Problémy distribuovaných týmů

Ačkoliv existují dobré důvody vzniku distribuovaných týmů a jistě i mnohé výhody, neobejde se taková struktura bez problémů, kterým je nutné věnovat pozornost, mají-li být týmy úspěšné.

3.3.1 Uvažování “my” versus “oni”

Zatímco s lidmi se kterými pracujeme v jedné kanceláři běžně komunikujeme, a to nejen nad pracovními tématy, potkáváme se u kávy, sbližujeme se a začneme takový tým lidí vnímat jako “my”. Toto vnímání může vést ke vzájemné rivalitě a obviňování jiných týmů z případných neúspěchů.

3.3.2 Držení týmu ve tmě

V mnohých případech je vývojářský tým vzdálený od product ownera a od budoucích uživatelů produktu. Product owner se tak stává klíčovou osobou, která perfektně rozumí vyvíjenému produktu a tyto informace zprostředkovává vývojářskému týmu.

3.3.3 Kultura je záhada

Pokud se v týmu sejdou lidé z různých kultur problémy se ještě zhorší, a to nejen ty s komunikací. Nerozumí tomu, jak lidé z jiných kultur přemýšlejí, jaké jsou motivace jejich jednání a v důsledku toho dochází k mnohým nedorozuměním a je velmi komplikované s takovými lidmi spolupracovat.

3.3.4 Komunikace

Zásadní problémem distribuovaných týmů, který více či méně zapříčiňuje všechny ostatní problémy je komunikace. Lépe řečeno její nedostatek. Ačkoliv je právě v distribuovaných týmech komunikace zcela klíčovým element, ve skutečnosti je její nedostatek jedním z nejběžnějších problémů. Tento problém lze dnes řešit nejrůznějších metod, technik a komunikačních nástrojů.

3.3.5 Černá skříňka

Tento problém spočívá v tom, že jeden tým nezná druhý, a tudíž nerozumí jeho způsobu práce, vidí pouze set požadavků, které slouží jako vstup a výstup, který na základě požadavků tým vyprodukoval. Zatímco při práci na jednom místě jsou týmy schopny vzájemně svou práci reflektovat po celý průběh procesu, v distribuovaných týmech panuje předpoklad, že si vzdálený tým na všechno přijde sám a dodá výsledek. Pro úspěšnou spolupráci je ovšem zapotřebí aby všechny distribuované týmy pracující na společném projektu pracovali jako jeden celek, nikoliv několik oddělených částí. (MESSER, 2016)

Následující kapitola se zaměří na to, jaké nástroje a techniky lze využít při řešení problému komunikace v distribuovaných týmech.

4 Techniky a nástroje využívané v distribuovaných týmech

Podle průzkumu společnosti VERSIONONE (2017) má 86 % oslovených firem ve své struktuře alespoň jeden distribuovaný tým, který používá agilní metodiky vývoje. Nejčastěji používané metodiky v těchto týmech jsou Scrum (58%) a Scrum / XP Hybrid (10%). Čisté XP používá méně než 1% týmů.

O využití metodiky Scrum v agilních distribuovaných týmech bylo napsáno již mnoho. Ale jak je to s párovým programováním? Lze vůbec využít takto specifickou techniku vývoje software, která vyžaduje úzkou komunikaci a spolupráci dvou programátorů, v distribuovaných týmech, ve kterých je právě komunikace často velkým problémem? K tomu, aby se problémy s komunikací v těchto týmech minimalizovaly a bylo snadné je řešit, je nutné co nejvíce dodržovat principy distribuovaného vývoje software a využívat vhodné nástroje, díky kterým je možné překonat vzdálenost, která jednotlivé členy týmu dělí, a umožnit pracovat stejně tak dobře, jako kdyby se celý tým nacházel na stejném místě.

4.1 Principy distribuovaného vývoje software

Vhodnou praktikou pro místně oddělené týmy je ad-hoc koordinace. Tato praktika nespoléhá na pravidelné týdenní nebo např. měsíční schůzky, ale vychází z předpokladu, že problémy se mají řešit ihned, jak se je podaří identifikovat. Díky této technice je možné snížit objem přepisování zdrojového kódu v důsledku špatného pochopení zadání nebo špatné koordinaci programátorů na naprosté minimum.

Další princip, který umožňuje práci v distribuovaných týmech je vhodné uspořádání pracovníků v týmu a návaznosti jejich činností na sebe. Tento princip se nazývá Flow a je převzatý z automobilového průmyslu. Spočívá v přeuspořádání pracovních postupů tak, aby práce na projektu jakoby “plynula” skrze organizaci a nikde se nezdržovala déle, než je nezbytně nutné.

Další z principů, který umožňuje velmi omezit problémy způsobené místně oddělenými programátory, je continuous integration. Tento princip se snaží omezit počet souběžně rozdělaných větví programu na naprosté minimum, ideálně jen na jednu aktuálně vyvíjenou větev. To umožní všem programátorům vždy pracovat na aktuální verzi programu bez nutnosti řešit velké množství konfliktů, které vznikají při slučování vývojových větví.

Tento princip úzce souvisí s principem automatického testování, který umožňuje nasazovat nové verze programu do produkčního prostředí velmi často. Ideálně by mělo dojít k nasazení nové verze programu nejméně jedenkrát denně, někdy i vícekrát. To by nebylo možné bez automatizovaného testování, které umožní týmu provádět testy funkcionalit několikrát denně bez vlivu na náklady či personální obsazení týmu.

Všechny tyto principy se spojují v technice nazvané distribuovaný vývoj software, která všechny tyto techniky spojuje do jednoho uceleného řešení. Díky rozvoji cloudových technologií je možné takto vyvíjet aplikace přímo v prostředí internetu bez nutnosti instalovat specializované robustní řešení. Díky těmto řešením mohou programátoři řešit pouze zdrojový kód aplikace. Komunikace může probíhat díky audiovizuálnímu přenosu naprosto přirozeně, programátoři mohou vidět zápis zdrojového kódu takřka v reálném čase. Po ukončení práce na konkrétním úkolu je možné v tomto prostředí software rovnou otestovat a v případě, že všechny testy projdou naprosto bez problémů tuto část aplikace i nasadit do produkčního prostředí.

4.2 Párové programování

Párové programování je jedna z technik agilního programování. Ač se to na první pohled nezdá, je velmi efektivní. Spočívá v psaní zdrojového kódu na jedné stanici dvěma programátory najednou. Podle serveru Agile Alliance (nedatováno) jeden z programátorů vystupuje v roli nazvané anglickým slovem “driver”. Tento programátor fyzicky zadává zdrojový kód programu do počítače. Druhý programátor, nazvaný jako “navigator” by podle této techniky měl sedět vedle svého kolegy a zdrojový kód kontrolovat, kontrolovat případné chyby, či případně vymyslet, zda nelze kód zapsat efektivněji. Oba kolegové se během práce na úkolu velmi často střídají.

Mnozí tvrdí, že tato technika je zbytečně drahá, protože vytěžuje dva programátory na jeden úkol, který by podle této logiky měl být x-krát tak drahý. To ovšem není pravda, protože dva programátoři pracují na jednom úkolu mnohem rychleji a s mnohem menší chybovostí než dva separátně pracující pracovníci. Případné chyby totiž snáze najde programátor, který program fyzicky nepíše.

Aby tato metoda dobře fungovala, musí spolu oba dva programátoři osobnostně velmi dobře vycházet, musí být oba dva podobně zainteresovaní do práce na projektu, nesmějí oba dva být velcí individualisté. Svými vlastnostmi by se měli vhodně doplňovat.

4.2.1 Párové programování v distribuovaných týmech

Z výše uvedeného textu by se mohlo zdát, že pro fungování této techniky je nutné mít oba dva programátory na jednom místě fyzicky. Dlouho to tak skutečně bylo, ale s rozvojem vysokorychlostního internetu a programů zprostředkovávajících přenos zvuku a obrazu to již přestává být potřebné. Stále více firem, zabývajících se vývojem SW, je mezinárodní, a proto by byla fyzická spolupráce dvou programátorů velmi limitujícím faktorem.

Tyto překážky se ovšem podařilo odstranit sdílením pracovní plochy programátorů, přenosem jejich hlasu či rovnou video přenosu z provozu kanceláře. Díky nástrojům pro vytváření virtuálních kanceláři se pracovníci z různých koutů světa mohou ve virtuální prostoru cítit skoro tak, jako by pracovali v jedné open space kanceláři. Samozřejmě pro určité pracovníky může způsobovat bariera způsobená přenosem videa na druhý konec světa určité potíže, ovšem benefity tohoto řešení rychle převáží. Jedním z nich a pravděpodobně tím největším je fakt, že program mohou psát lidé z odlišných civilizačních kultur, takže je více přizpůsoben klientům na globální úrovni a zároveň na odlišnost kultur připraven i po své funkční stránce.

4.3 Nástroje využívané v distribuovaných týmech

Existuje celá řada nástrojů, díky kterým je možné provádět softwarový vývoj v agilních distribuovaných týmech. Pro jednoduchost jsou rozděleny do několika skupin, které pokrývají různé potřeby jednotlivých členů distribuovaných týmů. Jedná se o aplikace pro koordinaci práce a vedení týmů, aplikace pro kolaborativní tvorbu dokumentů a zdrojových kódů, nástroje pro komunikaci (především pro chat, přenos zvuku, obrazu či pracovní plochy) a poté komplexní nástroje, které v sobě kombinují více výše uvedených kategorií.

Do první skupiny podle Adama Febera (2016) patří například aplikace Trello, Jira nebo Jell. Trello se vyznačuje velkou volností při tvorbě týmového workflow. Jira naopak velmi striktně vyžaduje práci podle předem stanoveného postupu. Jell slouží jako nástroj pro pořádání ranních stand up meetingů.

Mezi aplikace pro tvorbu kolaborativních dokumentů a zdrojových kódů patří aplikace Google Apps for Business, inVision, Evernote Skitch nebo editor zdrojových kódů Atom.io a doplňkem Motepair nebo Codenvy. Všechny tyto aplikace fungují na principu kontinuálních přenášení veškerých změn obsahu do klientských aplikací všech spolupracujících osob v reálním čase.

Mezi klasické nástroje pro komunikaci patří veškeré chatovací platformy (např. Google Hangout, Facebook Messenger, nebo například Slack). Mezi nejvyužívanější aplikace, které přenášejí zvuk a video, patří Skype, mezi aplikace přenášející obraz pracovní plochy patří například Team Viewer.

Samostatnou skupinu aplikací poté tvoří kompletní řešení, která v sobě kombinují funkcionalitu více těchto aplikací. Mezi nejrozšířenější v této skupině aplikací patří například Screenhero. Tento nástroj umožňuje sdílet obrazovku, komunikovat prostřednictvím zvuku i videa, plánovat úkoly, ale zároveň umožňuje přímo pracovat v prostředí vývojových programů či jiných aplikací, několika uživatelům zároveň a vidět tak i všechny úkony kolegy v reálném čase. V současnosti je Screenhero součástí většího ekosystému aplikací zapojených do nástroje Slack.

Všechny tyto aplikace mají několik společných znaků. Pro jejich funkčnost je potřeba sítě Internet, většina z nich je provozována jako webová aplikace s možností nainstalování desktop, či mobilní klientské aplikace.

Použití výše uvedených nástrojů pomáhá překonat vzdálenost, která může týmy dělit a umožňuje členům pracovat tak, jako by mezi nimi žádná překážka, způsobená vzdáleností, ani nebyla. Díky komunikačním nástrojům, nástrojům pro plánování úkolu a pro sdílení obrazovek, mohou členové týmu z různých koutů světa pracovat, bavit se, domlouvat, i dostávat stejně rychlou zpětnou vazbu tak, jako by seděli v jedné místnosti.

5 Závěr

Cílem této práce bylo analyzovat vliv inženýrské kultury na práci týmů, současně definovat agilní distribuované týmy a zaměřit se na problémy, které se mohou v těchto týmech vyskytovat. Součástí cíle bylo také popsat nástroje a principy, díky kterým je možné těmto problémům předcházet či je úspěšně vyřešit, pokud nastanou.

Cíl práce se podařilo naplnit. Byla definována inženýrská kultura včetně popisu všech hlavních specifik, které má na tým softwarového vývoje. Dále byly definovány distribuované týmy a jejich zapojení do procesu vývoje software, byly popsány druhy distribuovaných týmů, a problémy práce, které se vyskytují v takto postavených týmech. V závěru práce byly popsány nástroje, se kterými pracovníci v těchto týmech pracují včetně jejich kategorizace a popisu benefitů, které tyto aplikace mají pro distribuované týmy.

Díky této práci je vidět, že ve chvíli, kdy je správně nastavená fungující kultura v týmu a jsou využité vhodné techniky a nástroje pro práci, nezáleží, zda se kolega nachází přímo na vedlejší židli, či zda sedí na druhé straně planety. Tento fakt se dá aplikovat i na techniku párového programování, u které se nejprve zdálo, že je pro distribuované týmy naprosto nevhodná, ale s použitím vhodných nástrojů a dodržením principů distribuovaného vývoje software, je možné tyto problémy překonat a pracovat se spolupracovníky z různých koutů planety na jednom projektu zároveň bez větších problémů.

Zdroje

FEBER, Adam. Overcoming The Office: 13 Tools That Connect Our Remote Team. Chargify - the bullring blog [online]. 2016 [cit. 2017-12-16]. Dostupné z: https://www.chargify.com/blog/13-tools-that-connect-our-remote-team/

FOX, Brett, 2017. Why Your Engineering Culture Can Make Or Break Your Company [online]. Quora, Forbes [cit. 10.12.2017]. Dostupné z: https://www.forbes.com/sites/quora/2017/06/23/why-your-engineering-culture-can-make-or-break-your-company/#70ae53e1d5cf

MESSER, Hugo, 2016. The Top 5 Problems with Distributed Teams and How to Solve Them [online]. InfoQ. [cit. 20.11.2017]. Dostupné z https://www.infoq.com/articles/top5-problems-distributed

LAU, Edmond, 2012. What Makes a Good Engineering Culture? [online]. Quora [cit. 10.12.2017]. Dostupné z: http://www.slate.com/blogs/quora/2012/05/24/what_makes_a_good_engineering_culture_.html

PAHUJA, Savita a FRANZEN, Arjan, 2017. Engineering Culture and Distributed Agile Teams [online]. InfoQ. [cit. 20.11.2017]. Dostupné z https://www.infoq.com/articles/engineering-culture-distributed?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=09122017news&utm_content=other

SALAZAR, Alex, 2015. 9 Ways to Build a Great Engineering Culture [online]. [cit. 10.12.2017]. Dostupné z: https://stormpath.com/blog/how-to-build-engineering-culture

11th Annual State of Agile Report, 2017. VersionOne [online]. Alpharetta - Atlanta [cit. 2017-12-16]. Dostupné z: https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2

A FOUNDATION FOR BUSINESS AGILITY, 2014. Geographically Distributed Agile Teams [online]. The Disciplined Agile (DA) Framework. [cit. 20.11.2017]. Dostupné z http://www.disciplinedagiledelivery.com/agility-at-scale/geographically-distributed-agile-teams/

A FOUNDATION FOR BUSINESS AGILITY, 2017. How Geographically Distributed Are Agile Teams in Practice? [online]. The Disciplined Agile (DA) Framework. [cit. 20.11.2017]. Dostupné z http://www.disciplinedagiledelivery.com/category/geographic-distribution/