

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2018
Autoři – jméno, příjmení , xname	Jakub Aschmann (xascj00), David Galaš (xgald01), Jan Pecha (xpecj32)
Téma	Agile Outside the Development Team
Datum odevzdání	12.5.2018

Abstrakt

Tato týmová semestrální práce se zabývá současnými a budoucími možnostmi využití agilních technik v oblastech mimo vývoj informačních systémů. U jednotlivých metodik jsou v práci nastíněny možné oblasti jejich využití a jejich předpokládaný vývoj.

Klíčová slova

Agilní techniky, přístup, využití, princip, budoucnost

Obsah

1.	Úvod	1
2.	Agilní metodiky, vhodné pro použití mimo vývojový tým	1
2.1.	Scrum	1
2.2.	Kanban	2
2.3.	Lean Development	3
2.4.	Feature Driven Development	4
2.5.	Extrémní programování	5
2.5.1.	Párové programování	5
2.5.2.	Plánovací hra	5
2.5.3.	Celý tým	5
3.	Oblasti (mimo vývoje), kde je použití agilních metodik vhodné	5
3.1.	Využití Scrumu a Kanbanu na univerzitách	6
3.2.	Scrum v obchodním oddělení	8
3.3.	Agile v testování	8
3.3.1.	Testování v metodice Scrum	9
3.3.2.	Testování v metodice Extrémní programování	9
4.	Předpokládaný budoucí vývoj jednotlivých agilních metodik mimo vývojový tým	10
4.1.	Obecné principy agilních technik využitelné v budoucnu mimo vývojový tým	10
4.2.	Budoucí využití jednotlivých agilních metodik	10
4.2.1.	Scrum	10
4.2.2.	Kanban	11
4.2.3.	Lean Development	12
4.2.4.	Feature Driven Development	12
4.2.5.	Extrémní programování	12
5.	Závěr	13
6.	Zdroje	1

1. Úvod

Pojem "agilní" je v dnešním IT světě synonymem pro modernost a pružnost. Ve spojení s přístupem k vykonávání úkolů v rámci jednotlivých projektů je zpravidla považován za nástupce rigorózního, „zkostnatělého“ přístupu a v současnosti, jak uvádějí (VersionOne, 2017) a (Wijetilaka, 2016), jej téměř lze označit za běžný. Agilní přístup je uplatňován zejména v oblasti vývoje softwaru, jelikož svou povahou umožňuje rychle vykazovat průběžné výsledky odvedené práce a na jejich základě případně dále upravovat celkovou funkcionalitu vyvíjené aplikace (AltexSoft, 2018). Vývoj softwaru však není jedinou oblastí, kde je možné agilní přístup uplatnit. Cílem této práce je nalezení a popis možných oblastí využití agilního přístupu, mimo vývoj softwaru, a zmapování jednotlivých agilních metodik a postupů, které je možné mimo oblast vývoje softwaru využívat. K jednotlivým nalezeným oblastem budou následně přiřazeny jednotlivé agilní metodiky, které jsou v nich aplikovatelné. Závěrem budou zmapovány současné trendy, týkající se agilního přístupu. Pro naplnění zadaného cíle budou analyzovány doporučené zdroje k této semestrální práci a dostupné internetové zdroje, především pak odborné články zabývající se zadaným tématem.

2. Agilní metodiky, vhodné pro použití mimo vývojový tým

Na základě analýzy (VersionOne 2017) a (Goncalves, 2018) byly vybrány současné nejvýznamnější metodiky, pro které byly v práci hledány další možnosti využití:

2.1. Scrum

Specialitou Scrumu je série tzv. sprintů s fixní délkou trvání a hmatatelným výsledkem na konci každého sprintu. Během sprintů vznikají nejrůznější diagramy a grafy, zachycující postup prací, případně množství odvedené a zbývající práce, které jsou zpravidla viditelné všem členům týmu a slouží tak jako motivátory pro všechny členy týmu (Atlassian, 2018a).

V této metodice existují 4 hlavní prvky, které tvoří její jádro a bez nichž by nemohla správně fungovat. Tyto prvky jsou následující (Atlassian, 2018a):

- Denní standup meeting: Základní prvek metodiky, jedná se o rychlý, zpravidla 15minutový meeting, sloužící k vzájemné synchronizaci členů týmu.

- Plánování sprintu: Slouží k určení obsahu sprintu a úkolů, které je v jeho rámci potřeba vypracovat.
- Demo sprintu: Společný týmový meeting, kde jsou předváděny výstupy daného sprintu.
- Retrospektiva sprintu: Zhodnocení, co se v rámci sprintu povedlo, a na čem je do budoucna potřeba zapracovat, včetně zlepšujících opatření.

Dále jsou zde vymezeny 3 hlavní role, jejichž dodržování je za účelem efektivnosti práce klíčové. Role jsou následující (Atlassian, 2018a):

- Product owner: Zaměřuje se na jednotlivé požadavky na výsledný produkt, prioritizuje jednotlivé úkoly, zajišťuje, že každý člen týmu rozumí tomu, co má za úkol a určuje pořadí jednotlivých prvků, které je potřeba vyvinout.
- Scrum master: Působí jako coach celého týmu a jeho hlavním úkolem je plánování dostupných zdrojů (lidských i logistických) pro sprint. Je nutné, aby plně rozuměl zadané práci a mohl tak vhodně optimalizovat její průběh dle aktuální potřeby.
- Scrum tým: Vývojářský tým, čítající zpravidla 5-7 členů, kde jde především o kombinaci dovedností a zkušeností mezi jednotlivými členy týmu, aby došlo k vytvoření koherentní pracovní skupiny, jak bez dominantních členů, tak slabých článků

V současné době platí Scrum za jednu z nejpoblárnějších agilních metodik, uplatňovaných při vývoji softwaru, a dá se očekávat, že bude zastávat významnou roli i v nejbližších letech (Atlassian, 2018a).

2.2. Kanban

Metodika Kanbanu funguje dle (Atlassian, 2018b) na principu vizualizace práce od samotného zadání až po její dokončení. Veškerá práce probíhající podle této metodiky je zachycena na jedné přehledné tabuli, zpravidla dnes v digitální podobě, kde je dále členěna zpravidla do tří sekcí:

- ToDo: úkoly čekající na vypracování,
- Work-in-progress: rozdělané dílčí úkoly a funkcionality,
- Done: dokončené dílčí úkoly.

Každý jednotlivý úkol v těchto sekcích je reprezentován kartičkou neboli kanbanem, na které je napsán jeho obsah. Skrze tyto kartičky je následně dosahováno zmíněné vizualizace práce. Výhodou Kanbanu je dle (Atlassian, 2018b) především flexibilita, s jakou lze plánovat jednotlivé úkoly, dále redukce úzkých hrdel projektu prostřednictvím nastavení limitu pro množství úkolů ve stavu work-in-progress, a také poměrně krátká doba potřebná k provedení jednoho cyklu neboli průchodu daného úkolu z sekce ToDo do Done. Metodika také umožňuje neustálé dodávání hotové práce zákazníkovi, což zvyšuje konkurenceschopnost dodávaného produktu.

2.3. Lean Development

Podstatou této metodiky je optimalizace celkové efektivity vývojového procesu a dodávání co nejvyšší hodnoty koncovému zákazníkovi (McLaughlin, 2018). Toho je dle (Trivedi, 2017) dosahováno prostřednictvím 7 základních principů:

- Eliminace zbytečností: Eliminace praktik, které snižují kvalitu výsledného kódu, zbytečně natahují čas, potřebný k vývoji, a snižují hodnotu dodaného produktu.
- Zdůrazňování procesu učení: Snaha o maximální využití nabytých zkušeností, jejich sdílení v rámci týmu a kladení důrazu na poučení se z chyb.
- Odkládání rozhodnutí na poslední chvíli: Důležitá rozhodnutí jsou odkládána na co nejpozdější okamžik z důvodu flexibility vůči změnám v požadavcích ze strany zákazníka.
- Co nejrychlejší dodávky (produktu): Vývoj probíhá v krátkých iteracích s důrazem na rychlost, s jakou jsou jednotlivé části produktu dodávány.
- Motivovaný tým: Snaha o motivování, respekt a mentoring jednotlivých členů týmu, jelikož motivovaní lidé odvádějí nejlepší výsledky.
- Budování kvality: Jednotlivé prvky vyvíjené aplikace by měly být vyvinuty v co nejvyšší kvalitě, pokud možno na první pokus, aby se předcházelo vzniku defektů, které je následně nutno opravovat.
- Pohlížení na aplikaci jako na celek: Vyvíjená aplikace je vnímána jako jeden celek i přesto, že je vyvíjena a odlaďována po jednotlivých částech různými týmy.

Celkově jsou tak v rámci této metodiky postupně odstraňovány zbytečnosti a nedostatky na jednotlivých úrovních procesu vývoje (Trivedi, 2017).

2.4. Feature Driven Development

Princip této metodiky spočívá ve vývoji daného programu na základě jednotlivých „features“ neboli vlastností, kterými je obvykle malá část vyvíjeného programu neboli funkcionalita, která má pro koncového zákazníka určitou hodnotu (Ambler, 2014). Tyto vlastnosti jsou postupně přidávány do vyvíjené aplikace, až vznikne hotový celek. Tato metodika využívá 5 základních procesů, které jdou v daném pořadí za sebou a jsou dle (Karam, 2017) následující:

1. Vývoj celkového objektového modelu:

V této první fázi je vytvořen celkový objektový model vytvářeného řešení. Během sestavování modelu dochází k navrhování různých modelů jednotlivými týmy, složených z vývojářů a odborníků na danou oblast. Jednotlivé navrhované modely jsou průběžně revidovány, a nakonec je sestaven objektový model, který bývá zpravidla ideálním mixem několika původních návrhů.

2. Vytvoření seznamu vlastností

Dále je vyhotoven seznam vlastností v podobě user stories, který je porovnán s produkt backlogem. Zde jsou následně identifikovány vlastnosti, které mají pro klienta vysokou hodnotu a kterými se bude řídit projekt, přičemž doba vývoje jedné vlastnosti by neměla přesáhnout 14 dní, aby bylo možné klientovi ukazovat průběžné výsledky práce na produktu.

3. Plánování po vlastnostech

Následně je určeno, které konkrétní požadované vlastnosti budou implementovány. Ty jsou potom přiřazeny v podobě tříd jednotlivým programátorům.

4. Navrhování po vlastnostech

Zde jsou vytvořeny návrhové balíčky k jednotlivým vlastnostem, přičemž přesné pořadí jejich vývoje se odvíjí od rozhodnutí hlavního programátora. Jsou také započaty práce na návrhu vlastního řešení jednotlivých vlastností.

5. Vytváření po vlastnostech

Na závěr jsou naprogramovány jednotlivé třídy dle návrhu z předchozího kroku, které jsou následně otestovány. Pokud projdou potřebnými testy, je daná vlastnost přidána do celkového vyvíjeného produktu.

2.5. Extrémní programování

Metodika klade důraz na schopnost přizpůsobení se neustále se měnícím požadavkům klienta. Staví na 5 základních hodnotách, které jsou pro její využívání zcela klíčové a mezi které patří: jednoduchost, zpětná vazba, komunikace, respekt a odvaha. V rámci extrémního programování je definováno celkem 12 osvědčených praktik, které ukazují, jakých procedur by se měli lidé držet, pokud chtějí tuto metodiku úspěšně aplikovat (Powell-Morse, 2017a). Pro zjednodušení byly za základě (Powell-Morse, 2017a) vybrány na ukázkou pouze některé:

2.5.1. Párové programování

V rámci párového programování bývá jeden úkol svěřen dvojici programátorů, přičemž pravidelně dochází ke střídání „partnerů“ v rámci těchto dvojic.

2.5.2. Plánovací hra

Odehrává se často v podobě meetingu, který se koná v pravidelném a předem stanoveném intervalu, a během kterého se odehrává velká část plánování prací na projektu. Z prvků, kterých plánování dochází, jsou významné především plánování releasů a plánování iterací.

2.5.3. Celý tým

Tato technika zahrnuje do celého procesu kromě řadových členů týmu také klienty a zohledňuje během projektu jejich zpětnou vazbu a připomínky.

3. Oblasti (mimo vývoje), kde je použití agilních metodik vhodné

Agilní metodiky jsou v posledních letech velice populární, a to nejen ve vývojovém týmu. Postupně se Agile přenesl do dalších sfér, které nemusí nijak souviset se softwarovým vývojem, například do managementu, právních oddělení, škol a výroby automobilů. Michael A. Cusumano, který učí na MIT Sloan School of Management, definoval agilnost následovně:

“Agility comes in different forms, but basically it’s the ability to quickly adapt to or even anticipate and lead change. Agility in the broadest form affects strategic thinking, operations, technology innovation and the ability to innovate in products, processes and business models.” (Cusumano, 2011)

Mluví o tom, že se agile vyskytuje v různých formách a jeho hlavní vlastností je rychlá adaptace na změnu a očekávání změny. Dále rozebírá vrstvy podniku, které jsou ovlivněné agilním přístupem – strategické myšlení, technologickou inovaci a schopnost inovovat produkty, procesy a byznys modely. Rychlost a nepředvídatelnost dnešního světa využívá ke zdůraznění, jak moc je tento flexibilní přístup potřebný k úspěchu dnešních manažerů a jimi řízených podniků. Manažeři ve vedoucích pozicích musí být schopní reagovat na změny dnešního světa, které mohou přijít zcela nečekaně, a čím dříve je společnost schopná reagovat na změnu nebo problém, tím dříve dokáže z dané situace vytěžit maximum.

3.1. Využití Scrumu a Kanbanu na univerzitách

Marian Willeke a Scott Marsee, kteří učili na univerzitě v Ohiu, byli součástí malého týmu, který se věnoval vypracování osnovy online kurzů pro univerzitu. V tomto období již využívali prvky Scrumu a Kanbanu pro zvýšení své efektivity a lepší přehlednosti práce. Ze Scrumu přebrali prototypování a iterativní zpětnou vazbu, kterou propojili s vizualizací pracovního toku kanbanu, díky čemuž bylo každému jasné, na čem je třeba pracovat. Díky využití agilních metodik dokázali snížit potřebný administrativní čas na vytvoření kurzu z 20 na 10 hodin. Komunikace uvnitř týmu byla nezbytná pro zajištění této časové úspory a pro ověření, že každý člen týmu přesně ví, co má dělat, a co se od něj očekává. Experti v oblasti daných předmětů, pro které vytvářeli osnovy, byli nadšeni z jejich efektivity a jasné vize, čeho chtějí v daném kurzu dosáhnout. (Willeke a Marsee, 2016)

V roce 2014 byl ale jejich tým postupně rozpuštěn kvůli pokračujícímu růstu univerzity – více studentů, větší fakulta a více studijních programů. V důsledku těchto událostí musel jejich tým projít rekonstrukcí. Scott se stal programovým ředitelem a Marian se ocitl ve vedení oddělení akademických operací a dalších dvou podpůrných týmů. Při této změně a adaptaci na nové podmínky převzali myšlenky Agile Manifestu, které upravili pro potřeby univerzitního prostředí. Původní znění (Agile Manifesto, 2001) má 4 klíčové hodnoty:

- Jednotlivci a interakce před procesy a nástroji.
- Fungující software před vyčerpávající dokumentací.
- Spolupráce se zákazníkem před vyjednáváním o smlouvě.
- Reakce na změny před dodržováním plánu.

Tyto klíčové hodnoty a principy se zaměřují na lidi a jejich spolupráci v týmu. Vysvětlují, že lidé by spolu měli na projektech komunikovat, pomáhat si a věřit, že dodali kvalitní výstupy práce. Pomoci Agile Manifesto chtěli na univerzitě zajistit, že budou dodávat kompletní kurzy pro výuku a že nebudou mít žádná předem stanovená pravidla. Rozhodli se tedy upravit 3 klíčové hodnoty a jednu převzali v původním podobě. Prvořadě byly pro nich vztahy a zkušenosti před modely, naučné kurzy před komplexním designem, spolupráce před statusem a ponechali si reakci na změny před dodržováním plánu. (Agile Manifesto, 2001)

Tato změna způsobila řadu nových problémů, se kterými se museli dále potýkat:

- Komunikace utrpěla a již nebyla tak efektivní jako předtím.
- Při tvorbě kurzů chyběla spolupráce.
- Někteří zaměstnanci měli problém uchopit agilní myšlení.
- Experti na předměty, kterým se věnovali, byli zmatení novými rolmi lidí kolem nich.
- Kurzy musely být po dokončení upravovány kvůli chybám, které způsobovaly zmatek, namísto toho, aby byla vylepšována metoda výuky.
- Role Mariana a Scotta se změnila z vedoucích celého programu, kdy vykonávali všechna rozhodnutí, na pozici, kde ovlivňovali lidi, aby přijali agilní myšlení.

Problém komunikace a chybějící spolupráce řešili pomocí mentorování lidí, kteří navrhují kurzy. Museli se ujistit, že vnímají správně proces tvorby kurzu a potřebné agilní principy. Museli více komunikovat s product ownerem o postupu vývoje, skutečně dodržovat iterativní cykly a dodávat funkční části kurzu. Také chyběly pořádné a pravidelné standup meetingy pro product ownery, tým a scrum mastera. Pro nápravu se začali pravidelně střetávat scrum master a product owner, aby zjistili, jaké jsou potřeby a priority, na které je třeba se soustředit. Následně mohl scrum master apelovat na tým a pomáhat jeho členům pochopit stavy rozpracovaných úloh, pokud se scrum board dostal do neaktuální podoby vůči projektu. Důsledkem byly lépe načasované a propracované žádosti o osnovy kurzů od product ownerů a tým byl více aktivní v udržování scrum boardu v aktuálním stavu. Tyto změny napomohly vytvořit vztahy mezi lidmi, a tím se zlepšila komunikace a přehlednost všech zainteresovaných osob. (Willeke a Marsee, 2016)

Scottova a Marianova adaptace scrumu pro tvorbu kurzů byla velice efektivní a úspěšná, když pracovali v menším týmu a většinu úloh vykonávali svépomocně. Když byl původní tým rozpuštěn, tak byli přeřazeni do většího týmu, kde se přestali podílet na procesu tvorby, a

namísto toho pomáhali lidem kolem nich, aby se soustředili na agilní principy. Čím více lidé znali a využívali agilní přístup, tím lépe všichni fungovali a kvalita práce se zvyšovala. Postupně se jejich struktura pro vývoj kurzů rozšířila na další univerzity v Kansasu, Michiganu, Oregonu, Severní Karolíně a Texasu, kde se opět ukázalo, že agilní přístup v oblasti škol je možný, pokud se nezapomene na komunikaci, spolupráci a adaptivnost na změny.

3.2. Scrum v obchodním oddělení

Scrum byl také adoptován společností AccuRev, konkrétně v oddělení prodeje. Chtěli udržet své řešení jednoduché a vyhnout se zbytečným komplikacím, aby se nedostali do situace, kdy lidé nevědí, co mají dělat, a tak implementovali Scrumový trénink, kdy všechny své zaměstnance provedli školením, ve kterém je učili jednotlivé fáze scrumu – denní standup, plánování sprintu a retrospektivu sprintu. Pokračovali využitím znalostí získaných ze školení a zavedli denní standupy pro obchodní tým. Na těchto střetnutích diskutovali o zpětné vazbě od zákazníků, aby mohli vylepšovat své produkty. Délku jednoho sprintu stanovili na jeden kvartál, což je pro softwarový vývoj nepředstavitelná délka, v oblasti prodeje však dává smysl. Na závěr sprintu se všichni střetli a v rámci retrospektivy diskutovali o všech výsledcích za poslední kvartál, o procesech a jejich změnách, a přemýšleli o možnostech nadcházejícího sprintu. Implementaci Scrumu zakončili Scrum boardem, kde sledovali jednotlivé cíle a úlohy pro daný sprint napříč třemi stavy – backlog, rozpracované a dokončeno. Během implementace zjistili, že obchodní tým již pár agilních prvků přirozeně využívá a celý program byl považován za úspěšný. (Accurev, 2008)

3.3. Agile v testování

Agile má také rozsáhlé uplatnění v testování, kde je aplikován v rámci celého životního cyklu daného produktu. V rigorózních metodikách je testování prováděno až ke konci, před vydáním produktu, čímž se zvyšují náklady potřebné na opravy chyb, které byly v průběhu testování odhaleny. S agilním přístupem přišly společnosti na fakt, že čím dříve chybu odstraní, tím je to méně nákladné. Při agilním testování se s testováním začíná hned na úvod, kdy lze například testovat prvotní dokumentaci produktu, jeho analýzu a poukazovat na chyby, které se tam mohou vyskytovat. Tímto způsobem je možné předejít mnoha problémům, které by v budoucnu mohly zdržet projekt a tím snížit šance na jeho úspěšné dokončení. (ISTQB, 2014)

3.3.1. Testování v metodice Scrum

Scrum nedefinuje přesné techniky, kterých se musí využívat při vykonávání testů. Podoba testů z pohledu testera na projektu, kde se využíval Scrum, je velice detailní. Testování může probíhat v jednotlivých kolech, kdy testování začne týdnem integračního testování a kontrolují se jednotlivá propojení mezi systémy a obsah zpráv jejich komunikace. Následují dva týdny systémových testů, kdy je prověřena stabilita a funkčnost celého systému jako takového. Testování je dokončeno dvěma týdny akceptačních testů, které provádí zákazník a jeho testovací tým. Po celou dobu zároveň probíhá pravidelně komunikace se zákazníkem a jsou upravovány priority jeho požadavků. Na jiném projektu zas může být vyhrazen jeden tým pouze pro integrační testování a druhý tým může vykonávat systémové a akceptační testy přímo pod dozorem zákaznickova test leadera. Dalším případem může být projekt, kde jeden tým vykonává postupně jednotlivé úrovně testů. Všechny tyto přístupy využívají rozličné přístupy k testování, které mohou vést k úspěšnému dokončení produktu a jeho akceptaci zákazníkem. Tento závěr může být přesně jedním z důvodů, proč Scrum nedefinuje přístup k testování – spoléhá se na jednotlivé lidi na projektech, jejich schopnosti, zodpovědnost a že odvedou kvalitní práci pro zákazníka. Scrum je zkrátka efektivní, pokud se dodržují jeho hodnoty a principy. Práce v týmu, kde si kolegové navzájem pomáhají a otevřeně spolu komunikují, je pak mnohem příjemnější a v konečném důsledku i efektivnější. (ISTQB, 2014)

3.3.2. Testování v metodice Extrémní programování

Na druhé straně je Extrémní programování, které klade veliký důraz na testování. Přebírá praktiky Test Driven Developmentu, kde programátor nejprve napíše testy a až následně se věnuje psaní samotného kódu. Tento cyklus pokračuje, dokud nejsou pokryty všechny požadavky zákazníka a všechny testy skončí úspěšně. Programátor píše jednotkové testy, které jsou automatizované a spouští je po každé úpravě kódu. Jednotkové testy většinou testují pouze jednu třídu nebo malou skupinu tříd. Zákazník si vytváří akceptační testy, které testují funkcionalitu celého systému a vyhodnocují, zda bylo dosaženo požadované úrovně kvality dodaného systému. Pokud skončí všechny testy úspěšně, je testování považováno za dokončené. Extrémní programování často využívá párového programování, které by mohlo být uplatněno, mimo oblasti softwarového vývoje, například na operačních sálech, kde dva

chirurgové spolupracují při operaci pacienta nebo v dopravním leteckém průmyslu, kde se dva piloti podílejí na pilotování letadla. (Powell-Morse, 2017a)

4. Předpokládaný budoucí vývoj jednotlivých agilních metodik mimo vývojový tým

Vzhledem k současnému vývoji je zřejmé, že se myšlenky agilních metodik budou i nadále šířit do dalších oblastí lidských činností, které se netýkají pouze informačních technologií. Tyto metodiky se budou s velkou pravděpodobností využívat v různě modifikovaných podobách, nicméně určité základních principy by měly zůstat zachovány.

4.1. Obecné principy agilních technik využitelné v budoucnu mimo vývojový tým

Primární cíl agilních technik, kterým je uspokojit potřeby zákazníka jednoduchým, přesto současně kvalitním, a rychle dodaným produktem podle jeho požadavků, by měl být cílem všech společností a podnikatelů nejen v oblasti informačních technologií. Nemělo by však jít primárně o zisk, nýbrž o vyplnění zákaznických přání. V základu je kladen důraz na jednoduchost, komunikaci se zákazníkem, naslouchání jeho potřebám a požadavkům a neděláním zbytečné práce, ale jen toho, co zákazníkovi přinese očekávaný užitek. Bylo by jistě ku přínosu všech (zákazníků i dodavatelů a společnosti obecně), kdyby se agilní techniky v budoucnu uplatnily ve všech odvětvích, ve kterých je jejich nasazení možné a vhodné.

4.2. Budoucí využití jednotlivých agilních metodik

Vybrané metodiky mají poměrně velký potenciál, týkající se jejich budoucího využití. Níže zmíněné možnosti představují pouze určitou část z celé škály možných využití.

4.2.1. Scrum

Pokud se jedná o hmotný nebo nehmotný produkt či službu, kterou je možné vyvíjet postupně, je podle Scrumu zásadní provádět tuto dodávku (v oblasti vývoje informačních systémů označovanou jako „release“) v krátkých pravidelných intervalech tzv. „sprintech“ nebo „iteracích“. Po dodání zhotovené části/verze produktu si ji zákazník vyzkouší, zjistí, s čím je spokojený a co by si přál jinak, případně přidá, změní nebo zruší další požadavky (Atlassian, 2018a). Agilní techniky Scrumu by v budoucnu v upravené podobě šlo použít

například při návrhu a projektování rodinného domu. Zákazníkovi by byly v pravidelných cyklech dodávány výkresy a příslušná technická dokumentace a on by mohl postupně upřesňovat své požadavky. Ve fázi návrhu by bylo kdykoliv možné, aby zákazník požádal o přidání, odebrání nebo posunutí okna, zdi či celého pokoje a v rámci dalších cyklů by dodavatel, v tomto případě architekt nebo jejich tým, celý stavební plán přepracoval tak, aby bylo staticky proveditelné postavit takovou stavbu podle zákaznických přání. Těžko by se však takto pojaté agilní techniky uplatňovaly například při stavbě mrakodrapu. Pravděpodobně zcela nemožná by za současných technologických a stavebních možností byla konstrukce skeletu pomocí agilních metodik. S ohledem na statiku celé stavby by zkrátka nešlo jednoduše přidat pár dalších pater, byť by si to zákazník přál sebevíc. Jako další názorný příklad omezených možností měnit požadavky v průběhu projektu je možné uvést služby sochařství nebo kadeřnictví. V případě sochařství by dříve nebo později bylo po neustálých změnách požadavků jedinou zbývající možností opracovávaný kámen zahodit a začít tesat sochu (produkt) znovu od začátku do nového bloku kamene. V případě agilního kadeřnictví by zase po neustálém měnění požadavků na délku sestřihu hrozilo, že jedinou možností bude „holá hlava“. Obecně by bylo v budoucnu možné nasadit Scrum s jeho nedlouhými pracovními sprinty, pravidelnými týmovými setkáními a zpětnou vazbou v podstatě ve všech vývojových týmech, nikoliv jen v těch, které se zabývají vývojem informačních systémů. Jedná se například o týmy strojařů a designérů v automobilovém, leteckém a elektrotechnickém průmyslu nebo ve finančním reportování. (OPENVIEW, 2016)

4.2.2. Kanban

Kanban řízený tahem s využitím tzv. „kartiček“ a správou úkolů podle jejich stavu (udělat, rozpracováno a hotovo) je vhodný pro budoucí nasazení v mnoha odvětvích. Kanban se původně neobjevil v informačních technologiích, nýbrž v automobilovém průmyslu. Se způsobem řízení výroby Kanban přišla dle (Atlassian, 2018b) japonská automobilka Toyota a později se Kanban uchytil i při vedení vývoje softwarových produktů. Lze tedy právem očekávat, že Kanban nezůstane pouze v oboru informačních technologií, ale že se šířením jeho základních myšlenek a principů bude rozšiřovat i do dalších odvětví. Žádanou výhodou nasazení Kanbanu je i dříve zmíněné zvyšování konkurenceschopnosti produktu. Vedení podniku, který v budoucnu Kanban nebo jeho upravenou verzi zavede, poskytne kvalitní přehled o jednotlivých úkolech a aktuálním stavu jejich provedení. V budoucnu lze u

Kanbanu předpokládat ještě větší rozšíření v automobilovém průmyslu (Atlassian, 2018b) a podobně jako u Scrumu i širší spektrum působení v elektrotechnickém průmyslu, který je informačním technologiím poměrně blízký.

4.2.3. Lean Development

Obecná pravidla tohoto souhrnu by po malých úpravách šlo využít v široké škále oblastí. Nabízejí se například oblasti elektrotechnického a strojírenského průmyslu, kde je při výrobě komplexního produktu, složeného z mnoha sofistikovaných dílů, kladen velký důraz na základní pravidla Lean Developmentu, jakými jsou eliminace zbytečností, důraz na proces učení, odklad kritických rozhodnutí na poslední chvíli a co nejrychlejší dodávky produktu v krátkých iteracích.

4.2.4. Feature Driven Development

Metodika vývoje řízeného vlastnostmi by mohla být nasazena při tvorbě produktů či služeb na zakázku podle konkrétních přání zákazníků. Spíše než pro „sériovou výrobu“ je vhodná pro jedinečné produkty uspokojující průběžně upřesňované potřeby. V budoucnu by se dalo tohoto přístupu využít v oblastech, kde je uplatňována tvorba celkového modelu a seznamu vlastností, plánování vlastností, jejich návrh a samotné vytváření. Komunikace se zákazníkem je přitom klíčová všech fázích výroby. Konkrétním příkladem možného budoucího nasazení Feature Driven Developmentu může být výroba kuchyňské linky na míru s postupným osazováním skříněk a dřezu podle přání zákazníka. Dalším příkladem může být výroba specializovaného stroje s různými vyměnitelnými moduly.

4.2.5. Extrémní programování

Tuto agilní metodiku je možné využít nejen při psaní kódu, ale v podstatě ve všech odvětvích, kde se pracuje s rychle se měnícími dokumenty. Principy extrémního programování by tak v budoucnu mohli aplikovat například žurnalisti, právníci, marketing a PR. Výhody extrémního programování by se daly v budoucnu aplikovat například ve všech odvětvích, kde dochází ke tvorbě uživatelských příruček a manuálů. Vytvořené dokumenty by mohly být vytvářeny v páru a kontrolovány nebo případně na základě poznámek od dalších členů týmu upravovány či zamítány jako nevhodné.

5. Závěr

Práce je určena pro čtenáře, kteří chtějí získat znalosti o agilních přístupech a možných oblastech jejich využití. V úvodní části práce byl stručně definován pojem „agilní“, a byly uvedeny a krátce popsány některé z nejběžněji používaných metodik a technik patřících do této skupiny. V další části práce byly popsány jednotlivé případy z využití agilního přístupu v oblastech mimo vývoje softwaru a následně v závěrečné části byly nastíněny možnosti budoucího šíření do dalších odvětví. Cílem práce bylo nalezení dalších oblastí mimo vývoje softwaru, kde by bylo možné aplikovat agilní přístup a metodiky. V kapitolách jsme shrnuli možnost jejich využití v univerzitním prostředí, v obchodním oddělení, v testování, ve strojírenském průmyslu a v mnoha dalších. Implementace agilní metodiky pro potřeby univerzity byla vnímána jako úspěšná a proto byla rozšířena do dalších států. Společnost Accurev hodnotila adopci Scrumu pro svůj obchodní tým rovněž pozitivně. Naráželi sice na různé problémy, například pokles efektivity nebo neznalost agilních principů, jejich řešení však byla vždy podobná - školení zaměstnanců za účelem lepšího pochopení agilních metodik, podpora týmové komunikace a zlepšování přehlednosti práce. Z toho plyne závěr, že agilní metodiky je možné využívat v různých odvětvích a podmínkách, pokud je exekuce důkladně promyšlena a skutečně dodržuje agilní principy. Cíl práce považujeme za splněný, práce nabízí čtenářům základní informace o agilních metodikách a poskytuje informace o odvětvích, kde jejich využití není běžné. Omezením práce může být nejistota naplnění závěru týkající se využití agilních metodik v budoucnosti.

6. Zdroje

- ACCUREV, 2008. You're So Agile! Implementing Agile... in a Sales Team?. *Accurev*. [online]. [cit. 2018-04-13]. Dostupné z: <http://accurev.sys-con.com/node/1548765>
- AGILE MANIFESTO, 2001. Manifesto for Agile Software Development. *Agile Manifesto*. [online]. [cit. 2018-04-13]. Dostupné z: <https://www.agilemanifesto.org>
- ALTEXSOFT, 2018. Agile Project Management: Best Practices and Methodologies. *AltexSoft* [online]. [cit. 2018-05-06]. Dostupné z: <https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/>
- AMBLER, Scott, 2014. Feature Driven Development (FDD) and Agile Modeling. *Agilemodeling* [online]. [cit. 2018-04-17]. Dostupné z: <http://agilemodeling.com/essays/fdd.htm>
- ATLASSIAN, 2018a. Scrum. *Atlassian* [online]. [cit. 2018-04-17]. Dostupné z: <https://www.atlassian.com/agile/scrum>
- ATLASSIAN, 2018b. Kanban. *Atlassian* [online]. [cit. 2018-05-07]. Dostupné z: <https://www.atlassian.com/agile/kanban>
- CUSUMANO, Michael, 2011. How to Innovate When Platforms Won't Stop Moving. *MIT Sloan* [online]. [cit. 2018-04-13]. Dostupné z: <https://sloanreview.mit.edu/article/how-to-innovate-when-platforms-wont-stop-moving/>
- GONCALVES, Luis, 2018. What Is Agile Methodology. *Luis-Goncalves* [online]. [cit. 2018-05-06]. Dostupné z: <https://luis-goncalves.com/what-is-agile-methodology/>
- ISTQB, 2014. Foundation Level Extension Syllabus Agile Tester. ISTQB. [online]. [cit. 2018-04-14]. Dostupné z: <https://www.istqb.org/downloads/send/5-agile-tester-extension-documents/41-agile-tester-extension-syllabus.html>
- KARAM, Lea, 2017. Feature driven development. *Apiumhub* [online]. [cit. 2018-04-17]. Dostupné z: <https://apiumhub.com/tech-blog-barcelona/feature-driven-development/>
- MCLAUGHLIN, Mike, 2018. What Is Agile Methodology?. *Versionone* [online]. [cit. 2018-04-18]. Dostupné z: <https://www.versionone.com/agile-101/agile-methodologies/>

OPENVIEW, 2016. SCRUM one, SCRUM all: Why agile isn't just for technical teams. [online]. [cit. 2018-04-14]. Dostupné z: <https://labs.openviewpartners.com/scrum-for-non-technical-teams/>

POWELL-MORSE, Andrew, 2017a. Extreme programming: What is it and how to use it. *Airbrake.io* [online]. [cit. 2018-04-18]. Dostupné z: <https://airbrake.io/blog/sdlc/extreme-programming>

TRIVEDI, Amit, 2017. Lean Software Development: 7 Principles Everyone Should Follow. *Techjini* [online]. [cit. 2018-04-18]. Dostupné z: <https://www.techjini.com/blog/lean-software-development/>

VERSIONONE, 2017. *11th Annual State of Agile Report* [online]. [cit. 2018-05-06]. Dostupné z: <http://www.agile247.pl/wp-content/uploads/2017/04/versionone-11th-annual-state-of-agile-report.pdf>

WIJETILAKA, Shehan, 2016. Agile: The new normal in business growth planning. *Grow-strategy* [online]. [cit. 2018-05-06]. Dostupné z: https://grow-strategy.com/agile_new_normal_business_growth_planning/

WILLEKE, Marian a Scott MARSEE, 2016. Embracing the Agile Mindset for Organisational Change. Agile Alliance. [online]. [cit. 2018-04-15]. Dostupné z: <https://www.agilealliance.org/resources/experience-reports/embracing-agile-mindset-for-organisational-change/>