Semestrální práce ke kurzu

4IT421 Zlepšování procesů budování IT

LS2015/16

Agile Maturity Model

A Software Process Improvement framework for Agile Software Development Practices

Jakub Petr, XPETJ73

Kristýna Immerová, XIMMK00

Datum odevzdání: 9.5.2016

Abstrakt

Cílem této práce je v prvé řadě popsat Agile Maturity Model a jeho jednotlivé stupně ve vztahu ke zlepšování procesů vývoje softwaru. Následně si práce klade za cíl zhodnotit tento model, včetně jeho podobností s Capabilty Maturity modelem.

Tato práce umožní nejen studentům informatických oborů seznámit se s problematikou agilního vývoje a rámce pro jeho zlepšování dle Agile Maturity modelu a pochopit jejich fungování. Mimo jiné věříme, že zpracování této práce v českém jazyce napomůže šíření těchto vědomostí, jelikož materiály na toto téma jsou v českém jazyce dostupné pouze v malém množství.

Klíčová slova

Zlepšování procesů vývoje softwaru, Agilní vývoj softwaru, Agilní model zralosti

Obsah

1 Úvod do problematiky

Pro začátek by bylo dobré, aby si každý uvědomil změny, které v posledních více či méně letech nastaly. Vývoj softwaru se posunuje čím dál, tím více na mobilní zařízení. Skoro na každou činnost si člověk může vybrat aplikaci a v podnicích je už povinností mít zavedený podnikový informační systém.

Tyto změny, které mají velký spád, mají následky i v oblasti lidského přemýšlení nad softwarem a na požadavky na ně kladené. Byznys prostředí se může změnit pomalu ze dne na den, a proto i požadavky zákazníků na software se mohou tak rychle měnit. Navíc zákazník očekává, že vše bude vytvořeno a naimplementováno co nejrychleji, přičemž vše bude podle jeho představ. Je tedy těžké, aby zákazník cítil to zadostiučinění, které očekává.

Na všechny tyto otázky odpovídají agilní metodiky, o kterých bude v této práci řeč. Navíc je třeba, aby byly správně popsány vnitro podnikové procesy, jelikož s těmi budeme při vývoji, úpravě a implementaci pracovat. Aby byla udržena konzistence, podpořen agilní vývoj a zajištěno kontinuální zlepšování procesů, byl vytvořen tzv. Agilní model zralosti (Agile Maturity Model, zkráceně AMM). Oba zmíněné pojmy budou v práci vysvětleny a dále s nimi bude pracováno.

Ke splnění cíle práce bude dále popsán samotný Rámec pro zlepšování procesů agilního vývoje softwaru (tzv. Software process improvement Framework for Agile Software Development Practices). Je totiž důležité, aby si každý uvědomoval chyby používaného rámce a ty se snažil revidovat a napravovat. Kritický přístup by měl být zachován i v závěru práce, kde bude popisovaný rámec zhodnocen a případně zmíněny návrhy na jeho zdokonalení.

1.1 Agilní metodiky

Agilní metodiky vývoje jsou protikladem metod rigorózních, přičemž podstatný rozdíl mezi těmito přístupy můžeme jednoduše popsat takto: Zatímco rigorózní metodiky staví na jednom kompletním zadaní, podle kterého je za určitý časový úsek navrženo, vyvinuto a dodáno kompletní softwarové řešení, agilní metodiky dávají přednost vývoji po menších funkčních částech a průběžné spolupráci se zákazníkem na tvorbě zadání. Díky tomuto přístupu mohou lépe reagovat na případné změny v požadavcích zadavatele či jiná rizika a kvalita výstupu může být posuzována již v průběhu vývoje.

Manifest agilního vývoje zdůrazňuje čtyři hlavní principy, které mají všechny agilní metodiky společné:

"Jednotlivci a interakce před procesy a nástroji.

Fungující software před vyčerpávající dokumentací.

Spolupráce se zákazníkem před vyjednáváním o smlouvě.

Reagování na změny před dodržováním plánu." *(Beck, a další, 2001)*

Z množství agilních metodik můžeme jako nejznámější jmenovat Extrémní programování, Kanban či Scrum. Přestože tyto metodiky vznikaly již v 90. letech 20. století, pro množství nejen českých podniků jsou stále novinkou a to především pro velké korporace, pro které zásadní změna představuje riziko, které nejsou ochotné podstoupit ani na úkor možné vyšší kvality produktu, případně pro vládní instituce a podobně řízené podniky, které se spoléhají na formální moc smluv.

Nelze však předpokládat, že agilní metodiky přinesou okamžitý úspěch. Stejně jako na kteroukoliv jinou metodiku si i na ty agilní musí každý tým a organizace takříkajíc zvyknout a zjistit, jak přesně je využít co nejlépe ve svůj prospěch, přičemž primární je nejprve zajistit nezbytný chod projektů a následně zlepšovat své procesy a především učit se z vlastních chyb.

2 Rámec pro zlepšování procesů agilního vývoje softwaru

Dnešní vývojářské společnosti se zajímají o vývoj a správu svých softwarových produktů tak, aby dosáhli co nejvyšší efektivnosti a účinnosti. Proto tíhnou k tomu, aby měli zavedeny organizované a vhodné vývojářské postupy a praktiky. Teoreticky může být zmíněné organizovanosti a vhodnosti postupů dosaženo pouhým a hlavně rychlým implementováním světově rozšířených a užívaných metodik. Bohužel tomu tak je pouze teoreticky, jelikož prakticky je to velmi složitě proveditelné. Je tomu tak kvůli několika faktorům, mezi které může být zařazen organizační kontext, lidský faktor, firemní kultura a další. Navíc softwarové společnosti, které se soustředí na vývoj, stále mnohdy postrádají znalosti či zkušené odborníky pro identifikaci vhodných metodik pro danou společnost a podmínky, ve kterých se nachází.

Je všeobecně známo, že neexistuje jedna věc, která funguje nebo také vyhovuje všem. To samé platí i mezi metodikami, případně praktikami, které využívají vývojářské společnosti pro vývoj produktů. Jeden a ten samý nástroj, který by byl obecně užívaný, není možné najít hned z několik důvodů. Prvním z nich může být samotná lokalizace firmy, která se samozřejmě může lišit, protože neexistuje žádné centrální místo na zemi, kam by byl soustředěn veškerý vývoj všech aplikací a produktů. Druhým důvodem může být rozdílná kultura a z toho plynoucí odlišnosti požadavků jednotlivých vývojářů. Dalo by se najít dalších nespočet důvodů, proč není možné najít jeden všeobecně užívaný rámec, ale to není účelem této práce. Pro nastavení postupů vývoje produktů je třeba si vybrat z několika desítek možných metodik, které jsou více či méně užívány po celém světě. Samotný výběr a implementace metodiky ovšem nestačí. Aby bylo dosaženo požadovaných výsledků, musí být metodiky akceptována všemi zúčastněnými a dostatečně zažita. Z toho vyplývá i možné přizpůsobení metodik, protože i díky tomu bude dosaženo požadovaných výsledků.

Tím, že si daná firma implementuje metodiku vývoje softwaru, to ale nekončí. Od té doby, co se neustále mění stále více konkurenční byznys prostředí plné neustálých inovací, kulturních změn, požadavků, potřeb a dalších, je nutné neustále reagovat na všechny tyto změny a mít pro to nastaveny určité scénáře a postupy. Dále je třeba, aby metodiky a zažité postupy byly neustále revidovány a zlepšovány, aby docházelo k právě zmíněným neustálým inovacím, zvyšování efektivity a účelnosti. Pro toto zlepšování slouží Rámec pro zlepšování procesů agilního vývoje softwaru (Software Process Improvement Framework for Agile Software Development Practices), o kterém je tato práce, a který bude dále vysvětlený.

Tento rámec navrhli Chetankumar Patel a Muthu Ramachadran již v roce 2009 za účelem dalšího výzkumu v oblasti zlepšování procesů agilního vývoje. Zůstal však ve fázi výzkumu, v praxi není příliš využíván, což se bohužel dá říci i o dalším množství agilních modelů zralosti (Leppänen, 2013). Místo standardizovaného nástroje tak nyní zůstává pomůckou pro sebehodnocení týmů a jejich procesů.

Rámec má za cíl neustálou revizi softwarových procesů za předpokladu porozumění současných procesů. Měl by identifikovat problémy a slabiny, které v zavedených praktikách jsou a kontinuálně je měnit tak, aby bylo neustále dosahováno co nejvyšší efektivity a produktivity. Hlavním cílem celého rámce je zvyšování kvality procesů, které by mělo mít za následek alespoň jeden z následujících skutečností: Zlepšení kvality výsledného produktu, snížení vývojového času nebo snížení vývojových nákladů.

Zlepšení softwarových procesů můžeme dosáhnout několika způsoby, které budou zmíněny níže:

1. Zavedení nové vývojové metodiky
2. Identifikace a eliminace slabin exitujících procesů
3. Hodnocení (benchmarking) současného procesu a porovnání s předešlým nebo jinak souvztažným procesem (referenčním)

První způsob je realizován většinou tam, kde nejsou ještě stanoveny žádné procesní standardy a tím pádem není implementována žádná celopodnikově užívaná metodika. V takovém případě je to nejjednodušší a nejrychlejší cesta, jak efektivně a přitom s nízkými náklady implementovat metodiku, u které jsou představitelné následky a již dokumentované zkušenosti (např. z jiných společností, týmů apod.).

Další dva způsoby by měly být standardně implementovány v již zavedených metodice či metodikách, které daná firma používá. Benchmarking je ostatně používán v procesní standardizaci. Navíc je popsán v osvědčených praxích (best practice) implementace referenčních modelů, které jsou užívány právě při implementaci standardizovaných procesů metodik. Jedním z nejlepších a nejvíce využívaných referenčních modelů jsou znalostní modely, mezi které jednoznačně patří tzv. Capability and Maturity model Integration (CMMI).

Přesto, že již byla stručně popsána důležitost a důvody implementace jak metodik, tak Rámce pro zlepšování procesů agilního vývoje softwaru, je důležité se zamyslet i nad reálnou efektivností implementace a využívání tohoto rámce, jako takového. Studie ukazují, že 70% iniciativ na zavedení různých rámců pro zlepšování procesů selhalo a některé z nich ani vůbec nezačaly. To ukazuje na to, že stále přetrvávají problémy, které jsou mnohdy jen těžko překonatelné. Proto bude proces implementace a využití rámce detailně popsán v dalších kapitolách. Navíc budou zmíněny i různé tipy a triky, které mohou každému, kdo se o implementaci pokusí, pomoct. V neposlední řadě bude proces analyzován a podroben kritickému pohledu nezaujatého studenta se zkušenostmi s agilními metodikami.

2.1 Počáteční úroveň

V probíraném rámci se píše o pěti stupních procesů a prvním z nich je právě počáteční (neboli Initial). Stupně se odvíjejí od tzv. zralosti, kterou jsou procesy ve společnostech ohodnoceny. Zralost se odvíjí od stupně organizovanosti procesů, jejich standardizace, potřeby inovace,

měření výkonnosti apod. Dalo by se říct, že takto zralé procesy jsou v každé firmě, která používá agilní metodiku Scrum (Guo Yin, 2011). Sám název tak napovídá, jak jsou procesy ve společnostech s počátečním stupněm procesů nastaveny.

Procesy jsou tedy nestrukturované, z čehož vyplývá, že nejsou sepsány ani standardy, které by je popisovaly, ani nejsou stanoveny cíle či měřící techniky, které by nabízely zlepšování či změnu procesů. Pokud nějaké ze zmiňovaných věcí sepsány jsou, je to pouze strohý popis, který neobsahuje detailní specifikace. Z toho mimo jiné vyplývá, že takto nastavené procesy se mnohdy pravidelně neopakují, a proto to může např. firemní kultura tolerovat. Společnosti v tomto stupni zralosti dále nemají stanoven proces vývoje software (Patel, a další, 2016).

Z výše uvedeného vyplývá, že buď přímo v procesech, nebo při jejich řízení, bude docházet k problémům. Z nich můžou dále vyplynout další, které budou z takto nastavených procesů pouze vyplývat. Pro ilustraci budou tyto problémy vypsány níže, případně uvedeny pro názornost přímo v reálných příkladech.

Obecně se dá říct, že jde o problémy s termíny, komunikací, kvalitou řešení a náklady (Patel, a další, 2016). Tyto problémy plynou z toho, že nastavení procesů hodně závisí na jednotlivých lidech v organizaci, tudíž nemůže docházet k jejich přesnému plánování. Přesněji řečeno je každé řešení nastaveno právě jedním z řešitelů, a proto není možné průběh procesu opakovat jinak, než aby ho dělal právě jeden a ten samý řešitel. Proces potom může být řešen v podstatě pokaždé jiným způsobem. Postup není přesně stanoven a nemohou s ním být proto seznámeni všichni účastnící. I v důsledku toho dochází k zmiňovaným problémům.

2.2 Úroveň prozkoumaných procesů

Jak už může být předem jasné, druhý stupeň zralosti procesů – prozkoumané procesy (Explored level) - bude mít jistě nastaven podrobnější organizační, měrové či standardizační stránky věci. S tím souvisí i menší četnost problémů, kterým musí organizace čelit. To ovšem neznamená, že v průběhu procesů nemohou nastat žádná selhání. Níže bude popsáno předpokládané nastavení procesů a stejně jako v minulé kapitole budou uvedeny příklady selhání, která nejpravděpodobněji nastávají v těchto společnostech.

Pro takto nastavené procesy by měly být nastaveny cíle, které by měly být naplňovány. Vzhledem k tomu, že jedním ze základních procesů předchozího stupně zralosti bylo plánování, jedním z hlavních cílů je právě plánování projektů. Dalším cílem je zlepšení komunikace a samozřejmě kvality řešení, s kterou byl také problém. Dalším, ale ne posledním

hlavním cílem, je větší orientace na zákazníky a sponzory řešení, protože právě spolupráce s nimi umožňuje mimo jiné i přesnější plánování procesů. Posledním hlavním cílem je zlepšení technik a organizace ohledně požadavků na vývoj. (Patel, a další, 2016)

Z výše uvedených cílů tohoto stupně procesů může plynout, že by mělo docházet k lepšímu a přesnějšímu plánování, v důsledku čehož odpadají problémy s překračováním nastavených termínů a individuálním průběhem procesů podle přidělovaných řešitelů. Dále by mělo docházet k bližší komunikaci dodavatele a zákazníka, jelikož díky tomu se může předcházet problémům, které plynou ze strohé specifikace požadavků. Mimo jiné by to mělo pomoct k tomu, že budou lépe či dřív identifikovány slabší stránky procesů, požadavků, takže obecně řečeno vývoje. Navíc již dochází k prvním krokům v nastavování pojistek v plánování projektu, díky nimž je standardizována specifikace zadávání požadavků apod.

Vzhledem k tomu, že specifikace procesů je stále spíše v počáteční fázi, objevují se problémy, které mohou být způsobeny jednoduše lidskou zapomětlivostí, chabým nastavením praktik a postupů nebo nedostatečnou pozorností směrem k zákazníkovi. Jak již bylo zmíněno, specifikace procesů je stále v rané fázi, a proto postupy nejsou stále zažity a dochází k zapomínání lidí na jednotlivé kroky. Z toho plyne, že neplní všechny náležitosti nebo je plní pouze částečně. S tím souvisí i to, že standardizace programování a postupů při vývoji je velmi chabá. Z toho mohou plynout problémy s vzájemným nepochopením jednotlivých vývojářů. Z čehož dále plyne přetrvávající problém s komunikací. Navíc, vzhledem k tomu, že nejsou nastaveny zmíněné standardy, může nastat i situace, kdy zákazník není plně spokojen s dodaným řešením.

2.3 Úroveň definovaných procesů

V předchozí kapitole byly zmíněny nedostatky, které znemožňovaly, aby byl zákazník plně spokojen s dodaným řešením. Dále bylo uvedeno, že není dostatečně nastavena a zavedena komunikace ať už jde o komunikaci uvnitř týmu nebo směrem k zákazníkovi. Posledním hlavním problémem byla uvedena standardizace programování, která nebyla plně využívána. Všechny tyto problémy jsou proto uvedeny mezi hlavními cíli tohoto definovaného (neboli Defined level) stupně.

Cíle tohoto stupně zralosti tedy jsou následující:

- Spokojenost zákazníka
- Zlepšení komunikace

- Kvalita řešení (neboli software)
- Větší využití vývojářských praktik a standardizace vývoje

Prvním předmětem zlepšení je tedy obecně větší pozornost směrem k zákazníkovi. V důsledku této pozornosti je využívání vývoje řízeného testy, párové programování a dodávání řešení v pravidelných a opakovaných intervalech. Zvýšení pozornosti k zákazníkovi se dále projevuje v tom, že zákazník je přítomen při odhadech prací, zpřesňování požadavků i při hodnocení proběhlého sprintu a poskytuje zpětnou vazbu. Dochází tedy ke sbližování vývojářského týmu se zákazníkem a tím je dosaženo další inovace v oblasti procesů organizace dodavatele (Guo Yin, 2011). Cílem tohoto sbližování je mimo jiné i pomoct dodavatelské straně v nalezení nejenom problémů v procesech zahrnujících stranu zákazníka, ale také jejich řešení.

Toho může být dosaženo pouze tehdy, když je do dokumentu detailně sepsán proces, podle kterého je obecně postupováno. Právě k tomu již v této fázi dochází a může tak docházet k měření a vyhodnocování slabin jednotlivých sepsaných procesů (Patel, a další, 2016).

Dalším důsledkem, který plyne z uvedených cílů, je růst dohledu nad podnikovými vývojářskými a testovacími standardy. Navíc jsou tyto standardy orientovány na daného zákazníka, přičemž je stále udržována inovace v oblasti individuálních praktik jednotlivých lidí. Přesto, že jsou již více využívány standardy a to ve více oblastech najednou, stále nedošlo k tomu, aby byla např. sestavována databáze rizik a jejich ohodnocení. Navíc není využívána ani optimalizace kódu, která přispívá ke zkvalitnění vyvíjeného řešení a tím pádem i ke zvýšení spokojenosti zákazníka. (Patel, a další, 2016)

V tomto stupni zralosti procesu je dále počítáno s iteracemi, které jsou běžně využívány v agilních metodikách, a které jsou použity v rámci vydávání funkcionalit, automatizovaných testů, revizi priorit jednotlivých úkolů apod. Nové funkcionality jsou tak přidávány pravidelně v předem nastavených a dohodnutých intervalech. Zákazník tak ví, kdy mu bude dodána další funkcionalita. Tímto postupem je mimo jiné posíleno plánování a jeho přesnost. Pomocí automatizovaných testů je dosaženo nejenom zvýšení kvality řešení, ale navíc stálého ověřování různých agend řešení. Automatizací se navíc objevuje možnost snížení nákladů, což je také jeden z hlavních cílů jak agilních metodiky, tak probíraného rámce.

V tomto stupni zralosti je již dosaženo větší standardizace, která ale nesouvisí s obecnou standardizací řízení projektu, což stále neeliminuje všechny problémy, které mohou nastat.

Není proto zajištěno, aby docházelo k stejné kvalitě řešení ve všech oblastech vývoje (Guo Yin, 2011).

2.4 Úroveň zlepšených procesů

Když bude brán ohled na to, co bylo napsáno o procesech ve 3. stupni (tj. Managed level), je možné si všimnout, že tyto procesy již nabízí velmi konkrétní data o řízení projektu, nastavení procesů a jejich výkonnost či chybovost. Z toho vyplývá, že pokud bude cílem organizace, aby dosáhla vyšší zralosti procesů, musí využít tyto data k analýze a nalezení možností inovace a efektivnějších řešení. Mimo jiné může být měřena a analyzována spokojenost zákazníků, a tím to samozřejmě nekončí. Byly zmíněny hlavní oblastí měření, ale je především na managementu, jaké oblasti a způsoby měření si zvolí, a které je zajímají.

Cílem organizací, které operují s procesy tohoto stupně zralosti, proto budou následující:

- Posilnění pravomocí a schopností týmu a nastavení odměn
- Posilnění projektového řízení
- Sestavování rizik
- Naplňování dohodnutých termínů na 100%
- Zjednodušení

Cíle byly zmíněny, ale jde o to, aby byly také pochopeny a tím pádem, aby mohly být realizovány prostřednictvím smysluplných kroků. Základním principem je přijímaná odpovědnost napříč celým týmem. Už není hlavním smyslem přidělování odpovědností jednotlivým lidem nebo i celým týmem, ale tým již vnímá komplexní situaci kolem projektu a odpovědnosti sám přijímá. Vzhledem k tomu, že si tuto skutečnost uvědomuje, zároveň hledá taková řešení, která nabízí splnění požadované kvality a zároveň jednoduchou a přímočarou cestu jejího dosažení. Díky tomuto přístupu může tým důsledněji naplňovat očekávání zákazníka a zároveň pomocí optimalizace procesů dosahovat jednodušších a účelnějších postupů. (Patel, a další, 2016)

Navíc, aby bylo možno dosáhnout zmíněných cílů, je třeba spolupráce napříč účastněnými pracovníky na straně dodavatele. Pracovníci, ať už to jsou manažeři či vývojáři, se musí respektovat, spolupracovat a společně identifikovat problémy a hledat jejich řešení. Díky této spolupráci dochází mimo jiné k tomu, že na problém je hleděno z více stran a tím pádem může být nalezeno efektivnějšího řešení.

Dalším krokem k zlepšení procesů je sestavování rizik. Každý projekt je určitým způsobem odlišný. Nastanou i situace, kdy se objeví problém, který ještě řešen nebyl. Z tohoto důvodu je důležité sestavit a ohodnotit možná rizika a hned na začátku projektu se na ně připravit a vymyslet možné alternativy řešení, pokud problém nastane. Jenom tak může být dosaženo buď to, že riziko bude eliminováno nebo se mu tým vyhne. Navíc je důležité si všechny tyto materiály uchovat, aby bylo v budoucnu možné se na tyto minulé chyby podívat a poučit se z nich. Pomocí tohoto i výše uvedených postupů je mimo jiné možné eliminovat plýtvání zdrojů, času, techniky apod. Díky tomu je možné věnovat ušetřený čas na další činnosti, které jsou důležité vzhledem k projektu, vývoje kvalitního řešení apod.

V minulé kapitole byla také zmíněna optimalizace kódu, který píší vývojáři. V tomto stupni zralosti procesu nedochází jenom k optimalizaci kódu, ale rovnou k plánování podobných revizí a kontrol. Je proto dosaženo k stálé a opakované optimalizaci, která je kontinuální. Nejzákladnější optimalizace je např. ukládání dat do databáze. Může být několik způsobů, jak data ukládat, ale tým musí mít v zájmu to, aby dosahoval rychlého, přesného a přitom konzistentního ukládání dat a nedocházelo k nepotřebné redundanci dat apod.

V posledním bodu by mělo být zmíněno, že v tomto stupni zralosti jsou dohadovány akceptační kritéria, díky kterým jsou již předem stanoveny nejdůležitější body, které musí být v řešení obsaženy, a které jsou hlavním zájmem zákazníka. Tým dodavatele tak předem ví, na co se má především zaměřit. Jde tedy o to, aby např. nedocházelo k přílišné optimalizaci designu řešení, když zákazníkovi záleží na tom, aby bylo dosaženo plné automatizace některých účetních praktik.

Tímto stupněm zralosti je dosaženo téměř dokonalého průběhu procesů a implementace všech standardů agilních metodik (např. Scrumu). Jediné, čím se ještě organizace v této fázi moc nezabývají, je optimalizace procesů, jejich revize, měření a stálá inovace jejich průběhu. Navíc kromě řízení rizik ještě nedochází k předcházení dalších problémů, kromě především vývojové oblasti.

2.5 Úroveň zralých procesů

Projekty, které dosáhnou této úrovně zralosti procesů (Mature level) jsou již schopné průběžně dodávat kvalitní software dle požadavků zákazníka, mají dobře nastavené a fungující komunikační procesy, využívají různých metod ke zkvalitnění a zpřehlednění vlastní práce.

Cílem této úrovně je tedy především zajistit, že tomu tak bude i do budoucna a to skrze kontinuální zlepšování procesů.

Klíčové jsou pro tuto úroveň:

- Předcházení problémů a chyb
- Eliminace a řízení nejistot
- Navýšení výkonu projektu

Organizace má prostředky proaktivně identifikovat slabé a silné stránky projektu (Paulk, 1993). Za účelem předcházení problémů jsou pro každou funkcionalitu předem stanoveny akceptační testy, veškerý kód prochází unit testy a tým se snaží přicházet s novými způsoby testování. Dojde-li k chybám či problémům, jsou určeny jejich příčiny a chybná data jsou podrobena analýze za účelem budoucí eliminace podobných problémů.

Součástí zajištění budoucnosti projektu je též vyhledávání a implementace inovativních postupů a technologií a nahrazování zastaralých, na čemž by se měl celý tým zasadit.

Díky zvýšenému důrazu na kvalitu a hladký průběh vývoje tým opět zajišťuje spokojenost zákazníka, která je pro každý projekt stěžejní.

3 Zhodnocení rámce

Popsaný rámec podrobně popisuje zdokonalování agilních procesů ve firmě, přičemž neopomíná zájmy jednotlivých zúčastněných stran. Postupuje od budování základů, kterými jsou například standardizované vyplňování požadavků pro vývoj, na nichž poté staví funkční projekt, schopný dodávat kvalitní software, čímž zajišťuje základní cíl projektu. Následně vede k optimalizaci procesů, snaží se o předcházení rizik a minimalizaci nadbytečné práce, čímž přináší vyšší rentabilitu pro společnost a méně práce a přívětivější prostředí pro vývojářský tým. Finální úrovní zajišťuje budoucnost organizace a jejích projektů kontinuálním zlepšováním postupů a inovací v souladu s principem agilních metodik, flexibilitou vůči změnám požadavků a prostředí, a tím zvýšenou konkurenceschopností.

Rámec klade důraz na zapojení zákazníka do procesu, přesto nespecifikuje jeho přesnou formu, čímž nechává prostor pro potřeby rozdílných organizací a metodik.

Pro každou úroveň jasně specifikuje cíle, kterých je třeba dosáhnout a metriky, dle kterých je jejich dosažení vhodné hodnotit. Je-li to možné, doporučuje konkrétní přístupy vhodné k

dosažení těchto cílů, opět však nechává prostor přizpůsobení rámce specifickým požadavkům organizací a týmů.

Co z rámce sice vyplývá, ale blíže v něm není řešeno, je potřebná úroveň a finanční náročnost členů vývojového týmu a jeho vybavení. Je zřejmé, že k dosažení nejvyšší úrovně procesů je třeba dobře sestavených týmů zkušených lidí přístupných inovacím a sebevzdělávání se znalostmi nejen z oblasti vývoje, ale i plánování, řízení rizik a komunikace se zákazníkem a vedením firmy a stejně tak velmi pravděpodobně specializovaného softwaru na podporu procesů a hardwaru, který bude stačit tempu inovací. Tyto aspekty však mohou být finančně velice náročné a pro některé firmy tudíž hůře dosažitelné, rámec přesto nenabízí možná alternativní řešení pro takové situace.

Kritickým faktore pro úspěch jakéhokoliv projektu je taktéž lidský faktor v komunikaci a to jak týmové tak se zákazníkem. Problémem, který stojí v cestě postupu na další úroveň zralosti procesu, může být například nevhodně sestavený vývojářský tým s vnitřními neshodami nebo malá motivace zákazníka aktivně se podílet na vývoji v rámci agilní metodiky.

Jak již bylo řečeno v kapitole 1.1. Agilní metodiky, existují organizace, pro které inovace v různé míře představují risk, který nejsou ochotny podstoupit. Takové organizace mohou opět mít problém s dosažením vyšších úrovní.

Jako alternativu pro výše popsaný rámec je možné použití modelu CMMI (Paulk, 1993). Tyto dva modely nemohou zapřít jistou podobnost. CMMI model, který je mířený nejen na agilní metodiky ale procesy obecně, lze využít, například je-li firemní kultura méně přístupná agilním metodikám a vývojový proces tudíž spíše pouze obsahuje agilní prvky, či je potřeba rámec jinak upravovat. Protože CMMI model je vhodný pro aplikaci na projekt jako celek a agilní rámec poté detailně na část vývoje, není vyloučeno, aby tyto dva modely byly použity zároveň a navzájem se doplňovaly dle aktuální potřeby podniku.

Další možností je samozřejmě využití některé z mnoha dalších verzí agilního modelu zralosti publikovaných vědeckými pracovníky (Ambler, 2010), (Liu, a další, 2005), (Nawrocki, a další, 2001), (Packlick, 2007) nebo používaných konzultačními společnostmi, například rámec ThoughtWorks Studios (Humble, a další, 2011).

4 Závěr

Cílem této práce bylo především popsat a zhodnotit Agilní model zralosti a s ním související Rámec pro zlepšování procesů agilního vývoje softwaru. V úvodu práce byla přiblížena podstata a současný stav využívání agilních metodik. V druhé kapitole byl podrobně popsán Rámec pro zlepšování procesů agilního vývoje softwaru a jeho pět úrovní. Pro každou úroveň byly popsány její stěžejní cíle a faktory úspěchu s ohledem na všechny zúčastněné strany procesu, případně metody použité pro dosažení těchto cílů, jsou-li součástí rámce. Ve třetí kapitole byl Rámec pro zlepšování procesů agilního vývoje softwaru zhodnocen dle jeho obsahu a vhodnosti pro širokou škálu organizací. Přestože rámec velice dobře popisuje vývoj agilních procesů se zaměřením na výkonnost vývojového týmu, spokojenost zákazníka a rentabilitu pro společnost, nabízí málo řešení, či alternativních řešení mnoha možných problémů v oblasti kvality a kvantity zdrojů a spolupráce v rámci týmu, které některé agilní metodiky neopomíjejí a jsou stěžejní pro úspěšný postup na vyšší úrovně zralosti. Rámec navíc zůstává stále ve fázi výzkumu, doposud nebyl tento ani jiný rámec pro zralost agilních modelů standardizován. I přes tyto nedostatky byl popsaný rámec shledán vhodným pro sebeposouzení vývoje agilních procesů v rámci společností.

5 Bibliografie

Ambler, Scott W. 2010. The Agile Maturity Model (AMM). *Dr Dobb's.* [Online] 1. Duben 2010. [Citace: 8. Květen 2016.] http://www.drdobbs.com/architecture-and-design/the-agile-maturity-model-amm/224201005.

Beck, Kent, a další. 2001. Manifesto for Agile Software Development. *Manifesto for Agile Software Development.* [Online] 2001. [Citace: 26. Březen 2015.] http://agilemanifesto.org/.

Guo Yin, Alexandre Paulo. 2011. Scrum Maturity Model. [Online] Září 2011. [Citace: 8. Květen 2016.] https://fenix.tecnico.ulisboa.pt/downloadFile/395143154008/thesis.pdf.

Humble, Jez a Rolf, Russel. 2011. The Agile Maturity Model: Applied to building and releasing software. *ThoughtWorks.* [Online] ThoughtWorks, Inc., 2011. [Citace: 2016. Květen 2016.] http://info.thoughtworks.com/rs/thoughtworks2/images/agile_maturity_model.pdf.

Leppänen, Mauri. 2013. A Comparative Analysis of Agile Maturity Models. [autor knihy] Rob Pooley, a další. *Information systems Development.* New York : Springer Science + Biúsiness Media New York, 2013.

Nikitina, Natalja. 2014. *Software Process Improvement Framework.* [online] místo neznámé : KTH, School of Information and Communication Technology (ICT), Software and Computer systems, SCS., 2014. ISBN 978-91-7595-002-0.

Patel, Chetankumar a Ramachandran, Muthu. 2016. Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices. *ResearchGate.* [Online] Leden 2016. [Citace: 8. Květen 2016.] https://www.researchgate.net/profile/Muthu_Ramachandran/publication/45227382_Agile_Maturity_Model_AMM_A_Software_Process_Improvement_framework_for_Agile_Software_Development_Practices/links/09e4150c08b33558c1000000.pdf.

Paulk, Mark. 1993. Capability maturity model for software. *Encyclopedia of Software Engineering.* 1993.

Schweigert, Tomas, a další. 2013. Agile maturity model: analysing agile maturity characteristics from the SPICE perspective. [Online] © 2013 John Wiley & Sons, Ltd., 24. Říjen 2013. [Citace: 8. Květen 2016.] http://onlinelibrary.wiley.com/doi/10.1002/smr.1617. DOI: 10.1002/smr.1617. ISBN 10.1002/smr.1617.

VersionOne, Inc. © 2016. *Agile Project Management Software for Agile Development.* [Online] VersionOne, Inc., © 2016. [Citace: 8. Květen 2016.] https://www.versionone.com.

Lui K., Chan K. 2005. *A road map for implementing extreme programming*. Proceedings of international software process workshop (SPW 2005), Beijing, pp 474-481

Nawrocki J., Walter B., Wojciechowski A. 2001. *Towards the maturity model for extreme programming*. Proceedings of software quality (ECSQ 2002), LNCS 2349, pp 288-297

Packlick J. 2007. *The agile maturity map: a goal oriented approach to agile improvement*. Proceedings of AGILE 20007 conference, pp 266-271