

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2016/2017
Autoři	David Huňka xhundo00, Jan Mottl xmotj10, Filip Vacula xvacfo1
Téma	Standardizing Requirements Descriptions on Scrum Projects for Better Development and Testing Quality
Datum odevzdání	14. 5. 2017

Abstrakt

Práce se věnuje hlavním problémům a rizikům, které vyplývají ze zápisu požadavků při používání agilní metodiky Scrum. Ukazuje na nedostatky této metodiky z hlediska standardizace zápisu požadavků a navrhuje postup, jak tyto nedostatky odstranit. Dílčím tématem práce je objasnění důležitosti softwarových nástrojů pro správu požadavků a přehled několika z nich.

Klíčová slova

Scrum, řízení požadavků, agilní přístup, standardizace požadavků

Obsah

Abstrakt.....	1
Klíčová slova	1
Obsah	2
1 Úvod.....	3
2 Scrum.....	4
3 Definice a řízení agilních požadavků.....	5
3.1 Definice a řízení požadavků ve Scrumu	5
3.2 Decomposition	6
4 Standardizace popisu požadavků	7
4.1 Životní cyklus požadavku	7
4.2 Různé pohledy na požadavek.....	7
4.3 Nevýhody nestandardizovaných požadavků	7
4.3.1 Problémy s odhadem pracnosti	8
4.3.2 Nejasnosti a trhliny.....	8
4.3.3 Promarněný čas a snaha	8
4.4 Možnosti standardizace požadavků	8
4.5 Standardizace akceptačních kritérií.....	8
5 Agilní softwarové nástroje pro řízení požadavků	10
5.1 Nevýhody obecných nástrojů	10
5.2 Výhody specializovaných nástrojů	10
5.3 Postup výběru specializovaného nástroje	10
5.4 Přehled agilních softwarových nástrojů.....	11
6 Závěr	13
Literatura	14

1 Úvod

Používání agilních přístupů pro řízení projektů je dnes úplně běžnou praxí napříč celým spektrem zaměření společností. Právě proto je na místě se zabývat jejich nedostatky a zjišťovat, jak tyto nedostatky odstranit.

Klíčovou komponentou agilního projektového řízení je právě zápis požadavku, který má více či méně aktivní roli v průběhu téměř celého projektu, a proto může být často základem problémů a rizik na projektu.

Cílem semestrální práce je tedy přiblížit důležitost sběru a řízení požadavků v rámci projektů z oblasti IT využívajících metodiku Scrum a objasnit, jaké výhody plynou ze sjednocení formátu popisu požadavků. Vedlejším cílem práce je objasnit, proč je důležité se zabývat i softwarovými nástroji pro správu požadavků.

2 Scrum

Scrum je metodika řízení produktového vývoje. Vývoj podle scrumu se iterativně opakuje a aplikuje metodu přírůstků. Existuje několik základních rolí a praktik charakterizujících scrum.

Hlasem zákazníka je product owner. Ten tvoří seznam požadavků a zapisuje je do product backlogu. Z tohoto backlogu si vývojářský tým během plánování sprintu vybere požadavky a zařadí si je do sprint backlogu. Následuje samotný sprint, který trvá běžně od dvou do čtyř týdnů. V průběhu sprintu se tým každý den schází na stand-up meetingu a každý člen shrne, co zvládl minulý den a co plánuje na další. Scrum master v průběhu projektu kontroluje průběh a práci vývojářů. Na konci sprintu by měly být požadavky splněny a systém připraven na release. Sprint končí retrospektivou jeho průběhu a nadchází plánování sprintu nového – tvorba sprint backlogu a tak dále.

3 Definice a řízení agilních požadavků

Agilní metodiky mohou vytvářet dojem, že dokumentaci opomíjejí a nepovažují ji za nutnou, naopak má být zbytečná. Nicméně agilní přístupy hovoří spíše o tom, že se snaží vytvářet v první řadě software než vyčerpávající dokumentaci. Tudíž dokumentace nemá chybět, ale má být stručnější, výstižnější a nenabývat zbytečné délky. Ohledně dokumentace by se měla klást otázka, jakou hodnotu konkrétní dokument přináší. Otázka hodnoty je základní stavební kámen agilních metodik a Lean Thinking. (Moccia, 2012)

Koncept agilní definice a řízení požadavků (requirements definition and management, RDM) není novinkou, ale není jednoduché zakomponovat tradiční životní cykly požadavků do agilní činnosti. Pro funkční systém je potřeba pracovat s celým životním cyklem a ne se soustředit pouze na okamžik vývoje softwaru. Pokud se člověk podívá na dokumentaci z celkového pohledu ve vyšší hierarchii podniku, zjistí, že když vzniká funkční program v podobě kódu, často až polovina dokumentace, která vznikla v rámci konkrétního projektu, není nikdy použita. (Moccia, 2012)

To může být způsobeno komplikovaností. Například když vznikne softwarový systém, tak postupem času se dále vyvíjí. Stává se komplikovaným, takže je složitější ho pochopit, organizovat, řídit a udržovat. I pokud nastanou změny v samotné organizaci, které systémy slouží a změní se pravidla, regulace, nebo podnikatelské prostředí, systémy se musí dále přizpůsobovat. Toto vše působí růst komplikovanosti systémů do bodu, kdy ani nefungují správně a neplní to, co mají. Z tohoto důvodu jsou nejlepší a nejsnáze fungující systémy ty nejjednodušší a bez jakýchkoli součástí navíc. Tuto problematiku lze aplikovat i na definici a řízení požadavků. Stává se, že poskytovatel dostane specifikaci požadavků, která čítá stovky stran textu a je nadmíru obtížné takové požadavky naplnit a kontrolovat. (Moccia, 2012)

3.1 Definice a řízení požadavků ve Scrumu

Ve Scrumu probíhá přírůstkový vývoj systémů ve dvou až čtyřtýdenních sprintech. Než se spustí sprint, zaznamenávají se požadavky v product backlogu. Při plánování sprintu jsou požadavky z product backlogu rozčleněny do sprint backlogů. Členění pobíhá na základě diskuse vývojářského týmu, co má největší prioritu, co nejvíce spěchá a co má největší váhu. Práce je řízena product ownerem a celý proces je řízen a kontrolován scrum masterem. Podnik má za úkol plnění product backlogu a reakce na vznikající dotazy o dovysvětlení, jaký produkt má být vyvíjen a to před začátkem sprintu. Problém vzniká, protože organizace nedrží tempo a/nebo produkt backlog není dostatečně definovaný aby odpovídal vývojovým sprintům. (Moccia, 2012)

Agilní definice a řízení požadavků jsou určeny právě k překonávání těchto překážek, k předstihnutí vývojářů v organizaci, aby měli vždy rychlejší nárůst v backlogu než stihnou programovat řešení. Lze buď plnit backlog právě v čas kdy požadavky dojdou a nebo tvořit určitou zásobu požadavků, na které následně přijde čas ve vývoji. Každopádně je třeba scrumový backlog plnit tak, aby stíhal koloběh sprintů podle toho, jak dlouhé sprinty skutečně jsou a stejně tak důležitá je konkrétnost, jednoduchost a prioritizace zadání. Správného toku požadavků lze dosáhnout tak, že životní cyklus požadavků napodobí ten vývojový, ale vždy bude několik kroků před ním. Cílem je vytvořit proces, kde požadavky jsou kontrolovány, organizovány a komunikovány v dostatečné rychlosti a kvalitě. (Moccia, 2012)

Prvním krokem je naplnit backlog požadavků. To je seznam položek, které potřebují být nadefinovány za cílem naplnit product backlog. Konečná fáze jsou úkoly, vizualizace, funkční požadavky atp. Pomocí plánování a prioritizace určí tým řízení požadavků na základě podnikové strategie a plánů, co musí být nadefinováno a vyvinuto. Stejně jako vývojářský tak i požadavkový tým plánuje sprint, provede práci a zhodnotí výsledky. Pokud výsledky splňují očekávání, mohou být posunuty do product backlogu. (Moccia, 2012)

3.2 Decomposition

Decomposition (dekompozice, rozklad, rozpad, ...) je důležitý proces v RDM v rámci kterého je product backlog vykomunikován a vypilován společně s vývojovým týmem. Často vývojový a požadavkový tým zavede komunikační kanály, které dotáhnou položky backlogu do optimálního stavu (grooming the backlog, učesání backlogu). (Moccia, 2012)

Občas vzniká problém tím, že definice požadavků předběhne vývojové sprinty příliš. Pokud se tento časový náskok zvětší dostatečně, mohou nastat komplikace, když poté nebudou dostupní členové týmu, znalosti nebo celkové schopnosti týmu. Tehdy se v rámci dekompozice využije product backlog k další komunikaci a sdílení požadavků. (Moccia, 2012)

4 Standardizace popisu požadavků

V případě, že se metodika pro řízení projektů zabývá sběrem požadavků, je téměř jisté, že bude pro jejich zápis silně doporučovat, nebo přímo vyžadovat používání pevně stanovené formy a struktury. To je velice logický krok, jehož motivace se dá objasnit pohledem na to, kdo s požadavky v průběhu projektu přichází do styku a jakým způsobem s nimi manipuluje.

4.1 Životní cyklus požadavku

Ve většině případů vzniká představa o požadavku v mysli zákazníka, který se společně s vedoucím projektu snaží co nejlíp tento požadavek zformulovat a zapsat. Následně mohou být požadavky kontrolovány dalšími zástupci zákazníka, aby se zajistilo, že jsou správně zformulovány a vedou k naplnění cíle projektu.

Definicí požadavků však jejich životní cyklus zdaleka nekončí, právě naopak – s požadavky se pracuje po celou dobu trvání projektu. Po zformulování požadavků následuje ohodnocení jejich pracnosti a přiřazení priority, aby je bylo možné přiřazovat do časových úseků, na které je projekt rozdělen – konkrétně v metodice Scrum se například jedná o sprinty.

Při samotné implementaci požadavků se snad nejvíce projevuje kvalita formulace požadavku, protože kvalita implementace je většinou přímo závislá na kvalitě definice požadavku. S požadavky se však intenzivně pracuje i při testování jejich implementace a při akceptaci celého projektu.

4.2 Různé pohledy na požadavek

Z popisovaného zjednodušeného životního cyklu požadavku v průběhu projektu vyplývá, že s jediným zápisem určitého požadavku pracuje velké množství lidí, kteří na projektu zastupují různé a častokrát diametrálně odlišné role. Právě z tohoto důvodu je potřebné mít požadavky definované co nejdůkladněji a co nejsrozumitelněji, aby byl pohled všech lidí na určitý požadavek sjednocený a ideálně se nenabízel žádný prostor pro diskuzi. V případě, že je požadavek definován fádne, respektive nedůsledně, tak si každý člověk může vyložit stejný zápis požadavku různě a právě z důvodu této rozdílné interpretace požadavku často dochází k tomu, že je v rámci projektu dodáno něco jiného, než bylo původně požadováno – což může v extrémním případě ohrozit dosažení samotného cíle projektu.

V popisovaném životním cyklu byla navíc vynechána jedna specifická, ale často se vyskytující situace, kterou je přiřazení nového člověka na již rozběhnutý projekt. V takovém případě se nový člověk potřebuje v co nejkratším čase seznámit s požadavky a správně je pochopit, avšak není s projektem obeznámen tak důsledně jako jeho kolegové, kteří na projektu pracují delší dobu a vnímají tacitní kontext požadavků. Z tohoto důvodu je schopnost nového člověka na projektu správně interpretovat požadavky omezena, což je problém zejména u nedostatečně popsanych požadavků, případně u požadavků, které se ve svém popisu nedrží předepsané struktury.

4.3 Nevýhody nestandardizovaných požadavků

Nejčastější komplikace, způsobené žádnou nebo nedostatečnou úrovní specifikace požadavků, se dají shrnout do tří kategorií:

- a) Problémy s odhadem pracnosti
- b) Nejasnosti a trhliny
- c) Promarněný čas a snaha

4.3.1 Problémy s odhadem pracnosti

V případě, že popis požadavku není jasně standardizovaný, je mnohem složitější odhadovat pracnost, kterou si implementace požadavku vyžaduje. To například proto, že se popis požadavku dá poměrně jednoduše nesprávně interpretovat, takže výsledný odhad nemusí do značné míry korespondovat s původním požadavkem, protože místo toho koresponduje s jeho špatnou interpretací.

Na druhou stranu se můžou například vyskytnout nezodpovězené otázky, které vycházejí z nedostatečného popisu požadavku, přičemž na správný odhad je nutné tyto otázky nejdříve zodpovědět.

4.3.2 Nejasnosti a trhliny

Nejasnosti a trhliny ve vysvětlení podstaty požadavku vznikají ze stejných důvodů jako problémy s odhadem pracnosti, avšak nevznikají už u odhadování pracnosti, ale až později při jejich implementaci. Z projektového pohledu je tato komplikace mnohem horší, protože má zásadnější dopad na zdroje potřebné pro dokončení projektu než předešlá komplikace.

4.3.3 Promarněný čas a snaha

Poslední komplikace je úzce spjatá s dvěma předešlými – když se špatně vyloží nebo pochopí smysl požadavku, je kromě času a snahy na implementaci takového požadavku nutné přičíst i čas a snahu potřebnou pro zjištění správného popisu požadavku a provedení změny v implementaci, čímž se opět prodlužuje čas potřebný na dodání požadavku.

4.4 Možnosti standardizace požadavků

Metodika Scrum doporučuje zápis požadavků realizovat formou user stories, které mají jasně definovaný popis požadavku:

- **Jako** <uživatel/role>
- **chci** <nějakou funkcionalitu>,
- **abych mohl** <benefity funkcionality>.

Kromě toho by user story měla obsahovat ještě akceptační kritéria, na základě kterých se dá jednoznačně rozhodnout, jestli je požadavek splněn nebo ne. Na rozdíl od popisu user story však Scrum pro akceptační kritéria neuvádí žádnou jasnou strukturu, které by se měly držet. Právě zde se nachází prostor pro doplnění metodiky o vlastní prvek, kterým je právě určení standardní formy zápisu akceptačních kritérií user story.

4.5 Standardizace akceptačních kritérií

Elena Belkovskaya ve svém článku (Belkovskaya, 2016) doporučuje držet se při zápisu akceptačních kritérií následující formy:

- **Za předpokladu** <kontext, nebo předpoklad>
- **a** <další kontext nebo předpoklad> ,
- **když** <nastane událost, nebo uživatel vykoná akci> ,
- **pak** <očekávaný výsledek – co se má stát pro naplnění požadavku z user story> .

Počet vymezených předpokladů není v rámci jednoho akceptačního kritéria omezen. To znamená, že jediné akceptační kritérium může obsahovat jeden, dva, nebo i více předpokladů anebo kontextů.

Používání uvedené formy zápisu akceptačních kritérií zvyšuje úroveň standardizace zápisu požadavků. Díky tomu se předchází většímu množství rizik v porovnání se situací, kdy zápis akceptačních kritérií není žádným způsobem standardizován. Autorka konkrétně uvádí tyto benefity:

1. **Úplná jasnost.** Používání standardní šablony podněcuje všechny účastníky ke kladení otázek, které by je za jiných okolností nenapadly. Díky tomu je zodpovězeno více nejasností a popis požadavku je přesnější.
2. **Plné krytí.** Je zvýšena pravděpodobnost, že všechny možné průběhy scénáře, včetně alternativních, budou odhaleny a popsány ještě před zahájením vývoje.
3. **Vyšší produktivita.** Výsledkem předešlých dvou výhod je, že se zvýší produktivita práce celého týmu.
4. **Dokumentace každého požadavku, procesu nebo vlastnosti.** Když jsou požadavky dostatečně standardizovány, dají se pak jednodušeji použít jako základní dokumentace projektu. Lidé pak můžou hledat odpovědi na otázky přímo v user stories.
5. **Lepší konečný produkt.** Všichni členové týmu mají lepší možnost seznámit se s fungováním celého komplexního systému a pochopit, co zvyšuje kvalitu konečného produktu.
6. **Úspora času na komunikaci.** Obzvláště při outsourcingu projektů můžou všichni čerpat z výhod přesné specifikace požadavků a neplýtvat čas vyjasňováním nejasností.
7. **Vyšší kvalita implementace.** Celková kvalita implementace je na vyšší úrovni, protože jsou požadavky lépe pochopitelné a je zredukován prostor pro jejich špatné vyložení.
8. **Lehčí úpravy požadavků nebo scénářů.** Jestliže je pro psaní detailních požadavků vyhrazen dostatečný čas, jde pak jejich úprava provést téměř bez námahy.

5 Agilní softwarové nástroje pro řízení požadavků

Pokud se podnik rozhodne pro standardizaci popisu požadavků, je to rozhodně krok ke zlepšení efektivity a úspěšnosti projektů. Nicméně je také důležité věnovat patřičnou pozornost výběru vhodného softwarového nástroje pro správu požadavků.

5.1 Nevýhody obecných nástrojů

Mnoho společností využívá ke správě požadavků nástroje jako Excel nebo Word, které nejsou pro tento účel příliš vhodné. U větších projektů, kde se může vyskytovat velké množství user stories a akceptačních scénářů, by byly dokumenty s popisy požadavků vytvořené ve Wordu nebo Excelu velmi obsáhlé, nepřehledné a obtížně udržitelné. Vyhledávat v nich určité požadavky, aktualizovat je a zaznamenávat mezi nimi vztahy by jistě bylo poměrně obtížné a nepraktické. Z těchto a mnoha dalších důvodů nejsou tyto nástroje vhodnou volbou pro správu požadavků a je vhodné porozhlédnout se po specializovaných nástrojích (Shrivathsan, 2009).

5.2 Výhody specializovaných nástrojů

Specializované nástroje pro správu požadavků mají oproti obecným nástrojům řadu výhod. Mezi nejvýznamnější z nich patří:

- strukturované požadavky – Specializované nástroje umožňují definovat pro požadavky určitou strukturu. Takže například lze požadavku přiřadit určité atributy (žadatel, požadované datum splnění) a nástroj zkontroluje, jestli každý požadavek má tyto atributy skutečně vyplněné (Shrivathsan, 2009).
- úspora času – Specializované nástroje ušetří uživatelům spoustu času, protože automatizují mnoho úloh spojených se správou požadavků a práce s nimi je díky tomu rychlejší a pohodlnější (Shrivathsan, 2009).
- osvědčené postupy – Kvalitní nástroje obsahují osvědčené postupy spojené se správou požadavků. Podnik tak může koupit nástroje získat dokonce i určité know-how, které mu může ještě více pomoci se správou požadavků a úspěšností projektů (Shrivathsan, 2009).
- snadná spolupráce – Specializovaný nástroj umožňuje spolupracovat s interními i externími zainteresovanými stranami účinně a efektivně na rozdíl od obecných nástrojů, které nepodporují spolupráci specifickou pro úlohy týkající se správy požadavků (Shrivathsan, 2009).

5.3 Postup výběru specializovaného nástroje

Než se podnik rozhodne pro výběr specializovaného nástroje pro správu požadavků, musí si být vědom toho, že ani ten nejlepší nástroj na světě nevyřeší všechny problémy. Je potřeba si uvědomit, že kvalitní nástroj může velmi zefektivnit a usnadnit práci, avšak není klíčem k úspěchu. Tím je především existence nějaké metodiky pro řízení požadavků v

podniku, schopnosti a znalosti zaměstnanců a snaha o kontinuální zlepšování. Je-li si toho podnik vědom, může začít s výběrem vhodného nástroje (Beatty aj., 2016, s. 29).

Při výběru specializovaného nástroje je potřeba si nejprve důkladně stanovit kritéria pro výběr. Napříč podniky bude určitě mnoho kritérií stejných, ale každý podnik bude mít pravděpodobně také své specifické požadavky na nástroj, které musejí být při výběru zohledněny. Pokud by byly takové požadavky opomenuty, mohlo by to vést k volbě nevhodného nástroje, který by nemohl být efektivně nebo vůbec používán a znamenalo by to zbytečné plýtvání času i finančních prostředků.

Dalším krokem je analýza trhu a výběr specializovaných nástrojů pro správu požadavků, které budou následně podrobeny vyhodnocení na základě stanovených kritérií. Do první fáze vyhodnocení je potřeba vybrat větší množství nástrojů, aby nebyly přehlédnuty ty, které by mohly být pro podnik výhodné (Beatty aj., 2016, s. 6).

V první fázi vyhodnocení se stanoví několik základních kritérií, které jsou pro podnik klíčové a které by zvolený nástroj měl podporovat. Na základě těchto několika zvolených klíčových kritérií se vyřadí všechny nástroje, které je nesplňují. Pokud stále zbude příliš mnoho nástrojů, je dobré na základě dalších kritérií jejich počet snížit na nějaké rozumné číslo. Kdyby například 40 nástrojů prošlo do další fáze, bylo by to zbytečně časově i finančně náročné (Beatty aj., 2016, s. 6).

Ve druhé fázi vyhodnocení se nachází již podstatně menší počet nástrojů, které budou detailněji vyhodnocovány na základě veškerých kritérií stanovených podnikem. Lze si například ohodnotit jednotlivá hodnotící kritéria váhami podle důležitosti pro daný podnik a pro jednotlivé nástroje pak přiřazovat kritériím hodnoty podle toho, jak moc toto kritérium splňují. Třeba určit rozmezí hodnot od 0 do 3, kde 0 znamená, že nástroj kritérium nesplňuje vůbec a 3 znamená, že nástroj dané kritérium zcela splňuje. Výsledné hodnocení kritéria pak můžeme spočítat jako váhu x míru splnění a hodnocení nástroje jako sumu hodnocení všech kritérií. Spočítáme-li bodové hodnocení pro všechny nástroje v druhé fázi a jejich výsledky seřadíme od nejvyššího počtu bodů po nejmenší, tak zjistíme, jaký nástroj je nejvhodnější. Pokud jsou správně stanovena kritéria, jejich váhy a kritéria pro jednotlivé nástroje jsou pečlivě vyhodnocena, tak by měl být vybrán ten správný nástroj pro implementaci v podniku (Beatty aj., 2016, s. 7).

5.4 Přehled agilních softwarových nástrojů

Společnost Seilevel v roce 2016 zveřejnila výsledky velmi komplexního průzkumu trhu a vyhodnocení nástrojů pro správu požadavků. Zpráva s vyhodnocením je volně dostupná na internetu, takže ji každá firma může použít při výběru nástroje. Jak ovšem sama společnost Seilevel říká, tak podniky by neměly brát výsledky za samozřejmé, protože proces výběru je v každém podniku trochu jiný. Každý podnik si musí především stanovit svoje vlastní kritéria pro výběr a přiřadit jim vlastní váhy podle důležitosti (Beatty aj., 2016, s. 8). Vyhodnocení nástrojů společností Seilevel je však velmi komplexní a proto nabízí podnikům minimálně přehled o velkém množství nástrojů na trhu a detailnější popis těch nejlepších podle společnosti Seilevel.

Mezi čtyři nejlepší nástroje pro správu požadavků se zařadily hned tři, které velmi dobře podporují agilní přístupy. To ukazuje, jak jsou v současnosti agilní metodiky ve firmách rozšířené a softwarové nástroje pro správu požadavků proto zahrnují jejich podporu. Mezi zmíněné agilní nástroje patří:

1. TopTeam Analyst (1. místo) – TopTeam Analyst získal nejvíce bodů a to především za pozoruhodné modelovací možnosti, zobrazení pohledů a dashboardů unikátních pro

konkrétní zvolenou roli (product manager, zákazník atd.), přehledné uživatelské rozhraní a snadné ovládání, propracovaný administrátorský modul a správu dokumentů a mnoho dalšího. TopTeam Analyst obsahuje také agilní modul, díky kterému lze nástroj přizpůsobit i agilním metodikám (Beatty aj., 2016, s. 10).

2. Blueprint (3. místo) – Blueprint se umístil na třetím místě zejména díky obsáhlým modelovacím možnostem zahrnujících průběhy procesů nebo vizuální případy užití, integraci s MS Office Word, Excel a Visio pro import a export dokumentů, vysoce flexibilní datový model s množinou základních objektů, které mohou být přizpůsobeny využitím vlastních atributů a naplnit tak specifické potřeby organizací (Beatty aj., 2016, s. 13). Podle Ruth Zive, viceprezidentky marketingu z Blueprint Software Systems, dnes většina firem používá hybridní agilně-vodopádový přístup, který podporuje i Blueprint (Zive, 2016).
3. Jama (4. místo) – Mezi silné stránky nástroje Jama patří velmi flexibilní datový model s možností vytvoření vlastních typů položek a atributů a zachycení vztahů mezi nimi, podpora jakéhokoliv přístupu k vývoji softwaru včetně velmi dobře podporovaných agilních přístupů a robustní podpora rolí a oprávnění, které se snadno aktualizují a spravují (Beatty aj., 2016, s. 14).

6 Závěr

V práci byly přiblíženy nejčastější problémy, které mohou nastat v důsledku nedostatečně standardizovaných požadavků na projektu. Dále byl prezentován přístup metodiky Scrum k této standardizaci společně s jeho nedostatky. Následovalo doporučení pro odstranění těchto nedostatků prostřednictvím standardizace zápisu akceptačních kritérií společně s přínosy aplikace této šablony v praxi.

Také byly představeny výhody softwarových nástrojů pro správu požadavků, nevýhody používání obecných nástrojů, možný postup jejich výběru a přehled několika profesionálních nástrojů.

Původní cíl práce byl tedy naplněn.

Literatura

- ❖ ABEL, Anders, 2012. Requirements in Scrum. In: *Passion for Coding* [online]. 29.3. [cit. 2017-03-12]. Dostupné z: <https://coding.abel.nu/2012/03/requirements-in-scrum/>
- ❖ BEATTY, Joy, STOWE, Megan, CARDENAS, Amanda, REINHARDT, David a BARTLETT, Jonathan, 2016. Requirements Management Tool Evaluation Report. In: *Seilevel* [online]. [cit. 2017-03-12]. Dostupné z: <http://assets.cdnma.com/13314/assets/Website%20Downloads/2016-Seilevel-RequirementsTool-Evauation-Report-FINAL.pdf>
- ❖ BELKOVSKAYA, Elena, 2016. Standardizing Requirements Descriptions on Scrum Projects for Better Development and Testing Quality. In: *InfoQ* [online]. 24.10. [cit. 2017-03-11]. Dostupné z: https://www.infoq.com/articles/standardize-requirements-scrum?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=09272016news
- ❖ CHENG, Calvin, 2014. Why scrum? Why agile development? In: *Calvin's* [online]. [cit. 2017-03-12]. Dostupné z: <https://calvinx.com/2014/05/22/why-scrum-why-agile-development/>
- ❖ MOCCIA, Jason, 2012. Agile Requirements Definition and Management. In: *Scrum Alliance* [online]. 10.2. [cit. 2017-03-11]. Dostupné z: <https://www.scrumalliance.org/community/articles/2012/february/agile-requirements-definition-and-management>
- ❖ SHRIVATHSAN, Michael, 2009. Requirements Management Tools – Overview. In: *Product Management Insights: Practical Insights into Software Product Management* [online]. 30.7. [cit. 2017-03-12]. Dostupné z: <http://pmblog.accompa.com/2009/07/30/requirements-management-tools-overview/>
- ❖ ZIVE, Ruth, 2016. Why Agile Requirements Tools are the Future, According to Forrester. In: *blueprint* [online]. 2.5. [cit. 2017-05-14]. Dostupné z: <http://www.blueprintsys.com/blog/why-agile-requirements-tools-are-the-future/>