

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

User story (požadavky dle XP)

Charakteristika, jak psát dobré user stories, formát zápisu, odhadování

Jakub Resl

ROK : 2011

Obsah

1	Úvod.....	3
2	Co je extrémní programování.....	3
3	Plánování hry.....	3
4	User story	4
4.1	Shrnutí	6
4.2	Výhody a omezení.....	6
4.2.1	Výhody	6
4.2.2	Omezení.....	7
4.3	Hlavní aspekty user stories	7
4.3.1	Karta	7
4.3.2	Konverzace	7
4.3.3	Konfirmace	7
5	User story a plánování	8
6	User story během životního cyklu projektu	9
6.1	Iterace 0	9
6.2	Construction.....	9
6.3	Release.....	10
7	Formát zápisu	10
7.1	Šablony user story	10
7.2	Příklady.....	11
8	Doporučení	12
8.1	Postupy pro tvorbu user stories.....	12
9	Ukázka nástroje ExtremePlanner.....	13
10	Závěr	14
11	Zdroje.....	14

1 Úvod

Tato práce se zabývá tematikou user stories (uživatelských příběhů) v agilním přístupu vývoje softwaru a to především dle metodiky XP (extrémní programování). Jejím cílem je osvětlit problematiku user stories během životního cyklu projektu, převážně pak při jeho plánování. Dále se na následujících stránkách pokusím ukázat, v jakých formátech user stories psát a jaké zásady dodržovat při jejich tvorbě.

2 Co je extrémní programování

Extrémní programování (XP) je agilní metodologie vývoje softwaru, vytvořená skupinou lidí okolo Kenta Becka počátkem 90. let, která předepisuje určité činnosti všem účastníkům vývojového procesu. Jedná se o tradiční činnosti, které jsou však dovedeny do extrému. Díky tomu by mělo být extrémní programování schopné se lépe přizpůsobit měnícím se požadavkům zákazníků a dodávat software vyšší kvality [8]. V rámci XP projektu je třeba udržovat kvalitní komunikaci mezi všemi zainteresovanými subjekty. Pokud selže komunikace mezi programátorem a zákazníkem, dostaneme software, který zákazníkovi nemusí vyhovovat. Pokud selže komunikace mezi manažerem a programátorem, důležité informace o stavu projektu mohou manažerovi ztížit jeho rozhodování. XP přichází s postupy práce, které komunikaci podporují nebo i vyžadují. Patří mezi ně například párové programování, či plánování hry. Součástí komunikace jsou i takzvané uživatelské příběhy (dále user stories), které jsou součástí plánování hry.

3 Plánování hry

Plánování hry patří mezi jednu z 12ti základních praktik v XP. Zákazník a vývojový tým společně určují, jak dosáhnout maximálního ekonomického přínosu softwaru za co nejkratší čas. Celý proces začíná tím, že zákazníci vytvoří seznam funkčních požadavků, které od systému požadují. Každý tento požadavek se rozepíše jako tzv. uživatelský příběh, kde je v jazyku zákazníka funkce popsána. V dalším kole si tyto požadavky projde vývojový tým, a určí, jak rychle a za jakou cenu je schopný jednotlivé funkce vytvořit. Na řadě je opět zákazník a musí se rozhodnout, co je pro něj důležité a co má být implementováno jako první. Jednotlivé požadavky jsou pak seskupeny do jednotlivých iterací podle důležitosti [8].

Plánování hry se může v různých projektech odehrávat různě, ale základní pravidla jsou stále stejná. Je možno podle [1] je shrnout následovně:

- 1) Business přichází se seznamem požadovaných vlastností systému. Každá funkce je zapsána jako User story (uživatelský příběh), který dává vlastnostem jména a popisuje v hrubých obrysech, co je požadováno. User stories jsou typicky psané na kartičky.
- 2) Vývojáři odhadnou, kolik úsilí bude každá user story vyžadovat a kolik úsilí může tým produkovat v daném časovém intervalu (iteraci).
- 3) Business pak rozhodne, které příběhy realizovat a v jakém pořadí, ale i kdy a jak často vydávat produkční verze systému.

Samotné plánování lze vystihnout následujícími body:

- napsané „User stories“
- plánování vydání tvoří časový harmonogram
- časté vydávání malých změn
- měří se aktuální rychlost vývoje
- projekt je rozdělen do iterací
- každá iterace začíná plánováním
- rychlé schůze, nejlépe ve stoje
- „sprav to, když se to rozbije“

4 User story

V počítačovém programování je user story jedna nebo více vět v běžném nebo business jazyce konečného uživatele, který se snaží zachytit to, čeho chce uživatel dosáhnout. User stories se používají v agilních metodikách, mezi které samozřejmě patří i extrémní programování. Každá user story je omezená, zpravidla na velikost malého poznámkového papírku. Tím se dosáhne toho, že se v budoucnosti nebude zbytečně rozrůstat. User story by měly být psány pro zákazníky nebo přímo samotnými zákazníky, aby mohly být hlavním ovlivňovacím nástrojem vývoje softwaru. I samotní vývojáři mohou psát user story, ale jen v případech, které se přímo netýkají funkcionality, např. pokud jde o kvalitu či bezpečnost [7].

User stories jsou rychlý způsob nakládání s požadavky zákazníka bez nutnosti vytvářet formální požadované dokumenty a bez nutnosti provádět administrativní úkoly s nimi

spojené. Záměrem user story je být schopen reagovat rychleji a s menším stresem na rychle se měnící požadavky okolí.

Stejně jako tradiční projekty i projekty agilní začínají shromažďováním základních zákaznických požadavků. Znalost cílů vašich koncových uživatelů a zainteresovaných stran (stakeholders), je nutný první krok každého úspěšného projektu. Nicméně místo snahy o zachycení všech detailů předem, v agilním přístupu zjišťujeme pouze potřebné informace, abychom se mohli se zákazníkem bavit detailně později [9].

User stories slouží ke stejnému účelu jako případy užití (Use Case), ale nejsou stejné.

Používají se k vytváření časových odhadů pro release planning meeting. Jsou také používány místo velkých zadavatelských dokumentů.

Po té co jsou user story napsány, může dojít na release planning meetingu k vytvoření release plánu. Tento plán specifikuje, které user stories budou implementovány pro který systém a kdy budou příslušné releasy. Tímto dojde k vytvoření skupin user stories pro zákazníka, aby si vybral, které mají být implementovány v další iteraci. Vybrané user stories jsou následně přeloženy do jednotlivých programových úkolů (tasks), které mají být provedeny v průběhu iterace tak, aby byly user stories dokončeny. User stories jsou také překládány do akceptačních testů v průběhu iterace. Tyto testy běží během iterace a ověřují, zda jsou user story správně hotové a je možno pokračovat v práci.

Když přijde čas na implementaci user story, vývojář se sejdě se zákazníkem a získá podrobnější popis další komunikací s ním. Vývojáři odhadnou, jak dlouho se bude daná user story implementovat. Každá story dostane 1, 2 nebo 3 týdny odhadovaného ideálního vývojového času. Tato ideální doba vývoje znamená, jak dlouho bude trvat realizovat user story v případě, že nedojde k žádnému nedorozumění, nebude přidělen další úkol a vývojáři vědí přesně, co mají dělat. Doba delší než 3 týdny znamená, že bude potřeba user story rozdělit na menší části. Naopak čas kratší než jeden týden značí, že jsme na příliš detailní úrovni a je třeba některé story zkombinovat. Okolo 80 user stories plus mínus 20 je ideální číslo k vytvoření release plánu během release plánování [10]. Dalším rozdílem mezi user stories a požadavkovými dokumenty je zaměření na potřeby uživatelů. Měli bychom se snažit zabránit detailnímu specifikování použitých technologií, rozložení datové základny či algoritmů.

User story je symbolem a slibem, že se v budoucnu bude o zmíněné funkcionalitě diskutovat. Odsunutím detailní konverzace až těsně před začátek vývoje, se nám může podařit vyhnout se plýtvání, které je typické pro klasické projekty, kde jsou detailní požadavky shromážděny sice brzy, ale stávají se neplatnými ještě před zahájením samotného vývoje [9]. Navíc můžete

zjistit, že některé požadované funkce již nejsou po několika týdnech zákazníkem požadovány a dojde k úspoře nákladů na detailní analýzu.

4.1 Shrnutí

V centrální části agilních metodik, v našem případě v planning game, user stories definují, co má být vyvinuto v softwarovém projektu. Každé user story jsou zákazníkem přiřazeny priority, aby se rozlišilo, co je pro systém nejdůležitější a co má být rozděleno do jednotlivých úkolů (tasks). Následně je user story podrobena odhadu pracnosti vývojářů.

Když mají být user stories implementovány, měli by mít vývojáři možnost mluvit o nich se zákazníkem. K interpretaci krátké user story mohou být totiž potřeba další znalosti zákazníka. Může se též stát, že od doby sepsání user story dojde ke změně požadavků, proto je třeba jejich upřesnění přímo se zákazníkem.

Každá user story musí mít k sobě připojeno jeden nebo více akceptačních testů, které umožní vývojářům otestovat, kdy je user story hotová a umožní kontrolu zákazníkovi. Bez přesné formulace požadavků mohou vyvstat problémy až těsně před dodáním produktu, což způsobuje obvykle velké problémy. Jelikož se user story často upravují a používají zároveň více lidmi, je vhodné mít od každé user story více aktuálních kopií.

4.2 Výhody a omezení

Z vlastní podstaty user stories můžeme vyvodit několik výhod, na druhé straně je však třeba počítat i s určitými omezeními. Podle [7] jsou následující:

4.2.1 Výhody

- Je velmi krátká. Reprezentuje malou část obchodní hodnoty, která může být realizována v řádů dnů až týdnů.
- Umožňuje vývojáři a zákazníkovi projednávat požadavky po celou dobu životního cyklu projektu vývoje.
- Potřebují pouze velmi malou údržbu.
- Pracuje se s nimi pouze v době, kdy jsou potřeba.
- Udržují blízký kontakt se zákazníkem.
- Umožňují rozdělení projektů do menších částí (subprojektů)

- Jsou vhodné pro projekty, ve kterých se požadavky často mění nebo mohou být nejasné.
- Usnadňují odhad pracnosti vývoje.
- Vyžadují blízký kontakt se zákazníkem v průběhu celého projektu, aby se realizovaly nejcennější části vyvíjeného softwaru.

4.2.2 Omezení

- Ve velkých projektech mohou být obtížně proveditelné.
- Zákazníci mohou mít problémy s přiřazením priorit, pokud existuje hodně user stories.

4.3 Hlavní aspekty user stories

Jak již bylo řečeno, user stories jsou nejdůležitější součástí cyklu plánování. User stories mají tři kritické aspekty. Můžeme je nazvat karta, konverzace a konfirmace [5].

4.3.1 Karta

User stories jsou napsány na kartách. Kartička neobsahuje všechny informace o požadavku. Místo toho obsahuje jen tolik textu, aby bylo možné identifikovat požadavek a připomenout co je story. Karta reprezentuje požadavek a používá se při plánování. Jsou na ni napsány poznámky, které reflektují priority a náklady. Často jsou předány programátorům, když je story naplánována do vývoje, dále se předávají zpět zákazníkovi, když je story kompletní.

4.3.2 Konverzace

Požadavky sami o sobě jsou komunikovány od zákazníků k programátorům pomocí rozhovoru: výměna myšlenek, názorů a pocitů. Tyto konverzace přicházejí na řadu zvláště, když je story odhadována (obvykle během release planningu) a znovu na iteration planning meetingu, kdy je story naplánována do vývoje. Konverzace je především verbální, ale může být doplněna dokumenty. Nejlepšími doplňky jsou příklady, nejlepší příklady jsou spustitelné. Spustitelným příkladům říkáme konfirmace.

4.3.3 Konfirmace

Nezáleží na tom kolik diskuzí a kolik dokumentace se povedlo vyprodukovat, nemůžeme si nikdy být tak jistí, jak potřebujeme ohledně toho, co je třeba udělat. Proto potřebujeme konfirmaci či potvrzení. K tomu slouží již zmiňované akceptační testy.

Týmy, které nemají s těmito myšlenkami velké zkušenosti, se často snaží používat diagramy užití, tabulky, nákresy oken či vícestránkové dokumenty, které vypadají více konvenčně.

Může to být užitečné ve vzácných případech, ale ve většině projektů to ideální není.

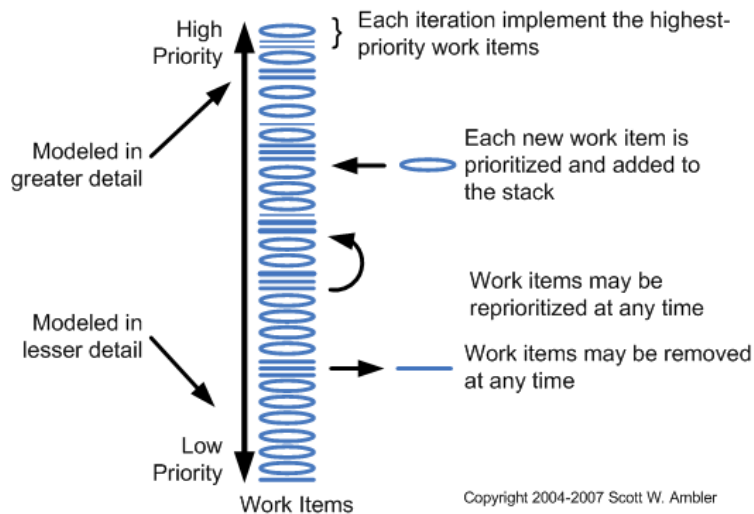
Doporučuje se začít kartami, konverzacemi a konfirmacemi a další dokumenty použít pouze pokud jsou opravdu potřeba.

Konfirmace poskytnutá akceptačními testy je to, co umožňuje jednoduché propojení karty a konverzace. Když se konverzace o kartě dostane k detailům ohledně akceptačních testů, zákazník a programátor vyřeší poslední detaily toho, co je třeba udělat. Když iterace končí a programátoři prokazují akceptační testy, zákazník zjistí, co tým udělal a co může udělat a může upřesnit, co je potřeba.

5 User story a plánování

Obrázek 1 znázorňuje agilní řízení změn, kde jsou pracovní položky včetně user stories seřazeny podle priority. Z toho vyplývá, že priorita přiřazená user story znamená, kdy budou vykonány práce k úspěšné implementaci požadavků. Jak již byl řečeno, stakeholdeři jsou zodpovědní za přidělení priorit k user story. Stakeholdeři též mají právo definovat nové požadavky, měnit názory ohledně stávajících požadavků a případně znovu prioritizovat požadavky, tak jak to zrovna uznají za vhodné. Nicméně jsou též zodpovědní za rozhodování a poskytování informací včas.

Vývojáři jsou zodpovědní za odhad úsilí, které bude třeba k implementaci všech věcí popsanych v user story. Zpočátku bývá strach, že to vývojáři nedokážou přesně odhadnout, což je často pravda, ale netrvá dlouho a lidé se naučí docela dobře odhadovat, když vědí, že se podle toho bude projekt řídit. Pokud je v týmu přijato párové programování, musí být user story proveditelná dvěma lidmi v jedné iteraci/sprintu. Proto pokud se pracuje např. v týdenních iteracích, tak každá story musí popisovat méně než týden práce. Samozřejmě pokud se párové programování nepoužívá, tak musí být user story schopen během jedné iterace implementovat jediný vývojář. Velké user stories, kterým se někdy říká eposy (epics), musí být rozděleny do menších stories, aby vyhovovali předchozím kritériím [6].



Obrázek 1 - Řízení změn

Výše byly zmíněny eposy, což jsou obrovské user story, hodící se pro věci, které budeme dělat v budoucnosti. Dále rozlišujeme témata, což jsou věci, na kterých se pracuje právě teď nebo v blízké budoucnosti a stories, které se berou na sprint [2]. Menší části musí zapadat do větších.

6 User story během životního cyklu projektu

Jak je možno vidět na obrázku 2, je zde několik zřetelných fází během životního cyklu. Jsou zde tři období, kdy se během životního cyklu pracuje s user stories [6].

6.1 Iterace 0

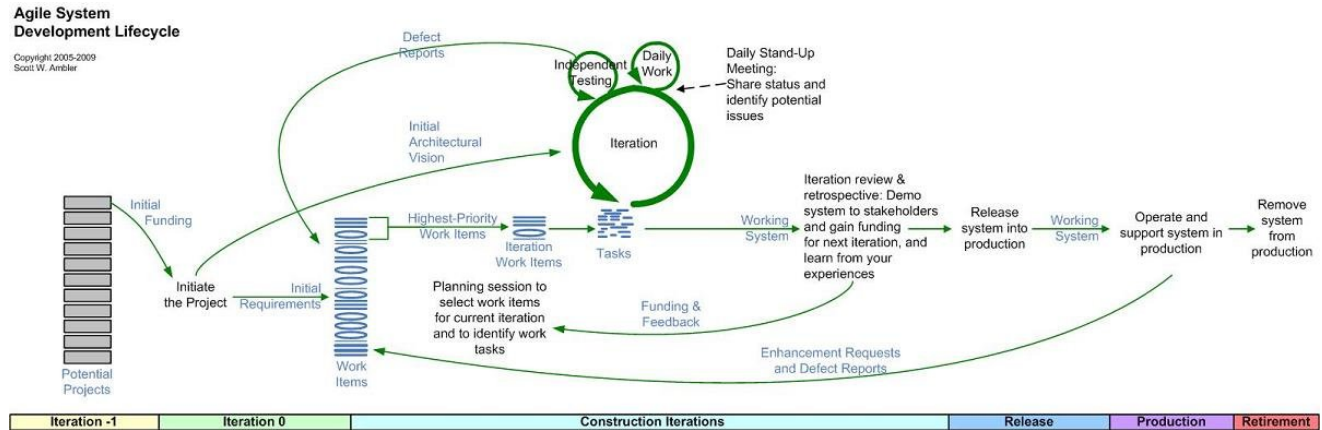
Během této fáze dochází ke shromažďování user stories jako části požadavků, které pomáhají určit rozsah systému.

6.2 Construction

Během konstrukční fáze se identifikují nové user stories nebo rozdělují existující stories, pokud je zjištěno, že jsou příliš objemné pro implementaci v jedné iteraci. Dochází též k reorganizaci priorit či k odstranění již nepotřebných user stories. Důležité je, že i user stories se během času vyvíjejí stejně jako jiné typy požadavkových modelů. Kromě toho mohou být požadavky na změnu identifikovány i v produkční fázi podpůrnými zaměstnanci a následně předány vývojářskému týmu, který pracuje na dalším releasu. Těmito požadavky dochází k tvorbě nových user stories.

6.3 Release

Někdy se stane, že nové use stories jsou identifikovány během fáze release. Ačkoli to není příliš obvyklé, jedná se především o požadavky na zpevnění systému a ne na novou funkcionalitu. Nicméně se to stát může a je třeba těmto user stories přiřadit prioritu a přidat je k ostatním jako jakékoli jiné.



Obrázek 2 - Životní cyklus agilního projektu

7 Formát zápisu

User stories, jsou zapsány na karty (např. 8x13 cm) s názvem a popisem, který zákazník formuloval. Pokud vývojář a zákazník zjistí, že user story není ideální (příliš velké, složité, nepřesné), je přepsán, dokud není uspokojivý. Nicméně v Extrémní programování (XP) user story nemusí být definitivní poté, co byly zapsány. Během vývoje může dojít ke změně požadavků a tím i k přepisu user story. Proto se na kartičkách často škrtá nebo se karty vyměňují.

7.1 Šablony user story

Pro zápis user story na kartičku, můžeme použít podle [2] následující formát:

Front of card

Story [Short Name]
As a [role]
I want [something]
So that [benefit]
Size ____
Priority ____

Back of card

Acceptance Criteria [Short Name]
Given [Context]
When [Event 1]
[Event 2] [Etc.]
Then [Outcome]
[Outcome 2] [Etc.]

Český překlad uvádím spíše pro úplnost a je přibližně následující:

Přední strana karty

Příběh [Krátký název]
 Jako [role]
 Chci [něco]
 Tak, že [prospěch]
 velikost ____
 priorita ____

Zadní strana karty

Kritéria pro přijetí [Krátký název]
 Hodnota, [kontext]
 Když [Událost 1]
 [Událost 2] [atd.]
 Pak [Výsledek]
 [Výsledek 2] [atd.]

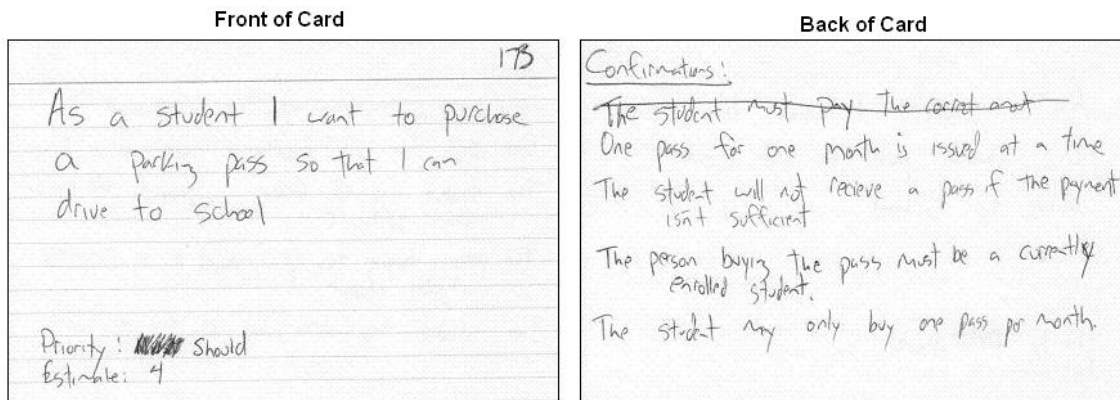
Zjednodušeně lze říci, že je nutné nadefinovat roli, cíle a přínosy. Může to vypadat následovně:

"As a <role>, I want <goal/desire> so that <benefit>"

7.2 Příklady

As a sales person, I want to add a new contact so that I can follow up later with prospects.

As a sales manager, view new contacts added by salesperson, so I can track leads [9]



Copyright 2005-2009 Scott W. Ambler



Obrázek 3 - Šablony pro user story

8 Doporučení

Na začátku projektu je důležité získat hlavní zúčastněné strany (stakeholders) k prodiskutování možných user stories na určité úrovni detailů. Zjištěním a zpracováním všech možných uživatelů systému na prvním místě zjistíme, jaké funkce budou jednotliví uživatelé potřebovat.

Není chybou, pokud některé user stories nejsou odkryty během prvního plánování. Klíčovou vlastností agilního vývoje je schopnost začlenit nové požadavky na funkcionalitu v okamžiku, kdy je objevena. To způsobuje velký tlak od zákazníků, kteří vědí, že mohou své požadavky měnit, pokud dojde například ke změně podnikatelského prostředí nebo pokud na něco důležitého minule zapomněli.

Pamatujte, že nejde o to, rozhodnout o tom, jak mohou být user stories implementovány na této úrovni, jedná se pouze o první fázi, která slouží k pochopení rozsahu celého systému. Později budou user story přiřazeny priority společně se zákazníkem a vyberou se ty, které budou součástí prvního releasu. Dále budou vybírány ještě menší podskupiny pro první iteraci (1-4 týdny) [9].

Rozhovory se zákazníky je vhodné zachovat nejen v procesu plánování iterace, ale i během samotného vývoje je důležité zůstat s nimi v kontaktu. Otázky mohou totiž zákazníka napadnout kdykoli.

Jedním z největších nepochopení user storie je jak se liší od tradičních specifikací požadavků. Největší rozdíl je v úrovni detailů. User story by měly pouze poskytovat dostatek podrobností k vytvoření přiměřeně rizikového odhadu, jak dlouho bude trvat jejich implementace.

8.1 Jak psát dobré user stories

Stakeholderi píšou user story: Důležité je pojetí, že user stories píšou stakeholderi, nikoli vývojáři. User stories jsou jednoduché, takže se je lidé naučí psát během několika málo minut, což právě dává možnost, aby je mohli psát sami zákazníci.

Použijte nejjednodušší nástroj: User story jsou často psány na kartičky, se kterými se velmi jednoduše pracuje a modeluje.

Pamatujte si nefunkční požadavky: Pomocí user stories je možné popsat širokou škálu požadavků. Například user story „The Students can purchase parking passes online“ je podobná use case zápisu, který by mohl být následující: „The Passes will be available online via a standard browser“, kde je již zahrnut i technický požadavek.

Uved'te odhad pracnosti: Jak můžete vidět na obrázku 3, je možné si k user story přidat odhad, jež vyjadřuje, jak dlouho bude trvat implementace. Jednou z možností je přiřadit k user story body, které indikují, jak dlouho bude páru programátorů trvat implementace. Důležité je určit si poměr mezi body a hodinami. Např. některý tým může mít poměr 2,5 hodiny na jeden bod a jiný třeba 3. Pokud bude user story na obrázku 3 dělat tým s poměrem 2,5, tak implementace zabere celkem 10 hodin.

Uved'te priority: User story by měla obsahovat prioritu. Pokud je u každé user story priorita, je možnost je podle toho řadit a udržovat aktivní ty user story, které mají nejvyšší prioritu. K vyjádření priority je možno použít například škálu od 1 do 10, kde 1 vyjadřuje nejvyšší prioritu. Dalším způsobem je vzít všechny user stories a každé přiřadit vlastní prioritu. Pokud budete mít 80 user stories, znamená to 80 úrovní priorit. Priority je samozřejmě možné v průběhu času měnit. Důležité je zvolit strategii která bude fungovat ve vašem prostředí a té se držet. Musí být dostatečně jednoduchá a účinná.

Volitelně přiřad'te jedinečný identifikátor: Jediným rozumným důvodem pro přiřazení identifikátoru je, pokud existuje návaznost user story na další artefakty, například akceptační testy [6].

9 Ukázka nástroje ExtremePlanner

V dnešní době lze user story samozřejmě spravovat i jinak než v podobě papírových kartiček. Pro ilustraci uvádím nástroj ExtremePlanner, který umožňuje zadávat user stories pouze pomocí jména (žádné další informace nejsou pro zadání nutné). Během času lze doplnit popis. Při použití výchozího nastavení se user stories promítnou do project backlogu. Bohužel se mi nepodařilo stáhnout žádnou zkušební verzi pro detailní vyzkoušení, proto je přiložen pouze obrázek 4 ilustrující administraci user stories pomocí tohoto nástroje [9].

The image shows two parts of the ExtremePlanner interface. On the left is the 'Add Story' form, and on the right is the 'Stories' list.

Add Story Form:

- Story** (tab)
- Name:** As a sales person I want to add new contacts
- Description:** (empty text area)
- Topic:** (empty text field)
- Type:** New Feature (dropdown menu)

Stories List:

Iteration: (any) Release: (any)

Estimated Effort: 36.0 points / Completed: 4.0

Buttons: Add Story, Delete, Schedule...

ID	Story Name	Topic	Typ
<input type="checkbox"/> S3	Add a product to my shopping cart	Shopping Cart	New Featu
<input type="checkbox"/> S1	View a list of products for sale	Catalog	New Featu
<input type="checkbox"/> S2	View details of a single product	Catalog	New Featu
<input type="checkbox"/> S7	Allow user to add a product review	Reviews	New Featu

Obrázek 4 - Správa user stories pomocí nástroje ExtremePlanner

10 Závěr

Na předchozích stranách bylo nastíněno několik doporučení, jak správně používat user stories, čímž byl splněn cíl této práce. Za nejdůležitější myšlenku, kterou je třeba mít na paměti, můžeme považovat držení se strategie ohledně user stories, která bude v daném prostředí fungovat nejlépe a bude dostatečně jednoduchá.

11 Zdroje

1. Brewer, J.: Extreme Programming FAQ, dostupný z <http://www.jera.com/techinfo/xpfaq.html>
2. Brown, C.: User Story Template, dostupný z <http://www.betterprojects.net/2011/03/user-story-template.html>
3. BUCHALCEVOVÁ, A.: Agilní metodiky a správa požadavků. Ostrava 04.06.2007 – 06.06.2007. In: Tvorba softwaru 2007. Ostrava : Ekonomická fakulta VŠB TU, 2007, s. 16–23. ISBN 978-80-248-1427-8
4. Buchalcevoová, A. Metodiky budování informačních systémů, Praha: Oeconomica 2009, 206 stran, ISBN 978-80-245-1540-3
5. Jeffries, R.: Essential XP: Card, Conversation, Confirmation, dostupný z <http://www.xprogramming.com/xpmag/expCardConversationConfirmation.htm>
6. Scott W. Ambler: Introduction to User Stories, dostupný z <http://www.agilemodeling.com/artifacts/userStory.htm>
7. Wikipedia : User Story, dostupný z http://en.wikipedia.org/wiki/User_story#User_stories_and_use_cases
8. Wikipedia : Extrémní programování, dostupný z http://cs.wikipedia.org/wiki/Extr%C3%A9mn%C3%AD_programov%C3%A1n%C3%AD
9. <http://www.extremepanner.com/resources/Agile-Requirements.html>
10. <http://www.extremeprogramming.org/rules/userstories.html>
11. <http://blog.mountangoatsoftware.com/advantages-of-the-as-a-user-i-want-user-story-template>
12. <http://www.allaboutagile.com/example-of-a-user-story/>