

User story požadavky dle XP

Bc. Martin Ješuta

3.12.2011

Obsah

1	Úvod	2
2	Extrémní programování	3
2.1	Principy	3
2.2	Popis procesu	4
3	User Stories	8
3.1	Jejich místo v XP	8
3.2	Příklady, rady a podněty z praxe	9
3.2.1	United States Navy's Sailor Assignment Problem	9
3.2.2	Charakter zákazníka	9
3.2.3	Volba User Stories	10
3.2.4	Zkušenosti z výuky Extrémního programování v laboratorních podmínkách	10
3.2.5	Šablona „AS A USER, I WANT“	10
3.2.6	Vlastnosti Story	11
3.2.7	Struktura story	13
3.3	Shrnutí a doporučení – Jak psát User Stories	14
4	Závěr	16
5	Seznam zdrojů	17

1 Úvod

Extrémní programování je agilní metodika pro vývoj software. Její nedílnou součástí jsou takzvané User Stories, které budou předmětem této seminární práce. Nejprve ve stručnosti popíši celou metodiku jakožto východisko celé práce, následně se pak budu věnovat přímo User stories. Pomocí řešerše dostupných zdrojů od teoretických stanovisek až po praktické zkušenosti z projektů, vytvořím závěry, které odpoví na to jak psát a strukturovat dobré User Stories. Tyto závěry jsou rovněž i cílem této práce.

2 Extrémní programování

Cílem kapitoly je popsat filosofii Extrémního programování a vytvořit tak východiska pro popis a analýzu User Stories. Popíši nejprve na jakých principech je Extrémní programování postaveno a následně projdu celý standardní proces s jeho vlastnostmi a pravidly.

2.1 Principy

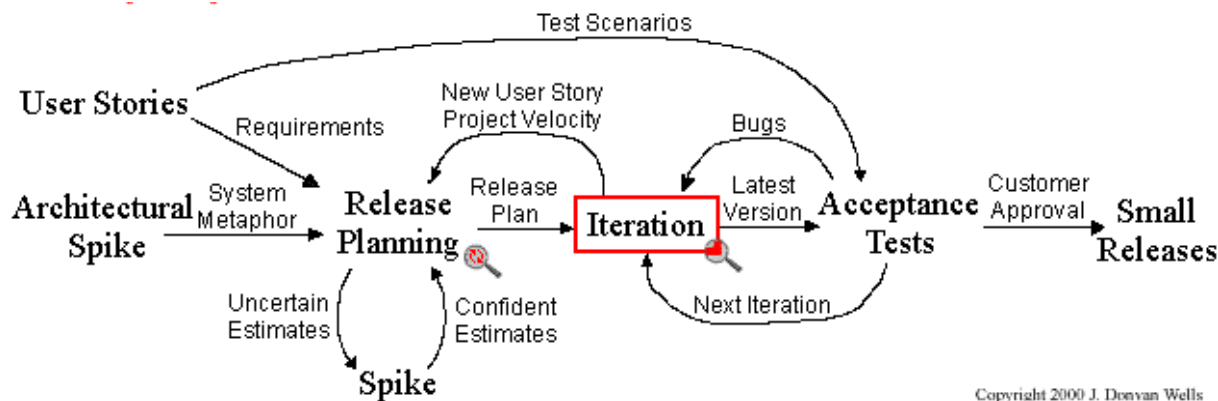
Extrémní programování se řadí mezi agilní metodiky. Poprvé bylo použito v roce 1996 a jeho duchovním otcem je Kent Beck. Staví na několika principech, které jsou stěžejní, a při studiu toho tématu jsem na ně narážel na více místech:

- základními hodnotami extrémního programování jsou jednoduchost, komunikace, zpětná vazba, odvaha a respekt (Kosa 2011)
- „*Zpochybnění uživatelských požadavků (stresses customer satisfaction)*“ (Wells 2009)
- „*Vyzívá vývojáře zodpovědně reagovat na změny uživatelských požadavků a to i v pozdních fázích životního cyklu (empowers your developers to confidently respond to changing customer requirements, even late in the life cycle.)*“ (Wells 2009)
- „*Manažeři, zákazníci a vývojáři jsou rovnocenní partneři v spolupracujícím týmu (Managers, customers, and developers are all equal partners in a collaborative team.)*“ (Wells 2009)
- „*...je o odpovědnosti týmu za celý kód, který je tvořen dle konzistentního vzorce tak, aby každý mohlo číst kód každého. Je o udržení integrovaného systému v běhu po celou dobu. (...is about team responsibility for all code, for coding in a consistent pattern so that everyone can read everyone's code, about keeping the system running and integrated all the time)*“ (Jeffries 2011)
- „*Extreme Programmers pracují v párech jako skupina, s čistým designem a důsledně testovaným kódem, kontinuálně zlepšují design, aby byl vždy v souladu s aktuálními požadavky. (Extreme Programmers work together in pairs and as a group, with simple design and obsessively tested code, improving the design continually to keep it always just right for the current needs.)*“ (Jeffries 2011)

Jistě by mohl někdo uvést i další principy a parametry. Tyto se jeví mně jako dobrý výchozí bod pro seznámení s extrémním programováním. Podrobnosti a konkrétnější pohled popíši v další kapitole, která již konkrétně prochází proces extrémního programování.

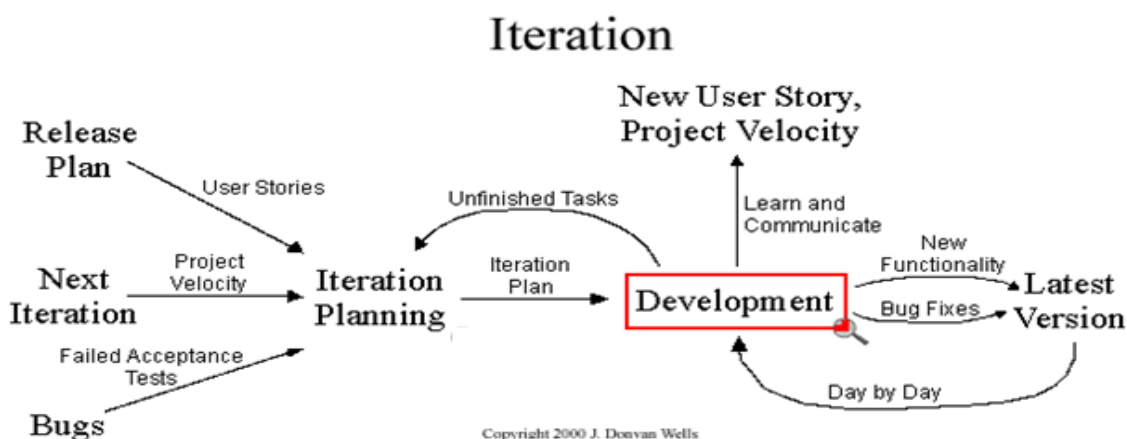
2.2 Popis procesu

Každý vyvíjený projekt Extrémního programování lze rozdělit na 5 oblastí, které mají svá neměnná pravidla. Jsou to Plánování, Řízení, Design, Programování a Testování. Tato pravidla se pak promítají do průběhu celého procesu, který ilustruje obrázek 1.



Obrázek 1

Vše začíná plánováním, zde je třeba vytvořit na základě User Stories Release plan. V něm strana vývoje určí, jak dlouho jednotlivé Stories budou v ideálním případě¹ trvat. Na základě toho zákazník vybírá, které Stories budou implementovány v další iteraci. Ty jsou následně podrobněji rozebrány v Iteračním plánu po technické stránce do jednotlivých programových úkolů a opět na základě Stories jsou navrženy akceptační testy. Vyskytne-li se nějaký náročný technický problém, využívá se takzvaných Spike Solutions². Jedná se o krátké experimentální kódy, které stojí mimo produkční kód a dávají rychlou odpověď na daný problém. Výsledkem Release planningu je definování projektu pomocí čtyř proměnných: Velikost, Čas, Zdroje a Kvalita. Tyto proměnné nemůže nikdo určit nezávisle, protože každý stakeholder má povědomí o parametrech jiné proměnné, ale zároveň jedna proměnná ovlivňuje druhou.



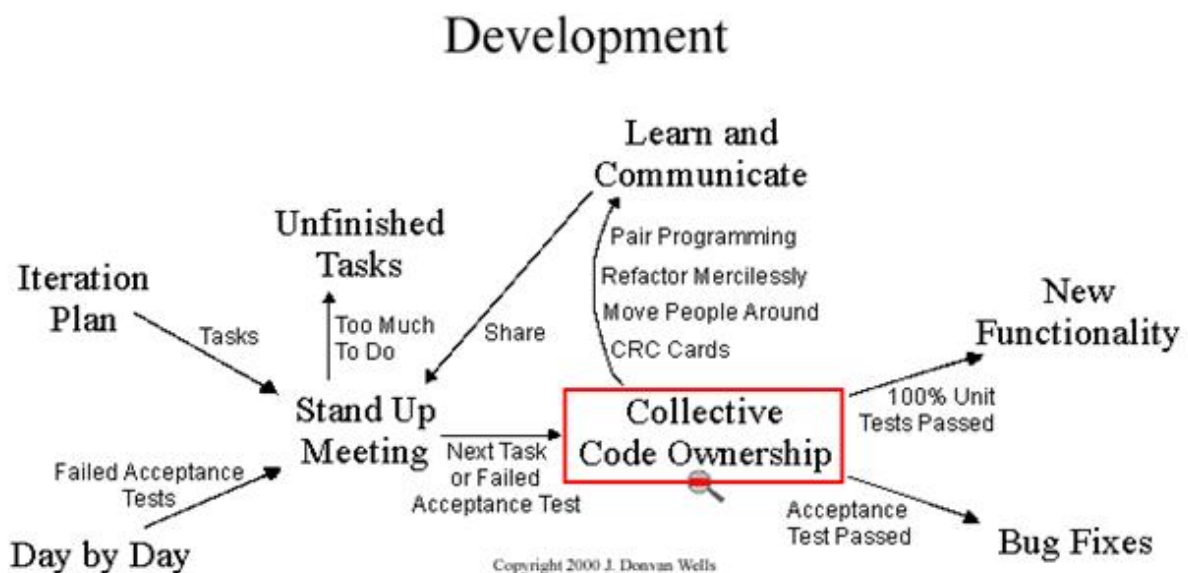
Obrázek 2

¹ Ideální případ je reálným odhadem práce, pokud nebude mít vývoj na starosti nic jiného.

² http://jamesshore.com/Agile-Book/spike_solutions.html

Řízení staví na základě šesti základních pravidel:

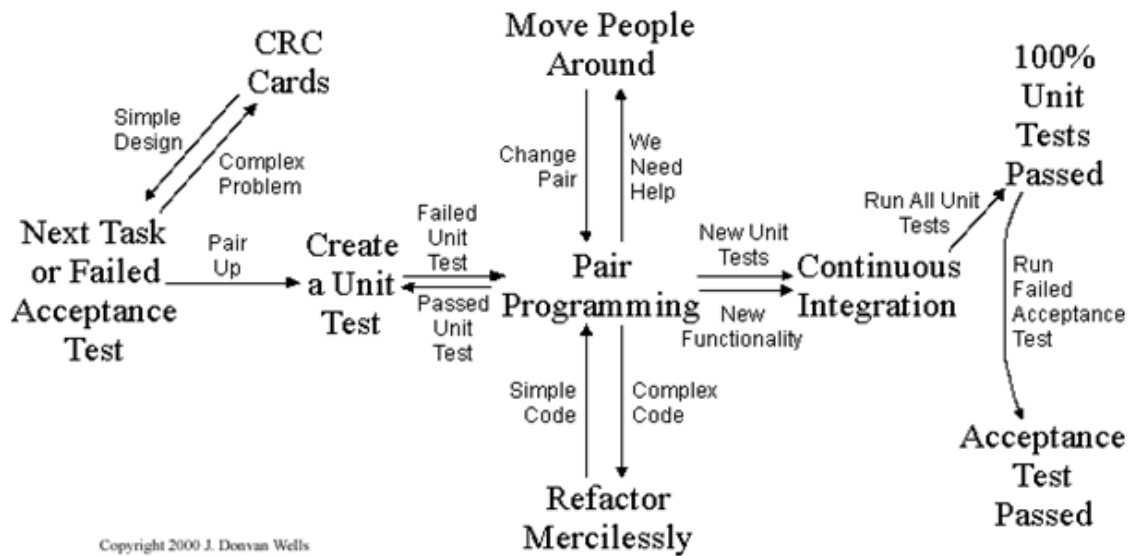
1. Dejte týmu vyhrazené otevřené pracoviště podporující snadnou komunikaci
2. Nastavte udržitelné tempo prací. Přesčasy znamenají zpravidla menší kvalitu. Přidávání lidí do projektu, který se zpožďuje, přinese spíše negativa než positiva.
3. Stand-Up Meeting každý den
4. Měřte rychlost projektu. Je dobré v každém okamžiku projektu a podle toho mít možnost jej revidovat a měnit plán dalších iterací
5. Nesoustředit znalosti na jedince, ale křížově trénovat celý tým
6. Opravit principy XP tam kde selhávají.



Obrázek 3

Design se řídí především jednoduchostí. Vychází ze Systému metafor, které tvoří architekturu budoucího systému tak, aby byla snadno pochopitelná a vysvětlitelná pro nové lidi. Jsou použity Class, Responsibility and Collaboration Cards (CRC cards), které representují objekty daného řešení. Tým nad nimi diskutuje a vytváří tak objektový model systému. Opět platí možnost využít Spike solutions, když se narazí na nějaký technický oříšek. Jednoduchost řešení udržuje i pravidlo žádné extra funkcionality, kód má být čistý a připravený na nečekané změny. Posledním pravidlem se plynule dostáváme do oblasti programování, protože jím je refactoring.

Collective Code Ownership



Obrázek 4

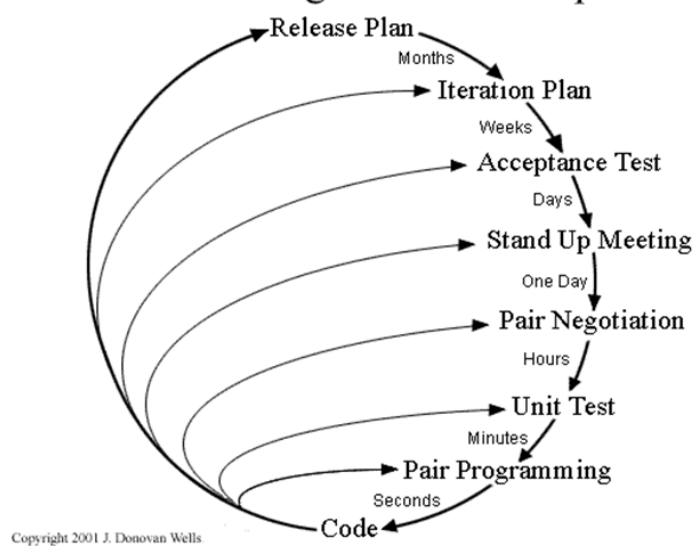
Programování vychází z designu, kde na základě modelů a User Stories se tvoří jednotkové testy. Následuje párové programování, které souvisí s již zmíněným refactoringem a s 5. pravidlem z řízení, tedy nesoustředit znalosti u jedince. Fragmenty kódu jsou neustále integrovány i po několika hodinách, tak aby každý mohl pracovat s co nejaktuálnější verzí. Práce končí ověřením jednotkovými a následně i akceptačními testy. Během celého procesu je k dispozici zástupce zákazníka, aby mohl v každém okamžiku upřesnit požadavky a případně rozhodnout priority dalšího postupu s ohledem na jeho byznys cíle. Důležitým rysem extrémního programování je kolektivní vlastnictví kódu. Každý vývojář zasahuje do jakékoli části kódu, opravuje chyby a přináší nové myšlenky.

Poslední oblastí je testování. Obsahuje čtyři pravidla:

1. Každý kód musí mít jednotkový test
2. Každý kód musí projít všemi testy před tím, než je puštěn dál
3. Při nalezení chyb je vytvořen test, aby se nemohla opakovat.
4. Akceptační testy, které jsou vytvořeny z User Stories během Iteration planning meetingu, se spouštějí tak často jak to jen jde.

Celý proces pak běží v jakési časové smyčce, kterou ilustruje obrázek 5.

Planning/Feedback Loops



Obrázek 5

Extrémní programování se z mého pohledu skládá z tří prvků. Prvním je iterativní proces tvorby software, druhým jsou pravidla, která se vztahují k jednotlivým oblastem vývoje a nakonec jsou to principy uvedené v úvodu této kapitoly.

3 User Stories

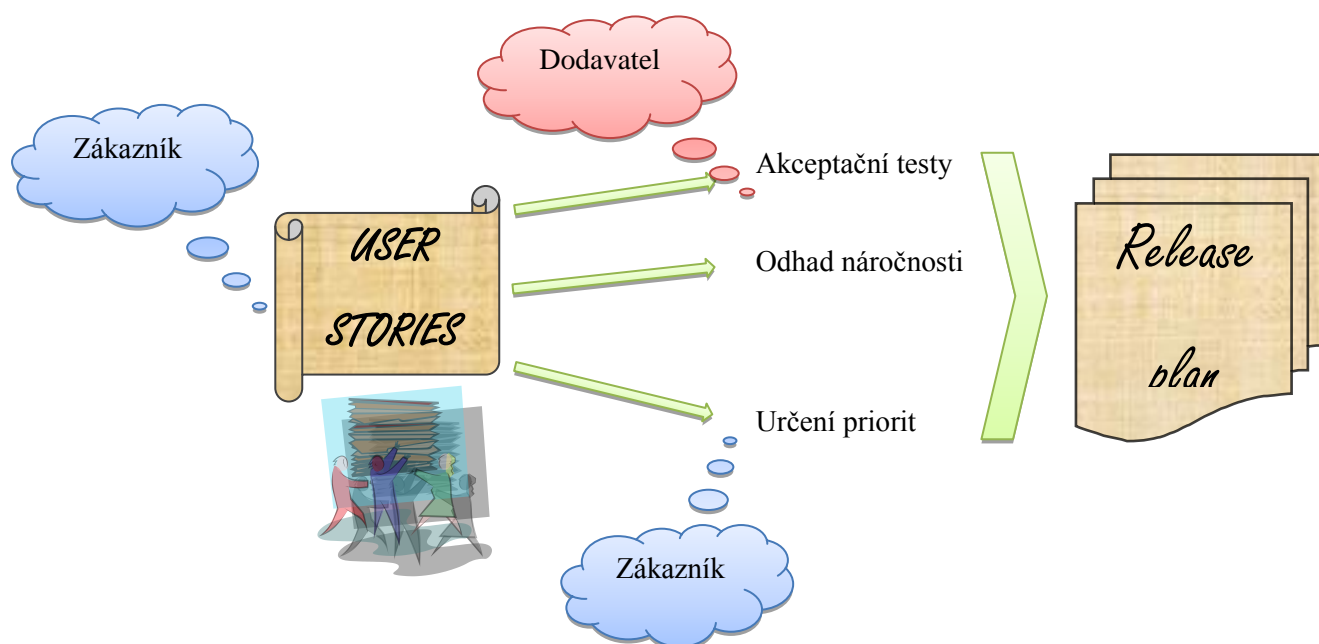
User Stories jsou účinným nástrojem a jedním z nejviditelnějších prvků extrémního programování. V následujících podkapitolách, v souladu s tématem práce, rozeberu nejen jejich pozici v extrémním programování, ale zejména jaké jsou na ně požadavky, jak se mají psát, jaké formy jsou v praxi nejčastější a nakonec vše shrnu do jakých si doporučení a případných návrhů šablon.

3.1 Jejich místo v XP

User Stories jsou sepsané jednotlivé požadavky zákazníka na navrhovaný systém. Od klasické dokumentace zpracující požadavky se Stories liší v několika zásadních věcech:

- Každá Story by měla být maximálně 3 věty dlouhá.
- Jsou oproštěny od zbytečných technologických detailů.
- Úroveň obecnosti je taková aby se dala odhadnout náročnost implementace.
- Jsou psány jazykem zákazníka.
- Soustředí se na definování potřeb zákazníka
- „*Story není dokončena, pokud neprojde všemi akceptačními testy.*“ (Kosa 2011)
- Náročnost jedné Story by neměla přesáhnout 3 týdny ideálního vývojového času³.

Obrázek 6 (Zdroj: Autor)



User Stories se následně použijí, jak bylo popsáno v kapitole 2.2 a jak ilustruje i obrázek 6. Tedy za spolupráce všech stakeholderů v takzvané Plannig game vzniká Release plan.

³ Ideální vývojový čas – „*ideal development time is how long it would take to implement the story in code if there were no distractions, no other assignments, and you knew exactly what to do*“ (Wells 2009)

3.2 Příklady, rady a podněty z praxe

3.2.1 United States Navy's Sailor Assignment Problem

V US NAVY mají námořníci právo každé dva roky žádat změnu jejich práce a to v konkrétní lokaci. Kvalifikace každého z nich však vyhovuje jen specifickým oblastem. US NAVY stálo před problémem vytvořit personální systém, který umožní efektivní přidělování pozic na základě určitých preferencí. Nejprve byla část projektu vytvořena tradičními technikami, po neúspěchu byla zvolena cesta extrémního programování s Vývojem řízeným testy.

Zástupce zákazníka sepsal několik User Stories, zde jsou uvedené ukázky:

1. *„Read Sailor Data: Be able to read the problem data file and construct any internal helper objects required. The problem data should be able to be extracted from the objects at any point of time.*
2. *Generate initial solutions: Be able to randomly or heuristically generate initial solutions.*
3. *Produce Crossover: Be able to produce a single point crossover of two randomly picked solutions, which should produce two different child solutions that are different from the parent, while maintaining the constraints.“* (Vejandla, Sherrell 2009)

V souvislosti s projektem je zmíněn pouze jediný problém a to, že zástupce zákazníka neupřesnil včas kritéria akceptačních testů. Doporučením v tomto ohledu tedy může být udržení těsné spolupráce celého týmu.

3.2.2 Charakter zákazníka

Extrémní programování jak ukazuje jeho teoretický model, tak i předchozí příklad, je silně závislé na kooperaci celého týmu. V souvislosti s tím (Grisham, Perry 2005) uvádí následující problémy s charakterem osoby zastupující zákazníka:

- Předpokládá se znalost byznys požadavků
- Předpokládá se schopnost efektivně vyjádřit své požadavky.
- Předpokládá se, že osobnost zástupce zákazníka bude sociálně adaptabilní vzhledem k týmu. Zejména ve smyslu, že bude schopen vyjádřit okamžitě svoji nespokojenost a požadavky týmu.

Častým problémem je typ zákazníka, který zpočátku vypadá spokojen a ve finální fázi projektu si začne stěžovat na vše. Řešením je: *„soustředit se na efektivní získávání požadavků a monitoring spokojenosti vývojářů s tímto získáváním. Zároveň je třeba adekvátně vést zákazníka k tomu jak správně komunikovat potřeby byznysu“* (Grisham, Perry 2005)

3.2.3 Volba User Stories

Zákazník během Planning game určuje, s ohledem na nejvyšší přidanou hodnotu byznysu, které Stories se budou vyvíjet v další iteraci. V této situaci, jak uvádí (Grisham, Perry 2005), může dojít ke konfrontaci s požadavkem, který je ze své podstaty nestabilní. V tu chvíli existují dvě cesty. První je odložení požadavku s tím, že v čase by mohl být požadavek více stabilní a tím i lépe pochopen a uchopen. Druhou cestou je práce na tomto požadavku s cílem mu co nejdříve co nejlépe porozumět. Správná volba je podmíněna co nejlepší informovaností zákazníka o případných nákladech a dopadech. (Grisham, Perry 2005) doporučují pro takovéto situace vytvářet risk analýzu, která by umožnila informované rozhodnutí zákazníkovi.

3.2.4 Zkušenosti z výuky Extrémního programování v laboratorních podmínkách

V Hannoverské universitě se rozhodli zprostředkovat svým studentům autentický zážitek z extrémního programování, pomocí kurzu, který věrně simuluje celý proces vývoje (Stapel, Lübke 2008).

Zkušenosti z tohoto kurzu přinesly podněty použitelné pro praxi a tak ty které se týkají User Stories uvádím zde.

Zkušenost ukázala, že pokud User Story píše pouze zákazník a nemá přitom potřebný technický background, tak je následná Planning game velice náročná, protože zákazník své požadavky popíše příliš obecně a většina se jich pak musí rozdělovat. Organizátoři kurzu po dvou letech zapojili do psaní User Stories i stranu vývojáře a ukázalo se to jako správný krok. Stories se nemusely tolik rozdělovat.

3.2.5 Šablona „AS A USER, I WANT“

Mike Cohn uvádí na svém blogu (Cohn 2011) jím preferovaný styl psaní User Stories, který pojmenoval AS A USER, I WANT. Jméno vychází ze samotného vzoru šablony, podle které se každá Story zapisuje takto:

AS A USER – Z jaké pozice, role, potřebuje a popisuje danou Story.

I WANT – Co chce, samotná Story.

SO THAT – Z jakého důvodu to požaduje.

Výhod této struktury je několik. Prvním důvodem je, že pokud je jasně dáno z jaké pozice je vznášen nějaký požadavek, tak jeho tvůrce se s ním dokáže lépe sžít. Druhý důvod souvisí s tím, že pokud bude backlog s User Stories jasně strukturován, tak je pro zákazníka snazší prioritizovat. Na první pohled vidí jaká role, co potřebuje a jaký z toho má pak prospěch. Příklad ilustruje obrázek 7.

As a/an	I want to...	so that...
moderator	create a new game by entering a name and an optional description	I can start inviting estimators
moderator	invite estimators by giving them a url where they can access the game	we can start the game
estimator	join a game by entering my name on the page I received the url for	I can participate
moderator	start a round by entering an item in a single multi-line text field	we can estimate it
estimator	see the item we're estimating	I know what I'm giving an estimate for
estimator	see all items we will try to estimate this session	I have a feel for the sizes of the various items
moderator	see all items we try to estimate this session	I can answer questions about the current story such as "does this include ..."
moderator	select an item to be estimated or re-estimated	the team sees that item and can estimate it

Obrázek 7 Ukázka User stories (Zdroj: Cohn 2011)

3.2.6 Vlastnosti Story⁴

Mike Cohn, zakládající člen Agile Alliance a Scrum Alliance, uvádí na svém blogu (Cohn 2004) několik prezentací, které popisují, jak dle něj mají User Stories vypadat. Vzhledem k jeho zkušenostem v této oblasti jsem si dovolil tyto poznatky převzít do této kapitoly a některé jeho názory přeložit a citovat:

Uživatel není nikdy jeden. Uživatelé nemají stejné cíle, proto je dobré používat uživatelské role.

Uživatelská role je: „...kolekce definujících atributů, která charakterizuje populaci uživatelů a jejich předpokládané interakce se systémem“

1. Pro lepší prezentaci uživatelských rolí je dobré používat persony. Jsou to imaginární představitelé dané populace. Mají jméno, oblíbené a neoblíbené věci/zvyky, čas kdy pracují, místo kde pracují a důvody proč dělají to, co dělají. Persony umožní lepší přiblížení vývojového týmu k uživateli: „*Přemýšlejme o softwaru jako nástroji pro řešení potřeb reálných lidí*“
2. Interview je nezbytný krok k tomu aby se ujasnili skutečné potřeby uživatelů: „*Většina uživatelů plně nerozpoznává své skutečné potřeby. Mít problém nikoho nepředurčuje k tomu, že ví jak jej řešit.*“ Interview je třeba vést pomocí neuzavřených otázek a v širším kontextu. Otázka nesmí směřovat k jediné možné odpovědi, ale dát prostor pro informovanou volbu.
 - Špatná otázka: “*Would you like it in a browser?*”

⁴ Kapitola čerpá a cituje z (Cohn 2004), zároveň jsou v ní promítnuty i další poznatky z prezentací uveřejněných na (Cohn 2011b)

- Lepší formulace: *“Would you like it in a browser rather than as a native Windows application even if it means reduced performance, a poorer overall user experience, and less interactivity?”*
- Ideální formulace: *“What would you be willing to give up in order to have it in a browser?”*

INVESTujme do User Stories, je sada šesti vlastností, které se dají vždy zlepšovat:

Independent – Nezávislost. Stories jsou vzájemně nezávislé. Pokud je závislost silná můžeme uvažovat o sloučení a naopak u slabé závislosti rozdělíme na základě jiného pohledu.

Negotiable – Právně nezávazné. Story není smlouva a nepředstavuje tak právní závazek. Z toho plyne i to, že Stories neobsahují detaily, bez kterých se vymezení smlouvy neobejdou.

Valuable – Přínosné. Stories vždy popisují, jaký přínos přinesou uživateli. Pokud je story psaná dle šablony „AS A USER, I WANT“ vždy ponese v závěrečné části hodnotu.

Estimatable – Odhadnutelnost. Stories jsou používány k plánování, musí tedy být odhadnutelná pracnost i časová náročnost.

Small – Malá. Tato vlastnost jde ruku v ruce s druhou vlastností Negotiable. Není žádoucí, aby stories byly delší než 3 věty, pak je namístě je rozdělovat.

Testable – Testovatelná. Testy demonstrují, že Story naplňuje zákazníkovi požadavky, proto by mělo být možné napsat na každou story jednoznačné akceptační testy.

Rady pro psaní Stories:

1. **Začněte s cíli.** Pro každou roli se ptejte: *„Jaké cíle uživatelé v systému mají?“*
2. **„Slice the cake“** Každý problém je třeba brát v kontextu všech úrovní pojmout jak UI, tak aplikaci, tak databázi – „skrz všechny vrstvy dortu“.
3. **Pište uzavřené story.** Story která dosáhne nějakého cíle dá uživateli pocit, že je něco hotové. Už se k tomu nemusí vracet: *„Uživatel cítí, že něco dokončil.“*
4. **Pište omezení na karty.** Stejně jako Stories, tak i omezení (Constraints) je dobré psát na karty. Omezení představují určité věcné limity dané story nebo situace. Měli by být časově zařazeny do, pokud možno, co nejbližší iterace. Samozřejmostí jsou testy, které ověří, zda je omezení ošetřeno.
5. **V nových projektech držte problematiku UI mimo co nejdéle.** V době psaní Stories není UI až tak podstatná a může jen odvést pozornost od závažnějších problémů. Případně se dá UI zahrnout formou omezení (Constraints) nebo jako vydělené Stories čistě na UI.

6. **Ne vše je story.** Existují tak specifické a rozsáhlé problémy že použití User Stories je kontraproduktivní. V těchto případech je vhodné využít jiná řešení: Use-case, User interface guidelines, List of business rules, Interface with another systém. Vždy však platí: „*Cokoli napíšete ať je stručné (lightweight)*“
7. **Zahrňte Uživatelské role do Stories.** Jak již jsem zmínil výše, je lepší vztahovat story k určité uživatelské roli s cílem: „*Pomozte každému, aby se dostal do kůže daného uživatele, když čte jeho story*“
8. **Pište pro jednoho uživatele.** Zde to autor myslí tak, že story by měla být psána pro jednoho uživatele, tedy v jednotném čísle. Příklad:
 - Správně: Uživatel vkládá dokumenty
 - Špatně: Uživatelé vkládají dokumenty. Není jisté, jestli si autor story uvědomil rozdíl mezi vztahy 1:N, M:1 a M:N.

Závěrem ze studia poznatků pana Cohna uvedu myšlenku, která prolíná veškerou jeho práci týkající se User Stories:

„*The story text we write on cards is less important than the conversations we have*“

„*Text napsaný na kartičky je méně důležitý než konverzace, kterou máme*“

3.2.7 Struktura story

Poslední tématem, které se zabývá teoretickým zázemím User Stories, je struktura Stories. Jak již bylo naznačeno výše, tak story by měla být papírová kartička, která nese několik informací. Tyto informace je dle (North 2011) ideální strukturovat na titulek, příběh, titulek scénáře a scénář.

Titulek by měl jednoznačně identifikovat story. Jeho forma by měla být taková, aby představovala činnost. Například: Recepční se přihlašuje do systému. Administrátor resetuje heslo. apod.

Příběh musí obsahovat roli, funkcionalitu a byznys přínosy. Jak bylo popsáno výše ideální je šablona: AS A USER, I WANT.

Titulek scénáře má především třídící vlastnost. Název titulku jednoznačně říká, v čem je tento scénář jiný než scénáře ostatní.

Scénář je formou zápisu akceptačních kritérií. Na základě šablony GIVEN/WHEN/THEN jsou postupně zapsány podmínky a kontext testu (given), akce uživatele (when do something) a reakce systému (then systém do). Počet scénářů by neměl přesáhnout více než 6, pak již by se mělo uvažovat o rozdělení story, pro její komplexnost.

Příklady této struktury ilustruje obrázek 8.

```
Title (one line describing the story)

Narrative:
As a [role]
I want [feature]
So that [benefit]

Acceptance Criteria: (presented as Scenarios)

Scenario 1: Title
Given [context]
  And [some more context]...
When [event]
Then [outcome]
  And [another outcome]...

Scenario 2: ...
```

Obrázek 8 Struktura User Stories

3.3 Shrnutí a doporučení – Jak psát User Stories

Na základě podnětů z předešlých podkapitol a mého úsudku jsem na schématu 1 nastínil doporučenou šablonu a na schématu 2 doporučený postup pro psaní User Stories. Postup začíná příchodem nového požadavku. Aby jej tým mohl zpracovat, musí být sepsána User story. Story se sepíše dle šablony uvedené v kapitole 3.2.5.. Následně story dostane strana dodavatele a píše scénáře dle šablony uvedené v kapitole 3.2.7 , které budou představovat scénáře akceptačních testů. Po celou dobu tvorby jsou dodržena pravidla INVEST a obecné dobré rady viz kapitola 3.2.6. Mezi dobré rady jsem zařadil i podněty z kapitol 3.2.2, 3.2.3 a 3.2.4 .

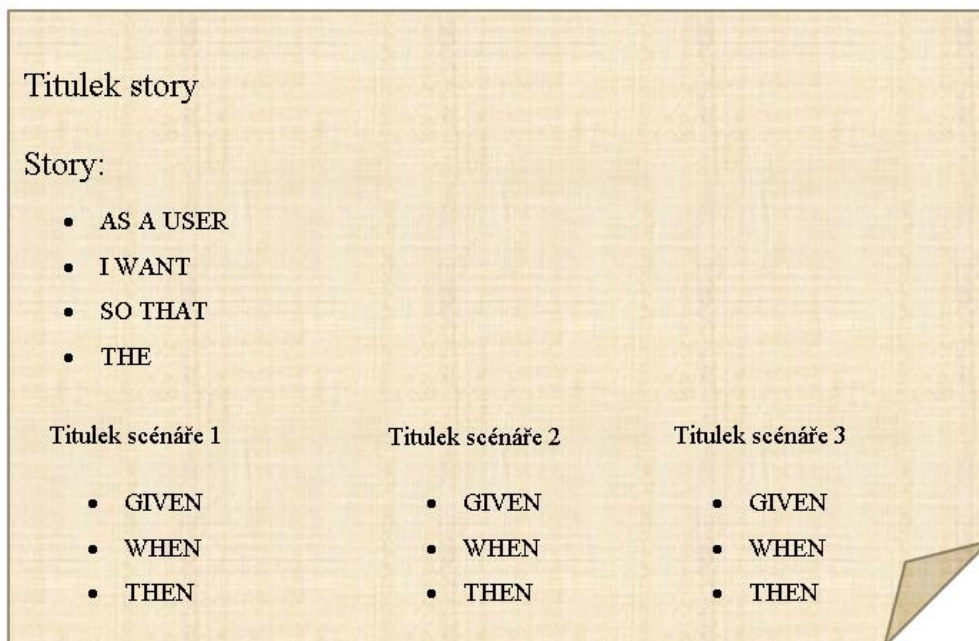


Schéma 1 (Zdroj: Autor)

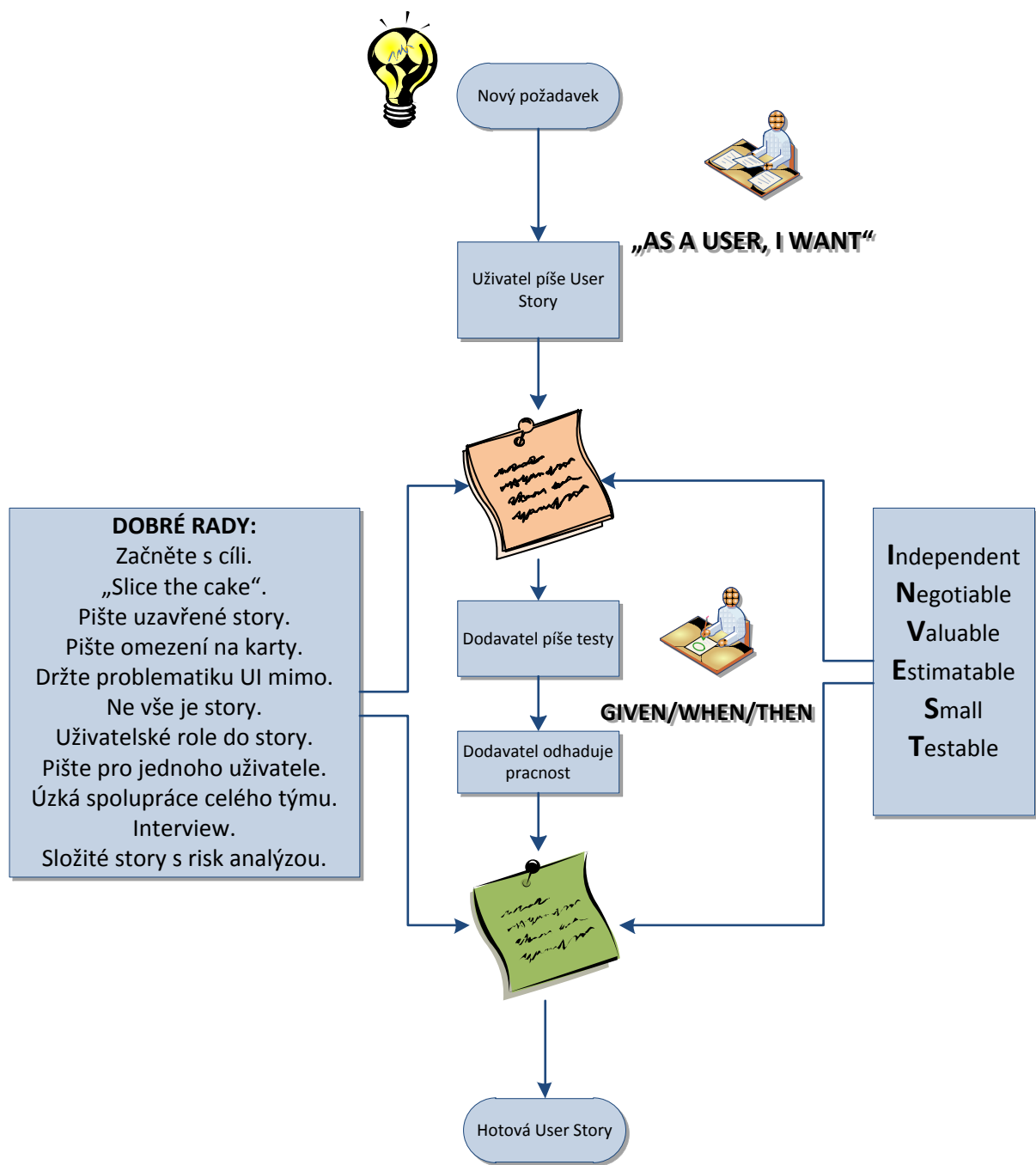


Schéma 2 (Zdroj: Auotr)

4 Závěr

V souladu s položenými cíli v úvodu práce jsem pomocí rešerše zejména internetových zdrojů popsal jak teoretická východiska, tak praktické zkušenosti s User Stories. Zde mi byly zejména významnou pomocí elektronické zdroje na VŠE, díky kterým jsem mohl popsat zkušenosti z praxe. Zároveň i stránky Mike Cohna, který se problematice agilního vývoje a User Stories věnuje dlouhodobě.

Výsledkem mého studia je tedy kompilace poznatků z výše uvedených zdrojů, které jsem následně shrnul do dvou schémat v kapitole 3.3. Schéma 1 ilustruje ideální formu šablony pro psaní dobré story a je tedy naplněním prvního cíle této práce, tedy nalezení vhodné struktury pro User Stories. Schéma 2 nastiňuje postup tvorby story s odkazy na základní principy a poznatky, které jsem ze zdrojů nasbíral.

5 Seznam zdrojů

(Wells 2009)	WELLS, Don. <i>Extremeprogramming.org</i> [online]. c2009, 28. 9. 2009 [cit. 2011-11-02]. Dostupné z WWW: < http://www.extremeprogramming.org/ >.
(Kosa 2011)	KOSA, Matúš. <i>Blog.websupport.cz</i> [online]. 15.3.2011 [cit. 2011-11-02]. WS Agilne: Extrémní Programování (XP). Dostupné z WWW: < http://blog.websupport.cz/2011/03/ws-agilne-extremne-programovanie-xp/ >.
(Jeffries 2011)	JEFFRIES, Ronald E. <i>Http://xprogramming.com</i> [online]. 2011 [cit. 2011-11-02]. Dostupné z WWW: < http://xprogramming.com/index.php >.
(Krupka 2006)	KRUPKA, Michal. <i>Krupka.inf.upol.cz</i> [online]. 14.3.2006 [cit. 2011-11-02]. Extrémní programování. Dostupné z WWW: < http://krupka.inf.upol.cz/other/XP060314.pdf >.
(Wake 2011)	WAKE, Bill. <i>XP123.com</i> [online]. c2011 [cit. 2011-11-02]. Dostupné z WWW: < http://xp123.com/ >.
(Vejandla, Sherrell 2009)	Pavan K. Vejandla and Linda B. Sherrell. 2009. Why an AI research team adopted XP practices. In <i>Proceedings of the 47th Annual Southeast Regional Conference (ACM-SE 47)</i> . ACM, New York, NY, USA, , Article 72 , 4 pages. DOI=10.1145/1566445.1566541 http://doi.acm.org/10.1145/1566445.1566541
(Stapel, Lübke 2008)	Kai Stapel, Daniel Lübke, and Eric Knauss. 2008. Best practices in extreme programming course design. In <i>Proceedings of the 30th international conference on Software engineering (ICSE '08)</i> . ACM, New York, NY, USA, 769-776. DOI=10.1145/1368088.1368197 http://doi.acm.org/10.1145/1368088.1368197
(Grisham, Perry 2005)	Paul S Grisham and Dewayne E. Perry. 2005. Customer relationships and Extreme Programming. In <i>Proceedings of the 2005 workshop on Human and social factors of software engineering (HSSE '05)</i> . ACM, New York, NY, USA, 1-6. DOI=10.1145/1082983.1083113 http://doi.acm.org/10.1145/1082983.1083113
(GRENNING 2002)	GRENNING, James. <i>Renaissancesoftware.net</i> [online]. April 2002 [cit. 2011-12-04]. Planning Poker . Dostupné z WWW: < http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf >.
(Cohn 2011)	COHN, Mike. <i>Mountain Goat Software : Mike Cohn's Blog - Succeeding with Agile</i> [online]. c2011 [cit. 2011-12-05]. ADVANTAGES OF THE “AS A USER, I WANT” USER STORY TEMPLATE. Dostupné z WWW: < http://blog.mountangoatsoftware.com/advantages-of-the-as-a-user-i-want-user-story-template >.

(Cohn 2004)	COHN, Mike. <i>Mountain Goat Software</i> [online]. Atlanta : February 10, 2004 [cit. 2011-12-05]. User Stories Applied. Dostupné z WWW: < http://www.mountangoatsoftware.com/system/presentation/file/35/UserStoriesXPAtlanta.pdf?1267636397 >.
(Cohn 2011b)	COHN, Mike. <i>Mountain Goat Software</i> [online]. c2011 [cit. 2011-12-10]. Presentations: User Stories. Dostupné z WWW: < http://www.mountangoatsoftware.com/presentations-user-stories?page=1 >.
(North 2011)	NORTH, Dan. <i>DanNorth.net : It's all behaviour</i> [online]. c2011 [cit. 2011-12-10]. What's in a Story?. Dostupné z WWW: < http://dannorth.net/whats-in-a-story/ >.