

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2014/2015
Autoři	- Jan, Berger, xberj58
Téma	Agilní vývoj ve společnosti Etnetera
Datum odevzdání	15.5.2015 23:59

Abstrakt

Cílem práce je popsat proces vývoje SW ve společnosti Etnetera v jednom menším týmu (4 – 5 lidí), který se věnuje vývoji a údržbě webové stránky (ifortuna.cz). Hlavní důraz bude kladen na popis implementace agilních metodik, především Scrumu.

Klíčová slova

Etnetera, Scrum, Agilní metodiky, vývoj softwaru

Obsah

1	Úvod	1
2	Představení společnosti Etnetera.....	2
	2.1 Zaměření.....	2
	2.2 Představení vybraného týmu	2
3	Agilní metodiky.....	3
	3.1 Příklady agilních metodik	4
4	Scrum	5
	4.1 Historie	5
	4.2 Model vývoje	5
	4.3 Role.....	5
	4.4 Artefakty	6
	4.5 Schůzky	6
	4.6 Shrnutí	7
5	Agilní vývoj ve společnosti Etnetera.....	8
	5.1 Důvody přechodu na agilní metodiky	8
	5.2 Současný stav	8
	5.2.1 Projekt Fortuna.....	9
	5.2.2 Agilní vývoj v týmu ETN Fortuna.....	9
	5.3 Směr do budoucna.....	12
	5.4 Zhodnocení.....	12
	5.4.1 Porovnání agilního a klasického vývoje	12
6	Závěr	14
7	Literatura.....	15

1 Úvod

Smyslem této práce je přiblížit agilní vývoj, tak jak se praktikuje v reálném prostředí vývojového týmu ve společnosti Etnetera. Práce se zaměřuje především na metodiku Scrum, která se oficiálně používá i když není striktně dodržována.

Největší přínos by měl být pro čtenáře, který se zajímá o metodiky vývoje softwaru nebo projektové řízení a zajímá ho, jak lze aplikovat metodiku Scrum nebo všeobecné agilní principy do praxe.

Pro porozumnění práce nejsou potřeba žádné specializované znalosti, ale je vhodné, aby čtenář měl povědomost o agilních a klasických (rigorózních) principech vývoje softwaru.

Vlastní práce je rozdělena do čtyřech částí. Nejprve je krátce představena společnost Etnetera včetně týmu, u kterého bude popisován vývoj. Následuje teoretická část s popisem agilních principů a část popisující metodiku Scrum včetně modelu vývoje, rolí, artefaktů a schůzek. V poslední a hlavní části bude popsán samotný vývoj v týmu Etnetera.

2 Představení společnosti Etnetera

Etnetera a.s. je česká společnost působící na poli vývoje webových aplikací. Je součástí Etnetera Group, byla založena roku 1997, její obrat činí 262 miliónů korun (údaj z roku 2013) a zaměstnává 165 zaměstnanců (údaj z 6/2014). (Etnetera, 2014)

Co se týče vnitřní struktury, Etnetera je rozdělena do osmi týmů, které do jisté míry fungují nezávisle na sobě. Týmy řídí jejich team leader a kromě nejvyššího jiný management není. (Etnetera, 2014)

2.1 Zaměření

Etnetera a.s. se zaměřuje především na zakázkový vývoj webových stránek a aplikací, ale nově i na vývoj aplikací pro mobilní telefony na platformě Android, Apple iOS a Windows Phone. (Etnetera, 2014) Řešení webových stránek stojí převážně na technologii Java a souvisejících technologiích s využitím CMS (content management system, systém pro zpravu obsahu) jménem jNP (jNetPublish) vyvíjeném také Etneterou. (Etnetera, 2010)

Kromě vývoje webových stránek se ostatní společnosti v rámci Etnetera Group zabývají mimo jiné online marketingem, sociálními sítěmi, e-learningovými aplikacemi a správou Google Apps. Etnetera Group tak kompletně pokrývá on-line služby. (Etnetera, 2014)

2.2 Představení vybraného týmu

Tato práce se bude věnovat především jednomu konkrétnímu týmu v rámci společnosti, jménem Synergy, jehož členem je i autor této semestrální práce. Synergy nyní čítá 15 lidí, rozdělených především do třech klientských týmů pro robustní e-shop Datartu, online sázení Fortuny a pro vývoj mobilních aplikací. (Etnetera, 2010) Vzhledem k tomu, že tyto tři týmy operují nezávisle na sobě, se práce zaměří převážně na tým spravující online sázení Fortuny. Tento tým bude dále nazýván „tým ETN Fortuna“.

3 Agilní metodiky

Důvod vzniku agilních metodik byla malá úspěšnost IT projektů (pod 35%) hlavně z důvodu malé zainteresovanosti zákazníků a nevyjasnění specifikace požadavků. Tyto faktory byly těžko adresovatelné rigorózními (klasickými) metodikami. Ty vyžadovaly detailní a hlavně neměnnou specifikaci požadavků a nulové zapojení uživatele od specifikace požadavků po akceptační testování. (Balada, 2015)

V roce 2001 proto 17 IT specialistů sepsalo v Utahu agilní manifest: (Beck, 2001)

- Objevujeme lepší způsoby vývoje software tím, že jej tvoříme a pomáháme při jeho tvorbě ostatním.
- Při této práci jsme dospěli k těmto hodnotám:
 - 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
- Jakkoliv jsou body napravo hodnotné, bodů nalevo si ceníme více.

Dále bylo definováno 12 základních principů agilních metodik stojících na agilním manifestu: (Beck, 2001)

1. Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného softwaru.
2. Víáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
3. Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.
4. Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.
5. Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.
6. Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.
7. Hlavním měřítkem pokroku je fungující software.
8. Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.
9. Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.

10. Jednoduchost - umění maximalizovat množství nevykonané práce - je klíčová.
11. Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.
12. Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším a následně koriguje a přizpůsobuje své chování a zvyklosti.

3.1 Příklady agilních metodik

Mezi agilní metodiky lze mimo jiné zařadit následující metodiky a přístupy:

Dynamic Systems Development Method, Adaptive Software Development, Feature-Driven Development, Extrémní programování, Lean Development, Scrum, Crystal metodiky, Agilní modelování. (Buchalcevová, 2009)

Z těchto metodik se v dnešní době používá hlavně Scrum a Extrémní programování (Balada, 2015) (průzkumy společnosti VersionOne za rok 2007 až 2012), ale nově se objevují Kanban a Scrumban. Kanban je jednou z technik vycházejících z výrobního procesu ve společnosti Toyota a Scrumban je kombinace Scrumu a Kanbanu. (Balada, 2015)

4 Scrum

Bude následovat popis pouze metodiky Scrum, protože tato práce má být především případovou studií agilního vývoje v jednom z týmů společnosti Etnetera, který Scrum ve svém týmu ETN Fortuna používá.

4.1 Historie

Principy metodiky Scrum použili nezávisle na sobě v devadesátých letech K. Schwaber a J. Sutherland, ale první zmínka o něm byla v rámci konference OOPSLA (popsáno v Sutherland, 1997) v roce 1995 a detailně byl popsán v roce 2001 (popsáno v Schwaber, 2002). (Balada, 2015)

4.2 Model vývoje

Scrum operuje s iterativním modelem životního cyklu vývoje, kdy se jednotlivé iterace nazývají Sprints a trvají vždy stejně dlouho, obvykle od jednoho týdne po měsíc. Výsledkem každého sprintu by měl být otestovaný a nasaditelný inkrement. (Schwaber, 2002)

Pro každý sprint je definován seznam požadavků (Scrum je pojmenovává user stories – uživatelský příběh), který se nazývá Sprint backlog. Sprint backlog vychází z Product backlog, což jsou všechny user stories pro daný produkt. User stories v Product backlog jsou řazeny podle priority, kterou jim přidělil zákazník. (Balada, 2015) User stories se přesouvají z Product backlogu do Sprint backlogu během Spring Planning meeting. Kromě toho se ještě tým schází společně se Scrum masterem každý den na schůzce zvané Daily Scrum.

4.3 Role

Ve Scrumu jsou definované tři hlavní role: Product Owner, Scrum Master a Development Team. Bude následovat jejich krátký popis:

Product Owner (vlastník)

Zastupuje zájmy zákazníka. Zodpovídá za správný popis a přiřazení priority položek v Product backlog a spolupracuje s týmem na časových odhadech délky vývoje user stories. Na konci Sprintu přebírá hotový inkrement. (Schwaber, 2002)

Scrum Master

Úkolem Scrum Master je zajistit správný chod vývojového týmu. Stará se o dodržování metodiky týmem a zajišťuje, že tým má ideální podmínky pro vývoj. (Schwaber, 2004)

Development Team (vývojářský tým)

Development Team se skládá z odborníků, kteří mají na starost dodání inkrementu na konci každého sprintu. Jsou jediní, kteří do něj přispívají. Členové se organizují sami a způsob vývoje je také na nich. (Schwaber, 2004)

4.4 Artefakty

Product backlog

Seznam všech user stories seřazených podle priority. Obsahuje všechny úkoly, které je potřeba na projektu vykonat. Každý user story by měl mít nějaký přínos pro uživatele a měl by i obsahovat informaci o tom, kdo pro koho se daná funkcionálita vytváří a proč. (Balada, 2015)

Sprint backlog

Seznam všech user stories pro daný sprint.

Burndown chart

Jedná se o graf, který zobrazuje vývoj zbývajících pracností ve Sprint backlogu v čase.

Samotné user stories se mohou nacházet ve třech stavech: připraveno, v řešení a hotovo. Členové vývojového týmu si vybírají připravené user stories, tím se dostávají do stavu *v řešení* a po dokončení nabývají stavu *hotové*. (Balada, 2015)

4.5 Schůzky

Sprint planning (naplánování Sprintu)

Schůzka sloužící na aktualizování priorit v Product backlog a zařazení user stories do Sprint backlog. Probíhá s účastí Product owner.

Daily Scrum (denní Scrum)

Každodenní schůzka, neměla by přesáhnout 15 minut a každý člen vývojového týmu by měl mluvit o následujících věcech: na čem pracoval předešlý den, na čem bude pracovat dnes a jestli existují nějaké překážky nebo omezení pro řešení jeho úloh.

Demo (ukázka)

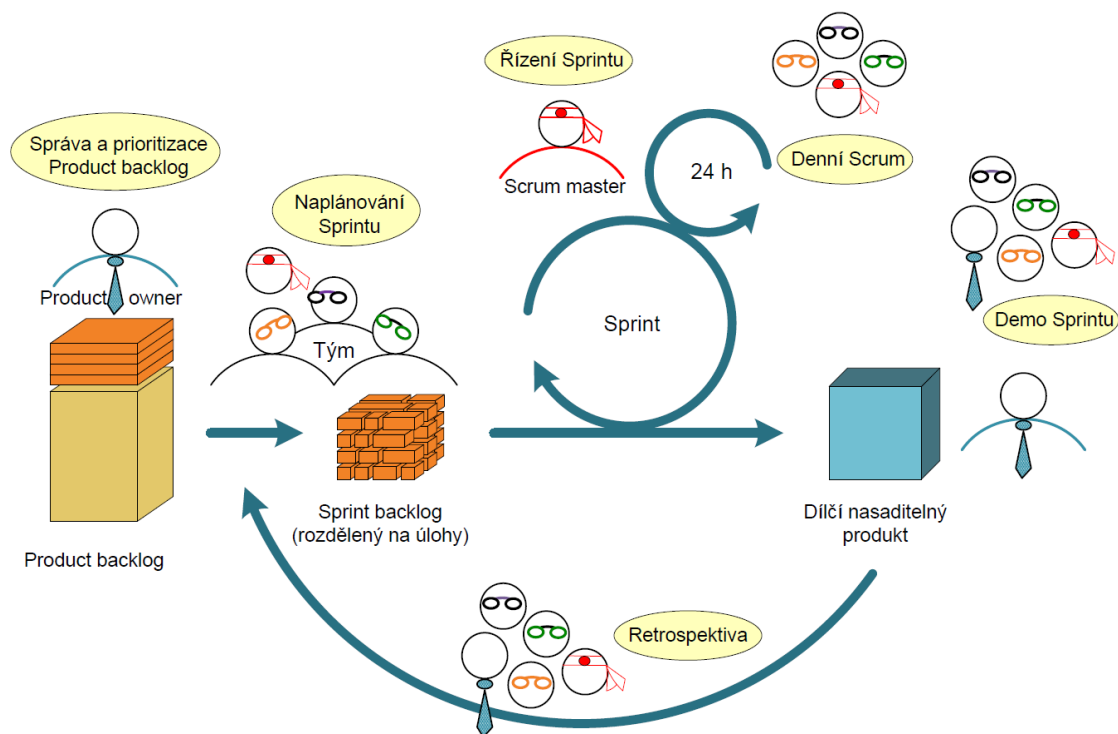
Jedná se o prezentaci inkrementu na konci Sprintu pro Product owner.

Sprint Retrospective (retrospektiva)

Zhodnocení Sprintu členy týmu i vlastníkem produktu. Každý by se měl vyjádřit, co během Sprintu fungovalo dobře a co naopak by zlepšil. Tato schůzka je zaměřena na hledání cesty k efektivnějšímu vývoji.

4.6 Shrnutí

Na obrázku Obrázek 4.1 je graficky znázorněn celý proces Scrum včetně všech rolí, artefaktů a schůzek.



Obrázek 4.1, Komplexní proces Scrum (zdroj: Balada, 2015)

5 Agilní vývoj ve společnosti Etnetera

Následující kapitola bude popisovat vývoj v týmu Synergy pomocí pracovního rámce Scrum. Úvodem budou stručně popsány důvody k přechodu na agilní přístupy z klasických vodopádových, následovat bude podrobnější popis vývoje v současné době včetně popisu souladu i nesouladu s pracovním rámcem Scrum. Závěrem poté bude popsán směr rozvoje vývoje do budoucna a subjektivní zhodnocení situace autorem.

5.1 Důvody přechodu na agilní metodiky

Začátky snah o přechod na agilní vývoj se v týmu Synergy datují od roku 2010, ale úplná snaha o implementaci metodiky Scrum začala začátkem roku 2011 a dá se říci, že byla úspěšně završena koncem téhož roku.

Do té doby se používala interní vodopádová metodika Etnetera založená na IPMA (International Project Management Association) standardech. Důvod přechodu však nebyl, že by vývoj takto nefungoval, šlo o hledání efektivnější cesty u vhodných klientů.

Situace před nástupem Scrumu vypadala tak, že pokud klient vyžadoval novou funkcionalitu, tak musel zpracovat podrobnou specifikaci, která se poslala do Synergy týmu, kde se ohodnotila pracnost a na základě náročnosti, se klient rozhodl, jestli jí bude chtít implementovat. Toto jednání vedlo k velkému mrhání času na obou stranách, ale hlavně značně prodlužovalo dobu od zadání požadavku do její implementace. Přičemž rychlost reakce na změny trhu (reakční doba) je pro činnost některých klientů naprosto zásadní.

Hlavní důvod přechodu na Scrum tedy je zkrácení doby odbavení požadavku od nápadu po nasazení.

5.2 Současný stav

Pracovní rámec Scrum se používá v týmu Synergy již čtyři roky a dle slov vedoucího týmu se neplánuje změna (dale o plánech do budoucna kapitola 5.3). Scrum není implementován doslovně a přesně, ale spíše je v duchu agilních přístupů modifikován a vyhovuje tak lépe aktuálním podmínkám.

5.2.1 Projekt Fortuna

Produkt

Tým ETN Fortuna se věnuje, jak již bylo řečeno, vývoji webového řešení online sázení společnosti Fortuna pro Českou Republiku (Fortuna, 2015b), Slovensko (Fortuna, 2015c) a Polsko (Fortuna, 2015a). Řešení stojí na platformě jNetPublish. V samotném produktu se v současné době přidává nová funkcionality nebo jsou opravovány případné chyby.

Nástroje pro podporu vývoje

Pro řízení projektu se používá software JIRA od společnosti Atlassian (více informací viz zdroj JIRA, 2001) hostovaný na doméně Fortuny. Tento produkt po správném nastavení plně podporuje rámec Scrum. Jako verzovací nástroj se používá Git (více informací viz zdroj Git, 2001).

5.2.2 Agilní vývoj v týmu ETN Fortuna

Role

Product Owner je zástupce společnosti Fortuna a úzce s Etneterou spolupracuje. Účastní se Demo, Sprint Planning i Retrospective.

Role Scrum mastera není v týmu úplně striktně dodržována. Sice plní svoji úlohu, řídí schůzky, komunikuje s klientem, ale je i součástí vývojového týmu a podílí se na inkrementu. Zároveň se stará o CI (Continuous integration), správu verzovacího prostředí a nasazovacího serveru.

Agilní Development team dodržuje Scrum dobře. Dá se říci, že všichni dělají vše a všichni spolu hojně komunikují. Nicméně se jednotliví členové specializují na různé oblasti vývoje. Mezi specializace patří frontend (Javascript, šablonovací nástroje, html), backend (Java, databáze) a testování (zátěžové, funkční).

Schůzky

Development team se skutečně jednou denně schází na Daily Scrum meeting v přesně daný čas a i formát schůzky odpovídá formátu metodiky. Schůzce se říká *Stand-up meeting* místo Daily Scrum a občas se schůzka protáhne, protože se řeší věci moc detailně. Jinak je ale každým členem vždy řečeno, co dělal minulý den, co bude dělat a jestli s něčím nemá problém.

Sprint planning probíhá vždy před začátkem sprintu a naopak Demo na konci, obě schůzky za účasti Product owner, tak jak to doporučuje pracovní rámec.

Trošku specifická je Retrospective schůzka. Neúčastní se jí jenom tým ETN Fortuna, ale celý vývojový tým i ze strany Fortuna tak, aby se hledala efektivita v celém cyklu.

Proces vývoje

Iterace Sprintu trvá v podání týmu ETN Fortuna dva týdny, na začátky se vždy do systému JIRA při Sprint planning schůzce přidají User stories. Vývojáři si je následně rozeberou a přiřadí jim v systému status *in progress*. Pro každou User story vznikne ve verzovacím systému nová větev zdrojových kódů.

Vzhledem k povaze projektu je občas do Sprint backlogu přidán User story během běžícího sprintu, protože je to klíčové pro činnosti klienta. Může se jednat o požadavek na funkcionalitu, kterou je nutné udělat okamžitě z důvodu konkurenceschopnosti. Nebo se sporadicky může jednat o požadavek na opravu kritické chyby. Takovéto požadavky vždy chodí buď se strany vlastníka produktu (PO) nebo z QA (quality assurance) oddělení společnosti Fortuna.

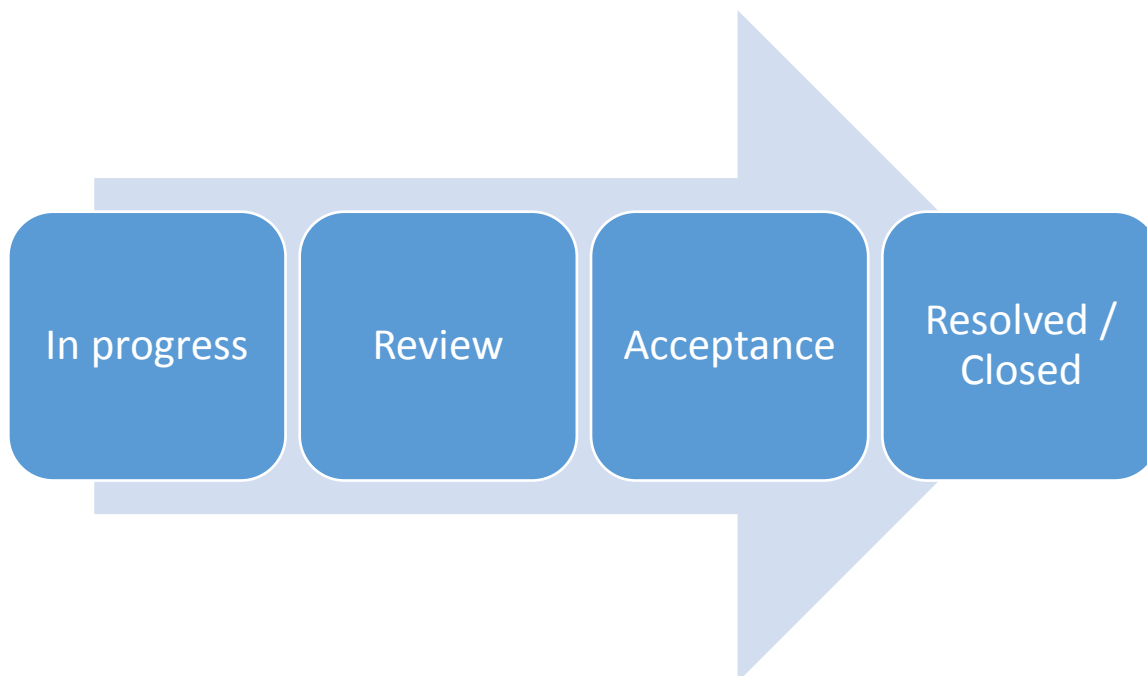
User stories, které jsou ve stavu *in progress* jsou samozřejmě komunikovány na Daily Scrum schůzkách nebo případně se zaměstnanci Fortuny pomocí systému JIRA.

Poté co je určitý *uživatelský příběh* dodělán, se v systému JIRA přesouvá do stavu *review*, čímž se indikuje skutečnost, že někdo jiný z týmu by měl provést *code review* (revize kódu druhým vývojářem).

Větev zdrojových kódů takto zkontrolovaného kódu se ve verzovacím systému spojí (*merge*) do hlavní vývojové větve.

Následně se User story v systému JIRA přesouvá do stavu *acceptance*, což je čekání na schválení Product owner nebo otestování QA oddělením společnosti Fortuna. Tomuto přístupu k akceptaci práce se v agilní terminologii říká *Daily demo*. Po schválení je User story přiřazen poslední stav a to buď *resolved*. Eventuálně je přiřazen stav *closed* v případě, že se chybu nepodařilo nasimulovat nebo z přidávání funkcionality sešlo.

Posloupnost stavů User stories je zobrazena na obrázku 5.1.



Obrázek 5.2, Stavby User stories (zdroj: Autor)

Úspěšně schválené a otestované User stories se stávají součástí následujícího inkrementu v podobě WAR (**W**eb **a**pplication **A**Rchive) souboru, který se automaticky nasazuje na produkční servery společnosti Fortuna na konci Sprintu.

Rozdíly ve vývoji oproti metodice Scrum

Většinou se vydávání inkrementů po každém Sprintu dodržuje, ale někdy je potřeba vydat inkrement i dvakrát za Sprint nebo naopak jednou za dva Sprints. Je to z důvodu povahy činnosti klienta, kdy musí flexibilně reagovat na konkurenci.

Další rozdíl je v tom, že se nezpracovává Burndown chart. Důvod je takový, že se zavádí Daily demo, kdy přesouvání User stories neřídí do finálního stavu *resolved* části vývojáři, ale Product owner (který může iniciovat testování). Product owner však tuto činnost nedělá důsledně každý den a proto by Burndown chart byl značně zkreslený.

Poslední nedodržení metodiky spočívá v tom, že u User stories často chybí informace, pro koho jsou určeny a co mají přinést. Většinou existuje pouze popis chyby nebo požadavků na funkcionalitu. Je to z důvodu, že tyto informace jsou dostatečně komunikovány s klientem. Stejně tak má vývojář dostatečnou doménovou expertnost, takže zadání rozumí. Nemá proto cenu vytvářet zbytečnou dokumentaci a vše je v souladu s jedním z principů agilní myšlení: „Snažte se maximalizovat množství práce, které není potřeba udělat“.

5.3 Směr do budoucna

Tým ETN Fortuna chce dlouhodobě zůstat u metodiky Scrum. Je s ním velká spokojenost u vedení Synergy týmu, u vývojářů a i u klienta. Plánuje se ještě určitě optimalizovat a dále přizpůsobovat pro potřeby týmů a klientů, ale v blízké budoucnosti se žádná konkrétní změna neplánuje. Jediné co lze očekávat v případě velkého zvětšení týmu je rozšíření na rámec Scrum of Scrums.

5.4 Zhodnocení

V této části práce dojde ke zhodnocení agilního vývoje, co se na něm autorovi líbí a co naopak ne. Bude se jednat především o subjektivní srovnání s klasickými metodikami, se kterými se autor setkal v tříměsíčním působení ve společnosti Unicorn, kde se používala metodika inspirované RUP (Rational Unified Process) pro vývoj a Prince2 (**PR**ojects **IN** Controlled **E**nvironments) pro projektové řízení.

5.4.1 Porovnání agilního a klasického vývoje

Na agilních metodikách je především dobré to, že nechávají jednotlivým členům vývojového týmu velkou volnost ve způsobu řešení implementace. To hlavně z důvodu přístupu, že úkoly členové řeší od analýzy problému, přes návrh a implementaci, až po testování a doladování. Z pohledu vývojáře je toto lepší než úzká specializace v klasických metodikách, protože to podporuje kreativitu a jednotlivé úkoly je zábavnější řešit. Z pohledu přínosu pro klienta to může vývoj v některých situacích urychlit.

Na druhou stranu agilní metodiky kladou větší nároky na odbornost a profesionalitu vývojářů, protože toho musí více vědět. Agilní vývoj ale i klade větší nároky na sociální stránku vývojářů. Lidé spolu musí často a rádi komunikovat, protože neexistují dlouhé stránky dokumentace, kde lze všechno najít, ale místo toho je třeba zajít k sousednímu stolu.

Osobní názor autora je takový, že agilní vývojáři musí mít mezi sebou dobré vztahy a určitě se dá říci, že Scrum komunikaci mezi lidmi podporuje a výsledkem je tým lidí, kteří ze zkušenosti autora jsou k sobě přátelštější, což je nesporná výhoda. Nutno dodat, že takovéto prostřední musí být reflektováno i firemní kulturou a podporou ze strany vrcholového vedení.

Nutné je samozřejmě i podpora ze strany zákazníků, která je podle slov vedoucího týmu Synergy kritická pro plnohodnotný a úspěšný agilní vývoj.

Klasické metodiky zase mají výhodu v tom, že jsou více robustní a více dokáží tolerovat chyby jednotlivce.

Autorův závěr z relativně krátké zkušenosti s klasickými i agilními přístupy je takový, že agilní vývoj vyžaduje schopnější a více kvalifikované lidi, které dále podporuje v seberozvoji.

6 Závěr

Hlavní cíl práce kromě formálního popisu agilních principů a metodiky Scrum byl popsat aplikaci Scrumu v praxi. V práci je vlastní případová studie poměrně podrobně popsána a lze se dozvědět důvody přechodu na agilní vývoj, jakým způsobem probíhá vývoj a v čem se úplně pracovní rámce nedodržuje (a proč).

Čtenář by si měl po přečtení této práce být schopen udělat rozumný obrázek o stavu vývoje v Synergy týmu a na základě toho se zamyslet v jakém prostředí je vhodné agilní přístup k vývoji aplikovat a na co se při tom zaměřit.

7 Literatura

BALADA, Jakub. 2015. *Agilní metodiky v komplexním prostředí*. Praha. Dostupné také z: http://isis.vse.cz/zp/index.pl?podrobnosti_zp=20701. Disertační práce. Vysoká škola ekonomická v Praze.

BECK, Kent, et al., 2001. *Manifesto for Agile Software Development* [online]. [vid. 2014-01-10]. Dostupné z: <http://agilemanifesto.org/>

BUCHALCEVOVÁ, Alena. 2009. *Metodiky budování informačních systémů*. Vyd. 1. Praha: Oeconomica, 205 s. Vysokoškolská učebnice (Oeconomica). ISBN 978-80-245-1540-3.

ETNETERA. 2014. *Etnetera* [online]. Praha [cit. 2015-04-22]. Dostupné z: <http://www.etnetera.cz/>

ETNETERA. 2010. *JNetPublish* [online]. Praha [cit. 2015-04-23]. Dostupné z: <http://www.jnetpublish.cz/>

FORTUNA. 2015. *Efortuna.pl* [online]. [cit. 2015-04-30]. Dostupné z: <http://www.efortuna.pl/>

FORTUNA. 2015. *Ifortuna.cz* [online]. [cit. 2015-04-30]. Dostupné z: <http://www.ifortuna.cz/>

FORTUNA. 2015. *Ifortuna.sk* [online]. [cit. 2015-04-30]. Dostupné z: <http://www.ifortuna.sk/>

Git. 2001. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-04-30]. Dostupné z: [https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software))

JIRA. 2001. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [cit. 2015-04-30]. Dostupné z: <https://en.wikipedia.org/wiki/JIRA>

SCHWABER, Ken. 2002. *Agile software development with Scrum*. Upper Saddle River: Prentice Hall, xvi, 158 s. Series in agile software development. ISBN 01-306-7634-9.

SCHWABER, Ken. 2004. *Agile project management with Scrum*. Redmond, Wash.: Microsoft Press, xix, 163 p. ISBN 07-356-1993-X.

SUTHERLAND, Jeffrey Victor. 1997. *Business object design and implementation: OOPSLA '95 workshop proceedings, 16 October 1995, Austin, Texas*. New York: Springer, viii, 165 p. ISBN 35-407-6096-2.