

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství



Diplomová práce

Projektový management a metodika RUP

Marek Pučelík

© 2011 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Marek Pučelík

obor Informatika

Vedoucí katedry Vám ve smyslu Studijního a zkušebního řádu ČZU v Praze čl. 17 odst. 2 určuje tuto diplomovou práci.

Název práce: **Projektový management a metodika RUP**

Osnova diplomové práce:

1. Úvod
2. Cíl práce a metodika
3. Teoretická východiska projektového managementu
4. Teoretická východiska metodiky RUP
5. Postup využívání metodiky RUP v projektovém managementu při tvorbě softwaru
6. Příklad z praxe
7. Závěr
8. Seznam použitých zdrojů
9. Přílohy

Rozsah hlavní textové části: 60 - 80 stran

Doporučené zdroje:

Doležal J., Lacko B., Máchal P. a kol.: Projektový management podle IPMA, Grada Publishing, Praha 2009

Fiala, P.: Řízení projektů. VŠE, Praha 2000

Němec, V.: Projektový Management. Grada 2005

Rosenau, M.D.: Řízení projektů, Computer Press, Brno, 2003

Svozilová, A. Projektový management. Grada, Praha, 2006.

Langrová, P, Šubrt, T.: Projektové řízení II. ČZU Praha, 2005

Šubrt, T., Langrová, P.: Projektové řízení I. ČZU Praha, 2005

Šubrt, T., Bartoška, J.: Projektové řízení III. ČZU Praha, 2007

Vedoucí diplomové práce: **doc. Ing. Tomáš Šubrt, Ph.D.**

Termín odevzdání diplomové práce: duben 2011



.....
Vedoucí katedry



L.S.



.....
Děkan

V Praze dne: 3. 2. 2010

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Projektový management a metodika RUP" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 8.4.2011

Poděkování

Rád bych touto cestou poděkoval panu doc. Ing. Tomáši Šubrtovi, Ph.D. za cenné rady a odbornou pomoc, kterou mi poskytl.

Projektový management a metodika RUP

Project Management and RUP

Souhrn

Diplomová práce Projektový management a metodika RUP pojednává o využitelnosti metodiky RUP při řízení projektů v malé softwarové společnosti. Na základě dostupné literatury jsou v literární rešerši popsány stěžejní aktivity projektového managementu. V praktické části této práce byla provedena analýza na základě získaných dat informačního systému vybrané malé společnosti. Výstupem analýzy je určení a pojmenování problémů v řízení projektů. Příčiny těchto problémů byly nalezeny v definovaných procesech společnosti a v nedodržování nastavených procesů projektového řízení v jednotlivých fázích softwarového vývoje. Práce obsahuje popis silných a slabých stránek metodiky RUP a návrh změn v procesech ve vybrané malé softwarové společnosti.

Summary

Diploma thesis Project Management and Methodology of Rational Unified Process (RUP) deals with availability of methodology RUP within project management in software company. On the basis of accessible literature there are described fundamental activities of the project management and discipline of RUP methodology in literature retrieval. In practical part of the diploma thesis it was analyzed on the base of acquired data of informational system within the software company. The output of analysis is determination and description of problem in project management. The cause of these problems were found in defined processes within society and in breaking defined processes of project management - in particular phases of software development. The diploma thesis covers description of positive and negative parts of RUP methodology and layout of changes in processes of the software company.

Klíčová slova: projekt, Rational Unified Process, metodika, projektový manažer, disciplíny RUP, Memos Software s.r.o., management

Keywords: project, Rational Unified Process, methodology, project manager, RUP disciplines, Memos Software s.r.o., management

OBSAH

1	ÚVOD	4
2	CÍLE A METODIKA PRÁCE.....	5
2.1	CÍLE PRÁCE	5
2.2	METODIKA PRÁCE	5
3	LITERÁRNÍ REŠERŠE	6
3.1	PROJEKTOVÝ MANAGEMENT	6
3.2	RATIONAL UNIFIED PROCESS	19
	PRAKTICKÁ ČÁST	30
4	MEMOS SOFTWARE.....	30
4.1	PŘÍKLAD PROJEKTU	30
4.2	INTERNÍ SYSTÉMY SPOLEČNOSTI.....	33
4.3	CÍLE A VYMEZENÍ PROBLÉMU.....	35
4.4	ANALÝZA	35
4.5	VÝSTUP Z ANALÝZY.....	36
5	ŘÍZENÍ PROJEKTŮ SPOLEČNOSTI MEMOS A VYUŽITÍ METODIKY RUP	38
5.1	PŘEDPROJEKTOVÉ VYJEDNÁVÁNÍ.....	38
5.2	SBĚR A SPRÁVA POŽADAVKŮ	45
5.3	ANALÝZA A NÁVRH.....	51
5.4	IMPLEMENTACE	57
5.5	TESTOVÁNÍ A KONTROLA	59
5.6	NASAZENÍ	64
5.7	KONFIGURACE A ZMĚNY.....	66
5.8	SPRÁVA PROSTŘEDÍ	68
5.9	UKONČENÍ PROJEKTU	70
6	CELKOVÉ HODNOCENÍ VYUŽITELNOSTI METODIKY RUP V MALÉ SOFTWARE SPOLEČNOSTI.....	73
6.1	SILNÉ STRÁNKY	73
6.2	SLABÉ STRÁNKY.....	73

7	PROCESY VE SPOLEČNOSTI MEMOS	75
7.1	HODNOCENÍ PROCESŮ VE SPOLEČNOSTI MEMOS	75
7.2	NÁVRH PROCESŮ VE SPOLEČNOSTI MEMOS	75
8	ZÁVĚR	79
	SEZNAM LITERATURY	80
	PŘÍLOHA A - EWAY (CRM SYSTÉM).....	82
	PŘÍLOHA B - REDMINE	92

1 Úvod

Na projektový management lze nahlížet z více úhlů pohledu, jedná se o složitý komplex problematiky oblasti řízení. Tato diplomová práce se zaměřuje na využívání ověřených postupů řízení projektů při vývoji softwaru.

Softwarová řešení se v dnešní době řídí pomocí projektového managementu. Osobou pověřenou k řízení projektu je projektový manažer. Ten zodpovídá za efektivnost procesů a správnost jejich výstupů. Tato práce se opírá o zkušenosti autora, kterých nabyl jako projektový manažer malé softwarové společnosti Memos Software s.r.o. (Memos).

Snaha zvýšit pravděpodobnost úspěchu projektu vede k tvorbě metodik softwarového vývoje. Metodika Rational Unified Process (RUP) nabízí dlouhodobě ověřené postupy, které mají základ v nejlepších praktikách vývoje softwaru. Autor této práce mohl na využívání metodiky RUP nahlížet jako člen projektového týmu v největší české softwarové společnosti Unicorn a.s. Na základě těchto zkušeností se diplomová práce zabývá hodnocením přínosu zavedení metodiky RUP do malé softwarové společnosti.

V teoretické části této práce jsou shrnuty a uspořádány informace o projektovém managementu a metodice RUP na základě dostupné literatury. Na tuto část navazuje v praktické části představení společnosti Memos, které zahrnuje pojmenování nedostatků ukončených projektů a určení jejich příčin. Dále jsou v této práci konfrontovány procesy navržené ve společnosti Memos a procesy definované metodikou RUP. Poté následuje zhodnocení použitelnosti metodiky RUP v malé softwarové společnosti a navržení procesů ve společnosti Memos snižující rizika neúspěšnosti projektu.

Přestože se tato práce soustředí především na integraci metodiky RUP, je dobré brát na zřetel, že projektové řízení neznamena jen používání metod a technik, byť základy by měl znát každý člen projektového týmu. Projektové řízení znamená především určitou filozofii a styl práce, určitý způsob myšlení.

2 Cíle a metodika práce

2.1 Cíle práce

Hlavním cílem této diplomové práce je nalezení možné využitelnosti metodiky RUP v malé softwarové společnosti.

Dílčí cíle práce jsou:

- Nalezení a pojmenování hlavních procesních nedostatků vybrané společnosti.
- Popis procesů společnosti.
- Navrhnutí úprav procesů společnosti

2.2 Metodika práce

V praktické části se tato diplomové práce opírá o procesy definované společností Memos Software s.r.o., která se zabývá vývojem software na zakázku. Tato společnost zaměstnává 40 lidí, čímž se řadí mezi malé podniky.

Data informačního systému společnosti Memos budou použita pro nalezení nedostatků v nastavených procesech společnosti. Součástí informačního systému jsou i data o implementovaných projektech. Nad daty projektů budou vykonány SQL dotazy pro nalezení zobecňujících vlastností výsledků všech uskutečněných projektů. Pro získání statistických výsledků byly použity funkce v aplikaci Microsoft Excel.

Práce je dále založena na metodě pozorování a vychází ze zkušeností autora. Na základě pozorovaných a popsáných procesů společnosti se porovnáním s procesy metodiky RUP určí slabá místa v nastavených procesech společnosti.

3 Literární rešerše

3.1 Projektový management

3.1.1 Úvod

Od dob, kdy lidé poznali dělbu práce a začali vytvářet skupiny k dosažení cílů, kterých by jako jedinci nedosáhli, se řízení stalo nezbytným pro zabezpečení koordinace individuálních úsilí. S růstem cílů společnosti rostl význam vůdců a organizátorů. Řízení neboli management se tak stává jednou z nejdůležitějších lidských činností.

Projektové řízení (i když se tento termín objevil až ve 20. století) průkazně existuje po celý dlouhý věk lidské civilizace. Dokumentují to velké stavby na začátku civilizace, které by bez řízení vzniknout nemohly. Důkazy tohoto tvrzení můžeme například najít v Egyptě v podobě známých pyramid. Pyramidy byly postaveny za třetí a čtvrté dynastie faraónů přibližně okolo roku 2600 př.n.l. Stupňovitá pyramida v Sakkaře a zvláště monumentální pyramidy v Gize nás přesvědčují o tom, že jejich stavitel musel mít od počátku jasno o tom, jak budou vypadat, odkud bude brát materiál na stavbu, jak ho bude dopravovat a kolik lidí bude na stavbu potřebovat.

Zmínky o projektovém řízení jsou i v Bibli, ve Starém zákoně, kde se popisuje stavba Šalamounova chrámu v Jeruzalému. Je tam popsáno, jaký materiál bude použit, jak je smluvně zajištěn, kdo bude na chrámu pracovat a jakou bude dostávat mzdu. Král Šalamoun uplatnil projektové řízení okolo roku 960 př.n.l. Podobných příkladů bychom mohli vyjmenovat celou řadu, jen namátkou můžeme ještě jmenovat stavbu Velké čínské zdi nebo stavby chrámů a amfiteátrů v Řecku, stavby silnic a akvaduktů ve starodávné říši římské. (Skalický, 2008)

3.1.2 Vymezení projektového managementu

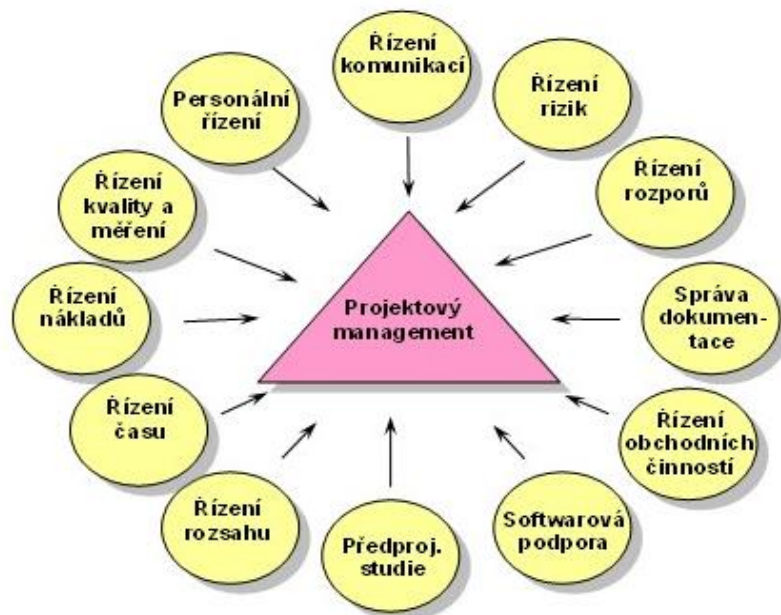
Hledáme-li rozdíly mezi běžně chápaným pojetím managementu a projektovým managementem, vyjdeme z obecných definic managementu. *“Projektový management je souhrn aktivit spočívající v plánování, organizování, řízení a kontrole zdrojů společnosti s relativně krátkodobým cílem, který byl stanoven pro realizaci specifických cílů a záměrů.”*

(Kerzner in Svozilová, 2007, s. 19). Jiná definice říká, že: *“Projektový management je aplikace znalostí, nástrojů a technologií na aktivity projektu tak, aby tyto splnily požadavky projektu.”* (PMBOK in Svozilová, 2007, s. 19).

Projektový management se liší od běžné formy operativního řízení zejména svou dočasností a v přidělení zdrojů pro jeho realizaci podle potřeb projektu. Pokud je dosaženo cílů u projektu, projekt končí, pokud je dosaženo cílů u operativního řízení, jsou nastaveny nové cíle a práce jednotky pokračuje. Pro projekt jsou plánovány a přiděleny pracovní, finanční nebo technologické zdroje podle jeho plánovaných potřeb a po jeho skončení jsou tyto zdroje spotřebovány nebo převedeny do jiných projektů. Zdroje pracovních jednotek podléhajících operativnímu řízení jsou kontinuálně plánovány a doplňovány, po ukončení potřeby jejich užívání mohou být uvolněny pro skladování, likvidaci nebo převod mimo společnost jiným způsobem, který však vždy nemusí znamenat jejich efektivní využití, neboť tyto zdroje nemusí patřit k primárním prvkům řízení a náklady na jejich nečinnost nejsou průběžně sledovány. (Svozilová, 2007, s. 20).

3.1.3 Znalostní báze projektového managementu

Pojem znalostní báze zavedl do projektového řízení Projektový institut (PMI). Celková báze znalostí projektového managementu byla rozdělena do devíti oblastí (PM Body of Knowledge). Byl položen základ pro metodiku projektového managementu, z které jsou odvozeny všechny další metodiky tím, že je přidána ještě další oblast znalostí, kterou PM využívá, nebo je některá vynechána, případně dvě sloučeny do jedné. (Skalický, 2008)



Obr. č. 1: Znalostní okruhy projektového managementu (Zdroj: Skalický, 2008)

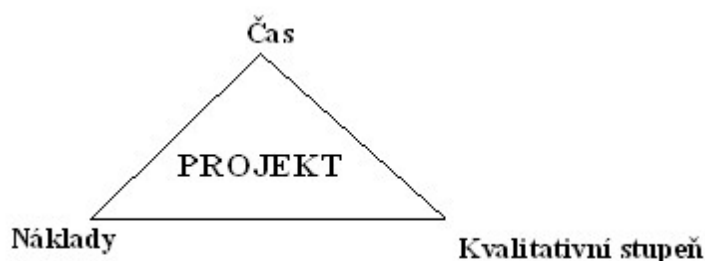
3.1.4 Projekt

Hlavními jevy a veličinami, které vytvářejí hranice projektového prostředí a které jsou ovlivňovány v průběhu projektu a které působí v řídicích a kontrolních procesech, jsou předmět projektu, čas a náklady. Dalšími veličinami pak jsou míra neurčitosti a rizika a kvalita realizovaných vstupů. (Svozilová, 2007, s. 20)

Nejdůležitějším prvkem projektového řízení je projekt. *“Projekt je jakýkoliv jedinečný sled aktivit a úkolů, který má*

- *dán specifický cíl, který má být jeho realizací splněn,*
- *definováno datum začátku a konce uskutečnění,*
- *stanoven rámec pro čerpání zdrojů potřebných pro jeho realizaci.”*(Kerzner, in Svozilová, 2007, s. 22)

Další definice uvádí, že: *“Projekt je dočasné úsilí vynaložené na vytvoření unikátního produktu, služby nebo určitého výsledku.”* (PMBOK in Svozilová, 2007, s. 22) Časová omezenost (temporary) znamená, že každý projekt má definován začátek a konec. Unikátnost (unique) znamená, že produkt nebo služba se nějakým významným způsobem liší od všech podobných produktů nebo služeb. (Skalický, 2008)



Obr. č. 2: Parametry projektu (Zdroj: Skalický, 2008)

3.1.5 Cíle projektu

Cíl projektu je nová hodnota – předmět, služba nebo jejich kombinace, která je výsledkem projektu a je reprezentována popisem určitého stavu, jenž má v budoucnosti existovat. (Svozilová, 2007, s. 78)

Cílem veškerého projektového snažení je vytvoření určitého unikátního produktu – předmětu, služby nebo jejich kombinace, která naplní očekávání zadavatele projektu a přispěje k dosažení jeho strategického nebo taktického cíle, který souvisí s jeho vlastními aktivitami. Produkt projektu je cíl, výsledek nebo jiný výstup projektu, který má být realizací projektu vytvořen. (Svozilová, 2007, s. 24)

Cíle projektu představují slovní popis účelu, jehož má být prostřednictvím realizace projektu dosaženo. Cíle projektu a jejich jednoznačná definice vytvořená před zahájením prací na projektu jsou předpokladem uzavření kontraktu, který správně a spravedlivě popíše obchodní vztah mezi zákazníkem a dodavatelem.

Vytvoření vhodných podmínek pro realizaci projektu ve fázi formulace jeho cílů lze příznivě ovlivnit použitím techniky SMART (Specific, Measurable, Assignable, Realistic a Time-bound). Cíle mají tedy být specifické a konkrétní, mají být opatřeny měřitelnými parametry pro rozpoznání, zda bylo cíle dosaženo. Dále mají být přidělitelné jedinému subjektu s odpovědností a autoritou k výkonu rozhodnutí. Dále mají být dosažitelné s použitím disponibilních zdrojů a realistické a časově ohraničené. (Svozilová, 2007, s. 79)

3.1.6 Rozpočet projektu

Rozpočet projektu je rámec pro čerpání zdrojů pro realizaci projektu. Každý projekt musí mít stanoven limit čerpání nákladů, který vychází z předpokládaného rozsahu využití materiálů, technologií a oceněného rozpisu prací v členění umožňujícím kontrolu skutečného postupu projektu.

Pro dosažení potřebné přesnosti odhadu budoucích nákladů projektu a kvůli možnosti efektivní kontroly musí být rozpočet rozpracován v detailech podle jednotlivých nákladových druhů. S ohledem na jednotlivé realizační složky a jejich časové rozložení jsou jednotlivé údaje zpravidla vyjádřeny ve finanční podobě.

Rozpočet projektu je součástí plánu projektu a obsahuje celou řadu ukazatelů, které je nutno při realizaci projektu dodržet. Návrh projektu provádějí zkušení manažeři projektu. (Skalický, 2008)

3.1.7 Organizační struktura projektu

Jako dočasné soustředění aktivit a zdrojů je každý projekt realizován v určitém prostředí – okolí projektu. Mezi projektem a okolím existují určité vzájemné záměrné, více nebo méně žádoucí vazby, které mohou být negativní, ale rovněž pozitivní. (Svozilová, 2007, s. 25)

Řízení lidských zdrojů projektu (Project Human Resource Management) obsahuje procesy pro nejefektivnější využití lidí účastnících se projektu. Zahrnuje všechny účastníky projektu - sponzory, zákazníky, investory, jednotlivé dodavatele a ostatní. S tím souvisí organizační plánování, které obsahuje identifikování, dokumentování a přidělení projektových rolí, odpovědnosti a vytvoření informační hierarchie. (Skalický, 2008)

Každý projekt má svého zákazníka, který má zájem na realizaci projektu a je jeho investorem nebo zadavatelem. Jedná se zpravidla o budoucího uživatele výstupů produktu projektu nebo o investora, pro nějž znamená realizace projektu zvýšení potencionální úspěšnosti na trhu prostřednictvím nového produktu nebo služby, která je předmětem projektu.

Zákazník projektu je tedy společnost nebo její část, která je zadavatelem projektu a již budou výsledky projektu sloužit pro naplnění určitého strategického záměru nebo změny.

Klíčovou osobou ve skupině zákazníka projektu je sponzor projektu, který je formálním nositelem zpravidla nejvyšší rozhodovací autority v projektu. Sponzor projektu je tedy manažer zákazníka, který má autoritu dostatečnou k rozhodování o fundamentálních aspektech projektu – předmětu projektu, rozpočtu a časovém rámci projektu.

Dodavatelem projektu je společnost nebo její část, která na základě kontraktu se zadavatelem projektu poskytuje realizační zdroje a know-how potřebné k dosažení požadovaného výsledku projektu.

Manažer projektu je osoba odpovědná za splnění cílů projektu při dodržení všech stanovených charakteristik projektu. (Svozilová, 2007, s. 27-29) Pracovní náplň projektového manažera se může hodně lišit podle podmínek konkrétního oboru i organizace (a někdy je na první pohled těžko čitelná), ale přesto v nich najdeme jisté společné úkoly, které jsou povinností většiny projektových manažerů, bez ohledu na obor či organizaci. (Svozilová, 2007, s. 47-49)

Řešení projektů probíhá vždy v širším prostředí, než jaké představuje projekt samotný. Projektoví manažeři se při práci na projektech musí řídit systémovým přístupem. To znamená, že musí o projektech přemýšlet vždy ve větším kontextu celé organizace.

Každá organizace má čtyři důležité rámce: strukturální, lidských zdrojů, politický a symbolický. Úspěšný projektový manažer musí porozumět všem těmto aspektům organizace. Strukturální rámec je zaměřen na role a povinnosti různých skupin, které tak naplňují cíle a politiky stanovené vrcholovým vedením firmy. Rámec lidských zdrojů se zaměřuje na dosažení harmonie mezi potřebami organizace jako celku a potřebami jednotlivých osob. Politický rámec je souhrnné označení organizační a osobní politiky a symbolický rámec se zaměřuje na symboly v dané organizaci a jejich význam.

Velký vliv na práci projektového manažera má také organizační struktura dané společnosti či organizace, zejména pak jeho autorita (množství pravomoci). Tři nejdůležitější modely představuje funkcionální, projektová a maticová organizační

struktura. Projektový manažer má největší pravomoci v organizaci s čistě projektovou strukturou, zatímco v maticové organizaci má střední úroveň a v čistě funkční organizaci jich má nejméně.

Na řízení projektů má velký vliv také organizační neboli firemní kultura. Práce na projektu mohou nejlépe probíhat v takové organizaci, kde se zaměstnanci umí s firmou hodně ztotožnit, kde se při pracovních aktivitách podporují skupiny, dále kde je silná integrace organizačních jednotek či oddělení, vysoká tolerance k rizikům, výkonnostní odměňování, vysoká tolerance ke konfliktům, a kde je organizace zaměřena na otevřenost systémů a umí správně vyvážit zaměření na lidi, jejich řízení a orientaci na výsledky. (Schwalbe, 2007, s. 84)

Patnáct důležitých součástí profese projektového manažera, které jsou pro dobré řízení projektu podstatné:

1. Definovat rozsah projektu
2. Identifikovat účastníky, rozhodovatele a eskalační postupy
3. Navrhnout podrobný seznam úkolů (strukturu rozpisu prací)
4. Odhadnout časové požadavky
5. Navrhnout prvotní vývojové diagramy pro řízení projektu
6. Stanovit požadované zdroje a finanční rozpočet
7. Posoudit požadavky na projekt
8. Identifikovat a posoudit rizika
9. Připravit nouzový plán
10. Identifikovat vzájemné závislosti
11. Identifikovat a sledovat kriticky důležité milníky
12. Účastnit se na revizích jednotlivých etap projektu
13. Zajistit potřebné zdroje
14. Kontrolovat proces řízení změn
15. Sestavovat zprávy o stavu projektu

(Schwalbe, 2007, s. 95)

3.1.8 Faktory úspěchu projektu

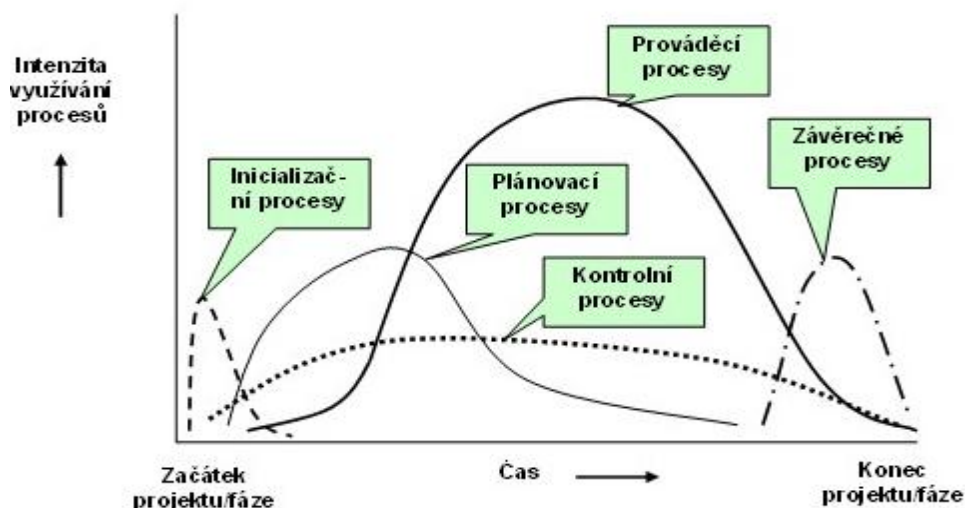
Studie společnosti Standfish group z roku 2001 se zabývá tím, jaké faktory nejvíce přispívají k úspěchu z oboru informačních technologií:

- podpora firemního vedení
- zapojení uživatelů
- zkušený projektový manažer
- jasně stanovené podnikatelské cíle
- minimální možný rozsah
- standardní softwarová infrastruktura
- pevné základní požadavky
- formální metodologie
- spolehlivé odhady
- ostatní kritéria, např. drobné milníky, odborní zdatní zaměstnanci

Důležité je, že studie uvádí hned na prvním místě podporu firemního vedení, zatímco zapojení uživatelů je až na druhém místě (ve starších studiích zaujímal právě tento faktor první místo). Dále poznamenejme, že firemní vedení či obecně vedoucí pracovníci mohou výrazně ovlivnit i několik dalších faktorů uvedených samostatně - mohou tak povzbudit zapojení uživatelů, definovat přesnější podnikatelské cíle, připojit k projektu zkušeného manažera, používat standardní softwarovou infrastrukturu a dodržovat vhodnou formální metodologii. Další významné faktory úspěchu souvisí s dobrým řízením rozsahu projektu s jeho časového plánu, tedy např. stanovení minimálního možného rozsahu, pevně daných základních požadavků a spolehlivých odhadů. Ve skutečnosti plných 97 procent úspěšných projektů bylo vedeno zkušenými projektovými manažery, kteří často dovedou dále ovlivnit všechny uvedené faktory a dále tak zvýšit pravděpodobnost úspěchu celého projektu. (Svozilová, 2007, s. 46-47)

3.1.9 Význam fází projektu

Životní cyklus projektu je souborem následných fází projektu, jejichž názvy a počet jsou určeny potřebami kontroly organizace, která je v projektu angažována. (Svozilová, 2007, s. 38)



Obr. č. 3: Projektová fáze (Zdroj: Skalický, 2008)

Každá projektová fáze představuje jeden nebo více určitých uzavřených celků (většinou se jedná o nějaký postupný cíl projektu). Cílem projektové fáze nebo jejím výstupem jsou určité, hmatatelné pracovní výsledky snadno kontrolovatelné, jako např. studie proveditelnosti, detailní konstrukční dokumentace nebo výroba prototypu. Tyto pracovní celky a tudíž i fáze jsou částmi celkového sekvenčního logického návrhu vlastního projektu. (Skalický, 2008)

3.1.10 Vymezení potenciálního projektu a iniciace projektu

Prvním krokem v řízení projektů je rozhodnout, které projekty vůbec začneme řešit. Je při tom velice důležité, aby do procesu plánování v informačních technologiích byli zapojeni i manažeři z jiných oddělení než jen z informačních technologií, protože odborníci na informační technologie mohou díky nim lépe porozumět celkové strategii organizace a identifikovat oblasti podnikání firmy, které tuto strategii podpoří. Dalším krokem v procesu plánování je provedení analýzy oblastí podnikání. Tato analýza načrtne

věcné či podnikatelské procesy, které jsou nejdůležitější součástí naplnění strategických cílů, a napomůže stanovit, jaké z procesů dovedou nejlépe využít výsledků informačních technologií. Poté přecházíme k dalšímu kroku, kterým je definování potencionálních projektů z informačních technologií, jejich rozsahu, přínosů a omezení. Posledním krokem plánovacího procesu u projektů z informačních technologií je výběr konkrétních projektů k řešení a přiřazení neboli alokace zdrojů. (Schwalbe, 2007, s. 151)

Je velmi obtížné dosáhnout úspěchu u velkého projektu z informačních technologií (a vlastně i u malého projektu), pokud si daná organizace neuvědomuje hodnotu informačních technologií jako takových. (Schwalbe, 2007, s. 85).

Zakládající listina projektu je dokument, který formalizuje existenci projektu, přiděluje manažerovi projektu autoritu pro použití zdrojů na naplnění požadavků spojených s realizací projektu. (Svozilová, 2007, s. 76)

Předběžná definice předmětu projektu je dokument, který srozumitelně a jednoznačně definuje všechny požadované cíle projektu, a to ve stavu aktuálních poznání vzhledem k vývojovému stupni projektu. (Svozilová, 2007, s. 77)

Hlavním účelem tohoto dokumentu je získat všechna potřebná schválení vyššího managementu pro realizaci projektu. Předběžná definice předmětu projektu standardně obsahuje popis problému, požadavek zákazníka, který je příčinou vzniku požadavku. Dále obsahuje hlavní cíl projektu a konkrétní cíle projektu, tedy podrobnější členění hlavního cíle do komponentů, které přesněji popíší rozsah řešeného tématu, jasný a jednoznačný popis vlastností výstupu a to na úrovni poznání v okamžiku sestavení dokumentu. Dále obsahuje kvantifikovaná kritéria, jejichž dosažením budou naplněny cíle projektu. A konečně seznam a popis potencionálních problémů a známých omezení, jimiž může být projekt zatížen. (Svozilová, 2007, s. 77)

3.1.11 Řízení rizik projektu

Řízení rizik je neustálý proces, který se odehrává v průběhu všech fází životního cyklu projektu, od počátečního nápadu po ukončení projektu. Znalosti týkající se řízení rizik mohou významně přispět k úspěchu budoucích projektů.

Často používaná metoda snižující neurčitost provázející každé jednotlivé riziko je založena na tzv. principu postupnosti, což znamená snížení neurčitosti odhadu tím, že je odhadovaná položka rozložena na menší části. Pokud má být snížena odchylka odhadů nákladů na projekt, měly by být vybrány položky nákladů s vysokou neurčitostí a rozloženy na menší části. Proces rozkládání je opakován do té doby, než odchylky jednotlivých komponent nákladů klesnou pod přijatelnou hranici.

Kvalitativní posudek rizika a příležitostí řadí rizika a příležitosti projektu podle jejich důležitosti z hlediska jejich dopadu na projekt a podle pravděpodobnosti jejich výskytu. Toto zařízení slouží k rozhodování o tom, jaká strategie bude pro zvládnutí každého jednotlivého rizika či příležitosti nejvhodnější. Kvalitativní posudek rizik a příležitostí poskytuje číselné hodnoty, které měří očekávaný dopad těchto rizik a příležitostí.

Řízení rizik projektu vychází z rizikového inženýrství. Moderní projektové řízení chápe pod zavedeným pojmem riziko jak negativní události, tak události pozitivní.

Riziko má svoji hodnotu, která se vypočte jako součin pravděpodobnosti, že riziko nastane a hodnoty předpokládané škody, kterou riziko způsobí. (Doležal a kol., 2009, s. 72-75)

3.1.12 Plánování

Cílem procesní skupiny plánování projektu je definice hlavních faktorů a sestavení plánových dokumentů projektu. Ve své podstatě se jedná o zpřesnění výstupů iniciační fáze – strukturování hlavního cíle do dílčích položek a konkrétních výstupů, rozdělení pracovních témat a produktů do dílčích částí, definice vnitřních rozhraní subsystémů, příprava metodik a inventarizace znalostí a dovedností budoucích členů projektového týmu, identifikace potřebných zdrojů a definice rizik a předpokladů omezení jejich dopadů. Činnosti procesu vrcholí v sestavení realistického časového rámce a rozpočtu projektu a přípravě detailních plánů na realizaci projektu. (Doležal a kol., 2009, s. 109)

Z procesu plánování jsou výstupem dva podrobné a závazné dokumenty:

- Definice předmětu projektu poskytuje všechny definice potřebné k popisu předmětu projektu a je základem komunikace mezi projektovým týmem a

zákazníkem projektu i pro věcnou komunikaci uvnitř projektového týmu.

- Plán projektu je dokument, který provází projekt v celém jeho životním cyklu. Je to souhrn toho, co musí být v průběhu projektu vykonáno, aby byl splněn cíl projektu a vytvořen předmět projektu tak, jak je popsáno v dokumentu Definice předmětu projektu. (Skalický, 2008)

3.1.13 **Komunikace**

Zavedení vhodné komunikace v projektu je jedním z kritických faktorů úspěšnosti projektu. Pro úspěch projektu je třeba zákazníka a akcionáře informovat o stavu projektu a řídit jejich očekávání. Jestliže tito lidé nejsou včas a dobře informováni o stavu projektu, je daleko větší pravděpodobnost vzniku problémů a těžkostí, které jsou důsledkem nepodložených očekávání těchto osob. Mnoho problémů vzniká spíše tím, že tyto osoby jsou překvapené průběhem projektu, než kvůli vlastním problémům.

Na všech projektech projektový manažer musí informovat o jeho stavu. To zahrnuje předávání informací od pracovníků projektového týmu projektovému manažeru a informování zákazníka, investora a akcionářů od projektového manažera. (Skalický, 2008)

3.1.14 **Vlastní řízení v průběhu projektu a koordinace**

Vlastní řízení v průběhu projektu a koordinace je souhrnem všech aktivit, které jsou zaměřeny na výkon, časování a sladění interakcí plánovaných prací v projektu a jejich integraci do podoby předepsané v definici předmětu projektu. Do aktivit této procesní skupiny je zahrnuta rovněž realizace veškerých změn, které jsou v průběhu projektu navrženy a schváleny k zapracování. (Svozilová, 2007, s. 175)

3.1.15 **Řízení kontroly a monitorování projektu**

Řízení kontroly projektu lze definovat jako proces, který zaručuje, že v rámci projektu budou realizovány přesně a pouze ty cíle, které jsou definovány v předmětu projektu.

Nejčastější metodou kontroly projektu je kontrola podle časového rozvrhu projektu (kontrola harmonogramu a kontrola podle rozpočtu projektu, kontrola nákladů).

Řízení projektu je proces neustálého sledování vývoje projektu a při vzniklých změnách a odchylkách znamená nutnost opravné akce, tzn. udělat cokoliv, co odvrátí negativní průběh projektu a přiblíží jeho vývoj zpět k plánu. (Skalický, 2008)

Vzhledem ke složitosti mnoha projektů i jejich výsledných produktů a také vzhledem k jejich důležitosti je třeba v každé etapě či fázi projektu přezkoumat stav jeho řešení. Projekt musí vždy nejprve projít úspěšnou kontrolou (revizí) jedné etapy hlavního projektu či produktu, a teprve poté může pokračovat další etapou. Protože organizace při řešení projektu zpravidla schvaluje také vydání dalšího objemu peněz, měla by po každé etapě následovat manažerská kontrola, která vyhodnotí úspěšný postup prací na projektu, a to jestli je i nadále ve shodě s cíli dané organizace. Tyto kontroly jsou velice důležité, hovoříme o ukončení debaty neboli kontrolním bodu (kontrolním dni), přičemž v něm sledujeme postup prací na projektu a stanovujeme, jestli bude řešení projektu pokračovat, zda se změní nebo jestli bude projekt ukončen. Každý projekt je jen jednou z mnoha součástí systému organizace jako celku. Změny v jiných částech organizace se mohou promítnout i do změn ve stavu projektu a naopak, změna stavu projektu může ovlivnit dění v jiných částech organizace. Jestliže každý projekt rozdělíme do etap, může vrcholové vedení firmy průběžně kontrolovat, jsou-li projekty stále slučitelné s potřebami zbytku společnosti.

Kromě formálních manažerských revizí je ve většině projektů důležité zapojit vrcholové vedení firmy i do jejich průběžného řešení, tedy do celého životního cyklu. Není rozumné čekat s podněty a vstupy od managementu až na konec příslušné etapy životního cyklu projektu či produktu. V řadě projektů provádí management firem pravidelné kontroly, například týdně nebo i denně, a sleduje tak postup jejich prací. Každý chce úspěšně plnit zadané pracovní úkoly, a pokud jsou do řešení projektu skutečně zapojeni členové vedení firmy, mají tak pod kontrolou plnění úkolů či cílů vlastního projektu i organizace jako celku. (Svozilová, 2007, s. 93)

3.1.16 Uzavření projektu

Uzavření projektu je činností, při které jsou ukončeny všechny aktivity projektu, předány a schváleny výstupy projektu, vypořádány a uzavřeny všechny jeho administrativní agendy. V této etapě jsou provedeny rovněž všechny nezbytné inventury a

hodnocení průběhu projektu, které se pak stává součástí know-how všech zúčastněných stran projektu. (Svozilová, 2007, s. 242)

3.2 Rational Unified Process

3.2.1 Úvod

Metodika Rational unified process (RUP) je v současnosti standardem mezi metodikami. Byla původně zařazována mezi rigorózní metodiku zejména z důvodu velké podrobnosti, ale postupně byla doplňována o agilní praktiky a v současné době představuje rámec, ve kterém je možné vytvořit metodiky pro všechny typy projektů od velmi formální metodiky až po velmi lehkou agilní metodiku. (Buchalceková, 2009, s. 121)

Metodika RUP poskytuje ukázkový přístup k zadaným úkolům a odpovědnostem v celé organizaci. Jejím cílem je zajistit vývoj vysoce kvalitního softwaru, který uspokojí potřeby koncových uživatelů v očekávaném čase a rozpočtu.

RUP je procesní produkt. Je vyvíjen a spravován společností Rational Software a zahrnuje sadu nástrojů na vývoj softwaru. Je k dispozici od společnosti IBM na CD-ROM nebo je možné jej získat prostřednictvím internetu.

RUP také funguje jako výrobní systém, který může být přizpůsoben a rozšířen tak, aby vyhověl potřebám kterékoliv organizace. (Kruchten, 2003, s. 45-48)

3.2.2 Historie RUP

Kořeny metodiky RUP sahají do roku 1967, kdy vznikl Ericssonův model, jehož radikální pojetí vycházelo z faktu, že se složité systémy mají modelovat jako množiny vzájemně propojených bloků. Dnes je tento postup znám jako vývoj komponentovaného softwaru.

Dalším závažným krokem bylo zveřejnění specifikace SDL institutem pro mezinárodní standardy. Tento jazyk byl navržen, aby zachycoval chování telekomunikačních systémů. Systémy byly modelovány jako množina komponent, které spolu komunikují prostřednictvím zasílaných signálů. (Jazyk SDL byl prvním objektově orientovaným standardem modelování a používá se dodnes.)

Na Ericssonově modelu byla založena metodika Objectory, která byla považována za systém se svými vlastními zákonitostmi. Toky pracovních činností v procesu tvorby softwarového produktu byly vyjádřeny pomocí množiny diagramů.

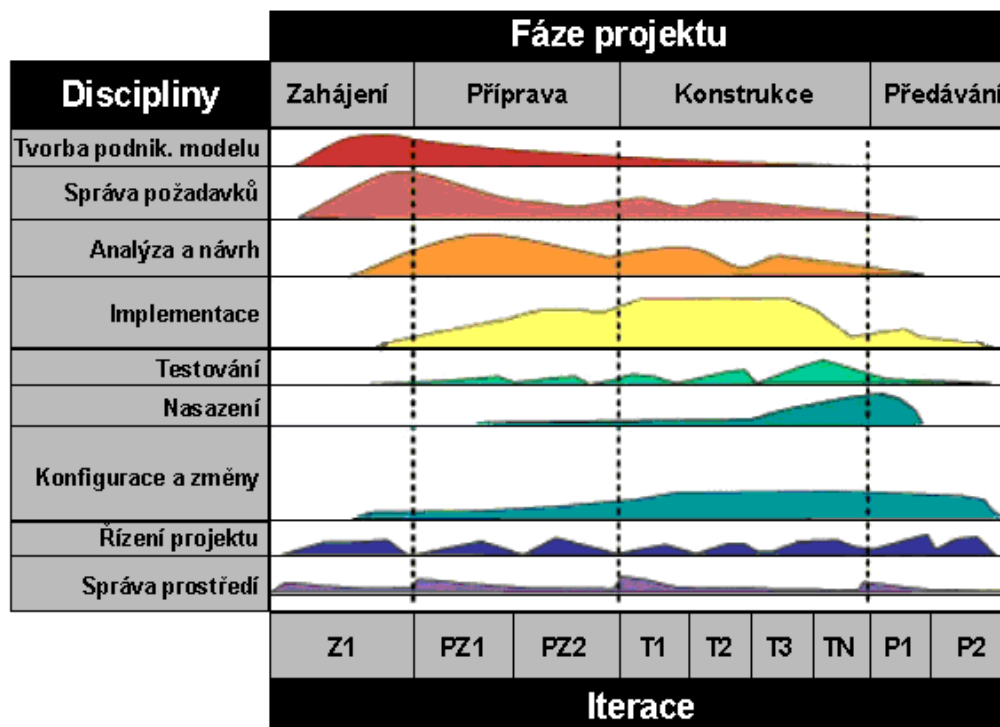
Významné bylo pojetí firmy Rational, která se v roce 1995 začala zabývat prací nad sjednocením metodiky Objectory s dalšími pokročilými metodami tvorby softwaru. Výsledkem byla architektura pohledu 4+1 založená na čtyřech samostatných pohledech a na jednom sjednocujícím pohledu případů užití. Tento model je dodnes základem pojetí RUP i UML (unified modeling language - grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů).

Výsledkem sjednocení prací na postupech Objectory a Rational byla metodika ROP (Rational Objectory Process). Tato metodika nabídla významné zlepšení především v oblastech, v nichž byla metodika Objectory slabá. Jako hnací mechanismus metodiky ROP bylo použito riziko. (Arlow, Neustadt, 2008, s. 53-58)

3.2.3 Vymezení metodiky RUP

Metodika RUP je založena na nejlepších praktikách softwarového vývoje, kterými jsou: iterativní vývoj, řízení požadavků, použití komponentové architektury, vizuální modelování, kontrola kvality softwaru a řízení změn. RUP je softwarovým inženýrským procesem, který představuje disciplinovaný ucelený přístup přiřazující úkoly a odpovědnosti v organizaci zabývající se vývojem softwaru.

Proces vývoje softwaru je popisován v rámci dvou dimenzí, které odpovídají osám na obrázku č. 4. Horizontální osa reprezentuje dynamický pohled na proces, který je zde vyjádřen pomocí cyklů, fází, iterací a milníků. Osa vertikální představuje statické hledisko procesu, což je popis činností, artefaktů, pracovníků a pracovních toků. Na obrázku č. 4 je znázorněno na vertikální ose 9 tzv. disciplín, které reprezentují logické seskupení činnosti. Graf obrázku potom ukazuje podíl jednotlivých disciplín v různých fázích projektu.



Obr. č. 4: Fáze a disciplíny RUP (Zdroj: Buchalcevova, 2009)

Životní cyklus projektu je rozdělen na jednotlivé cykly. Předmětem každého cyklu je nová verze produktu. Jeden vývojový cyklus je v RUP tedy rozdělen do čtyř po sobě jdoucích fází. Jedná se o fázi zahájení, přípravy, konstrukce a předávání.

Ve fázi zahájení by měly být definovány cíle projektu a požadavky, měl by se vytvořit harmonogram projektu (plán iterací) a v neposlední řadě by měly být odhaleny náklady a definována rizika. Součástí této fáze může být vytvoření modelu nebo jednoduchého prototypu, na kterém je možno si ověřit, zda zvolená technologie a zvolené nástroje dovedou klíčové požadavky splnit. Tato fáze končí rozhodnutím, zda je projekt za daných požadavků, dostupných technologií, zdrojů a rozpočtu možné realizovat.

Cílem fáze přípravy je definovat architekturu systému. V této části by měl být vytvořen prototyp, který ověří všechny architektonické principy a umožní tak zpřesnění plánu realizace systému. Měly by zde být definovány komponenty, které je třeba vyvinout pro opakované použití.

Obsahem konstrukční fáze je návrh a realizace systému včetně testování. Bývá prosazován, pokud je to možné, paralelní vývoj.

Cílem fáze nasazení je zajistit, aby uživatelé mohli systém používat. Součástí této fáze je tedy školení uživatelů, předání dokumentace, vytvoření help-desku atd. Každá fáze je uzavřena milníkem, ve kterém se posuzuje, zda byly splněny cíle fáze nebo nikoliv a rozhoduje se o dalším postupu. (Buchalceková, 2009, s. 121-124)

3.2.4 RUP jako produkt

Mnoho společností si uvědomuje důležitost správně definovaného a zdokumentovaného vývoje softwaru k tomu, aby byly jejich softwarové projekty úspěšné. Před několika lety získávali vývojáři znalosti a sdíleli je s ostatními. Toto sběratelské know-how často přineslo metody, knihy, tréninkové programy, které však nebyly aktualizovány z dlouhodobějšího hlediska, čímž se staly zastaralými a nepotřebnými.

V kontrastu s těmito nepotřebnými poznámkami je RUP navržen a vyvinut tak, jako jakýkoli jiný softwarový nástroj. RUP má stejné charakteristiky jako dlouhodobé softwarové produkty:

- IBM umožňuje pravidelné aktualizace.
- Používá se online WEB technologie pro on-line přístup uživatelům.
- Může být přesně nastaven dle specifických potřeb jednotlivé organizace.
- Je propojen s mnoha softwarovými nástroji na vývoj v IBM Rational Suites. Vývojáři tedy mohou mít přístup k procesnímu manuálu pomocí nástrojů, který používají.

Tento přístup zabývání se procesem jako softwarovým produktem má následující výhody:

- Proces není nikdy zastaralý, v pravidelných intervalech dostávají společnosti nová vydání, v nichž získávají vylepšené a aktualizované techniky.
- Všichni členové projektu mají přístup k poslední verzi svého procesního nastavení na intranetu.
- Prohlížeče a vyhledávače umožňují vývojářům okamžitě mít k dispozici manuál nebo mít přístupy včetně nejposlednějších dokumentových šablon, které by měli použít.

- Odkazy poskytují navigaci z jedné části procesu do jiné a následně se rozvětvují na nástroje softwarového vývoje, na externí reference nebo na dokumenty a šablony.
- Pro společnosti jsou zahrnuty lokální, projektové a procesní zlepšení nebo speciální procesy.
- Každý projekt nebo oddělení může řídit svoji vlastní verzi nebo část procesu. (Kruchten, 2003, s. 53-56)

3.2.5 Iterativní vývoj a metodika RUP

Výhody metodikou RUP doporučeného iterativního přístupu:

- Umožní nahlédnout do seznamu změn požadavků. Obvykle se často požadavky mění. Změna požadavků bývá primárně zdrojem problémů s projektem, který způsoboval pozdní vyřešení, chybějící plány, nespokojené zákazníky a frustrované vývojáře.
- V procesu RUP není integrace jeden velký „třesk“ na konci, namísto toho jsou součástí integrovány progresivně. Iterativní přístup je většinou proces neustálé integrace. Celkové řešení se rozpadá například na 6 až 9 menších iterací, které začínají s daleko menšími částmi k integraci.
- Iterativní přístup umožňuje zmírnit rizika dříve, protože iterace je všeobecně jediná doba, kdy jsou rizika objevena a určena. Znamá rizika by měla být minimalizována a nová neočekávaná rizika by měla být lépe odhalitelná.
- Ulehčují opakované použití. Navržené kontroly včasné iterace dovolují tvůrcům identifikovat neočekávaný potenciál znovupoužití a vyvíjet běžný kód pro pozdější iterace.
- Chyby jsou objeveny v dřívějších iteracích tak, jak se produkt posouvá od samotného počátku přes rozpracování až do konečné fáze. Chyby v aplikaci jsou odhalovány v průběhu celého projektu, čímž se snižuje riziko chybovosti při předávání výsledného produktu či služby.
- Vývojáři se mohou učit průběžně, čímž dochází k vyváženému využívání lidských zdrojů v průběhu celého projektu. V neiterativním vývoji by jednotliví členové projektového týmu museli čekat na zahájení fáze spadající do jejich oboru činnosti,

zatímco ve vývoji iterativním je možné zapojit do procesů všechny členy projektového týmu.

- Na konci každé iterace by měla být získána zpětná vazba, která by zkvalitnila vývojový proces. Součástí hodnocení na konci každé iterace je také analýza, jejíž součástí jsou možnosti změn ve společnosti a v procesech pro lepší proveditelnost při další iteraci.
- V metodice RUP je iterativní přístup velice kontrolován, iterace je plánována v číslech, trváních a cílech. Úkoly a odpovědnosti účastníků jsou určeny. Je zachyceno objektivní měření progresu. (Kruchten, 2003, s. 84-102)

3.2.6 Řízení požadavků

Řízení požadavků je systematický přístup ke zjišťování, organizování, komunikaci a řízení požadavků změn na systém nebo aplikaci.

Výhody efektivního řízení požadavků zahrnují:

- Lepší kontrolu nad komplexností projektů

Běžnými příčinami ztráty kontroly nad projektem je nedostatek porozumění mezi účastníky projektu nad zamýšleným chováním systému a plánovanými požadavky.

- Zlepšení softwarové kvality a spokojenosti zákazníka

Základem měření kvality je schopnost systému zpracovávat předpokládané výstupy. Toto může být hodnoceno pouze, pokud všichni účastníci projektu stejně rozumí budovanému a testovanému řešení.

Účastník projektu je jakákoliv osoba nebo zástupce organizace, která má podíl v projektu. Účastník projektu může být koncový uživatel, nákupčí, dodavatel, developer, projektový manažer nebo kdokoli jiný, kdo je dostatečně informován nebo jehož požadavky musí být projektem splněny.

- Omezené náklady na projekt a zpoždění

Zpoždění projektu je minimalizováno kontrolou chyb v požadavcích.

- Zlepšení týmové komunikace

Managementu usnadňuje účast uživatelů v procesu a zajišťuje jim plnění požadavků systému. Správně řízené požadavky vytvářejí běžné porozumění projektových potřeb a povinností mezi účastníky projektu: uživateli, zákazníky, managementem, návrháři a testery. (Kruchten, 2003, s. 52-54)

3.2.7 Využití komponent

Softwarové komponenty mohou být definovány jako netriviální součást softwaru, modul, balení nebo subsystém, který plní svoji funkci a má jasná pravidla a může být integrován do správně navržené architektury/struktury. Toto je fyzická realizace abstrakce v designu návrhu.

RUP podporuje vývoj skládající se z komponentů několika způsoby:

- Iterativní přístup umožňuje vývojářům průběžně identifikovat komponenty a rozhodnout se, který z nich vyvíjet, který znovu použít a který koupit.
- Zaměření se na softwarové struktury umožňuje struktury členit. Struktura vypočítá komponenty a způsoby, jak mají být integrovány zcela stejně jako základní mechanismy a vzorce, který ovlivňují.
- Koncepce jako jsou sady programů, subsystémy a vrstvy používané při analýze a návrhu umožňují organizovat komponenty a specifikovat rozhraní.
- Testování je nejprve organizováno kolem samostatných komponent a poté je postupně rozšířeno k širšímu okruhu integrovaných komponent.

(Kruchten, 2003, s. 36)

3.2.8 Modelace a UML

Podstatná část metodiky RUP se zaměřuje na vývoj a tvorbu modelů.

Modely pomáhají porozumět a formovat jak problém, tak jeho řešení. Model je zjednodušená realita, která nám napomáhá zvládat komplexní systém, který nemůže být ve své podstatě obsáhnut.

Unified Modeling Language (UML) je grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci artefaktů softwarově intenzivních systémů. UML nabízí

standardní způsoby při navrhování projektů, pokrytí koncepčních článků jako jsou byznys procesy a systémové funkce, stejně dobře konkretizuje články, které jsou specifické pro programovací jazyk, databázová schémata a pro znovu používané softwarové komponenty.

UML je běžným jazykem vyjadřujícím různé modely, ale neříká, jakým způsobem vyvíjet software. Na základě těchto okolností Rational vyvinul RUP ruku v ruce s UML, aby byla práce UML smysluplně doplněna.

RUP je manuálem, pro efektivní využití UML pro modelaci. Popisuje, které modely se potřebují, z jakého důvodu se potřebují a jak je sestavit. (Kruchten, 2003, s. 38)

3.2.9 Procesní nastavení

Proces by neměl jen bezmyšlenkovitě kopírovat, generovat zbytečnou práci a produkovat artefakty, které mají zanedbatelnou přidanou hodnotu. Namísto toho by měl být veden jednoduše, při plnění úkolů rychle produkujících předvídatelně kvalitní software.

Organizace by měla doplnit proces vlastními příklady, specifickými pravidly a procedurami.

RUP je procesní rámec, který může organizace modifikovat, přizpůsobovat a rozšiřovat o specifické potřeby, charakteristiky, omezení, o historii organizace, kultury a domény.

RUP nabízí dva procesní základy (první pro malé projekty a druhý pro větší a obecnější projekty). Tyto základy přináší soubor procesních komponent, ze kterých je možné si vybrat a pokrýt tak mnoho přidaných oblastí (real-time, systémový design, systémový engineering a design uživatelského rozhraní), různých technologií (IBM Websphere, Microsoft .NET, J2EE) a programovacích jazyků, nástrojů a technik. (Kruchten, 2003, s. 45)

3.2.10 Nástroje podpory

Efektivní procesy by měly být podpořeny adekvátními nástroji. RUP je podporován širokou paletou nástrojů, které automatizují kroky mnoha aktivit.

Tyto nástroje jsou používány k tomu, aby vytvořily a udržely různé artefakty (modely) procesu softwarového engineeringu, například vizuální modelování, programování a testování.

Do rodiny produktů Rational se řadí tyto nástroje:

- Rational Ada Developer - integrované vývojové prostředí navržené pro aplikace založené na Ada-case
- IBM Rational Application Developer - vývojové prostředí založené na Open Source projektu Eclipse ve verzi 3
- IBM Rational ClearCase - nástroj pro správu verzí, obdoba např. CVS, Subversion
- IBM Rational ClearQuest - produkt pro sledování závad a změn v rámci softwarového vývoje aplikace
- IBM Rational Data Architect - nástroj pro datové modelování a integrační návrh, který usnadňuje návrh všech typů databází, od jednoduchých až po komplexní a složité relační databáze
- IBM Rational Functional Tester - umožňuje automatizovat funkční a regresní testování aplikací napsaných nejen na platformě Java, ale také v prostředí .NET a webových aplikací
- IBM Rational Manual Tester - nástroj pro správu (vytváření) manuálních testů
- IBM Rational Method Composer - flexibilní procesní platforma obsahující procesy a nástroje pro použití během řízení životního cyklu IT (IT Lifecycle Management)
- IBM Rational Performance Tester - produkt pro testování (nejen) zátěže a výkonu webových aplikací
- IBM Rational Portfolio Manager - poskytuje možnost monitorovat a řídit rizika, problémy a zdroje napříč portfoliem stejně jako umožňuje plánovat, řídit a hodnotit každý projekt nebo celé portfolio
- IBM Rational PurifyPlus - run-time řešení pro analýzu softwaru navržené tak, aby pomáhalo vývojářům a programátorům v produkci kvalitních a spolehlivých

zdrojových kódů. Spolehlivost je zajišťována přes dvě základní funkce; detekování chyb operační paměti a tzv. memory leak detekci

- IBM Rational RequisitePro - nástroj pro správu požadavků v průběhu vývoje projektu
- IBM Rational Robot - nástroj automatizující testování aplikací vytvořených v různých vývojových prostředích a jazycích zahrnujících HTML, DHTML, Microsoft Visual Studio .NET, Microsoft Visual Basic, C++, Java, Oracle Developer/2000, PeopleSoft a Sybase PowerBuilder. Představuje jediný nástroj poskytující plnou nativní podporu jazyků postavených na platformě .NET, jako jsou VB.NET, C# a J#
- IBM Rational Rose XDE Developer - umožňuje softwarovým designérům a architektům vytvářet na platformě nezávislé modely softwarové architektury, vytvářet nebo editovat znovupoužitelné šablony a vzory. To vše s využitím notace UML (Unified Modeling Language)
- IBM Rational Software Architect - integrované prostředí pro vývoj a tvorbu aplikací (nejen) orientovaných na služby za pomoci modelového vývoje UML
- IBM Rational Software Modeler - uživatelsky nastavitelná vývojová aplikace, založená na UML modelové architektuře vývoje aplikací
- IBM Rational TestManager
- IBM Rational Unified Process
- IBM Rational Web Developer - nástroj pro tvorbu, testování a umístování webových stránek a služeb podporujících různé programové prostředí. (Vach, 2007)

3.2.11 Využití RUP

Asi milion uživatelů z celého světa pracujících ve více než třech tisících společnostech využívají RUP (údaj z roku 2003). Tito uživatelé používají různé aplikační domény pro rozsáhlé či malé projekty. To představuje všestranné použití a obrovskou přizpůsobivost RUP.

Způsob, jak tyto organizace využívají RUP, se velmi liší. Někteří jej využívají oficiálně, zahrnují své vlastní společnosti do procesu RUP, který dodržují. Jiní jej používají spíše neformálně, jako druh instruktora v softwarovém strojírenství, ti RUP berou jako soubor rad, šablon a návodů, které mohou využívat. (Kruchten, 2003, s. 45)

Praktická část

4 Memos Software

MEMOS Software s.r.o. (dále Memos) je společnost zabývající se vývojem software na zakázku. Hlavním zaměřením jsou webové a desktopové aplikace, vývoj pod MS Outlook, mobilní řešení a služby v oblasti business intelligence - datových skladů. Systémy dodávají obzvláště na technologiích firmy Microsoft.

MEMOS Software nabízí služby především v těchto oblastech:

- Webové aplikace (intranet, web portály)
- Aplikace v rámci MS Office (MS Outlook a MS Exchange)
- Mobilní a SMS software
- Konzultace v IT
- Databázová řešení (datawarehouse/business intelligence)
- Webdesign a internetový marketing (SEO)

Software společnost dodává náročným zákazníkům po celém světě. Pracovníci Memosu mají zkušenosti s klienty nejen z České republiky, ale i z USA, Velké Británie, Dánska a Jižní Afriky. Zákazníky Memos jsou mimo jiné společnosti jako T-Mobile, Scania, Orco Group, Plzeňský Prazdroj, Člověk v tísni, Česká Konsolidační Agentura a White & Case.

4.1 Příklad projektu

Příkladem projektu typickým pro společnost Memos je projekt CallCentrum2, kterým společnost Memos řešila business požadavek a modernizaci pracovního prostředí společnosti LeasePlan Česká republika.

4.1.1 Klient - společnost LeasePlan

LeasePlan ČR zajišťuje operativní leasing a správu pro více než 21.000 vozidel, což neoficiálně představuje téměř 40% podíl na trhu operativního leasingu v ČR. V současné době se o spokojenost přibližně 1000 klientů stará více než 110 zaměstnanců.

LeasePlan Česká republika s.r.o., dceřiná společnost holandské společnosti LeasePlan Corporation N.V., vznikla v roce 1995. V červnu roku 1996 uvedl LeasePlan ČR spolu s prvním dodaným vozidlem na český trh zcela nový leasingový koncept - otevřenou kalkulaci. Celá skupina LeasePlan Corporation N.V. zabezpečuje provoz více než 1,3 milionu vozidel ve 30 zemích a náleží jí tak ve světovém měřítku první místo v oboru. (*Základní informace o společnosti LeasePlan.* [online])

4.1.2 Požadavek klienta

Ve společnosti LeasePlan ČR se doposud pracovalo v systému NOLS (New Operating Leasing System), který byl, i přes slovo „New“, velmi zastaralý. Novým zaměstnancům nějaký čas trvalo, než se v systému NOLS dokázali orientovat. Klient proto požadoval aplikaci pro správu požadavků na 21 000 vozidel s intuitivním ovládáním a přívětivým grafickým rozhraním k uživatelům.

Obchodním impulsem pro uskutečnění tohoto projektu byla možnost propojení správy požadavků s funkcionalitou vyhledávání nejvhodnějších dodavatelů v místě, kde se klient společnosti LeasePlan s vozidlem nachází.

Klient dále žádal o provedení ve dvou formách. První formou byla desktopová aplikace typu Windows Forms pro interní použití společností LeasePlan. Druhou formou byla webová aplikace přístupná i uživatelům mimo společnost LeasePlan, tedy dodavatelům, klientům apod.

Klient požadoval i napojení na stávající systém NOLS. (Pučelík, 2011)

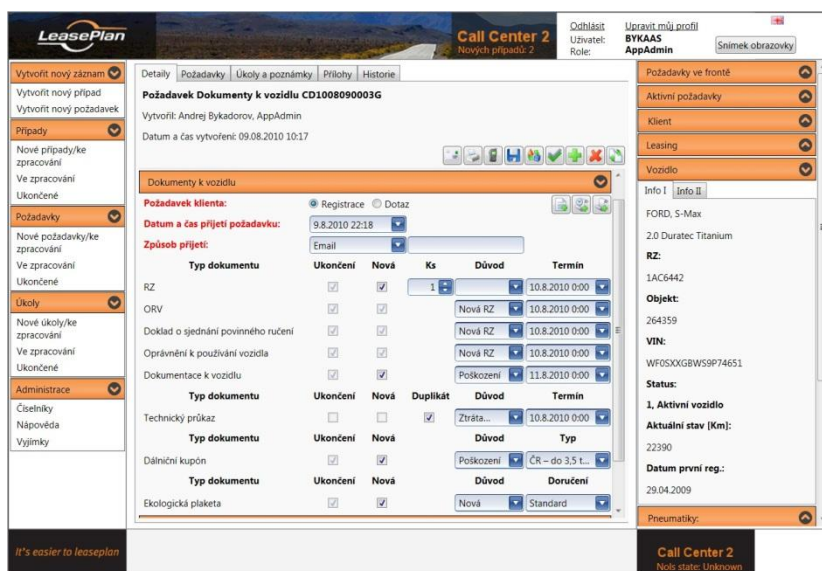
4.1.3 Technický pohled na projekt

Aplikace je vytvořena použitím moderní Microsoft WPF technologie. Robustní a rozšiřitelné jádro aplikace napsané v C# bylo rozděleno do několika vrstev.

Na požadavek zákazníka databázová vrstva poskytuje data, která jsou uložena v XML formátu, protože jsou v mnohých případech synchronizována do jiných interních systémů firmy a tato forma dat ulehčuje komunikaci mezi aplikacemi.

Business logika a prezentační logika samotné aplikace je také uložena v databázi ve formě xml definic. Jádro aplikace zpracovává požadavky a dynamicky generuje uživatelské rozhraní. V těchto xml definicích jsou i akce, které se spouští po úkonech uživatele. Z definic uživatelského rozhraní (s akcemi), které je definováno v xml formátu, může být jádrem generováno WPF standardní rozhraní „tlusté“ aplikace tak, jako běžné Windows aplikace anebo jako webové stránky přístupné uživatelům přes webový prohlížeč. Jako databáze je použit databázový server Microsoft SQL Server. Všechna data, či už samotná prezentovaná data anebo definice uživatelského rozhraní, jsou validována xsd definicemi.

Dále bylo navrženo a implementováno propojení se systémem NOLS přes IBM MQ server. Vybraná data aplikace jsou tedy synchronizována do interního systému přes MQ server. Stejnou cestou jsou v systému NOLS vykonávané akce spouštěné z uživatelského rozhraní aplikace. (Pučelík, 2011)



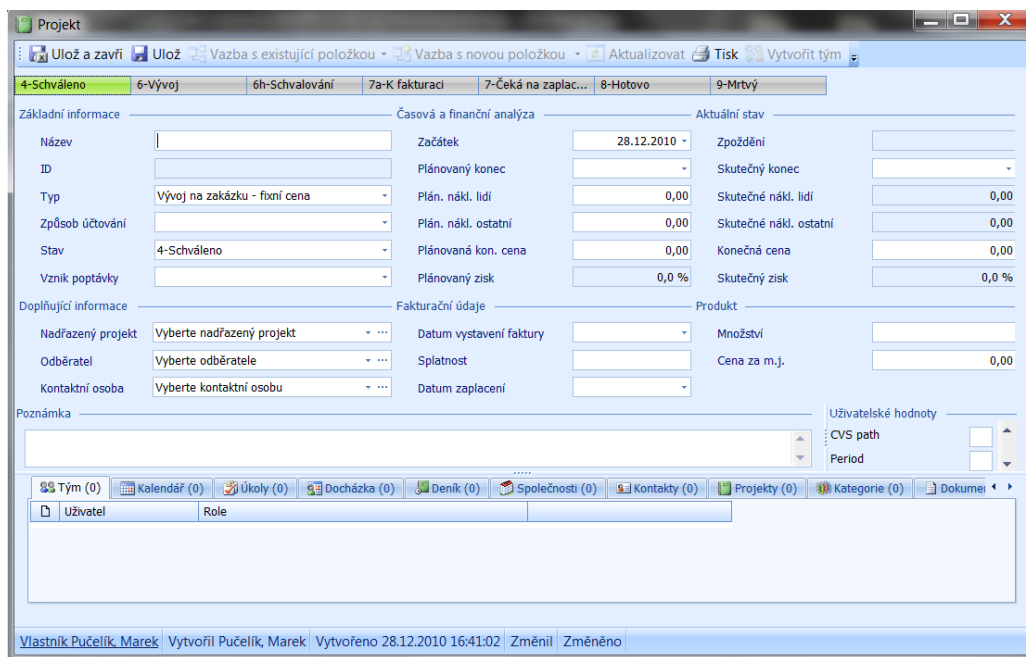
Obr. č. 5: Výsledek aplikace (Zdroj: autor, 2011)

4.2 Interní systémy společnosti

Společnost Memos Software používá ke své činnosti několik interních systémů, které většinou byly vytvořeny vlastním vývojem.

4.2.1 eWay – CRM

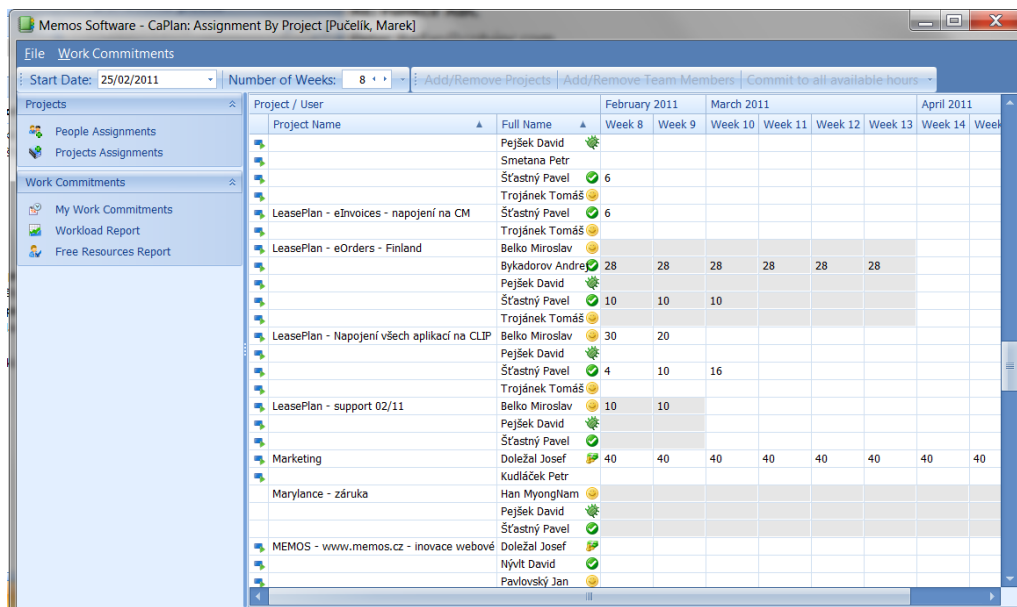
Informační systém eWay-CRM je zasazený přímo do aplikace Microsoft Outlook. Z hlediska projektů se jedná o nástroj, který eviduje veškeré podstatné události v projektu, např. sleduje vývoj projektu od prvního kontaktu se zákazníkem (příležitost) až po splacení poslední faktury zákazníkem a zhodnocení projektu. Dále uchovává veškeré dokumenty, které jsou nutné pro jednotlivé stavy projektu. Eviduje docházku u zaměstnanců přidělených k projektu atd. (viz. Příloha A)



Obr. č. 6: eWay (Zdroj: autor, 2010)

4.2.2 CaPlan

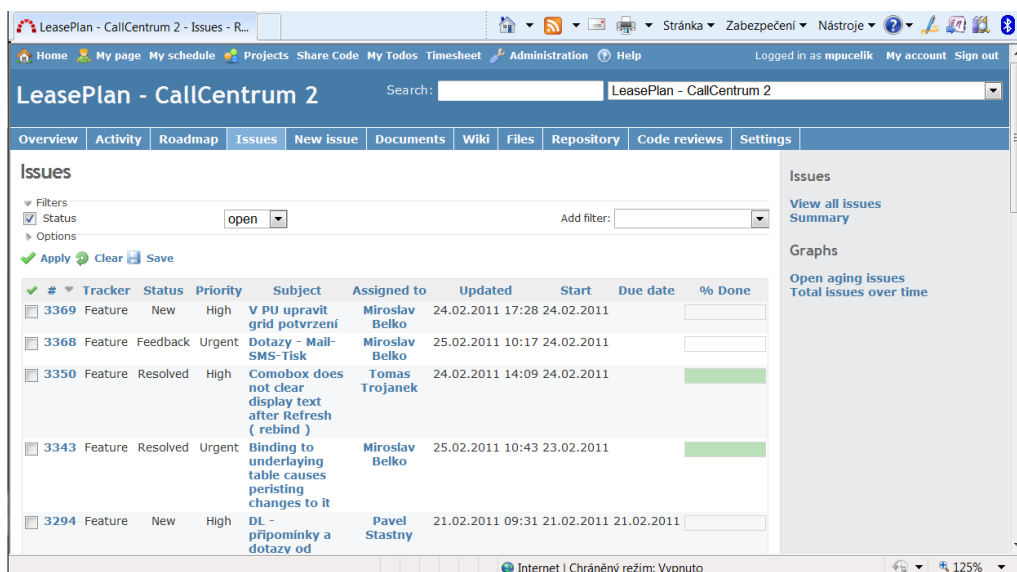
Nástroj CaPlan (Capacity PLANer) je používán pro plánování kapacit na projektech. V případě, že je zapotřebí určitých zaměstnanců na projektu, je umožněno projektovému manažerovi zvolenou kapacitu si v tomto nástroji „zarezervovat“.



Obr. č. 7: Caplan (Zdroj: autor, 2011)

4.2.3 Redmine

Nástroj Redmine je webová aplikace pro flexibilní podporu projektového řízení. Ve společnosti Memos je hlavním prostředkem komunikace projektového týmu. V Redmine se sleduje průběh zadaných požadavků na vývoj nových funkcionalit nebo na opravy chyb. (viz příloha B)



Obr. č. 8: Redmine (Zdroj: autor, 2011)

4.3 Cíle a vymezení problému

Praktická část diplomové práce je zaměřená na výsledky práce společnosti Memos Software. Na tyto výsledky je nahlíženo z více úhlů pohledu a cílem práce je co nejobjektivnější zhodnocení nastavených procesů a jejich využívání.

Práce se nejdříve zabývá získáním a zpracováním kvantitativních dat z informačních systémů společnosti. Dalším zdrojem dat této práce je roční pozorování procesů ve společnosti.

Otázky k zodpovězení jsou nastaveny následovně:

- Jsou pracovní procesy a postupy společnosti Memos Software s.r.o. efektivní?
- Jsou jednotlivé cíle projektů nastavovány dle současných možností společnosti?

4.4 Analýza

4.4.1 Kvantitativní data

Zdrojem kvantitativních dat pro praktickou část je interní systém eWay společnosti Memos. Data získaná ze systému můžeme rozdělit na příležitosti a projekty.

Příležitosti lze vymežit podle eWay jako proces od úvodní komunikace se zákazníkem až po podepsání smlouvy (včetně).

Projekty jsou v systému eWay charakterizovány jako procesy od podepsání smlouvy až po ukončení projektu.

Kvantitativní data pro tuto práci byla získávána za období od 1. 7. 2007 do 1. 1. 2011. Tedy 42 měsíců.

Ke dni 1. 1. 2011 bylo v systému evidováno celkem 3141 záznamů. Z těchto záznamů bylo 913 příležitostí a 2228 projektů. Pokud se stane z příležitosti projekt, ID příležitosti se použije pro ID projektu.

Z 2228 projektů bylo 176 supportních a 16 interních. Pro tuto analýzu tedy bude uvažováno 2036 projektů.

Z reportů provedených nad těmito daty lze vyčíst následující skutečnosti:

- 26% příležitostí je převedeno na projekt.
- Průměrné zpoždění projektu je 55 dnů.
- Medián zpoždění opožděných projektů je 24 dnů.
- Průměrná plánovaná délka projektů je 78 dnů.
- „Marže M1“ je 56%. Marže M1 se získá jako rozdíl mezi cenou a náklady na lidský faktor výroby softwaru.
- 71% projektů má konečnou cenu nižší než 200 000 Kč.

4.4.2 Kvalitativní data

Kvalitativní data byla získána jednoletým pozorováním a participací na práci ve společnosti. Autor měl příležitost sledovat užití procesů v roli sběratele požadavků, IT analytika, IT projektového manažera a QA manažera. Měl tedy možnost získání objektivního pohledu nad procesy ve společnosti, a to z pohledu různých rolí.

V následující podkapitole jsou vypsány výstupy z analýzy. Přiblížení získaných kvalitativních dat je v kapitolách zabývajících se jednotlivými procesy.

4.5 Výstup z analýzy

Z kvantitativních dat je možné vyvodit závěry o největším problému společnosti při vývoji softwaru. Tímto problémem jsou zpožděné projekty.

Z kvalitativního pozorování je možné určit další slabé stránky společnosti (z hlediska řízení projektů):

- Nezohledňování rizik projektů.

V projektech nejsou dokumentována a především zohledňována rizika projektu.

- Chybí poučení z historických projektů.

Ve společnosti Memos není možnost vyhledat dokumenty z historických projektů, které by při přípravě nových projektů mohly být pomocným vodítkem.

- Nepřesně odhadnutá cena a pracnost projektů.

Cena jednotlivých, cenově nižších projektů, je většinou určena zpracovatelem nabídky. Záleží tedy na jediném specifikátorovi, jakou bude mít projekt cenu.

- Přílišná stručnost specifikací.

Některé specifikace jsou příliš stručné a bývá obtížná dohoda se zákazníky ohledně hranic a rozsahu projektu. Tyto problémy vycházejí na povrch až po podepsání objednávek a mohou být tedy frustrující nejen pro zákazníky, ale i pro společnost Memos, a to v případě více práce při nízké ceně.

- Nejednotný způsob tvorby specifikací.

Každou specifikaci nebo nabídku má na starosti vždy jeden zaměstnanec. Každý zaměstnanec má zažitý jiný způsob tvorby specifikace nebo nabídky. Ta samá věc může být podána od různých zaměstnanců jiným způsobem.

- Nedostatečné testování.

V celé společnosti byl zaměstnán pouze jeden pracovník na pozici testera. Tento pracovník byl navíc zaměstnán na poloviční úvazek. Při technické a časové náročnosti jednotlivých projektů nebylo možné výsledky práce řádně otestovat. Většinou k projektu tester vůbec nebyl pozván, protože z hlediska nákladů a možností času bylo pro projektového manažera vhodnější si aplikaci otestovat vlastními silami. V současné době dochází k přijímání nových zaměstnanců na pozici testerů a jejich začleňování do procesů.

Z výstupů analýzy jsou zřejmé některé slabé stránky společnosti. Návrhem pro zlepšení činností procesů by mohla napomoci standardizace veškerých procesů pro vývoj softwaru ve společnosti. K tomuto se mohou použít metodiky. V ČR je nejznámější a nejpoužívanější metodikou Rational Unified Process.

5 Řízení projektů společnosti Memos a využití metodiky RUP

V následujících kapitolách jsou popsány disciplíny metodiky RUP a procesy ve společnosti Memos. Na základě konfrontace rozdílů v přístupech k řešení těchto disciplín bude možné získat náhled nad danou problematikou.

5.1 Předprojektové vyjednávání

Předprojektová fáze má za účel prozkoumat příležitost pro projekt a posoudit proveditelnost daného záměru. Někdy bývá do této fáze zahrnována i vize, základní myšlenka, že by se nějaký projekt mohl realizovat. Obecně by se v této fázi měla získat odpověď na strategické otázky projektu – odkud jít, kam dojít, jakou cestu zvolit a zda má vůbec smysl projekt realizovat. (Doležal, Máchal, Lacko, 2009, s. 160)

Jak je již v literární rešerši několikrát zmíněno, projekt je jedinečný. V softwarovém vývoji na zakázku to platí samozřejmě také. Je velmi obtížné z hlediska obchodu nabídnout potenciálnímu zákazníkovi přesná data o možné funkcionalitě, termínech, kvalitě a především ceně požadovaného projektového produktu. Celá tato fáze je tedy charakteristická nejistotou. Zákazník nemá jistotu, že za své peníze získá požadovaný produkt a dodavatel nemá jistotu, že se zákazník rozhodne pro jeho řešení.

Problémovou oblastí tedy často bývá určení pracnosti a tedy i ceny konečného řešení. Nejlepší metody pro danou situaci jsou určeny jak snahou započítat všechny vlivy do odhadů, tak i potřebou vyhnout se zdrojům chyb. U některých projektů je definována funkčnost a pak je potřeba soustředit se na odhad rozvrhu práce, který je pro výrobu dané funkčnosti potřebný. U jiných projektů je k dispozici rozpočet a časový rámec vývoje, pozornost se tak může soustředit na odhad množství funkčnosti, který lze za těchto podmínek dodat.

Pokud to daný projekt umožňuje, je vhodné používat dostupné výpočty. Pro správný výpočet je nutné najít smysluplný parametr měřítka objemu práce, který je vysoce korelovaný s velikostí softwaru.

Nejméně přesnou metodou odhadů je tzv. znalecký úsudek., a i přes to je individuální úsudek experta zdaleka nejčastějším přístupem k odhadům, který se v praxi používá. Odhady pracnosti bývají přesnější, když mají základ v něčem konkrétním.

Historická data v kombinaci s kalkulacemi jsou oproštěna od předsudků, které mohou podkopat odhady více založené na úsudcích.

Pro převod počtů na odhady se používá kalibrace. Odhady je možné kalibrovat z historických dat, projektových dat nebo z dat z odvětví. Historická a projektová data jsou velice užitečná a mohou pomoci vytvořit velmi přesné odhady. Data z odvětví jsou dočasná náhrada v případě, kdy není možné použít data historická ani projektová.

Základní metodou odhadování je dekompozice. Tato metoda spočívá v tom, že je odhad každé části vytvořen samostatně a následně jsou všechny tyto jednotlivé odhady poskládány zpět do úhrnného odhadu.

Odhadování pomocí analogie je jednoduchá myšlenka založená na předpokladu, že je možné udělat přesné odhady nového projektu jeho porovnáním s podobným starým projektem.

Přístup známý jako fuzzy logika lze použít pro odhadnutí velikosti projektu v řádcích kódu. Odhalovatelé jsou většinou schopni ohodnotit vlastnosti jako velmi malé, malé, střední, velké a velmi velké. V další práci zde pomohou historická data.

Při odhadování v rané fázi projektu nebo pro odhady velkých neznámých jsou užitečné metody skupinových úsudků odborníků. Tato jednoduchá metoda na zvyšování přesnosti odhadů spočívá ve skupinové revizi těchto odhadů.

Provádět několik druhů odhadovacích prací, které ručně příliš pohotově udělat nelze, umožňují softwarové nástroje. Ale skutečnost, že odhad pochází ze softwarového nástroje pro odhady, neznamená, že je zcela přesný. Odhad může být kalibrován špatnými daty nebo může být špatně použit nastavovací prvek atd.

Žádná jednoduchá metoda odhadování není dokonalá, a proto je užitečné používat v mnoha případech více postupů najednou. Nejsofistikovanější komerční produkty používají obvykle alespoň tři různé přístupy k odhadům a pak hledají mezi těmito odhady soulady či odlišnosti. Pokud odhady vzájemně konvergují, jedná se pravděpodobně o odhad dobrý. Ale v případě, že se rozcházejí, patrně bylo něco přehlédnuto a je potřeba se odhadu ještě věnovat. Tato metoda platí stejně pro odhady velikosti, práce i vlastností.

Na špatně odhadovaných projektech se odhadování soustředí na přímé odhady ceny, práce, rozvrhu, s malým či žádným ohledem na velikost vytvářeného softwaru. Odhady projektu jsou často měněny, ale obvykle se jedná o odezvu na skluzu v rozvrhu v pozdní fázi projektu. U dobře odhadovaných projektů se pozornost při odhadování soustředí jinam a milníky změn odhadů jsou jinde.

Dobře definovaný proces pro vytváření odhadů, který je přijat na úrovni celé organizace a poskytuje vodítka jednotlivým projektům, se nazývá standardizovaná procedura. Ta chrání před špatnými praktikami v odhadech, jako jsou například rychlé odhady nebo hádání. Chrání před změnami v odhadech způsobenými rozladěnostmi investora z určitého výsledku. Podporuje důslednost v procesu odhadování a v případě zvláště špatného odhadu umožní projít si zpětně veškeré kroky, aby mohla být procedura postupně zlepšena. (McConnell, 2006, s. 93-199)

5.1.1 Co nabízí metodika RUP

Metodika RUP se specializuje na vývoj softwaru, a tudíž neposkytuje žádné procesní návrhy k řešení této disciplíny ani nezohledňuje obchodní vztahy mezi zúčastněnými stranami.

5.1.2 Přístup společnosti Memos

První kontakt mezi zákazníkem a společností Memos zajišťuje obchodník, který se snaží vyhledávat vhodné obchodní příležitosti. Filtruje poptávky a následně vybírá ty, které jsou pro Memos lákavé, s ohledem na hlavní obchodní zaměření, technologie, kapacitní a jiné možnosti společnosti. V případě, že je nalezena vyhovující poptávka, obchodník ihned zakládá do eWay projekt, který označí stavem „lead“ (příležitost).

Obchodník je odpovědný za vyplnění odběratele a kontaktů v eWay a za vznik poptávky, k čemuž mu vypomáhá asistent.

Obchodník kontroluje a pravidelně konzultuje budoucí volné zdroje v Caplan a při schůzkách projektových manažerů, ve spolupráci s technickým ředitelem. K tomu je zapotřebí ještě další spolupráce s projektovým manažerem, analytikem, architektem, programátory a testery tak, aby měl obchodník stále přehled, co může zákazníkovi nabídnout.

Obchodník vytváří cenovou nabídku, do které shrne sumarizaci hlavních požadavků, hrubý odhad ceny a času. Nabídka se vytváří na základě aktuální cenové politiky. Dále je pak cenová nabídka ukládána do eWay.

Ve společnosti Memos se cenové nabídky odvíjejí od zkušeností obchodníka. Tedy využívá se především metoda odhadu dle znaleckého úsudku, případně se využívá metody skupinového úsudku (viz. 4.1.1). Tyto odhady pracovníci jsou však nejméně přesné.

Nabídnutá cena zahrnuje určitá rizika. Cena projektu by měla být nezávislá na nákladech pro programátory. Není možné vytvořit specifikaci, která by obsahovala veškeré detaily projektu. Představy zákazníka a představy projektového manažera se mohou lišit. Zákazník může předpokládat, že obsah specifikace se dá upravovat výhodněji pro něj, čemuž se dá zabránit přesnou specifikací a podepsanými zápisy z jednání.

Pro malé projekty jsou vytvářeny specifikace již v rámci nabídky. Zákazník si s dodavatelem domluví jednu či více schůzek, kde zákazník přednese své požadavky a dodavatel vytvoří specifikaci, kterou zároveň ohodnotí z hlediska časové náročnosti, požadovaných zdrojů a z toho vyplývající ceny. Tuto specifikaci přiloží k nabídce řešení a zákazník buď nabídku potvrdí svým podpisem, nebo ji zamítne.

U velkých projektů se používá více postupů řešení problému odhadnutelnosti parametrů, času a nákladů projektu. Prvním takovým postupem je outsourcing analytika a jednoho či více vývojářů. Ti mají předem domluvený počet hodin, které stráví na daném projektu a záleží na zákazníkovi, jak tento čas využije. Tento postup je využíván především u supportních smluv.

Druhou možností je určit počet hodin na analýzu a podání nabídky na analýzu, která je smluvně nezávislá na případném následném vývoji. Zákazník si tedy objedná pouze analýzu a podle té se poté rozhodne, zda se daný vývoj uskuteční.

Třetí možností je zahrnutí analýzy do celého projektu. Tato možnost je nejčastější, ale bohužel nejobtížnější z hlediska určení ceny výsledného produktu.

Za závazné alokování zdrojů prostřednictvím aplikace Caplan je zodpovědný technický ředitel. U projektů do 50 tisíc včetně se vytváří objednávka, u projektů nad 50 tisíc se sepisuje smlouva.

Oba dokumenty má na starosti asistent. Po zhotovení je předává obchodníkovi, který má za úkol zajistit podpis mezi oběma stranami. Za společnost Memos musí smlouvu nad padesát tisíc podepsovat dva jednatelé.

Asistent dále ukládá podepsanou verzi smlouvy, resp. objednávky do eWay a do šanonů.

Poté obchodník v eWay vyplní předpokládaný konec projektu, potvrdí konečnou cenu a typ nákladů. Po schválení úvodní specifikace a podpisu smlouvy zákazníkem, změní obchodník status objednávky v eWay na „accepted“. Následně pošle finančnímu řediteli termín dokončení a částku a upraví konečnou cenu v eWay.

Ve chvíli, kdy je smlouva podepsána, projekt přebírá projektový manažer. Implementace zatím neproběhla, projektový manažer tedy vyplní v eWay plánovanou marži. Plánované náklady lidí se exportují z Caplan, začátek a konec projektu se určí ve spolupráci s technickým ředitelem.

Samozřejmostí je, že harmonogram si nese svá rizika. V tomto případě je nejlepší rizika eliminovat tím, že se dělá harmonogram dvojnásobný. (Bykadorov, 2008)

5.1.3 Tvorba podnikového modelu

Pro správné pochopení struktury a dynamiky obchodu zákazníka všemi zúčastněnými je vhodné vytvořit podnikový model, který by měl být předpokladem pro požadavky na systém.

Dle Wiegerse (2008, s. 57) by pro snížení zmatků a nedorozumění vývojáři měli být seznámeni s podnikatelskými aktivitami zákazníka a jeho používanou terminologií. Vývojář by měl znát podnikatelský záměr, který je vývojem nového softwaru sledován.

Sledování uživatelů přináší členům projektového týmu znalosti o vnitřních procesech zákazníka, a tím i kontext potenciálního využití nové aplikace. Často je toto sledování přínosnější než dlouhé přednášky vyššího managementu o tom, jak se co dělá. Management může mít neaktuální informace a může být zaslepen ideálními předpoklady pro činnosti, které v praxi nakonec vypadají zcela jinak.

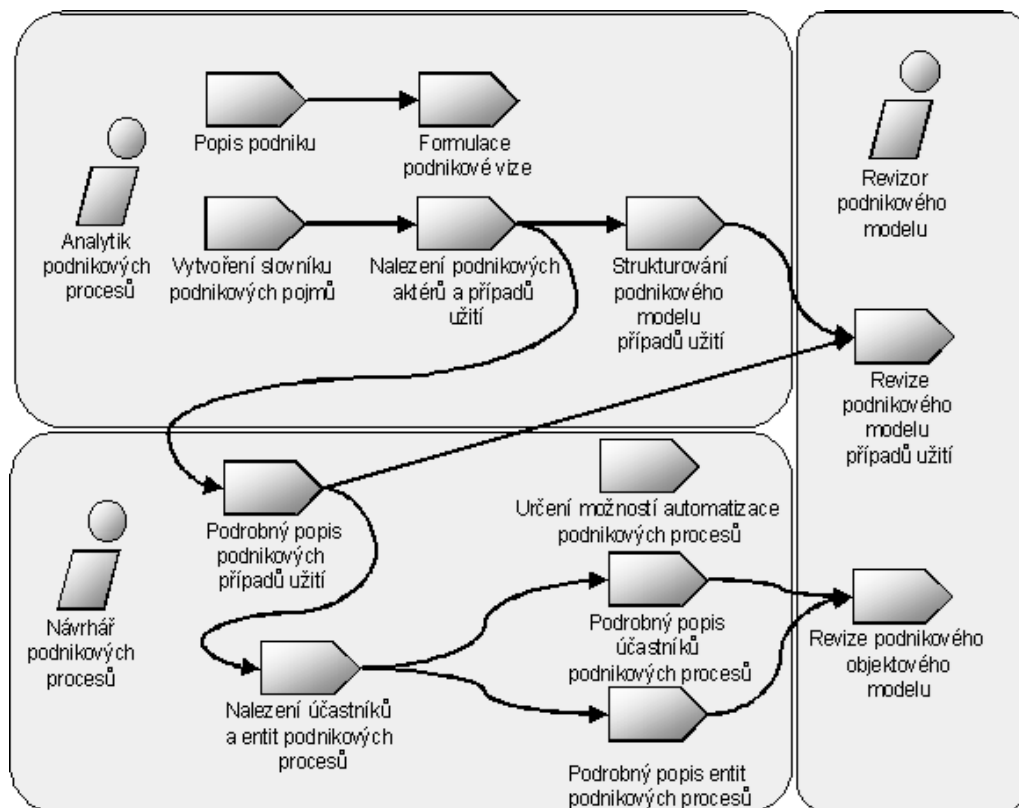
S předcházejícím bodem souvisí sbírání informací o stávajícím systému. V tomto procesu sbírání informací je mnoho skrytých požadavků, které zákazník považuje za

samozřejmé, a které tedy nezohledňuje v nové specifikaci a mohou být příčinou pozdějších sporů o nové funkčnosti aplikace.

5.1.4 Co nabízí metodika RUP

Metodika RUP propojuje modelování podnikových procesů a softwarový návrh na úrovni řízení projektu. Pro lepší komunikaci se využívají významové slovníky.

Obrázek č. 9 poukazuje na specifické činnosti pracovního postupu tvorby podnikového modelu. Detaily pracovního postupu metodiky RUP jsou modelovány jako role (na obrázku vlevo např. systémový analytik) a úlohy či aktivity (ve střední části obrázku, např. tvorba vize). Šipky jsou relacemi, které znázorňují tok prací od jedné úlohy k další. Symboly role, aktivity a šipek jsou použity i v následujících obrázcích.



Obr. č. 9: Disciplína tvorba podnikového modelu v metodice RUP

(Zdroj: Aldorf, 2008)

Pracovní postup tvorby podnikového modelu v sobě zahrnuje činnosti jako je: popis podniku a jeho podnikové vize, vytvoření slovníku a nalezení podnikových rolí, strukturování podnikového modelu a další. Tyto aktivity jsou zakončené revizí modelu.

Pro obchodní modelování mohou být použity techniky softwarového inženýrství. Pro tvorbu podnikového modelu se využívají prostředky UML.

Požadavky na software vyplývají z velké části z modelů podnikání, pokud je systém svázaný s konkrétním podnikovým procesem.

Klíčové artefakty metodiky RUP pro podnikové modelování:

- Dokument obchodní vize - jedná se o dokument, který vymezuje a definuje cíle podnikatelského úsilí
- Obchodní use-case model - jedná se o model podnikání, kde je určena funkce používaná jako základní vstup pro identifikaci rolí a výstupů v organizaci
- Obchodní analytický model - je objektový model, který popisuje realizaci případů obchodního použití

Ostatní artefakty metodiky RUP pro podnikové modelování:

- Hodnocení cílové organizace - popis současného stavu organizace, v níž mají být systémy nasazeny
- Obchodní pravidla - zahrnují podmínky, které musí být splněny
- Doplnující obchodní specifikace - je dokument, který představuje definice podnikání, které nejsou zahrnuty do obchodního use-case modelu.
- Slovníček obchodních pojmů - zahrnuje definice důležitých pojmů používaných v podnikání
- Dokument obchodní architektury - jedná se o ucelený přehled architektonicky významných aspektů podnikání z mnoha úhlů pohledu.

(Kruchten, 2003, s. 170)

Rational Software nabízí nástroje Rational Rose a Rational XDE pro návrh a podporu obchodních modelů. (Kruchten, 2003, s. 180)

5.1.5 Přístup společnosti Memos

Společnost Memos nevyužívá možnosti zpracovat podnikový model zákazníka. Slabina spočívající v nezpracovávání podnikových modelů se projevuje především u velkých a dlouhodobých projektů. Například při změnách v projektovém týmu společnosti se noví členové musejí zapracovávat do projektu. I velice jednoduché a základní podnikové modely by jednoduše zobrazovaly důvody vzniku požadavku zákazníka a tedy i důvody vzniku projektu.

U menších projektů by mohlo záviset na zvážení projektového manažera, zda využít těchto modelů.

Z tohoto pohledu by modelování podnikových procesů zákazníka vedlo ke zvýšení efektivnosti a především ke zlepšení porozumění v rámci týmu, ale i porozumění mezi členy týmu a zákazníkem.

5.2 Sběr a správa požadavků

Wiegiers (2008, s. 58-63) definuje požadavky a náležitosti požadavků z hledisek znalostí, správy požadavků, specifikace požadavků, sběru požadavků a řízení projektu.

Z hlediska znalostí je vhodné poskytnout analytikům dostatečné školení. Člen projektového týmu s rolí analytika by měl mít charakterové vlastnosti jako je trpělivost, schopnost účelné komunikace a schopnost navázat dobré mezilidské vztahy.

Zástupci uživatelů (produktoví šampioni) a vyššího managementu zákazníka by měli být seznámeni s hodnotou jednotlivých požadavků a práce s nimi. S tímto souvisí i informování o rizicích plynoucí ze zanedbání práce s požadavky.

Dalším nástrojem pro snížení nedorozumění je projektový slovník, který pokud je úplný a využívaný, výrazně snižuje možnosti v nedorozuměních.

Podstatné je sledovat stav každého požadavku. Je potřeba vytvořit databázi, ve které bude mít každý samostatný funkční požadavek svůj záznam a ukládat všechny důležité atributy, jako je například jeho stav (navržený, schválený, implementovaný, otestovaný a podobně), což je důležité pro přehled o aktuálním počtu požadavků daného stavu.

Poté je nutné sledovat stabilitu jednotlivých požadavků. Zapisovat si počet směrných dokumentů a počet schválených změn, které se odehrály v průběhu každého týdne. Příliš časté změny poukazují například na nedostatečné pochopení problému, špatně definovaný rozsah projektu, příliš rychlé změny podnikatelských požadavků a další.

Komerční nástroje pro správu požadavků umí vytvořit databázi s mnoha různými typy požadavků. U každého požadavku je možné nadefinovat si atributy, sledovat jeho aktuální stav a uložit odkazy na jiné požadavky nebo softwarové výstupy. Tyto nástroje mohou pomoci automatizovat výše uvedené úkoly.

V neposlední řadě je důležité vytvořit si spojovací matici, což obnáší vytvoření tabulky, ve které bude u každého funkčního požadavku odkaz na kód, jež ho implementuje, a testy, které ověřují správnost této implementace. Tabulka dokáže propojit funkční požadavky s obecnými požadavky, ze kterých vzešly a s dalšími souvisejícími požadavky.

Se správou požadavků úzce souvisí řízení softwarového projektu. Z hlediska řízení je dobré dodržovat následující postup:

Zvolit si vhodný životní cyklus projektu. Vybrat z několika typů životních cyklů ten, který se projektu nejlépe přizpůsobí. Pokud je to možné, je dobré implementovat funkce po skupinách tak, aby se mohly nové verze vydávat průběžně a přinést zákazníkovi nějaký užitek v co nejbližší době.

Upřesňovat termíny a plány svého projektu postupně podle toho, jak se budou vyjasňovat požadavky. První odhady nákladů a termínů založených na nepřesných požadavcích bývají nejisté, ale s upřesňováním představ o požadavcích se budou vylepšovat i tyto odhady.

Kdykoliv jsou do projektu zapracované nové požadavky, je dobré se ubezpečit, zda se dají při aktuálním rozpočtu ještě dodržet slíbené termíny a nesnížit při tom kvalitativní parametry. Pokud to není možné, musí se vyjednat nové podmínky.

V rámci řízení projektových rizik by se měla najít, analyzovat a popsat rizika spojená s jednotlivými požadavky. Poté se pomocí například brainstormingu pokusit najít způsob, jak tato rizika snížit nebo eliminovat.

Sledováním úsilí, které tým věnuje vývoji a správě požadavků, je možné odhadnout, jestli práce probíhají podle plánu, a popřípadě zlepšit rozdělení zdrojů na budoucích projektech. Lépe pak lze posoudit návratnost investic do řízení požadavků.

Na základě projektové retrospektivy neboli posmrtné analýzy projektu je možné poučit se ze zkušeností starších projektů. Na základě studia problémů a postupů, které byly v souvislosti s požadavky na starších projektech použity, mohou analytik a projektový manažer v budoucnosti zvolit o něco jistější kurz.

Z hlediska psaní specifikace je dobré využívat šablon, vybrat si v organizaci šablonu, podle které se softwarové požadavky budou dokumentovat. Šablona poskytuje jednotnou kostru pro popis funkcí a dalších informací, které se požadavků týkají. Není vhodné vymýšlet šablonu novou, lepší je upravit si nějakou z již hotových šablon. Mnoho organizací vychází ze šablony popsané ve standardu IEEE830-1998. Pokud se pracuje na více projektech různých typů nebo velikostí najednou, například na novém velkém projektu, a zároveň na malém vylepšení nějakého ze stávajících projektů, je dobré pro každý projekt vybrat co nejvhodnější šablonu. Šablony i procesy by měly počítat s budoucím růstem.

Pro usnadnění budoucího upřesňování specifikace a jistotu, že všichni účastníci znají důvody pro zahrnutí toho kterého požadavku do specifikace, je dobré najít zdroje jednotlivých požadavků, tedy vysledovat pečlivě každý požadavek až k jeho zdroji. Takovým zdrojem může být případ užití nebo nějaký jiný zákazníkův podnět, obecný systémový požadavek, podnikatelské pravidlo, nebo jiný vnější zdroj. Zdroje požadavků by mělo být možné najít pomocí odkazů nebo na ně může být vyhrazen samotný atribut.

Pro přehlednost je dobré označit každý požadavek jedinečným identifikátorem. Systém označení musí být dostatečně robustní na to, aby vydržel přidávání nových požadavků, mazání a změny. Označení požadavků pomáhá s jejich identifikací a sledováním změn.

Nezávisle na specifikaci požadavků by měla být dokumentována podnikatelská pravidla. Jedná se například o firemní předpisy, vládní nařízení a výpočetní algoritmy. Jejich životnost většinou překračuje životnost projektu. Některá podnikatelská pravidla povedou ke vzniku funkčních požadavků, jež se budou starat o jejich dodržování.

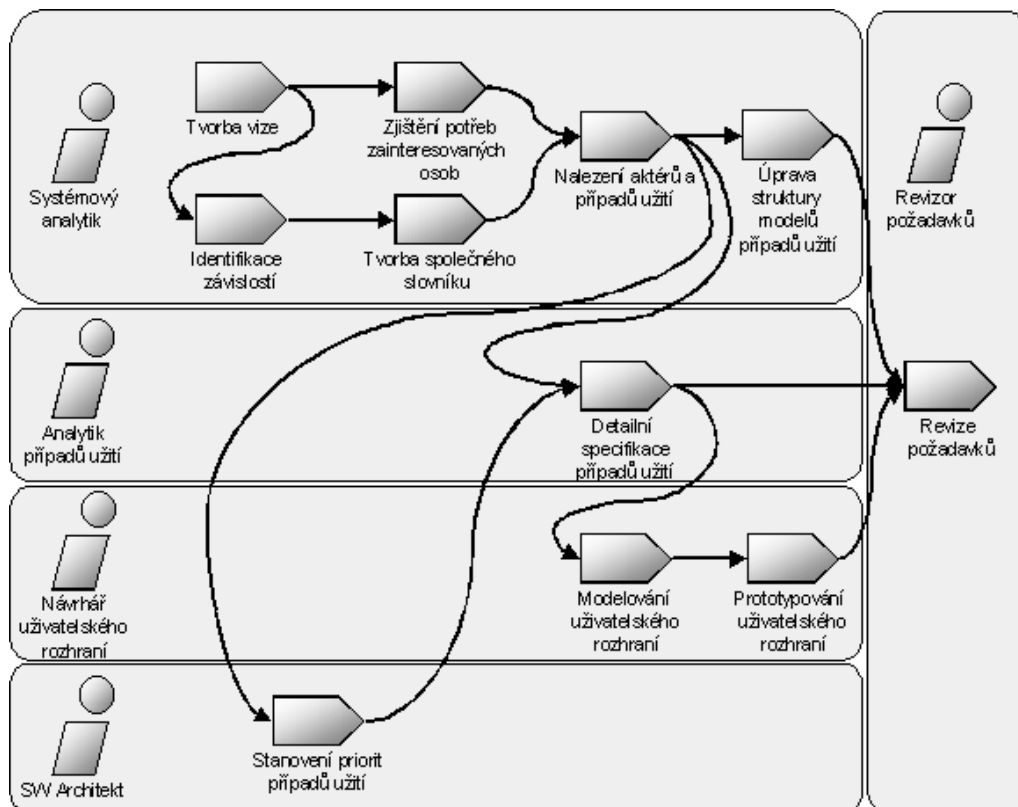
V takovém případě jsou mezi požadavky a odpovídajícími pravidly vytvořené odkazy, aby se snadno dalo podle jednoho najít druhé.

Podstatné je nezůstávat u diskuze nad funkcemi, ale zamyslet se i nad kvalitativními měřítky, které systému pomohou uspokojit zákaznicka očekávání. Mezi kvalitativní parametry patří například výkon, efektivita, spolehlivost, použitelnost a mnoho dalších. Zvolené kvalitativní parametry se vypíší do specifikace. Zákaznickovy názory na důležitost jednotlivých kvantitativních parametrů mohou vývojářům usnadnit rozhodování.

5.2.1 Co nabízí metodika RUP

Požadavky by se daly definovat jako kritéria, jejichž splnění podmiňuje úspěch projektu. Podstatné je požadavky odpovídajícím způsobem dokumentovat, systematicky evidovat a zpracovávat jejich případné změny. Většina práce při definici a specifikaci požadavků probíhá ve fázích zahájení a rozpracování, což znamená v úplném počátku celého projektu. Správa požadavků je proces, který zajišťuje shodu mezi dodavatelem a zadavatelem v případě, že se změní požadavky na systém. (Arlow, Neustadt, 2007, s. 74) Potřeby na řízení správy požadavků zahrnují tedy týmové úsilí o vytvoření a údržbu dohod mezi zúčastněnými stranami a vývojovým týmem o funkcích systému. (Kruchten, 2003, s. 184)

Na obrázku č. 10 je možno vidět specifické úlohy pracovního postupu specifikace a definice požadavků v metodice RUP.



Obr. č. 10: Disciplína sběr a správa požadavků v metodice RUP (Zdroj: Aldorf, 2008)

Pracovní postup správy požadavků v sobě zahrnuje aktivity: vyhledávání aktérů a případů užití, detailní specifikace případů užití, úpravy struktury modelů případů užití, stanovení priorit jednotlivých požadavků, nebo například sledování požadavků až k případům užití.

V typickém projektu, by měly být požadavky rozděleny podle důležitosti. Systém by měl zahrnovat hlavní funkce, podrobné funkční a nefunkční požadavky a případy užití. Požadavek lze definovat jako specifikaci toho, co by mělo být implementováno. Z tohoto hlediska se rozlišují dva typy požadavků.

- Funkční požadavky, jež určují, jaké chování bude systém nabízet
- Nefunkční požadavky, které specifikují vlastnosti nebo omezující podmínky daného systému

Modelování případů užití je jednou z forem inženýrství požadavků. Skládá se z následujících aktivit: nalezení hranic systému, vyhledávání aktérů, nalezení případů užití

(specifikace případů užití, určení alternativních scénářů) a opakování postupu. (Arlow, Nestadt, 2007, s. 82-92)

Klíčovými artefakty pro disciplínu sběr a správa požadavků jsou: dokument s vizí, use-case model a doplňující specifikace. Klíčové potřeby a funkce jsou specifikovány v dokumentu s vizí, ten poskytuje kompletní představu softwarového systému ve vývoji a podporuje komunikaci jak mezi finančním oddělením, tak mezi vyšším managementem. Use-case model by měl sloužit jako komunikační médium, jako smlouva mezi zákazníkem, uživatelem a systémem vývojářů s důrazem na funkčnost systému. Doplňkem k výše uvedeným artefaktům jsou vyvinuty artefakty následující:

- slovník - je důležitý především proto, že definuje společnou terminologii, která je používána konzistentně v celém projektu
- prototyp - slouží jako základ pro uživatelské rozhraní (Kruchten, 2003, s. 194)

V průběhu projektového cyklu by měly být používány nástroje (IBM Rational RequisitePro a IBM Rational Software Modeler), které sledují změny v požadavcích tak, aby byl rozsah těchto změn efektivně udržen. První náčrty uživatelského rozhraní se zaměřují na klíčové potřeby a cíle uživatelů ve vizuálním formování uživatelského rozhraní pro účely vytvoření představy o systému a jeho použitelnosti. Rational nástroje podporují zachycení, vizuální modelování a správu požadavků, jejich atributy.

5.2.2 Přístup společnosti Memos

Ve společnosti Memos není nastaven žádný proces pro uchovávání požadavků. Záleží tedy na každém členovi projektového týmu, jak s jednotlivými požadavky naloží (například, kde a jak si požadavky zákazníka uchovává). Pokud nemá projektový manažer zažitý svůj určitý systém zpracování požadavků, nemá možnost se v určitých případech odvolávat na schválené a dodané požadavky ze strany zákazníka.

Zákazník tedy ani nemá možnost až do přečtení úvodní specifikace vyzorovat, že jeho požadavky byly mylně pochopeny. V těchto případech vznikají společnosti zbytečné náklady na zpracování specifikace, aniž by byla poté schválena zákazníkem.

U malých projektů je specifikace součástí první nabídky, tudíž pro aplikaci metod sběru požadavků je velmi omezený prostor. Sběr požadavků tedy u malých projektů

probíhá formou emailového dotazování nebo po telefonu. Společnost se snaží minimalizovat náklady na zpracování nabídek, protože 74% všech nabídek není realizováno.

U větších (dlouhodobých) projektů jsou využívány schůzky mezi produktovým šampionem na straně zákazníka a analytikem nebo projektovým manažerem na straně společnosti Memos.

Disciplína sběr požadavků je ve společnosti Memos součástí disciplíny analýza a návrh. Výstupem těchto fází je úvodní specifikace.

5.3 Analýza a návrh

Některé požadavky se hodí vyjádřit hned v několika podobách, např. v podobě textové a grafické. Různé pohledy odhalí postřehy a problémy, které by z jednoho úhlu pohledu byly jen těžko viditelné. Větší spektrum pohledů navíc napomáhá účastníkům projektu dospět ke společné představě toho, co nakonec získají.

Z hlediska analýzy požadavků je podstatné nakreslit si kontextový diagram, což je jednoduchý a analytický model, na kterém lze vidět, jakým způsobem zapadá do svého prostředí. Vymezuje hranice mezi vyvíjeným systémem a vnějšími entitami, například uživateli, hardwarovým zařízením a jinými informačními systémy.

Když si vývojáři nebo uživatelé nejsou jistí přesnou podobou požadavků, je dobré udělat si technický prototyp a prototyp uživatelského rozhraní, což je částečná, zkušební nebo předběžná implementace, na které budou myšlenky a možnosti vidět lépe. Vyzkoušením prototypu uživatelé ostatním účastníkům pomáhají vytvořit lepší společnou představu o řešeném problému.

Dále je nezbytné analyzovat proveditelnost požadavků, pečlivě si rozmyslet, zda se každý z navrhovaných požadavků dá za přijatelnou cenu implementovat, a jestli v plánovaném operačním prostředí dosáhne přijatelného výkonu. Součástí je uvědomit si veškerá rizika spojená s implementací všech požadavků – včetně konfliktů s jinými požadavky, závislostí na vnějších okolnostech a technických překážek.

Požadavky by měly být rozděleny podle priorit a nemělo by být opomenuto zajištění poměrné priority implementace všech funkcí, případů užití i jednotlivých

požadavků. Podle priority se určí, ve které verzi se jednotlivé skupiny funkcí nebo požadavku objeví. Při změnách požadavků je naplánována každá změna na některou z budoucích verzí a nemělo by být opomenuto do plánů příslušné verze zapracovat úsilí, které si implementace této změny vyžádá. Rozdělení priorit v průběhu projektu je upravováno podle toho, jak se mění zákaznickovy potřeby, podmínky na trhu a podnikatelské cíle.

Dalším užitečným postupem je vytvořit si model požadavků. Grafický analytický model popisuje požadavky na vysoké úrovni abstrakce, čímž se liší od prototypů uživatelského rozhraní nebo podrobné psané specifikace. Model dokáže odhalit chybné, nejednotné, chybějící nebo nadbytečné požadavky. Mezi modelovací nástroje patří například diagramy datových toků, E-R diagramy, stavové diagramy, mapy dialogů, diagramy tříd, sekvenční diagramy, diagramy interakcí, rozhodovací tabulky a rozhodovací stromy.

Vytvořením datového slovníku je umožněno všem lidem, kteří na projektu pracují, používat jednotné definice dat. Ve fázi analýzy požadavků by měl datový slovník obsahovat i definice dat z aplikační domény, čímž by se usnadnila komunikace mezi zákazníky a vývojářským týmem.

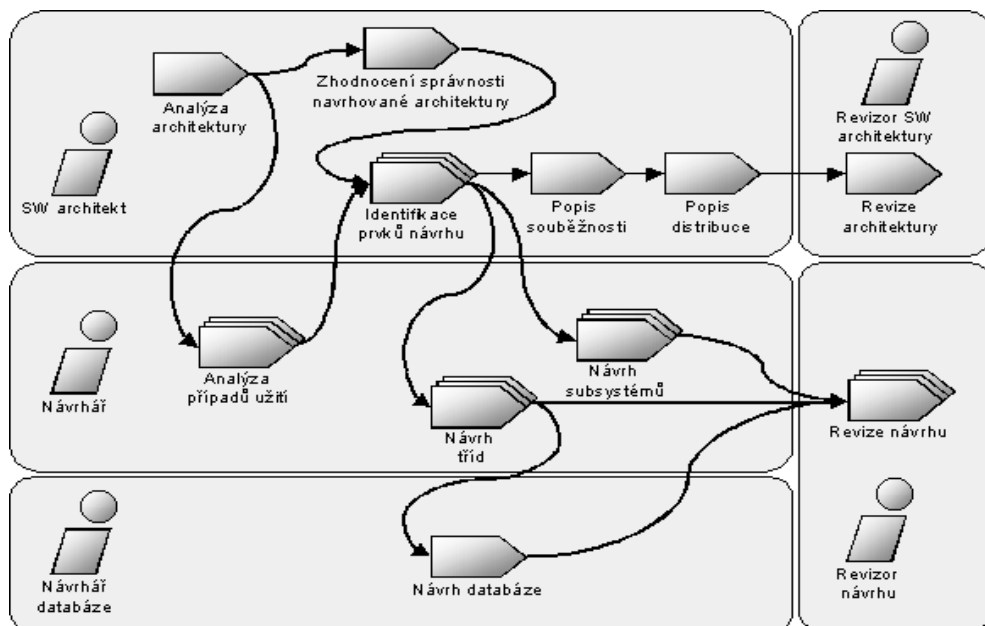
Požadavky na složitý výrobek, který se skládá z několika menších částí, se musí správně rozdělit mezi jednotlivé softwarové, hardwarové a lidské podsystémy. (Nelsen, 1990 in Wiegers, 2008) Toto rozdělení většinou provádí designér systému nebo architekt. (Wiegers, 2008, s. 60)

5.3.1 Co nabízí metodika RUP

Analýza a návrh překlenuje propast mezi požadavky a implementací. Hranice mezi analýzou a návrhem může být velice nejasná. Tento pracovní postup používá případy užití k identifikování množiny objektů, která je rozdělena do tříd podsystémů a balíčků. (Kruchten, 2003, s. 198-206)

Analýza se většinou provádí na konci fáze zahájení. Záměrem analýzy je tvorba analytického modelu. Tento model se zaměřuje na to, co systém musí udělat, avšak nezabývá se detaily, jakým způsobem to vykonat. (Arlow, Neustadt, 2007, s. 137)

Na obrázku č. 11 jsou znázorněny specifické úlohy pracovního postupu analýzy a návrhu.



Obr. č. 11: Disciplína analýza a návrh v metodice RUP (Zdroj: Aldorf, 2008)

Povinnosti v oblasti analýzy a návrhu jsou rozděleny mezi role software architekt, návrhář a návrhář databáze a zahrnují analýzu architektury, analýzu případů užití, návrh databáze aj. Tyto aktivity jsou zakončeny revizí návrhu.

Výsledkem fáze analýzy a návrhu je vytvoření analytického modelu, který je abstrakcí využívající tři architektonické pohledy:

- logický pohled, který představuje dekompozici systému do množiny logických elementů (tříd, podsystémů, balíčků a spolupráce).
- procesní pohled, který mapuje procesy elementů a zobrazuje vlákna systémů.
- vývojářský pohled, který mapuje procesy k určení spouštěcích uzlů. (Kruchten, 2003, s. 214)

Není snadné vytvořit obecný analytický model, protože každý systém je jiný. Pro tvorbu úspěšných analytických modelů existují určitá pravidla, a sice:

- Analytický model by měl být vytvořen v obchodním jazyce.
- Každý diagram by měl objasnit určitou důležitou část požadovaného chování připravovaného systému.

- Soustředit se na zachycení případů užití z určité perspektivy. Příliš detailů je zde na škodu.
- Jasně rozlišovat mezi problémovou doménou (obchodními požadavky) a doménou řešení (podrobné úvahy na téma návrhu).
- Snažit se minimalizovat vzájemné vazby.
- Pokud se zdá, že v modelu existuje přirozená a vynucená hierarchie abstrakcí, je na místě prozkoumávat dědičnost. (Arlow, Neustadt, 2008, s. 138-146)

Analytické třídy umožňují modelovat statickou strukturu systému, zatímco realizace případů užití slouží k popisu spolupráce instancí analytických tříd.

V průběhu realizace případů užití během analýzy by měly být sledovány tyto cíle:

- Snažit se zjistit interakci jednotlivých analytických tříd z důvodu realizace chování specifikované případem užití
- Snažit se zjistit, jaké zprávy musejí instance zmiňovaných tříd posílat instancím jiných tříd, aby bylo možné určené chování realizovat. Tyto skutečnosti odhalují klíčové operace analytických tříd, klíčové atributy těchto analytických tříd a důležité reakce mezi analytickými třídami.
- Snažit se aktualizovat model případu užití, model požadavků a také analytické třídy na základě informací získaných během realizace případu užití. Všechny modely by měly být udržovány konzistentně a v souladu s ostatními. (Arlow, Neustadt, 2008, s. 148)

Během realizace případu užití v procesu analýzy je nezbytné soustředit se na zachycení klíčových atributů, operací a relací mezi analytickými třídami. Kromě toho nemusí být vytvářena realizace všech případů užití. Lze vybrat jen klíčové případy a pracovat pouze s nimi. Metodika RUP je metodikou iterativního postupu, je tedy možné se k realizacím v případě potřeby vracet.

Během postupného upřesňování analýzy se modelování stále více zaměřuje na návrh. Aktivity analýzy a návrhu mohou do určité míry probíhat paralelně. Je však velmi důležité rozlišovat mezi artefakty obou aktivit.

Metodika RUP doporučuje místo dvou týmů analytiků a návrhářů jen jeden tým odpovědný za tvorbu artefaktu - zpracováním požadavků počínaje analýzou, návrhem a implementací konče. Metodika RUP místo sestavování týmů pro určené aktivity sestavuje tým pro tvorbu výsledků a milníků. Metodika RUP se zaměřuje více na cíl než na úlohu.

Analýza se zabývá především tvorbou logického modelu připravovaného systému, který zachycuje funkce, jež tento systém musí poskytovat, aby uspokojil požadavky uživatelů. Smyslem návrhu je přesná specifikace způsobu, jak takovéto funkce implementovat.

Během návrhu rozhodují návrháři objektivě orientovaného systému o strategických otázkách, např. o prezentaci objektů, o distribuci objektů a o tvorbě příslušného navrhovaného modelu. Vedoucí projektu a inženýr projektu by měli vytvořit zásady, podle nichž se lze vypořádat s taktickými otázkami týkajícími se návrhu.

Jedním z cílů metodiky RUP je konstatování, že určité osoby přebírají vlastnictví určité části systému a odpovědnost za ni po celou dobu analýzou počínaje a implementací konče. Osoby (nebo tým) odpovědné za tvorbu konkrétní části objektivě orientované analýzy svou práci často zdokonalí až do podoby návrhu nebo dokonce vytvoří zdrojový kód. Výhodou tohoto přístupu je skutečnost, že brání komunikačním potížím mezi analytiky, návrháři a programátory, což může být v objektivě orientovaných projektech poměrně běžné.

V metodice RUP je architektonický návrh aktivitou, která startuje celý proces návrhu. Provádí ji jeden nebo více architektů.

Architektonický návrh spočívá ve vytvoření náčrtu architektonicky významných artefaktů. Ty jsou pak vstupní branou k aktivitám podrobnějšího návrhu. Architektonický návrh obvykle není samostatným krokem. Protože metodika RUP je procesem iterativním, tento návrh probíhá v pozdní fázi, kdy se vyvíjí podrobnosti systémové architektury.

Aktivita navrhnutí podsystému spočívá v rozbití systému na maximálně autonomní a na sobě nezávislé součásti, tyto součásti nazýváme podsystémy. Interakce mezi nimi zprostředkovávají rozhraní. Cílem návrhu podsystémů je minimalizace vazeb v systému. Vznikají vhodná rozhraní, která zajišťují, že všechny podsystémy správně realizují chování předepsané pomocí rozhraní.

Aktivita navrhnutí případu užití spočívá ve vyhledávání návrhových tříd, rozhraní a komponent, jež se díky vzájemné komunikaci projevují chováním specifikovaným pomocí případu užití. (Arlow, Neustadt, 2007, s. 136-206)

Primárním artefaktem analýzy a návrhu je designový model. Skládá se ze sady spolupracujících prvků, které poskytují systému chování. Toto chování je odvozeno od use-case modelu a od nefunkčních požadavků.

Designový model se skládá ze spolupracujících tříd, které mohou být sloučeny do balíčků a subsystémů, jež mají pomoci organizovat model a poskytnout kompoziční stavební bloky v rámci modelu. (Kruchten, 2003, s. 205)

Z hlediska metodiky RUP jsou vhodnými nástroji pro analýzu a návrh:

- IBM Rational Software Architect
- IBM Rational Rose XDE Developer
- IBM Rational Web Developer
- IBM Rational Application Developer
- IBM Rational Data Architect

5.3.2 Přístup společnosti Memos

Dle interního informačního systému společnosti Memos (www.wiki.memos.cz) analýza začíná po odsouhlasení projektu a podepsání kontraktu. Jejím cílem je především poskytnout detailní a jednoznačné podklady pro vývoj.

Výstupem analýzy a návrhu je detailní specifikace. Ta by měla obsahovat důvody vzniku projektu, požadavky zákazníka a ucelený popis aplikace. Pro tvorbu specifikace jsou používány nástroje (např. Axure). Specifikace je jediným prostředkem pro vymezení obsahu projektu.

Vytváří se technická detailní specifikace, na které se podílí projektový manažer, architekt, analytik a supervizor. Při tomto typu spolupráce určuje architekt, co se bude brát z knihovny a co se bude muset naprogramovat. Všechny zainteresované osoby, architekt,

supervizor a tester jsou seznámeny se základním harmonogramem a podílejí se na vývoji detailního harmonogramu.

5.4 Implementace

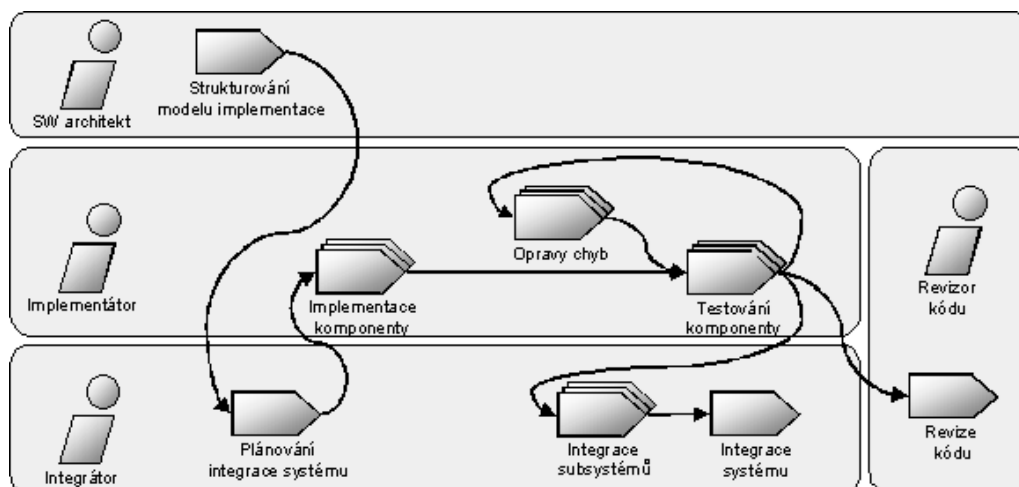
Implementace je proces uskutečňování projektu za účelem dalšího použití.

5.4.1 Co nabízí metodika RUP

Implementace spočívá v převodu návrhového modelu do spustitelného kódu. Implementace je hlavním pracovním postupem konstrukční fáze. Pracovní postup implementace je zaměřen hlavně na tvorbu spustitelného kódu. Vedlejším produktem této aktivity může být implementační model, přestože tento model není výsledkem explicitní modelovací aktivity. (Arlow, Neustadt, 2007, s. 461)

Hlavním rysem RUP je přístup k přírůstkové integraci v celém životním cyklu projektu. Během konstrukční fáze je vystavěn strukturní prototyp, který se stává konečným systémem v této fázi. (Kruchten, 2003, s. 218)

Z Obrázku č. 12 vyplývá, že pracovní postup implementace zahrnuje architekta, systémového integrátora a implementátora, kteří pracují na těchto činnostech: strukturování modelu implementace, plánování integrace systému, implementace a testování komponent a integrace systému. Tyto aktivity jsou zakončeny revizí kódu.



Obr. č. 12: Disciplína implementace v metodice RUP (Zdroj: Aldorf, 2008)

Klíčovým artefaktem implementace je implementační model. Tento model se skládá z diagramu komponent, jenž modeluje závislost mezi softwarovými komponentami, které utvářejí systém, a z diagramu nasazení, který modeluje fyzické výpočtové uzly, v nichž je poté software nasazen, a relace mezi nimi.(Arlow, Neustadt, 2007, s. 464) Současně je možné vytvořit několik prototypů chování, které nabízí řešení, např. v uživatelském rozhraní.

Klíčové artefakty implementace jsou následující:

- Implementace subsystému - sbírka implementovaných prvků a dalších subsystémů, která obvykle vzniká rozdělením modelu na menší části.
- Implementace prvků - jedná se o část softwarového kódu (zdroj, binární, nebo spustitelný), nebo soubor, který obsahuje informace nebo souhrn dalších prvků, například aplikace skládající se z několika spustitelných zdrojů)
- Začlenění vytvořeného plánu - tento dokument definuje pořadí, ve kterém jsou prvky a subsystémy prováděny. (Kruchten, 2003, s. 222)

Technika, která spojuje dohromady design a realizaci úsilí o vývoj softwaru je podporovaná nástroji, jako je Rose nebo XDE. (Kruchten, 2003, s. 229)

5.4.2 Přístup společnosti Memos

Implementace ve společnosti Memos začíná potvrzením specifikace zákazníkem. Projektový manažer ve fázi vývoje mění status projektu v eWay na „development“. Programátor ve spolupráci s architektem pracují na správném vývoji díla. Architekt kontroluje kvalitu kódu na základě předepsaných standardů (tzv. coding standards), kvalitu databáze (primární klíče, vazby, komentáře) a je odpovědný za celkovou kvalitu kódu. Součástí jsou pravidelné schůzky mezi projektovým manažerem, architektem, supervizorem, programátory a testery.

Dále by měla probíhat pravidelná testovací kolečka (u velkých projektů týdně). Projektový manažer ve spolupráci se supervizorem a architektem vytvářejí testovací scénáře podle detailního testovacího plánu z technické specifikace. Tyto scénáře eviduje projektový manažer. Tester kontroluje bezchybnost aplikace a je za její hladký průběh zodpovědný. Současně s tím supervizor s programátory a architektem vytvářejí

projektovou dokumentaci. Projektový manažer ve spolupráci s administrátorem zakládá projekt v Cruise Control (nástroj pro automatickou tvorbu build, tedy spustitelného programu). Dále nastavuje analýzu kódu pomocí nástroje CSharp analyzáru.

Projektový manažer kontroluje průběh projektu a jeho stav ve srovnání s harmonogramem. Tedy, zda je v pořádku docházka všech zaměstnanců, jakým způsobem se vyvíjí rozpočet projektu ve srovnání s plánem, jaký je stav Redmine, a je-li vše správně zálohováno (záloha kódu v SVN, záloha SQL skriptů, záloha dokumentů v eWay).

V případě, že je ve smlouvě ukotveno, že platby budou probíhat průběžně, projektový manažer nechává podepsat předávací protokol s poměrnou částí z celkové ceny. Tento předávací protokol ukládá do eWay a následně předává asistentovi, který protokol ukládá do šanonu.

V případě, že zákazník požaduje změnu ve specifikaci, projektový manažer konzultuje náročnost této změny s architektem, technickým ředitelem a supervizorem a v případě potřeby mění cenu a termín dokončení. Změny ceny projektový manažer ukládá do eWay a informuje o tom finančního ředitele.

5.5 Testování a kontrola

Vzhledem k rozsahu a složitosti dnešního softwaru je jeho profesionální a efektivní testování naprosto nezbytné, a to především z důvodu příliš vysokých rizik výskytů softwarových chyb.

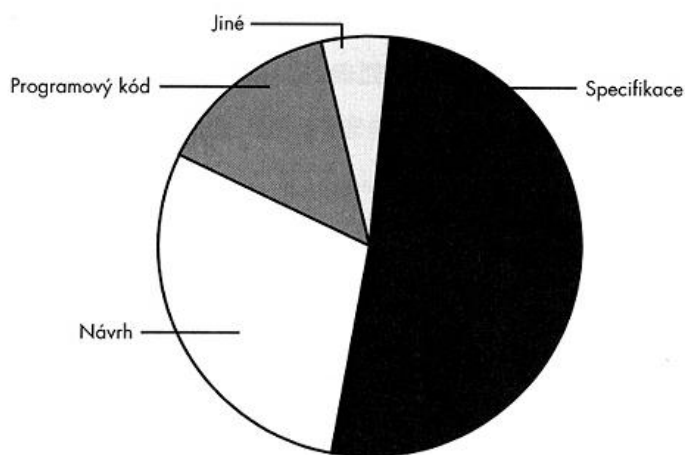
Pro účely většiny softwarového průmyslu se hovoří o softwarové chybě tehdy, pokud je splněna jedna nebo více z následujících podmínek:

- software nedělá něco, co by podle specifikace produktů dělat měl
- software dělá něco, co by podle údajů specifikace produktu dělat neměl
- software dělá něco, o čem se produktová specifikace nezmiňuje
- software nedělá něco, o čem se produktová specifikace nezmiňuje, ale měla by se zmiňovat
- software je obtížně srozumitelný, těžko se s ním pracuje, je pomalý, nebo jej koncový uživatel nepovažuje za správný

Příčinou většiny chyb jsou chyby ve specifikaci. V řadě případů specifikaci jednoduše vůbec nikdo nenapíše. Další příčinou může být specifikace, která není dostatečně podrobná, nebo se neustále mění, nebo ji dostatečně neprojednali všichni členové vývojového týmu.

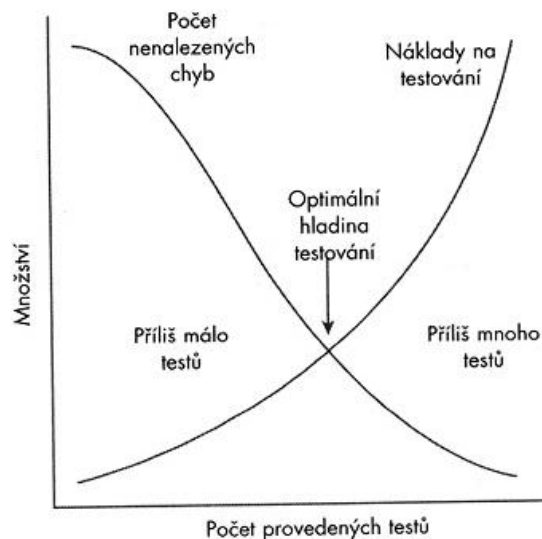
Druhým největším zdrojem chyb je návrh. V návrhu dochází k chybám ze stejných důvodů, proč vznikají i ve specifikaci: návrh je uspěchaný, často se mění, nebo není dostatečně prodiskutován.

Jako příčina chyb v programovém kódu se dá zpravidla vysledovat složitost softwaru, nedostatečná dokumentace, časová tíseň nebo chyby z nepozornosti.



Obr. č. 13: Příčiny chyb v softwaru (Zdroj: Patton, 2002)

Softwarové testování může prokázat existenci chyb, ale v žádném případě nemůže zaručit, že chyby neexistují. Může být provedeno jakékoliv množství testů, nalézání a oznamování chyb, ale v žádném okamžiku nemůže být zaručeno, že již žádné další chyby nejsou. Každý softwarový projekt má svoji efektivní hladinu testování (viz následující obrázek). (Patton, 2002, s. 10-36)



Obr. č. 14: Optimální hladina testování (Zdroj: Patton, 2002)

Do procesu testování je vhodné zahrnout i testování požadavků. Testování požadavků má na starosti správnost formulace požadavků tak, aby splňovaly požadované kvalitativní parametry i nároky zákazníků. Některé požadavky působí při prvních čteních specifikace správně, ale jakmile se vývojáři pustí do jejich implementace, začnou se v nich objevovat chyby. Nejednoznačnost a neurčitost požadavků je často odhalena při psaní testovacích scénářů. Pokud má požadavková dokumentace fungovat jako spolehlivý základ pro návrh systému a jeho pozdější testování, je nutné všechny problémy opravit.

Z tohoto důvodu je vhodné provádět revizi požadavkové dokumentace. Formální revize dokumentovaných požadavků je jeden z vůbec nejcennějších zvyků pro řízení kvality softwaru. Malý tým lidí s různými specializacemi (např. zákazník, vývojář, analytik a tester) může efektivněji nacházet chyby ve specifikaci, analytických modelech a všech souvisejících informacích. Pomáhají i neformální předběžné revize během vývoje požadavků. Revize požadavkové dokumentace nepatří mezi postupy, které by se snadno implementovaly, poskytují ale velkou přidanou hodnotu.

Je dobré testovat požadavky. Testovací scénáře pro funkce systému lze napsat podle uživatelských požadavků; měly by popisovat očekávané chování systému za daných podmínek. Hotové scénáře je třeba konzultovat se zákazníkem pro nabytí jistoty, že doopravdy popisují požadované chování systému. Pomocí testovacích scénářů se ověřuje správnost analytických modelů a prototypů. (Wiegiers 2008, s. 65).

Stanovení kritérií pro přijetí systému uživateli by mělo být nedílnou součástí specifikací pro snížení rizika a naplnění uživatelských potřeb.

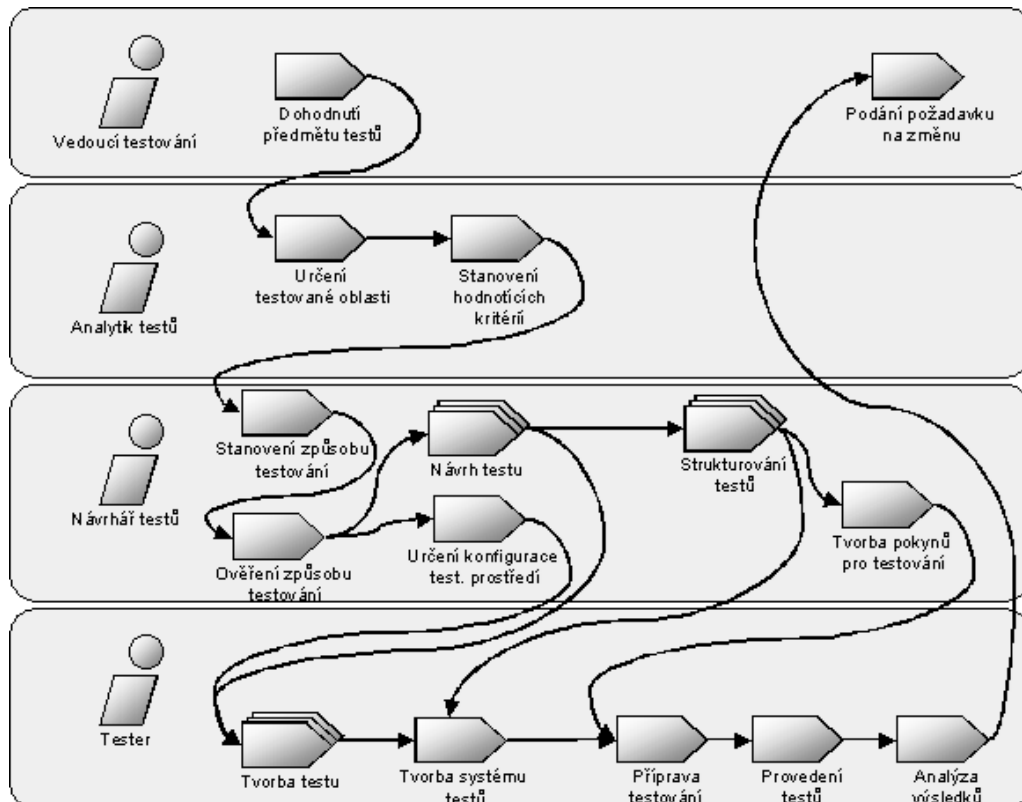
Požadavky na testování a kontrolu:

- Revize dokumentace požadavků
- Testování požadavků

S testováním souvisí i kvalita. Kvalitou je souhrn charakteristik, které se týkají schopnosti vyhovět daným a předpokládaným potřebám. Řízení kvality zahrnuje veškeré činnosti v rámci všech funkcí řízení, které definují strategii kvality, cíle a odpovědnosti, a uvádějí je do praxe pomocí plánování, kontroly, zabezpečení a zlepšení kvality v rámci systému řízení kvality. (Doležal, Máchal, Lacko, 2009, s. 106)

5.5.1 Co nabízí metodika RUP

Z obrázku č. 15 jsou patrné specifické úlohy pracovního postupu testování.



Obr. č. 15: Disciplína testování a kontrola v metodice RUP

(Zdroj: Aldorf, 2008)

Pracovní postup testování zahrnuje tyto aktivity: určení testované oblasti, stanovení kritérií, stanovení způsobu testování, návrh testování, provedení testů aj.

Testování umožňuje hodnocení kvality vznikajících produktů. Testování v iteračním procesu se vyskytuje ve všech fázích životního cyklu projektu, což umožňuje včasnou zpětnou vazbu na kvalitu výrobků a slouží ke zkvalitnění navrženého a vyvíjeného produktu.

Testování je velmi podstatným mechanismem zpětné vazby. Každý zúčastněný nese odpovědnost za kvalitu. V iterativním procesu testování poskytuje zpětnou vazbu v čase měření kvality vývoje, umožňuje tak zlepšit kvalitu systému.

Testování pracovního postupu poskytuje projektu mechanismus zpětné vazby umožňující měření kvality, identifikaci chyb a nalezení řešení dříve, než se stane neopravitelným. Testovací aktivity začínají v počáteční fázi projektu testovacím plánováním a hodnocením vyskytujících se událostí a pokračují v projektu až do konečné fáze. (Kruchten, 2003, s. 315)

Klíčové artefakty testování jsou:

- Plán testování obsahuje informace o účelu a cílech testování daného projektu. Určuje strategii, která má být použita, vymezuje zdroje nezbytné k provádění testování a definuje konfiguraci, které je v testování zapotřebí.
- Případy testování stanovují podmínky o provedení testování.
- Testovací scénáře jsou manuální nebo automatizované postupy, které tester používá k provedení testování.
- Model pracovní zátěže je speciální druh modelového případu pro testování výkonu, identifikuje proměnné a definuje jejich hodnoty použité v různých typech výkonnostního testování. (Kruchten, 2003, s. 236)

Nástroji automatizovaného testování a kvality jsou:

- IBM Rational Functional Tester
- IBM Rational Performance Tester
- IBM Rational Manual Tester
- IBM Rational Robot

- IBM Rational TestManager
- IBM Rational PurifyPlus

5.5.2 Přístup společnosti Memos

Dle informačního systému společnosti Memos je za bezchybnou aplikaci zodpovědný projektový manažer, kterému se zodpovídá tester a programátoři.

Ve společnosti Memos nebylo do konce roku 2010 testování při plánování a realizaci projektů příliš zohledňováno. Je neobvyklé, aby softwarová společnost využívala pouze jediného testera na 40 zaměstnanců. Většinou si aplikaci otestovali projektoví manažeři, kteří se na tuto činnost nespécializovali,

Od začátku roku 2011 se v tomto směru začaly vytvářet nové procesy pro testování. Byli získáni noví zaměstnanci na pozici testerů. Testeři však stále testují pouze funkcionalitu a funkční chybovost softwaru. Nepoužívají tedy žádné jiné specifické metody pro účinnější testování.

Součástí testování je i řízení kvality

Co představuje pro společnost Memos pojem kvalita?

- *„Neslibuji, co nemůžu splnit, k tomu mi slouží kapacity a metodologie, plánování*
- *Co slíbím, splním*
- *Dělám to tak, jako bych si to měl poté koupit*
- *Znalosti (známe svůj produkt, službu)*
- *Celý proces obsluhy musí být kvalitní (nejenom produkt), jedám podle metodiky vývoje a metodiky projektů*
- *Včasná komunikace - problém se dá odstranit komunikací.“ (Bykadorov, 2008)*

5.6 Nasazení

Disciplína nasazení se v životním cyklu projektu objevuje na konci konstrukční fáze.

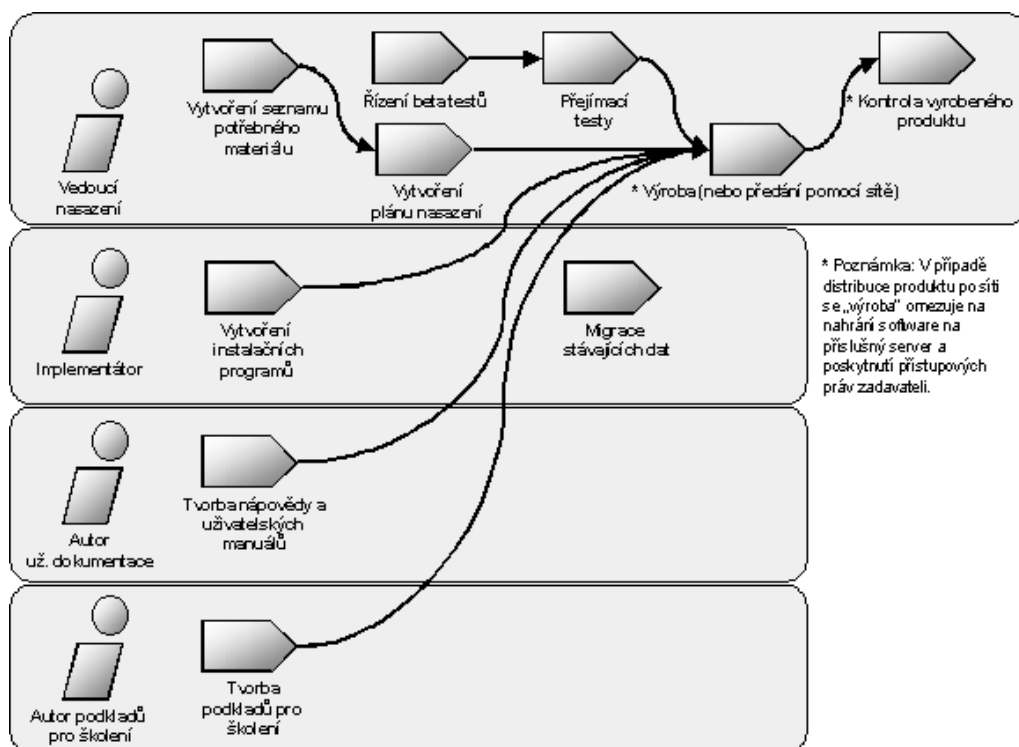
5.6.1 Co nabízí metodika RUP

Disciplína nasazení se zabývá všemi artefakty předávaných konečnému uživateli nebo zákazníkovi, jakož i ostatním zúčastněným.

Disciplína nasazení popisuje aktivity související s beta testováním a doručováním softwarem.

Nasazení může být instalováno buď prodejcem (v případech komplexních distribuovaných systémů), nebo uživatelem (v případě krabicového softwaru).

Tato disciplína je závislá na druhu produktu, který byl vytvořen a na obchodním kontextu. (Kruchten, 2003, s. 280)



Obr. č. 16: Disciplína nasazení v metodice RUP (Zdroj: Aldorf, 2008)

Klíčové artefakty pro disciplínu nasazení jsou:

- Spustitelný software, instalační skripty, nástroje, soubory, průvodce a licenční informace.
- Podpurné materiály, jako jsou uživatelské materiály a materiály údržby.
- Tréninkové materiály

- Výstupy z testování (Kruchten, 2003, s. 275)

5.6.2 Přístup společnosti Memos

Ve společnosti Memos se automaticky nevytvářejí žádné dokumenty, pokud nejsou explicitně vyžádány zákazníkem. Zákazník tedy bez vyžádání neobdrží například technické ani uživatelské dokumentace.

Zákazník obdrží objednaný software, a jeho nasazení dle obchodních smluv. V případě, že je potřeba instalovat software přímo u zákazníka, hladký průběh zajišťují projektový manažer a programátor, popřípadě i administrátor. Projektový manažer nechává podepsat předávací protokol, který ukládá do eWay a dává pokyn asistentovi k fakturaci.

5.7 Konfigurace a změny

Změny se vyskytují v každém projektu, mohou být důsledkem požadavků zákazníka, polehčujících okolností, změn legislativních předpisů, nedostupnosti zdrojů atd.

Z hlediska správy požadavků je důležité stanovit si proces pro řízení změn, tedy vytvořit proces, kterým se bude řídit navrhování, analýza a řešení změn, a navrhované změny zpracovávat výhradně podle tohoto procesu.

Dále pak vytvořit komisi pro řízení změn, kterou by měla představovat malá skupinka účastníků, jež bude přijímat návrhy na změny, bude je vyhodnocovat, rozhodovat o jejich přijetí a určovat jejich prioritu a verzi, ve které budou implementovány.

Pro každou změnu by měla být provedena analýza důsledků, která pomáhá komisi pro řízení změn s informovaným rozhodováním. Každou navrhovanou změnu je potřeba vyhodnotit a zjistit, které další požadavky, části návrhu, zdrojové kódy a testovací scénáře by se kvůli ní musely změnit.

Nezbytné je verzovat požadavkovou dokumentaci. Směrná verze požadavkové dokumentace je verze, která se má implementovat v nějaké konkrétní verzi systému. Po podepsání této verze musí každá změna projít předepsaným procesem řízení změn. Další možností řešení je použít nějaký běžný verzovací software.

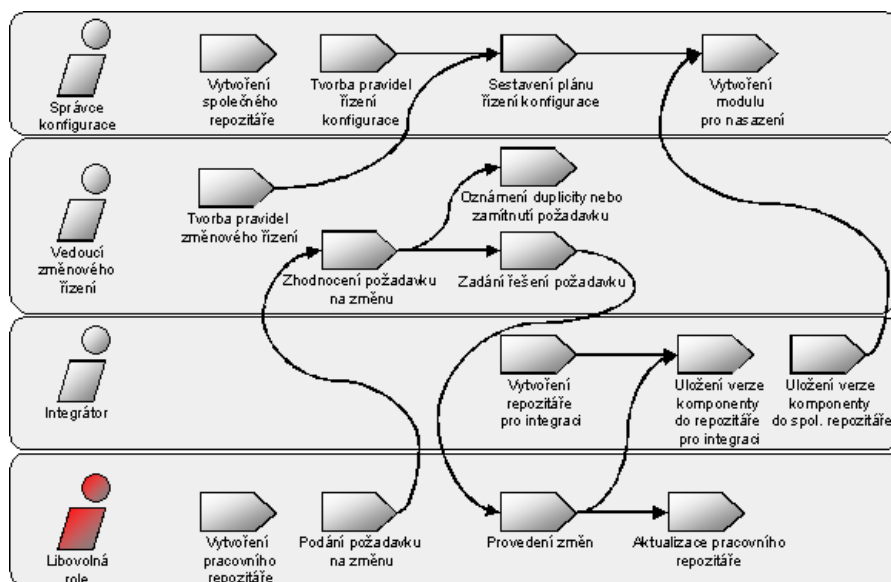
Nemělo by být opomenuto vedení historie změn, což zahrnuje zapisování dat, kdy došlo ke změnám specifikace požadavků a jaký byl důvod těchto změn. Zapisovat to

můžeme pomocí verzovacího softwaru nebo pomocí nějakého komerčního nástroje pro správu požadavků. (Wiegers, 2008, s. 63)

5.7.1 Co nabízí metodika RUP

Účelem disciplín konfigurace a řízení změn je zachování integrity projektových artefaktů tak, jak se vyvíjejí v reakci na požadavky na změny.

Konfigurační management se zabývá produktovou strukturou, identifikací elementů, verzí a validních konfigurací. Management změn zahrnuje procesy, kterými se mění artefakty konzistentním způsobem.



Obr. č. 17: Disciplína konfigurace a změny v metodice RUP (Zdroj: Aldorf, 2008)

Na obrázku č. 17 je možno vidět specifické úlohy pracovního postupu konfigurace a řízení změn, které v sobě zahrnují tyto činnosti: tvorba pravidel změnového řízení, hodnocení požadavků na změnu, provedení změn a mimo jiné vytvoření modulu pro nasazení změn.

Klíčové artefakty konfigurace a řízení změn jsou následující:

- Plán řízení konfigurace, popisuje zásady a postupy, které mají být použity v projektu (verze, varianty, postupy pro řízení změn). Plán řízení konfigurace je součástí plánu vývoje softwaru.

- Požadavky na změny: mohou být velmi různorodé. Každá žádost o změnu je spojena s původcem a příčinou. (Kruchten, 2003, s. 254)

Nástroje Rational ClearCase a Rational ClearQuest automatizují aspekty této disciplíny. (Kruchten, 2003, s. XY)

5.7.2 Přístup společnosti Memos

Pokud jsou ze strany zákazníka požadovány změny v softwarovém produktu, pak je pro tyto účely vytvořen dodatek původní smlouvy. Tento dodatek vytváří projektový manažer a prochází na straně společnosti Memos schválením obchodníkem.

Pokud není sjednána systémová podpora, zákazník dostává přístup do help desku, kde eviduje chyby v relaci s projektovým manažerem. Pokud jsou chyby v rámci reklamace, projektový manažer přiděluje programátorovi chybu k opravě. Pokud se jedná o nový požadavek, který nebyl ve specifikaci, je třeba ho nejdříve ocenit, zaslat zákazníkovi cenu emailem a vložit komentář do help-desku.

Pokud zákazník cenu schválí, projektový manažer přiřadí programátorovi nový požadavek a informuje asistenta o fakturaci po dokončení úpravy.

Systémová podpora je standardně nabízena už při uzavírání obchodu, platí se ročně 18% z ceny projektu. Systémová podpora obsahuje určitý počet hodin supportu „zdarma“.

5.8 Správa prostředí

V každé společnosti je třeba stanovit pravidla pro jednotlivé činnosti a dát k dispozici příslušné nástroje, čímž se zabývá správa prostředí.

Z hlediska projektového řízení je třeba, aby si projektový tým vybudoval efektivní informační systém, který bude poskytovat informace pro kvalitní řízení projektu. (Doležal, Máchal, Lacko, 2009, s. 241)

5.8.1 Co nabízí metodika RUP

Cílem disciplíny správy prostředí je poskytnout odpovídající podporu pro rozvoj organizace v nástrojích, procesech a metodách. Vývoj softwaru organizace se obvykle

5.8.2 Přístup společnosti Memos

Nástroje používané v metodice RUP jsou připravené speciálně pro svůj úkol. Ve společnosti Memos jsou využívány nástroje více univerzální. Vzhledem k ceně nástrojů metodiky RUP (cena za licenci pro jednoho uživatele na jeden nástroj se pohybuje v rozmezí několika set až tisíců dolarů) jsou nástroje využívané ve společnosti Memos cenově dostupnější.

Společnost Memos používá především nástroje eWay (viz Příloha A) a Redmine (viz Příloha B).

5.9 Ukončení projektu

Ukončení projektu neznamená ukončení veškerých aktivit souvisejících s projektem. Po ukončení nastává fáze vyhodnocení a produkt projektu přechází do fáze provozu, kdy je třeba dále plnit závazky dodavatelské organizace (záruka, servis) - nikoli však už jako projekt, který ukončením skončil. (Doležal, Máchal, Lacko, 2009, s 266)

Přesné určení okamžiku ukončení projektu je nezbytným předpokladem pro vyhodnocení, zda projekt splnil svůj trojimperativ a další náležitosti (rozsah, kritéria úspěchu, atd.)

Ukončení projektu bývá obvykle realizováno určitým postupem, definovaným ve smlouvě, navrženým projektovým týmem, nebo definovaným v organizačních směrnících.

I v rámci vlastního ukončení by měl projektový tým provést vlastní ohlédnutí zpět a vyhodnotit průběh projektu.

5.9.1 Co nabízí metodika RUP

V této disciplíně metodika RUP nenabízí žádný zásadní (specializovaný) návrh řešení. Částečně byla tato disciplína zohledněna v kapitole nasazení (viz. 5.7.1)

5.9.2 Přístup společnosti Memos

Za ukončení projektu a zajištění fakturace je dle informačního systému společnosti odpovědný projektový manažer.

Pro zajištění toho, aby faktura byla zaplácena, je nutné dodržet tyto kroky:

- musí být vyhotoven předávací protokol
- zdrojové kódy dostane zákazník až po zaplacení projektu

Pro otestování platební schopnosti zákazníka je vhodné vybírat zálohy předem.

Podklady pro účetní se skládají z textu faktury (název projektu, číslo smlouvy), částky, odběratele, počtu dní splatnosti (obvyčejně 15 či jinak dle smlouvy). Asistent dále nechává podepsat fakturu jednatelem a zadává do eWay potřebné údaje (datum vystavení faktury, počet dní splatnosti), poté odesílá fakturu zákazníkovi a ukládá ji do eWay. Finanční ředitel zajistí kontrolu pravidelných plateb a jejich včasnou fakturaci u supportních projektů. Dále asistent změní stav v eWay na „waiting for payment“.

Ve chvíli, kdy je projekt ukončen, je odpovědný za úplnost všech údajů v eWay technický ředitel, kterému se v tomto směru zodpovídá projektový manažer. Po skončení projektu se sejde projektový manažer s programátorem, popř. jinými lidmi a do „knowledge base“ zanesou kritické části projektu, každý na základě svých vlastních poznatků. Projektový manažer informuje obchodníka a marketingového ředitele o projektu, jeho funkcionalitě a vlastnostech. Tato aktivita je takto popsána v procesech společnosti, bohužel se nevyužívá a tedy získané poznatky z projektu nemohou využívat ostatní projektové týmy v budoucnu.

Marketingový ředitel vytváří případové studie, které kontroluje projektový manažer. Technický ředitel zadá ve spolupráci se správcem Microsoft partner programu zákaznickou referenci. Asistent kontroluje příchozí platby a mění stav projektu v eWay na „closed“

Technický ředitel a projektový manažer s architektem by měli společně vyhodnotit úspěšnost projektu a popřípadě učinit opatření do budoucna. Nejdůležitější závěry by měly být rozhodnuty na schůzkách managementu. Výše uvedení spolu vyplní informace o průběhu projektu. Technický ředitel jej uloží do eWay a zároveň kontroluje, zda jsou v eWay všechny dokumenty a náležitosti.

V eWay by na konci projektu měly být dokumenty:

- Cenová nabídka
- Zápisy z jednání
- Objednávka (smlouva)

- Faktura
- Uživatelská a technická dokumentace
- Předávací protokol
- Úvodní specifikace
- Detailní specifikace

Ve chvíli, kdy je projekt ukončen, musí být všechny náležitosti vyplněny v eWay. Projekt je svázán s fakturační adresou firmy a alespoň s jedním kontaktem (telefon a email).

Je dobré, aby se v průběhu projektu sešel projektový manažer s obchodníkem a prodiskutovali možné nové funkce a projekty, které mohou navazovat na stávající projekt. Cílem je nabídnout zákazníkovi další možné projekty, aby byla zvýšena výtěžnost zákazníka. Ten už není novým neznámým objektem, dá se zde mluvit již o určité důvěře, a pokud je zákazník s prací spokojen, lépe se mu prodává další projekt. (Bykadorov, 2008)

6 Celkové hodnocení využitelnosti metodiky RUP v malé softwarové společnosti

6.1 Silné stránky

Jednou z nejsilnějších stránek metodiky RUP je přehledně navržený model. Přehlednost spočívá v zobrazení jednotlivých rolí, jejich aktivit a výstupů (artefaktů) z jednotlivých aktivit. Každá role tedy přesně ví, za co je v projektech zodpovědná, jaké výstupy by měla generovat a jakými činnostmi tyto výstupy získávat. Metodika tímto přístupem nabízí především vysokou míru systematičnosti a snižuje rizika projektu pouhým určením rolí a jejich odpovědnosti jednotlivým členům projektového týmu v celém životním cyklu vývoje a jeho iteracích.

Metodika je přístupná na základě oprávnění všem zainteresovaným osobám v projektech, a to díky webovým stránkám v interní podnikové síti. Zaměstnanci se nemusí učit ovládat novou aplikaci díky standardnímu webovému rozhraní s možností využívání všech výhod internetového prohlížeče.

Společnost IBM komerčně dodává celistvé řešení metodiky RUP s mnoha materiály, návody, studii, a především nástroji pro jednotlivé disciplíny metodiky. Mimo to však společnost poskytuje i velké množství volně dostupných materiálů, jejichž obsahem jsou doporučené aktivity pro vývoj softwaru.

IBM je velkou společností, která dlouhodobě patří k hlavním strůjcům ICT světa. Z tohoto pohledu je na několik let zajištěn rozvoj, podpora a možné rozšiřování metodiky vzhledem k zázemí společnosti. Využíváním a podporou populárního jazyka UML společnost IBM zajišťuje dlouhou budoucnost ve využívání metodiky v mnoha společnostech na celém světě, čímž je dán prostor pro další zdokonalování.

6.2 Slabé stránky

Jednou z nejslabších stránek metodiky RUP je nutnost přizpůsobení potřebám konkrétní společnosti z důvodu rozsáhlosti a vysoké míry formálnosti. Metodika přesně specifikuje jednotlivé činnosti v průběhu vývoje, které však nemusí být v projektech efektivní a přínosné. Metodika RUP definuje přes 30 rolí, které většina malých společností

nemá možnost z důvodu počtu zaměstnanců přesně obsadit a musí tedy role slučovat, což může jít i proti smyslu metodiky. Dále je definováno více než 100 výstupů z aktivit (artefaktů), jejichž vytvářením a správou by projektový tým ztrácel mnoho času a jeho činnosti by mohly být kontraproduktivní a mohly by zdržovat celý vývoj softwarového produktu.

Metodika RUP pokrývá pouze vývoj softwaru. Nezabývá se tedy mimo jiné vytvářením rozpočtu projektu a obchodními vztahy mezi zúčastněnými stranami. Neřeší dokonce ani provoz, údržbu a vyřazení z provozu vytvořeného díla. Pro tyto činnosti musí být definovány vlastní případně použity převzaté procesy, které však musí být kompatibilní s metodikou RUP.

Metodika RUP se odkazuje pouze na nástroje z rodiny Rational (potažmo IBM). Využívání všech doporučených a specializovaných nástrojů je však pro většinu společností cenově nedostupné. Ceny samostatně prodávaných licencí se pohybují pro jeden nástroj od několika set až po několik tisíc dolarů. Metodika RUP a nástroje podporované touto metodikou jsou prodávány zvlášť.

Metodika RUP určitým způsobem může na pracovníky s tvůrčími schopnostmi působit příliš svazujícím dojmem.

Předpokladem využívání této metodiky je určitý stupeň znalostí všech zúčastněných stran. Například je nutná na straně dodavatele softwarového řešení znalost UML. Na straně odběratele je tato znalost přínosná.

7 Procesy ve společnosti Memos

V předcházejících kapitolách byly popsány procesy ve společnosti Memos, které jsou průběžně vytvářeny a upravovány od roku 2003, kdy společnost vznikla. V následující kapitole budou tyto procesy hodnoceny a v další kapitole bude proveden návrh na zefektivnění procesů při vývoji softwaru.

7.1 Hodnocení procesů ve společnosti Memos

Projektové procesy ve vývoji softwaru jsou ve společnosti Memos popsány v interním systému. Nutno dodat, že některé nejsou vyžadovány a není ani nastaven žádný mechanismus pro kontrolu plnění výstupů procesů.

Nevyužívá se disciplína podnikového modelování, která by měla být využívána především u velkých projektů. U malých projektů by tato aktivita mohla zvyšovat zbytečně náklady a konečná cena projektu by se zvyšovala.

Vynechaná je i disciplína sběr a správa požadavků. Ve společnosti není nastaven žádný standardizující proces pro správu a sběr požadavků. Tento chybějící proces je příčinou řady neshod se zákazníky, kteří v průběhu projektu, či při předávání argumentují tím, že o požadavku dříve hovořili. Pro minimalizaci nedorozumění by bylo vhodné použít nějaký nástroj pro správu požadavků.

Tvorba disciplíny testování a kontrola je ve společnosti na začátku a procesy nejsou dosud popsány.

Ostatní disciplíny jsou podobné disciplínám doporučené metodikou RUP (viz. Předcházející procesy).

7.2 Návrh procesů ve společnosti Memos

V následujících kapitolách jsou navrženy úpravy, které doplňují současné procesy ve společnosti Memos popsané v předcházejících kapitolách. Tyto procesy jsou navrženy s ohledem na výstup z analýzy a jsou navrženy tak, aby byly co nejvíce obecné a standardizované pro všechny typy projektů.

7.2.1 Předprojektová fáze

Cílem předprojektové fáze je dohoda obou zúčastněných stran na požadovaném řešení a následném stanovení ceny. S ohledem na analogické projekty by měla být snaha minimalizovat veškerá rizika. Obchodník by měl již v počátečních komunikacích s potenciálním zákazníkem přizvat projektového manažera zodpovědného za oblast zákaznických požadavků. Projektový manažer by měl mít zpracovanou šablonu a dotazy na zákazníka, vycházející z analogických projektů. Ze získaných odpovědí a představ zákazníka by se obě strany měly shodnout na rozsahu a hlavních cílech projektu. Pro stanovení ceny by se z analogických projektů získala požadovaná data. Projektový manažer by měl zpracovat a předložit softwarovému architektovi a vývojáři alespoň rámcové požadavky zákazníka za použití standardizovaného jazyka UML. Na základě analogických projektů, dekompozice požadavků s návrhem architektury řešení a metody skupinového úsudku odborníků (projektový manažer, obchodník, softwarový architekt a vývojář) by byla stanovena cena a vlastnosti požadovaného řešení.

Nevýhodou tohoto řešení je zvýšení nákladů a potřebných zdrojů na předprojektovou fázi, v systému eWay označovanou jako příležitost. Výhodou je standardizovaný postup v přípravách projektů, stejná komunikace v dokumentech (UML), členové řešitelských projektových týmů jsou součástí předprojektové fáze a získané znalosti využijí následně v řešení projektu. Dalšími výhodami je získání šablon návrhů řešení pro odlišné projekty, které při správném využívání náklady snižují. A v neposlední řadě by byla získána optimální cena a reálnější odhad doby trvání projektu než v případě znaleckého úsudku.

7.2.2 Projektová fáze projektu

Po podepsání obchodních smluv si projektový manažer zvolí po dohodě s technickým ředitelem projektový tým s ohledem na velikost projektu, ceny a dohody se zákazníkem. Dále vytvoří projektový plán.

Každý projekt by měl začít zahajovací schůzkou, na které by měly být zmíněny informace o cílech projektu a rolích jednotlivých členů týmu s přidělenými odpovědnostmi.

Společnost by měla začít používat u velkých projektů nástroj na řízení požadavků, který by umožňoval náhled zákazníkovi. Standardizované prostředí by minimalizovalo komunikační šumy mezi projektovým týmem a zákazníkem. Zákazník by navíc měl kontrolu nad plněním dohodnutých požadavků.

Nejdůležitějším výstupem přípravné fáze projektu je specifikace. Jejím cílem je především poskytnout detailní a jednoznačné podklady pro vývoj a řízení projektu. Tyto podklady by měly zahrnovat především:

- Seznam požadavků – funkcí
- Identifikace procesů organizace zadavatele
- Identifikace procesů požadovaného systému s uvedením jednoznačné vazby na procesy organizace
- Identifikace uživatelských rolí
- Přiřazení funkcí rolím (UML Use Case)
- Analytický model tříd (UML Class Diagram)
- Popis podstatných procesů (UML Activity Diagram, UML Sequence Diagram)
- Možnost návrhu komponent; rozdělení funkcí do komponent (packages); navržení rozhraní komponent
- Návrh architektury systému (návrh použití sítě, serverů, webových služeb, komunikačních protokolů).
- Definice potenciálních rizik

Tuto specifikaci by měl zákazník písemně potvrdit.

Konečný výčet požadovaných výstupů musí odsouhlasit projektový manažer společně s technickým ředitelem a softwarovým architektem s ohledem na rozsah, cenu a zdroje projektu. Tyto osoby jsou zároveň zodpovědné za tvorbu těchto výstupů a jejich správnost. Dle jednotlivých rolí by měly být rozděleny odpovědnosti za psaní dokumentace. Tester by měl psát uživatelskou dokumentaci. Vývojář by se měl podílet s architektem na technické dokumentaci.

Již při zahájení projektové fáze by se měl určit způsob a četnost testování. Projektový manažer a softwarový architekt by měli testerovi předat dostatečné informace o budovaném softwarovém produktu. Na základě komunikace mezi testerem a vývojářem v Redmine by se v aplikaci měla otestovat tak, aby v ní nebyly chyby. Tester by měl být tedy důležitou součástí projektového týmu a měl by zodpovídat za bezchybnost aplikace.

7.2.3 Poprojektová fáze

Pro zvyšování znalostí ve všech aspektech a fázích projektu je třeba zajistit řízení znalostí a tyto dokumentovat tak, aby se získané nové poznatky a zkušenosti mohly využít v příštích projektech. Po každém projektu je tedy třeba získané informace předávat všem zaměstnancům společnosti. Tím se minimalizuje riziko, že se chyby v dalších projektech budou opakovat.

8 Závěr

Vývoj softwarového řešení je složitý proces, který vyžaduje účinné a efektivní řízení. Měl by se zaměřovat na veškeré aktivity potřebné pro dodání kvalitního systému zákazníkovi. Proces vývoje by však neměl být zbytečně složitý. Využívání metodik pomáhá projektovým týmům zvýšit pravděpodobnost úspěchu a zvýšit efektivitu prací na projektu.

Hlavním cílem této diplomové práce bylo nalezení možné využitelnosti metodiky RUP v malé softwarové společnosti. Byly popsány silné a slabé stránky metodiky RUP. Přednosti tkví především v zohledňování veškerých disciplín týkajících se vývoje softwaru a v přehlednosti modelů. Proti využívání této metodiky jako produktu hovoří přílišná formálnost a velké množství rolí a artefaktů, které může malá softwarová společnost jen velmi obtížně obsáhnout. I přes tyto slabé stránky mohou být role, procesy, aktivity a artefakty popsané v metodice RUP vhodným vzorem pro nastavení procesů v malé softwarové společnosti. Tímto zjištěním byl splněn hlavní cíl této diplomové práce.

V praktické části této práce byla provedena analýza na základě získaných dat z informačního systému společnosti Memos. Výstupem analýzy je určení a pojmenování problémů v řízení projektů. Příčiny těchto problémů byly nalezeny v definovaných procesech společnosti a v nedodržování nastavených procesů projektového řízení v jednotlivých fázích softwarového vývoje.

Dále byly zpracovány návrhy úprav jednotlivých disciplín pro minimalizaci rizik spojených s projektovým řízením vývoje softwaru ve společnosti Memos.

Každý softwarový projekt je ze své podstaty jedinečný. Pro jeho úspěšné dokončení je vhodné přizpůsobovat projektové řízení a konfigurovat jednotlivé procesy projektu s cílem minimalizovat rizika. Metodiky pomáhají procesně zajistit jednotlivé disciplíny a fáze projektového vývoje. Využití metodiky může být dobrým návodem, jak předcházet problémům, nezaručuje však úspěšné dokončení projektu. Je třeba brát na zřetel například lidský faktor a s ním spojená možná selhání. Dá se předpokládat, že se v budoucnosti bude více využívat metodik při softwarovém vývoji a že se metodiky budou rozšiřovat i do malých softwarových společností.

Seznam literatury

ALDORF, F. *Metodika RUP*. [online]. 2005 [cit. 2011-14-01]. Dostupné z www: <<http://objekty.vse.cz/Objekty/RUP#abstrakt>>

ARLOW, J. – NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací*. 1. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-1503-9.

BUCHALCEVOVÁ, A. *Metodiky budování informačních systémů*. 1. Vyd. Praha: Oeconomica, 2009. ISBN 978–80–245-1540-3.

BYKADOROV, A. *Informační systém Memos*. [online]. 2008 [cit. 2011-12-03]. Dostupné z www: <<http://www.wiki.memos.cz/>>

DOLEŽAL, J. – MÁCHAL, P. – LACKO, B. a kol. *Projektový management podle IPMA*. 1. vyd. Praha: Grada Publishing, 2009. ISBN 978-80-247-2848-3.

KRUCHTEN, P. *Rational Unified Process: An Introduction*. 3. vyd. Boston: Addison Wesley, 2003. ISBN 0-321-19770-4.

McCONNELL, S. *Odhadování softwarových projektů*. Brno: Computer Press, 2006. ISBN 80-251-1240-3.

PATTON, R. *Testování Softwaru*. 1. vyd. Brno: Computer Press, 2002. ISBN 80-7226-636-5.

PUČELÍK, M. *Call centrum*. [online]. 2011 [cit. 2010-15-02]. Dostupné z www: <<http://www.memos.cz/case-study/call-centrum/>>

SCHWALBE, K. *Řízení projektů v IT*. 1. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1526-8.

SKALICKÝ, J. *Historický vývoj projektového řízení*. [online]. 2008 [cit. 2010-03-10]. Dostupné z www: <<http://lear.fek.zcu.cz/web/foxisapi.dll/lmsunifor.page?ID=TEXTY&VZ=176&T=1283>>

SVOZILOVÁ, A. *Projektový management*. 1. vyd. Praha: Grada Publishing, 2007. ISBN 80-247-1501-5.

VACH, D. *IBM Rational*. [online]. 2007 [cit. 2010-09-11]. Dostupné z www: <<http://www.itexpert.cz/ibm-rational-rup/>>

VRANA, I. – RICHTA, K. *Zásady a postupy zavádění podnikových informačních systémů*. 1. vyd. Praha: Grada Publishing, 2005. ISBN 80-247-1103-6

WIEGERS, K. E. *Požadavky na software: od zadání k architektuře aplikace*. 1. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-1877-1

Základní informace o společnosti LeasePlan. [online]. 2007 [cit. 2010-05-02]. Dostupné z www: <<http://www.leaseplan.cz/o-nas/-//SYSactTab-10112d884ba9404facb7f10a83226bfe/>>

Příloha A - Eway (CRM systém)

EWay-CRM je informační systém, zasazený přímo do aplikace Microsoft Outlook. Je určený pro všechny firmy, které chtějí efektivněji řídit svůj obchod a následnou realizaci projektů/zakázek.

Obchodní aktivity pod kontrolou

- Vždy aktuální informace o tom, kolik je rozpracovaných obchodních případů, kdo na nich dělá a v jakém jsou stavu.
- Historie komunikace s klienty na kartách
- Nabídky, objednávky a další dokumenty vždy po ruce. Systém si o ně dokonce dokáže říct sám.
- Připomínání termínů důležitých úkolů a událostí.

Přehledné řízení projektů / zakázek

- Snadná definice workflow (pracovních postupů). Díky tomu všichni zaměstnanci vyřizují projekty podle definované šablony a systém důležité fáze projektů pohlídá sám.
- Projektový deník na pár kliknutí. Okamžitě tak lze zjistit aktuální dění na projektu.
- Finanční stránka projektu je také vždy po ruce. Díky možnému provázání s účetnictvím lze v eWay-CRM důležité údaje počítat z přijatých či vydaných faktur bez nutnosti data jakkoliv ručně přepisovat.

Reporting - důležitá zpětná vazba

EWay-CRM obsahuje reportingový modul spolu s předdefinovanými sestavami. Např.:

- Obratovost a ziskovost zákazníků.
- Výkonnost konkrétního obchodníka či celého obchodního týmu.

- Sumarizace práce na projektu.
- Ziskovost zakázek za poslední rok.
- Pracovní výkazy zaměstnanců, apod.

Provázanost s Microsoft Outlook

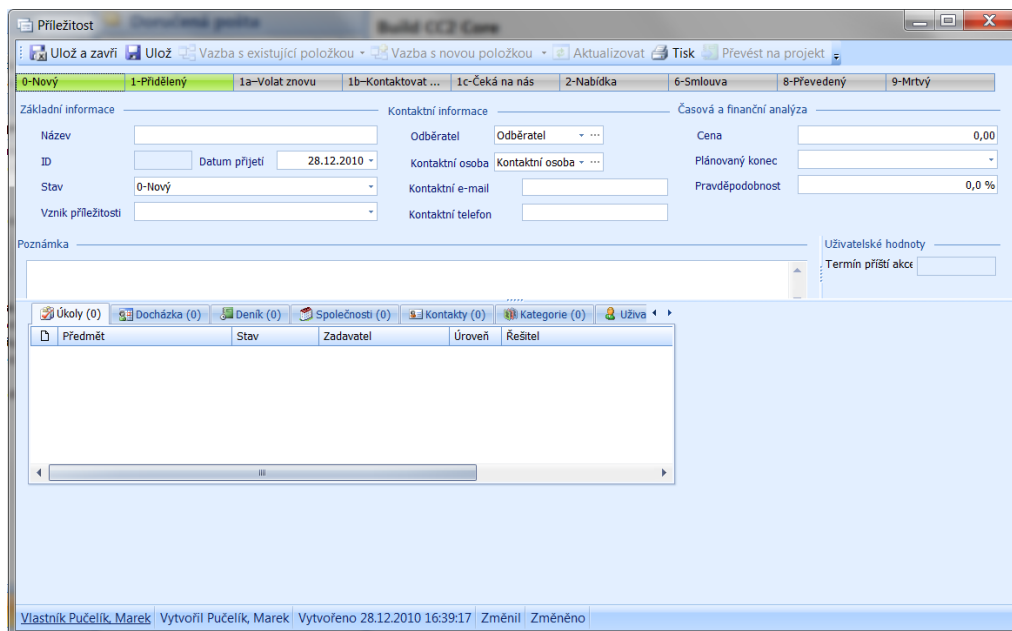
EWay-CRM je zásuvný doplněk pro Microsoft Outlook a uživatelé tak těží z těsného provázání těchto dvou aplikací. Tato integrace s sebou přináší celou řadu výhod:

- Jedno prostředí pro práci - pošta, obchod i realizace projektů je v jedné aplikaci
- Odpadá přepínání mezi více systémy či dokonce přepisování dat z jedné aplikace do druhé
- Díky stejnému uživatelskému rozhraní uživatelé nepoznají rozdíl mezi Outlookem a eWay-CRM
- Nenáročné na zaškolení pracovníků - návyky, které si uživatelé osvojili při práci s Outlookem, není potřeba měnit, vše funguje v eWay-CRM úplně stejně

eWay moduly

Příležitosti

Tento modul je určený pro obchodníky a obchodní manažery, kterým umožňuje evidovat informace o rozpracovaných obchodních případech. Obchodníci zde mají prostor pro zapisování průběhu prací na příležitosti a pro plánování dalších aktivit.



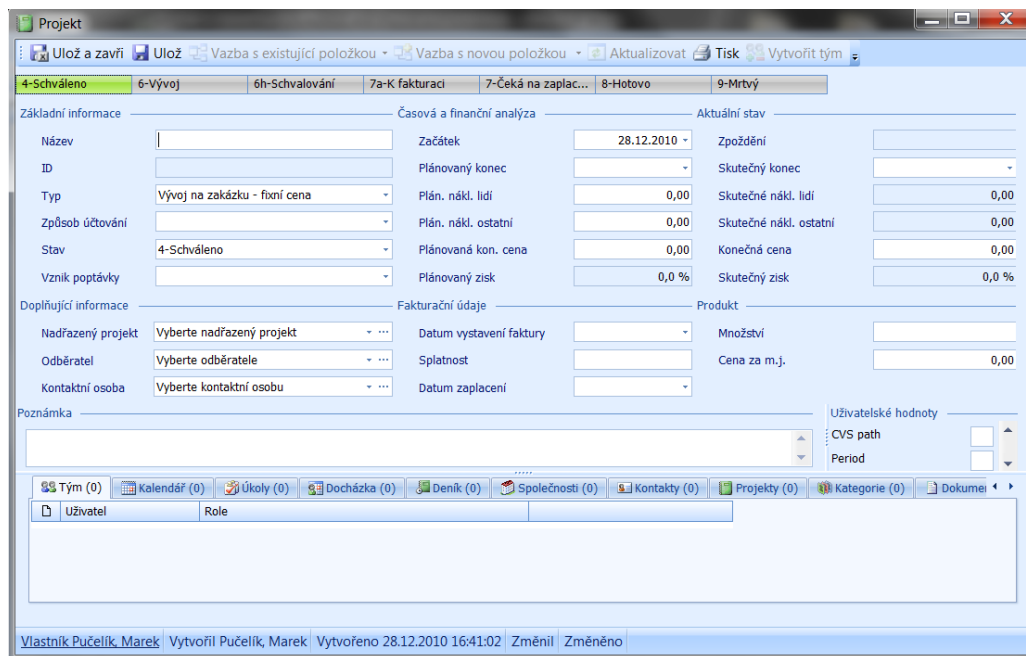
Obr. č. 1: Modul příležitosti v eWay. (Zdroj: autor)

Projekty

Modul Projekty v eWay-CRM slouží pro evidenci a správu projektů skrz celý jejich životní cyklus.

Projektoví manažeři mohou v eWay na kartě projektu sledovat vztah mezi plánovanými a reálnými údaji a ve spojení s dalšími moduly ihned zjistit, jaké jsou náklady členů celého týmu. Projektoví manažeři mají také možnost vést si přímo v eWay podrobný projektový deník, díky němuž sami při realizaci dalších zakázek zjistí, kde posledně udělali chybu, a mohou se jí příště vyvarovat.

K projektům v CRM systému eWay lze samozřejmě přikládat nabídky, smlouvy, zápisy z jednání či evidovat veškerou e-mailovou komunikaci mezi oběma stranami. Zcela jiného rozměru nabývá tento eWay modul ve spojení s reporty, kde je možné na jeden klik zjistit srovnání napříč jednotlivými projekty, efektivitu projektových manažerů či zjistit, které týmy jsou nejnákladnější.



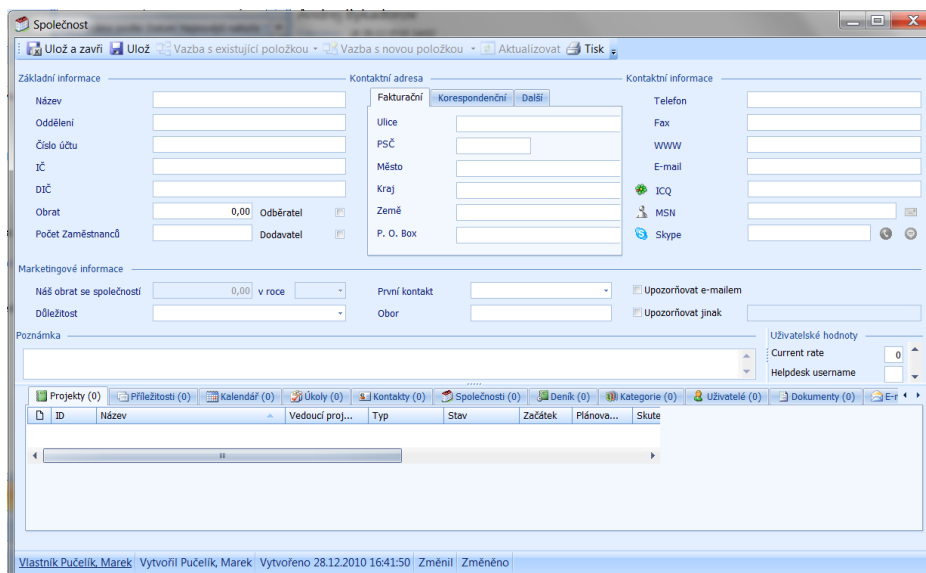
Obr. č. 2: Modul projekty v eWay. (Zdroj: autor)

Společnosti

Modul Společnosti eviduje v eWay-CRM podnikatelské subjekty, ať už stojí na straně dodavatele, odběratele či se jedná o firmy jiného charakteru. Obchodníci tak mají díky eWay na jednom místě k dispozici kompletní přehled o zákazníkovi, historii běžících či realizovaných projektů a všechny aktuální poptávky.

CRM systém eWay umožňuje automaticky evidovat e-mailovou komunikaci se společnostmi, respektive jejími zaměstnanci, a přikládat dokumenty, smlouvy či předávací protokoly.

Důležitou funkcí eWay je také možnost přikládání poznámek ve formě deníku. V případě, že je kolega z obchodního oddělení na dovolené, není potřeba mu lámat heslo na jeho počítači, abyste si přečetli historii obchodního vztahu. V eWay modulu Reporty lze společnosti řadit podle významu, například podle generovaných příjmů či indexu atraktivity.



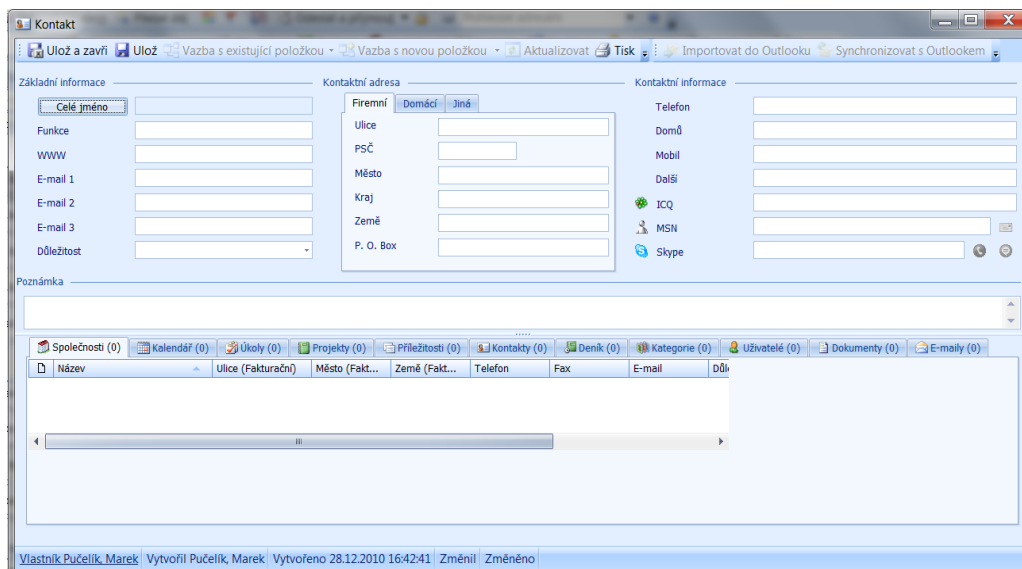
Obr. č. 3: Modul společnost v eWay. (Zdroj: autor)

Kontakty

Modul kontaktů je určen ke kompletní správě a evidenci veškerých obchodních kontaktů v eWay. Jedná se o odlišný modul než ten, který je v aplikaci Microsoft Outlook, ale eWay je s ním plně kompatibilní a nabízí funkci pro obousměrnou synchronizaci záznamů.

Díky tomu je zajištěno, že není nutné kontakt spravovat na více místech a navíc je možné používat i řešení třetích stran pro synchronizaci položek s mobilními telefony, PDA či Blackberry.

Kromě základních údajů a kontaktních informací je možné ke kontaktu do eWay opět ukládat poznámky, evidovat historii elektronické komunikace či přikládat dokumenty. Zajímavou funkcí pro urychlení práce je možnost vepisování informací o ICQ, Live Messenger nebo Skype a přímo z karty kontaktu v CRM systému eWay pak osobě psát či telefonovat.



Obr. č. 4: Modul kontakt v eWay. (Zdroj: autor)

Dokumenty

Modul Dokumenty slouží pro kompletní správu všech souborů, které je v souvislosti s řízením projektů a obchodních aktivit do systému eWay-CRM možné přikládat.

Hlavní výhodou je, že tyto dokumenty jsou přístupné přímo v MS Outlook a již není třeba hledat je v jiných správčovských aplikacích.

Podobně jako u modulu Deník, i zde se pravá síla tohoto modulu objeví ve spojení s dalšími položkami – projekty, společnostmi či obchodními kontakty. Projektoví manažeři mohou do CRM systému eWay vkládat specifikace, zápisy z jednání a předávací protokoly, obchodní manažeři naopak objednávky, smlouvy, atp.

Součástí tohoto eWay modulu jsou i zásuvné moduly do Microsoft Word a Microsoft Excel.

Uživatelé

V modulu Uživatelé je zobrazen seznam všech uživatelů CRM systému eWay. U každého uživatele je možné evidovat veškeré podrobnosti, zjišťovat, na kterých projektech pracuje a jaká je jeho role v nich, sledovat jeho docházku či spravovat mzdy.

Citlivost zde obsažených údajů lze zabezpečit nastavením bezpečnostní politiky.

Reporty

eWay-CRM obsahuje modul Reporty, který slouží pro rychlý přehled nad daty systému. V CRM systému eWay jsou přednastaveny některé sestavy dat.

Zajímavým reportem, kterým lze v eWay zobrazit, je přehled projektů dle jejich stavů. V přehledné tabulce lze jednoduše zjistit, kterému projektovému manažerovi svěřit další úkoly, protože má nejméně živých projektů.

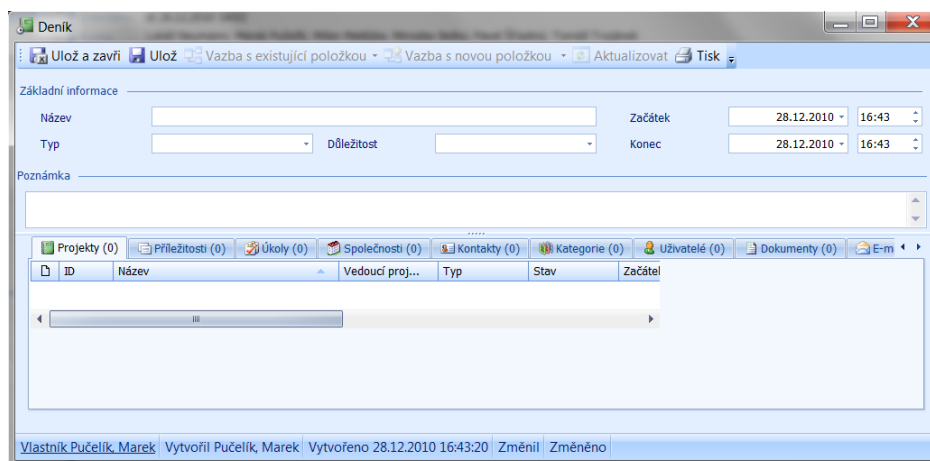
Pracovníci obchodního oddělení při řízení vztahů se zákazníky jistě ocení datové sestavy, které se týkají výše obrátu jednotlivých společností či indexu atraktivity. Jejich aktivity tak mohou být přesně cíleny a budou se moci soustředit na nejvíce perspektivní klientelu.

Kromě předdefinovaných sestav dat je v eWay možné vytvářet své vlastní reporty za využití technologie Microsoft Reporting Services. Obecně lze říci, že z CRM systému eWay je možné dostat všechna data, která jsou v něm uložena.

Deník

Modul Deník spravuje veškeré poznámky a události všech uživatelů systému eWay. Pro zjednodušení správy a zvýšení přehlednosti může uživatel vybírat z několika typů záznamů, které jsou posléze v seznamech odlišené jinými ikonami.

Obchodní manažer naopak ocení sílu deníku ve spojení s moduly Společnosti nebo Kontakty, protože bude schopen do eWay vkládat poznámky o schůzkách, telefonátech či o celkovém vývoji obchodního vztahu.



Obr. č. 5: Modul deník v eWay. (Zdroj: autor)

E-mailly

E-mailly, došlé do aplikace Microsoft Outlook, je možné ukládat i do CRM systému eWay. Záznam o elektronické komunikaci se archivuje v systému a zůstane uchován, i když jej uživatel smaže ze své poštovní schránky.

eWay-CRM umožňuje automatickou evidenci příchozích a odchozích e-mailů a automatické svazování s osobami, kterých se týkají. V eWay tak je možné bez zvýšeného úsilí evidovat komunikaci s klientem a v případě, že je osoba, která s ním jedná, na dovolené, jeho kolega může plně zastoupit jeho práci.

Pokud se po několika letech zákazník rozhodne provést úpravy projektu, je získání dva roky starých informací z elektronické pošty záležitostí pouhého otevření karty příslušného kontaktu a klepnutí na záložku e-mailly v CRM systému eWay. Toto navíc bez ohledu na to, zda pracovník, který měl projekt před lety na starosti, ve firmě pracuje, či nikoliv.

Docházka

Modul Docházka umožňuje uživatelům eWay zaznamenávat čas, který strávili prací na konkrétním projektu, a to i s přesným popisem vykonávané činnosti.

Zcela jinou dimenzi pak modul Docházka nabývá ve spojení s modulem Mzdy. V CRM systému eWay tak lze jednoduše sledovat reálné náklady lidí na jednotlivých projektech v čase a za pomoci modulu Reporty pak třeba zjistit, které projektové týmy přinášejí společnosti nejvyšší marže.

Díky záznamům v eWay modulu Docházka je snadné vyúčtovat zákazníkovi vykonanou práci a vzniklé náklady. K tomu opět výborně poslouží modul docházky ve spojení s reporty.

Mzdy

Modul mezd v CRM systému eWay zajistí dokonalý přehled o mzdách zaměstnanců. Finanční ředitel má možnost do eWay modulu Mzdy pohodlně zadat platy všech pracovníků, ať už je jejich ohodnocení fixní nebo hodinové. eWay je možné napojit na jakýkoliv účetní systém, čímž je zautomatizována část administrativních procesů nezbytných pro chod společnosti.

Hlavní výhodou modul mezd přináší ve spojení s moduly Docházka a Projekty, kdy je na jeden klik možné zjistit aktuální náklady všech lidí na projektu.

Dovolená

Díky modulu Dovolená může mít manažer společnosti přehled o kapacitních možnostech svého oddělení a zautomatizovat proces žádosti o dovolenou. Schvalovací proces je plně nastavitelný. Skrz emailové notifikace nadřízeným pracovníkům je žádost o dovolenou velmi jednoduchá. Nahlédnutím do karty uživatele systému jeho manažer snadno zjistí, kolik dní dovolené si již vybral a kolik mu jich ještě zbývá.

Školení

V eWay modulu Školení lze zadávat a sledovat informace o různých kurzech a školeních, kterých se zaměstnanci účastní.

Na základě těchto informací pak lze efektivněji vytvářet týmy pro konkrétní projekty. Je zde také možnost generovat reporty s informacemi o kurzech absolvovaných zaměstnanci, což lze využít např. při outsourcingu.

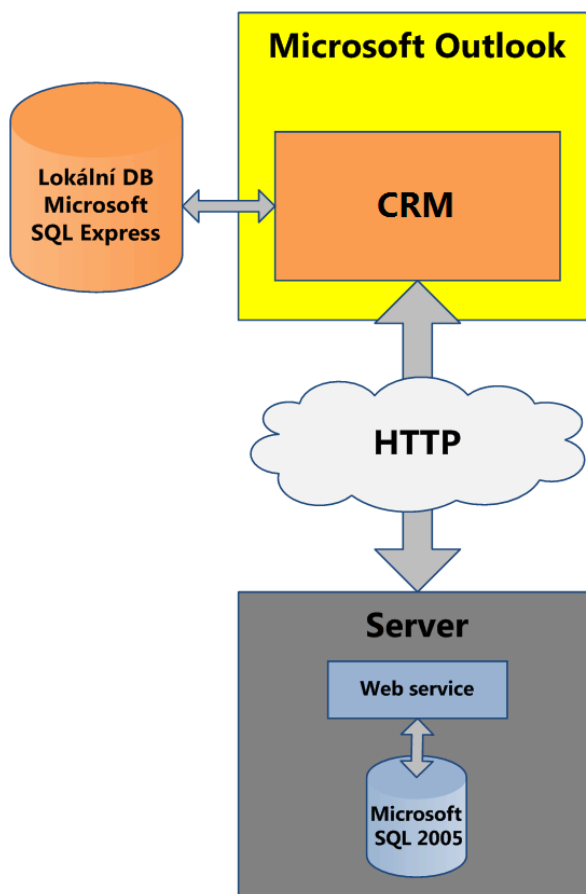
Znalosti

Modul Znalosti je určen pro sledování schopností zaměstnanců. Je velmi snadno dohledatelné, která osoba disponuje jakými kvalitami a podle toho tvořit optimální týmy či vhodná pro outsourcing.

Tento modul je možné také použít ve spojení s reporty.

Základní architektura eWay

Jako databáze je použit MS SQL Server 2005. Microsoft SQL Server 2005 Express je poměrně schopné řešení pro evidenci malých databází a Microsoft ji poskytuje zdarma. Není tedy nutné dokupovat jakékoliv licence při instalaci CRM systému na klientské stanice. Následující obrázek představuje základní architekturu systému, tak jak je využita v CRM systému eWay. Jediný rozdíl je v již zmíněné klientské databázi za Microsoft SQL Server 2005 Express.



Obr. č. 6: Architektura eWay. (Zdroj: autor)

Centrální databáze je umístěna na serveru a obsahuje veškerá data. Je použit opět Microsoft SQL Server 2005, nyní již však nikoliv edici Express, ale vyšší verzi. To z důvodu potřeby evidovat větší množství dat, než na klientském PC.

Webová služba slouží pro jednotný přístup k databázi. Tato mezivrstva je nasazená kvůli centrální správě, zejména pak bezpečnostní politice. V případě potřeby nasadit jakýkoliv jiný systém, který bude s eWay spolupracovat pak nebude nic jednoduššího, než využívat právě tuto webovou službu. Bude tak zajištěno, že žádná jiná aplikace nebude moci s databází provádět nepřípustné operace. Protokoly HTTP / HTTPS zajišťují komunikaci mezi klientskou aplikací a webovou službou.

Klientská část CRM zajišťuje synchronizaci mezi lokální databází a centrální databází, synchronizaci s Microsoft Outlookem a zobrazuje data. Uživatelské rozhraní je navrženo tak, aby bylo maximálně podobné uživatelskému rozhraní Outlooku. Příloha A byla převzata z <http://www.eway-crm.cz/>

Příloha B - Redmine

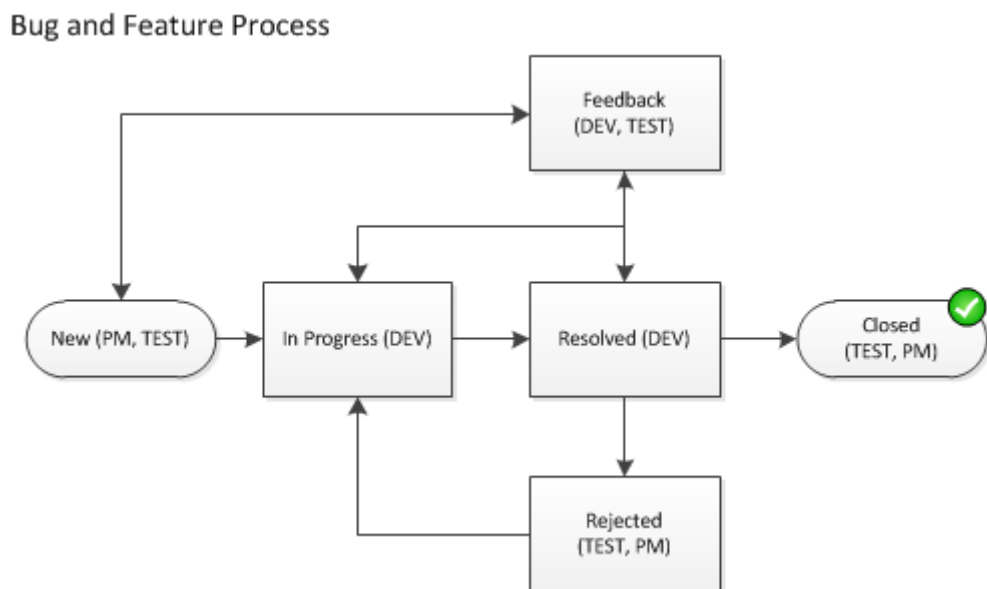
Redmine se používá jako prostředek pro komunikaci v rámci týmu. A to především mezi projektovým manažerem, vývojářem a softwarovým architektem v činnostech vývoje a mezi projektovým manažerem a manažerem testování. Manažer testování používá Redmine jako prostředek komunikace se svými testery.

Redmine umožňuje sledování stavů projektu, jeho verzí a testovacích kruhů. Při rychlém pohledu na jednotlivé úkoly je možné odhadovat potřebný čas a pracnost pro dokončení projektu nebo jeho verze.

Typy úkolů a jejich stavy

Funkce

Funkce se používá hlavně na začátku projektu. Odpovědná osoba má za úkol vytvořit některé části projektu týkající se specifikace (popis problému).



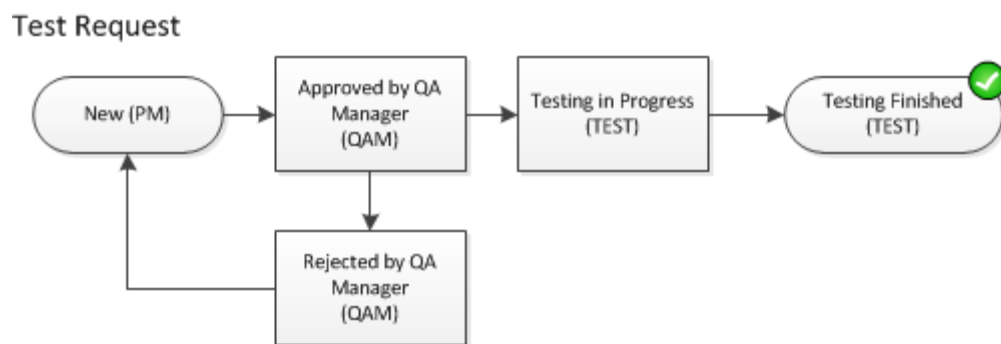
Obr. č. 7: Proces funkce a chyb (tzv. bugů) (Bykadorov, 2008)

Bug

Bug Softwarová chyba je obvyčejný termín používaný k popisu chyby, selhání, nebo chyby v počítačovém programu nebo systému, který produkuje nesprávné nebo neočekávané výsledky, nebo způsobí, že se tyto výsledky budou chovat nezamýšleným způsobem. Nejvíce chyb vyplyne z chyb a omylů v programu zdrojový kód nebo jeho designu, a někdy může být způsobeno tím, že kompilátory produkuje nesprávný kód.

Poptávka na testování

Projektový manažeři vytvářejí pro každé testovací kolečko tzv. „test request“. Tím se žádá QA manažera o přidělení kapacity na testování aplikace.



Obr. č. 8: Poptávka na testování. (Bykadorov, 2008)

Příloha B byla převzata z wiki.memos.cz