

Podpora činností softvérového manažéra so zameraním na riadenie softvérového projektu

ZOLTÁN PULAI

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
blackbrain@ynet.sk*

Abstrakt. Vývoj väčších softvérových systémov si vyžaduje vysokú mieru spolupráce medzi viacerými členmi projektového tímu s rôznymi zodpovednosťami. Proces vývoja je často rozložený v čase a priestore a je vykonávaný v rámci špecializovaných pracovných skupín. Tieto vlastnosti vyžadujú nájdenie vhodných odpovedí na otázky, týkajúce sa deľby práce, komunikácie, koordinácie a spolupráce počas plánovania, vývoja a údržby softvérových systémov. V predkladanej práci si zhrnieme, na aké činnosti môže využiť softvérový manažér podporné nástroje. Podrobnejšie si rozoberieme tematiku plánovania a sledovania úloh. Na základe analýzy vlastností dvoch nástrojov si zhrnieme rôzne prístupy k riadeniu softvérového projektu.

Úvod

Skupina zručností expertného vývojára je tvorená programovacími trikmi a tipmi ako aj znalosťami z metodológií návrhu, konfiguračného riadenia a znalosťami súčasných vývojových technológií a nástrojov. Skupina zručností všeobecného, nie softvérového manažéra je tvorená úplne odlišnými znalosťami. Všeobecní manažéri ovládajú ako vytvárať rozpočet, ako robiť pohovory s budúcimi zamestnancami, ako vytvárať pravidelné hodnotenia produkcie a podobne. Projektoví manažéri vedia ako riadiť rozsah, čas, náklady, kvalitu, ľudské zdroje, komunikáciu a riziká v projekte. Dobrí manažéri softvérových projektov by mali ovládať niektoré zo zručností expertných vývojárov, všeobecných manažérov aj projektových manažérov. Zručnosti softvérového manažéra by mali pokrývať päť hlavných oblastí: *vytváranie odhadov, plánovanie, sledovanie a riadenie, manažment rizík a meranie.*

Vytváranie odhadov

Ľudia si pod odhadovaním často predstavujú „dohadovanie“ alebo „expertný názor“, ale tieto pohľady sú skôr výsledkom aktuálneho stavu v praxi, ako samotným opisom efektívneho nastavenia nákladov a rozvrhu. Skúsení technickí manažéri spravidla prechádzajú tromi základnými krokmi pri vytváraní podkladov pre plánovanie: najskôr odhadnú rozsah vyvíjaného systému, následne vypočítajú úsilie, potrebné na vytvorenie daného rozsahu a nakoniec vytvoria rozvrh na základe odhadnutého úsilia. Dobrý odhad by mal byť skôr počítaný, ako „hádat“.

Plánovanie

Pri plánovaní by mal manažér pokryť nasledovné oblasti:

- Odhadnúť predpokladané celkové úsilie a rozvrh projektu
- Rozhodnúť z koľkých ľudí sa bude skladať projektový tím a aké bude mať zloženie
- Zvoliť vhodný model životného cyklu projektu
- Zvoliť vhodné technické postupy na zistenie požiadaviek, vytvorenie návrhu, na spravovanie výstupov z projektu, na vytváranie kódu, testovanie softvéru a na zachytenie skúseností s projektom pre znovupoužitie na iných projektoch
- Identifikácia vhodných postupov na pre zaručenie kvality, potrebných pre dosiahnutie požadovanej kvality rámci stanovených nákladov a stanoveného času.
- Vytvorenie sady indikátorov, ktoré poskytnú prehľad o projekte počas jeho realizácie

Mnoho z uvedených aktivít sú zanedbávané na väčšine projektov.

Sledovanie a riadenie

Na typickom projekte je manažment takmer funkcionalita typu čierna skrinka: na začiatku sa vytvoria plány, ale počas vykonávania projektu sa málokedy vie, čo práve prebieha a je nutné akceptovať, čokoľvek sa nakoniec vytvorí. Na dobrom projekte by mal mať manažér dobrý prehľad – v každom okamihu by mal poznať aktuálny stav nákladov, rozvrhu, kvality a funkcionality projektu. Ukázalo sa, že najlepšie projekty boli vždy charakterizované vysokou mierou sledovania a riadenia. Typickými prvkami sledovania a riadenia projektov sú zoznam úloh, správy a stretnutia ohľadom stavu projektu, zhodnotenia pri dosiahnutí míľnikov, správy o rozpočte a podobne.

Manažment rizík

Softvérové projekty sú ohrozované riziká, ktoré vyplývajú zo zmeny požiadaviek zákazníka, príchodu nových technológií, nestability niektorých nástrojov, nespoľahlivých partnerov, neskúseného personálu a z rôznych iných zdrojov. Ak manažér aktívne neriadi riziká, neriadi vôbec svoj projekt.

Meranie v softvérovom projekte

Jedným z kľúčov k dlhodobému úspechu spoločnosti je získavanie historických údajov na analýzu kvality a produktivity. Ak sa získavajú údaje o vynaloženom úsilí (človekomesiace), rozvrhu (kalendárny čas), veľkosti programu (napríklad počet riadkov kódu alebo podobne) a počte chýb, môžu tieto údaje slúžiť ako dobrý základ pri plánovaní projektov v budúcnosti.

Je mnoho činností a povinností, ktoré musí softvérový manažér vykonávať pri riadení softvérového projektu. Na väčšinu týchto úloh však existujú podporné nástroje, ktoré vo veľkej miere automatizujú jednotlivé činnosti a zvyšujú efektivitu práce. My sa v ďalšom texte zameriame len na jednu oblasť – na sledovanie a riadenie úloh pri vytváraní softvérových systémov.

Kolaboratívne riadenie úloh

V poslednom čase jasným trendom, že sa do vývoja zapájajú čoraz väčšie a rozložené tímy. Tieto zmeny si vyžadujú nové IT riešenia pre kolaboratívne riadenie úloh, pri ktorých sú členovia tímu zapojení do procesu riadenia a zodpovednosti za priebeh projektu zásadným spôsobom.

Globalizácia a narastajúca konkurencia nútia spoločnosti k decentralizácii vývoja. Z tohto dôvodu sa vytvárajú rozložené tímy z členov s rôznymi skúsenosťami a technickým pozadím na zvládnutie zložitých inžinierskych postupov. Decentralizácia však prináša so sebou aj radu nových ťažkostí. Ako môžu členovia tímu dosiahnuť jednotné porozumenie úlohy, identifikovať závislosti medzi úlohami, zostať informovaní o aktuálnom stave a výsledkoch a spoločne vyvíjať požadovaný výsledok? Obyčajne sú znalosti rozložené v tíme, dohody medzi členmi sú komunikované len vzájomne a nie s celým tímom a zdržania sú rozpoznané často príliš neskoro. Chýba jednotný pohľad, čo vyžaduje časovo náročné stretnutia s celým tímom na synchronizáciu a riadenie úloh v rozloženom inžinierskom procese. Počas priebehu procesu sa môžu zmeniť podmienky, čo môže vyvolať prispôbovanie konečných termínov, pridelenie úloh a vytvorenie ďalších úloh s pridelením rôznych zodpovedností. Cieľom je kombinovať metódy a výsledky z riadenia pracovného toku a výskum v oblasti CSCW (Computer Supported Collaborative Work) s metódami na zlepšenie a koordináciu rozloženého inžinierskeho procesu. S konceptom pracovného toku definovaného používateľom je snaha podporiť kolaboratívne riadenie úloh v tímoch. Sú poskytnuté flexibilné koordinačné mechanizmy na vytvorenie, prispôbenie a synchronizáciu pracovného toku v závislosti od aktuálnej situácie. Členovia tímu sú informovaní o stave procesu, o jeho priebehu, o existujúcich prekrytiach a hroziacich konfliktoch.

Scenár

Nasledovný aplikačný scenár ilustruje proces softvérového vývoja v rozložených tímoch. Vývojový tím pozostáva z dvoch partnerov A a B z rôznych spoločností. Predpokladajme, že v tíme A traja z členov (Peter, Paul, Mary) a v tíme B dvaja z

členov (Jim a John) riešia systémovú platformu. Práve bol dosiahnutý míľnik a teraz treba dokončiť nasledujúci komponent v najbližších deviatich mesiacoch. Na úrovni projektu je proces rozdelený na dve časti, každý z nich obsahuje rôzny počet úloh. Plnenie úloh je kontrolované na projektovej úrovni. Postupnosť činností na dosiahnutie jednotlivých cieľov nie je preddefinovaná a každý tím je zodpovedný za dokončenie svojich cieľov. Táto pracovná etapa je ovplyvnená systematickým zlučovaním výsledkov a vyžaduje rozsiahle a časovo náročné prispôsobovanie sa medzi tímami prostredníctvom telefónu, e-mailu a projektových stretnutí.

Predpokladajme, že sa objavil problém pri práci s veľkým množstvom dát vo vyvíjanej aplikácii a zároveň klient požaduje implementáciu dodatočnej funkcionality – napríklad funkcionality drag&drop.

Popísané úlohy sú prejednané dvoma vývojovými tímami na spoločnom stretnutí. Pripraví sa návrhy na riešenie úloh a vo vývojovom dokumente sa zaznačí, že Jim a John z tímu B pripraví riešenie problému s veľkými dátami a Peter a Mary z tímu A vyvinú funkcionality drag&drop. Obe úlohy sa zdajú byť na prvý pohľad navzájom nezávislé. Ak by sa predsa našla nejaká závislosť, tímy sa dohodnú, že sa o tom budú navzájom informovať. Oba tímy pracujú na svojich úlohách v zažitom cykle návrhu, implementácie, odlaďovania a testovania. Funkcionalita drag&drop sa bude implementovať ako podpísaný aplet. Keďže Peter nemá veľa skúseností s podpisovaním apletov, prehľadáva vývojárske dokumenty ale požadované informácie nenájde. Snaží sa dovolať Jimovi, ktorý má v danej oblasti veľa skúseností, ale nedarí sa mu to. Preto mu Peter pošle patričný e-mail. Keďže mu Jim na mail neodpovedal ani na nasledujúci deň, skontaktuje sa s Johnom, ktorý mu oznámi, že Jim je na dovolenke. Napriek tomu ho John odkáže na relevantné dokumenty na internete. Peter po preštudovaní daných dokumentov tie nahrá do úložiska vývojárskych dokumentov, aby boli v budúcnosti dostupné aj pre ostatných.

Medzitým je drag&drop funkcia dokončená. Patričné dokumenty sú aktualizované a komponenty sú naložené na testovací server. Všetci vývojári sú o zmenách informovaní prostredníctvom e-mailu.

Jim preberá integračné testovanie – úlohu na projektovej úrovni. Ihneď rozpozná chybu súvisiacu s funkciou drag&drop – pravdepodobne niektorá zmena v prístupových metódach nebola komunikovaná. Navyše niektoré pomocné funkcie na prácu s dokumentami s rôznymi názvami ale rovnakou sémantikou boli vyvinuté duplicitne. Jim vyhotoví odpovedajúce testovacie protokoly. Prostredníctvom e-mailu požiada oba vývojové tímy aby vzájomne vyriešili novú požiadavku – chybu v prístupovej metóde a synchronizáciu pomocných funkcií.

Požiadavky

Analyzovaním prezentovaného scenára sme zistili, že oba tímy sú samo organizujúce. Ona tímy súčasne pristupujú k rovnakým informáciám, týkajúcich sa úloh, výsledky projektu sú rozdelené v a medzi tímami, a často sú problémy riešené len vzájomne medzi dvoma členmi pomocou e-mailu. Na riešenie problémov a synchronizáciu úloh sú potrebné spoločné stretnutia. Je taktiež dôležité, aby členovia tímu boli informovaní o zmenách v dokumentoch, aby sa mohli daným zmenám prispôbiť pri svojej práci.

Na základe týchto zistení sme zistili, že členovia tímu potrebujú podporu pri plánovaní a koordinácii úloh, pri manažmente členov projektu a zdrojov, a pri prispôsobení práce a synchronizácii procesov v závislosti od súčasného stavu procesu. Najskôr uvedieme požiadavky na riadenie úloh.

Požiadavky na riadenie úloh	Dôležitosť
Zdieľaný pracovný priestor pre vývojové dokumenty (špecifikácie, testovacie údaje, protokoly, ...)	++
Zdieľaný pracovný priestor s automatickým rozpoznávaním zmien dokumentov, ktoré by mohli mať vedľajší efekt	++
Zdieľané úlohy so zdieľaným pracovným priestorom za účelom rozpoznať vzťahy a konflikty čím skôr	++
Jednoduchá možnosť rozpoznania neukončených úloh a aktivít	++
Zdieľané zoznamy úloh	++
Možnosť štruktúrovania, pridelovania a kombinovania úloh	++
Verziónovanie vývojových dokumentov za účelom sledovania cesty vývoja	+
Možnosť priebežne meniť stav úlohy za účelom informovania ostatných členov	+
Synchronizácia medzi plánom projektu a súčasným stavom	+
Vyhľadávanie v úlohách zamerané na obsah	+
Kontrolný zoznam na kontrolu výsledkov	++

Tabuľka 1. Manažment úloh

Členovia tímov väčšinou nemajú detailne špecifikované prístupové práva. Napríklad v tíme s piatimi vývojármi každý z nich má práva na vytvorenie informácií súvisiacich s úlohou, môže čítať dokumenty vytvorené ostatnými členmi, a podobne. Iba zákazník a dodávateľ majú špecifickú rolu. Identifikovali sme nasledovné požiadavky:

Požiadavky na manažment členov tímu a zdroje	Dôležitosť
Dynamická prispôsobiteľnosť zloženia tímu na základe požiadaviek	++
Dynamické pridelovanie/zmena zdrojov s ohľadom na vykonávanie úlohy	++
Jednoduchý koncept rolí založený na aktivitách členov tímu	++
Podrobne špecifikované prístupové práva k vývojovým dokumentom	+
Pridelovanie úloh	++

Tabuľka 2. Manažment členov tímu a zdrojov

Na koordináciu zúčastnených členov tímu sú potrebné služby pre komunikáciu, prispôsobenie spolupráce a jednaní. Členovia tímu by mali byť informovaní

o aktuálnom stave procesu, o posledných zmenách, o zmenách v projektovom pláne a prienikoch v rozvrhu. Požiadavky sú zhrnuté v nasledovnej tabuľke:

Požiadavky na zabezpečenie koordinácie a tímového povedomia	Dôležitosť
Fórum pre koordináciu a prispôsobenie sa v rámci tímu a medzi tímami	++
Viditeľnosť komunikácie a vzťahov spolupráce	++
Diskusné fórum na dohodnutie implementačných alternatív	++
Virtuálne schôdze a zdieľané pracovné prostredie za účelom riešenia konfliktov	++
Udalosťami riadené včasné zistenie porúch	++
Decentralizovaná koordinácia dôležitých termínov a zdrojov orientovaná na úlohy	++
Fórum na dohodnutie protiopatrení na zaručenie dodržania dôležitých termínov	++
Virtuálne stretnutia za účelom synchronizácie priebehu úloh	+
Notifikačný mechanizmus pre automatickú notifikáciu o zmene stavu úlohy	++
Používateľské adresáre	+
Viditeľnosť dohôd medzi partnermi	+
Fórum na diskusiu odlišností	+

Tabuľka 3. Zabezpečenie kooperácie a tímového povedomia

Pre zjednodušenie riadenia úloh a zvýšenie efektivity je dôležitá jeho integrácia do pracovného prostredia používateľa.

Na podporu kolaboratívneho riadenia úloh sa využívajú používateľom definované pracovné toky. Organizačné ciele sú prevedené na individuálne a tímové ciele. V popredí sú autonómnosť členov tímu a vzťahy v tíme a tie určujú pracovný plán. Hlavnými cieľmi sú jednoduchý spôsob popísania úloh a postupov, ktorý zahŕňa štruktúrované úlohy s podúlohami, sledovanie cieľov, poverovanie, koordináciu procesov a riadenie udalostí a konfliktov.

Riadenie projektu pri agilnom vývoji softvéru

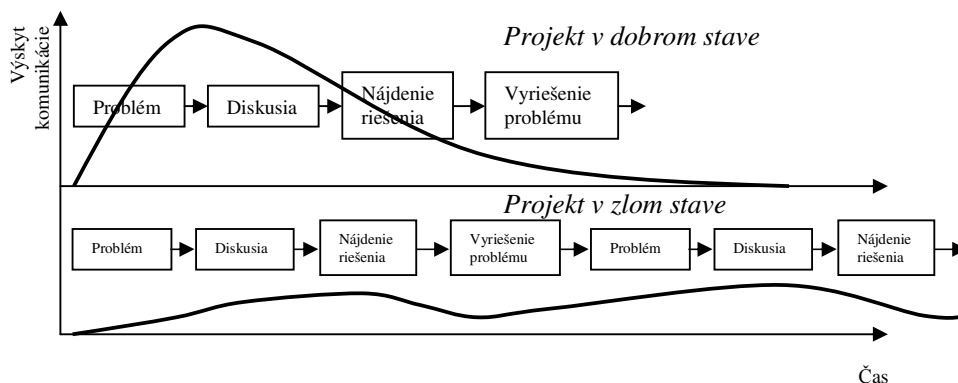
V poslednom čase sú agilné metódy vývoja softvéru čoraz populárnejšie, pretože čas sa vývoj softvérového produktu stále skracuje. Avšak bežné techniky na riadenie projektu, založené na dokumentoch, sú často nepoužiteľné pri takýchto metódach vývoja. Čas, potrebný na vývoj, by sa mal skracovať tým, že sa z procesu vynechá tvorba dokumentov; detailné návrhy, detailné špecifikácie, detailné správy a správy o postupe. Manažéri musia vedieť sledovať kvalitu a postup vývoja bez týchto dokumentov. Navyiac by mali byť manažéri schopní zachytiť rôzne stavy projektu ako

dobrý, normálny alebo zmätený stav a to bez spomínaných dokumentov. Taktiež môžu nastať v projekte závažné problémy, bez toho aby o nich manažér vedel.

Preto bol zavedený nový postup na podporu riadenia projektu pri agilnom vývoji softvéru, založený na sledovaní a vyhodnocovaní komunikácie. Základnou myšlienkou je, že dobrá komunikácia medzi členmi tímu vedie k dobrým výsledkom a dobrému priebehu vývoja. Komunikácia je nevyhnutná a je jednou z najdôležitejších nástrojov vývojára. Je aj jednou z dôležitých ukazovateľov stavu projektu. V ďalšom texte rozumieme pod komunikáciou všetky spôsoby dorozumievania sa: schôdze, konverzáciu, dokumentáciu, inšpekcie a podobne.

Model komunikácie

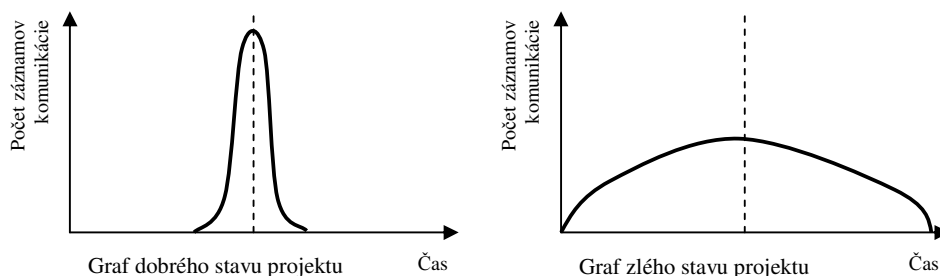
Stavy projektu zahŕňajú rôzne stavy: dobrý, normálny, zmätený, stály. Komunikácia je rôzna pri projekte v stálom stave a projekte v zmätenom stave. Základnou myšlienkou opísaného prístupu je, že miera rozptýlu výskytu komunikácie je závislá od stavu projektu. Napríklad ak je projekt v dobrom stave, komunikácia na riešenie problémov sa vyskytuje často. Členovia sa zapájajú do diskusie za účelom nájdenia najlepšieho riešenia. Po nájdení riešenia sa riešenie implementuje. Keďže je riešenie založené na dostatočne rozsiahlej diskusii, problém je vyriešený úplne. Výsledkom toho je, že komunikácia na vyriešenie problému nie je rozptýlená. Naopak pri projekte v zmätenom stave sa proces komunikácie, diskusie, nájdenia riešenia a implementácie riešenia niekoľkokrát opakuje. Keďže diskusia o probléme nie je dostatočne rozsiahla, nepodariť sa nájsť dobré riešenie. Keď sa napriek tomu riešenie implementuje, nielenže sa zachová pôvodný problém, ale môžu sa objaviť nové problémy. Výsledkom je rozptýlená komunikácia na dlhom časovom úseku. Uvedené situácie sú znázornené na obrázku 1.



Obrázok 1. Komunikácia pro rôznych stavoch projektu

Na analýzu komunikácie môžeme využiť dva prístupy. Prvý z nich je použiteľný na analýzu komunikácie, ktorá je už kategorizovaná, ako napríklad séria správ o chybách s počtom chýb. Metóda je založená na štatistickom rozptyle a miere

fluktuácie okolo strednej hodnoty výskytu komunikácie. Príklad modelu je znázornený na obrázku 2. Pri grafe dobrého stavu projektu je rozptyl komunikácie v čase nízky. To znamená, že vývojári diskutujú intenzívne o jednom probléme v krátkom časovom intervale. Pri druhom grafe je rozptyl komunikácie vysoký, diskusia o probléme sa naťahuje v čase.



Obrázok 2. Rozptyl komunikácia pri v rôznych stavoch projektu

Výhodou tohto prístupu je menšia výpočtová náročnosť. Tým pádom je vyhodnotenie rýchle a manažéri môžu sledovať stav projektu v reálnom čase.

Druhý prístup analýzy komunikácie je vhodný pre komunikáciu, ktorá nie je kategorizovaná ako napríklad záznamy komunikácie vývojárov alebo záznamy e-mailov. Pri väčšine projektov je veľké množstvo takejto nekategorizovanej komunikácie. Tento prístup je komplexnejší a skladá sa z dvoch procedúr: kategorizácia komunikácie s podobným predmetom a výpočet rozptylu v rámci zväzku. Keďže tieto procedúry sú pomerne komplikované, uvedieme si len základné kroky, ktoré sa vykonávajú v rámci jednotlivých procedúr.

Prvá procedúra má za úlohu kategorizovať komunikačné záznamy a skladá sa z dvoch krokov:

- Vyhodnotenie podobnosti medzi záznamami v modeli vektorového priestoru
- Zoskupenie záznamov na základe podobnosti komunikácie do zväzkov

Druhá procedúra vykonáva výpočet rozptylu v rámci zväzkov a pozostáva z nasledovných krokov:

- Mapovanie záznamov do časových radov pre každý zväzok (tému)
- Vyhodnotenie stavu projektu na základe rozptylu v rámci zväzkov

Aj keď je výpočtová náročnosť tohto spôsobu vysoká, je nevyhnutná na analýzu nekategorizovanej komunikácie.

Pri oboch prístupoch k analýze komunikácie je potrebné zachytávať komunikáciu medzi členmi tímu.

Nástroj na podporu riadenia projektu

Na základe modelu komunikácie bol vyvinutý nástroj na podporu riadenia softvérových projektov. Používateľ si môže zvoliť, aký spôsob analýzy komunikácie chce použiť. Ak je komunikácia už kategorizovaná, môže sa použiť prvý, jednoduchší spôsob analýzy. V prípade nekategorizovanej komunikácie je potrebné použiť druhý spôsob analýzy napriek vysokej výpočtovej náročnosti.

Nástroj bol nasadený na rozsiahle projekty s otvoreným zdrojom (open-source). Na základe výsledkov pri nasadení na takéto rozsiahle projekty sa potvrdilo, že zmeny stavu projektu odpovedali zmenám výskytu nových chýb a zmenám počtu neopravených chýb.

Aj keď manažéri nemajú k dispozícii dokumenty, záznamy komunikácie môžu indikovať stav projektu automaticky pomocou použitia nástroja. Navyše indikácie zmeny stavu projektu v čase pomocou nástroja môžu byť veľmi užitočné pri riadení projektu.

Záver

Od manažéra softvérového projektu sa vyžadujú skúsenosti z mnohých oblastí a mali by ovládať aj zručnosti, ktoré sú pre manažérov nesoftwareových projektov zbytočné. Manažéri by pri riadení projektu mali pokryť týchto päť hlavných oblastí: *vytváranie odhadov, plánovanie, sledovanie a riadenie, manažment rizík a meranie*. S týmito oblasťami je spojených mnoho činností, niektoré špecifické len pre softvérové projekty. Na väčšinu týchto činností však existujú podporné nástroje, ktoré vo veľkej miere automatizujú jednotlivé činnosti a zvyšujú efektivitu práce. My sme sa zamerali na oblasť plánovania a sledovania úloh a popísali sme dva prístupy k tejto oblasti.

Prvou z nich je kolaboratívne riadenie úloh. Na podporu kolaboratívneho riadenia úloh sa využívajú používateľom definované pracovné toky. Organizačné ciele sú prevedené na individuálne a tímové ciele. V popredí sú autonómnosť členov tímu a vzťahy v tíme a tie určujú pracovný plán. Hlavnými cieľmi sú jednoduchý spôsob popísania úloh a postupov, ktorý zahŕňa štruktúrované úlohy s podúlohami, sledovanie cieľov, poverovanie, koordináciu procesov a riadenie udalostí a konfliktov.

Druhou skúmanou oblasťou bolo riadenie softvérového projektu pri agilnom vývoji softvéru. Metóda sa zakladá na sledovaní rozptylu komunikácie v rámci softvérového projektu. Sleduje sa hlavne množstvo a rozptyl výskytu komunikácie a na základe štatistického vyhodnotenia sa určí stav, v akom sa projekt nachádza.

Použitá literatúra

1. McConnell, S.: The Software Manager's Toolkit. *IEEE Software* July/August 2000
2. Kolvenbach, S., Wolfgang, G., Klöckner, K.: With Collaborative Task Management towards Lightweight Distributed Engineering Processes. *Proceedings*

of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, 2006.

3. Hanakawa, N., Okura, K.: A project management support tool using communication for agile software development. *Proceedings of the 11th Asia-Pacific Software Engineering Conference, 2004*

Annotation

Supporting project managers with concentrating on controlling the software project

Development of larger software systems demands intensive collaboration among multiple team members with different responsibilities. The development process is often distributed across time and space and is performed within and between specialized workgroups. This necessitates finding appropriate answers to questions related to division of labor, to communication, and to coordination and cooperation in the planning, development and maintenance of software systems. In presented paper, we review the activities of software manager, which can be supported by software tools. We describe more in detail the issue of task planning and tracking. Upon the analysis of two tools, we summarize different approaches for the software project controlling.