

Podporné prostriedky pre riadenie softvérového projektu so zameraním sa na sledovanie úloh, ľudské zdroje, matica zodpovednosti

PETER ČIMO

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
xcimo[zavináč]is.stuba[.]sk*

Abstrakt. Na riadenie softvérového projektu sa v dnešnej dobe kladie čoraz väčší dôraz. Ľudia poverení vedením takýchto projektov však majú k dispozícii viacero podporných prostriedkov a techník, ktoré im môžu túto úlohu pomôcť zvládnuť a udržovať poriadok a prehľad nad projektom. Obsahom tejto práce je nielen prehľad týchto prostriedkov, ale aj skúsenosti a postrehy autora z praxe a jeho pohľad na túto problematiku.

Úvod

Tak ako je činnosť obchodov stále viac závislá na informačných technológiách, tak sú manažéri softvérových projektov pod zvyšujúcim sa tlakom priniesť kvalitný softvér načas a v limitovanom rozpočte. Tak okrem technickej odbornosti musia mať aj nutnú manažérsku zručnosť aby mohli viesť a riadiť vývoj v softvérových projektoch [8]

Nástroje pre vývoj softvéru bývajú primárne zamerané na podporu technickej stránky práce. Bez ohľadu na použité nástroje, sú niektoré dôležité aspekty vývoja ako použité postupy alebo veľkosť tímu netechnického charakteru. Napríklad, rozdelenie projektu medzi viac ľudí v tíme vyzdvihne dôležitosť koordinácie a kooperácie [5]

Akonáhle na jednom projekte začne pracovať viac ľudí, je potrebné pri vývoji dodržiavať určitý systém a prácu začať rozdeľovať. Jedna otázka je ako prácu rozdeľovať, odpoveď na túto otázku však nieje jednoduchá a stačila by na samostatnú prácu. Okrem toho nezapadá do témy tejto práce preto sa jej nebudem príliš venovať. Zaujímavejšou otázkou alebo skôr témou je, samotné sledovanie jednotlivých úloh, ktoré boli pridelené vývojárom alebo inej pracovnej sile v softvérovom projekte. Pozrime sa teda bližšie na úlohu ako organizačnú jednotku. [7] definuje úlohu ako postupnosť pokynov, ktoré sú chápané ako základné jednotky práce. Podľa [11] sa

úlohou nazýva presne definovaná jednotka práce v softvérovom procese, ktorý vykonáva riadenie, s viditeľným posunom do určitého stavu projektu. Úlohy majú počiatočné kritériá ako aj kritériá pre svoje ukončenie.

Každá úloha v softvérovom projekte má určité vlastnosti. Medzi tieto vlastnosti patria napríklad:

- osoba alebo zoznam osôb zodpovedných/pridelených/pracujúcich na tejto úlohe
- časový harmonogram na splnenie tejto úlohy - od,do,prípadne počet človeko-dní vyhradených pre túto úlohu
- priorita úlohy - niektoré úlohy môžu byť dôležitejšie a uprednostnené pred ostatnými, niektoré zas naopak môžu byť len okrajové
- závislosť na niektorej inej úlohe - niektoré úlohy nemôžu byť splnené pred ukončením inej úlohy
- a ďalšie (rozpočet, miesto, atď.)

Matica zodpovednosti




Pre rôzne vlastnosti úloh existujú rôzne techniky ich zobrazenia pre ich lepšiu čitateľnosť a prehľadnosť. Pre osoby pracujúce na projekte sa napríklad hodí technika pomenovaná matica zodpovednosti. Táto matica ukazuje presné zapojenie každej funkčnej sekcie v každej úlohe alebo aktivite na všetkých úrovniach [9]. Je to štruktúra, ktorá závisí od organizácie projektu slúžiaca na jej špecifikáciu s cieľom pomôcť zabezpečiť, aby každý element v rámci projektu, mal pridelenú zodpovednú osobu [4]. Jednoduchšie povedané táto matica nám znázorní vzťah medzi úlohami a osobami v rámci projektu. Tieto vzťahy môžu byť rôzne, osoba môže byť za danú úlohu priamo zodpovedná, môže na úlohe len asistovať, môže pri úlohe vystupovať ako konzultant alebo môže nad úlohou vykonávať kontrolu.

Podľa môjho názoru, matica zodpovednosti môže byť vynikajúcou pomôckou pri projektoch najrôznejšieho rozsahu vďaka svojej škálovateľnosti. Jej slabé miesto je však pri projektoch s veľkým pohybom pracovných síl, kde sa osoby často menia. Keď sa na jednej úlohe vystrieda viacero zodpovedných pracovníkov, len ťažko sa potom dá určiť, ktorý z nich nesie aký podiel zodpovednosti za úlohu. Tento problém ide niekedy eliminovať rozdelením úlohy na menšie podúlohy, ale nie vždy je úlohu možné takto rozdeliť.

Výhodou tejto techniky je nepochybne aj jej jednoduchosť na realizáciu. K jej zostrojeniu netreba žiaden špecializovaný softvér, stačí k tomu obyčajný tabuľkový editor, ktorý je dnes zahrnutý do väčšiny balíkov kancelárskeho softvéru. **Error! Reference source not found.** zobrazuje príklad matice zodpovednosti.

Podporné prostriedky pre riadenie softvérového projektu so zameraním sa na sledovanie úloh, ľudské zdroje, matica zodpovednosti 3

Artifact	Project Owner	User	Project Manager	System Analyst	Software Analyst	Use Case Architect	Developer Specifier	Test Roles	Administrator Roles
Vision	A		C	C	W	C		C	C
Use Case Model	A		C	C	W				
Software Architecture Document	A					W		C	C
Use Case Specification	A		C		C	C	W		C
Use Case Realization						C	C	W	
Component						C		W	C

 = Write / responsible for
  = Contribute / review
  = Formally Accept

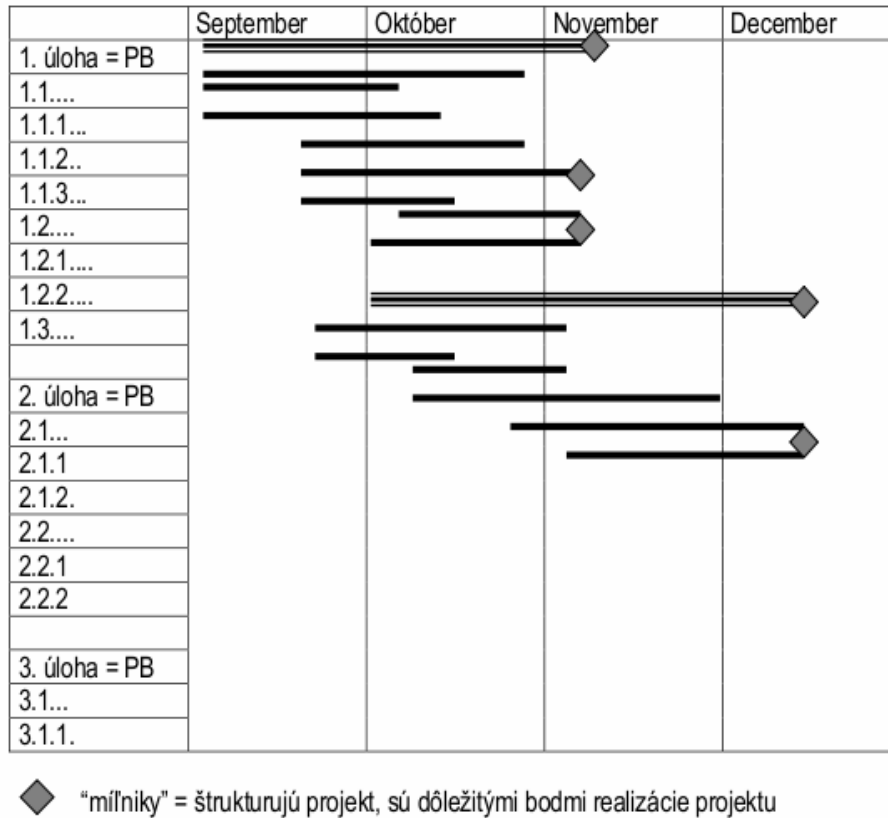
Obr. 1. Matica zodpovednosti [2].

Ganttov graf

Ďalšou vlastnosťou úloh bolo ich umiestnenie v čase - na časovej osi. Táto vlastnosť, nám umožňuje zaraďovať úlohy do pracovného plánu. Pracovný plán popisuje technické aktivity, ktoré majú byť vykonané počas rôznych fáz projektu [3].

Existuje niekoľko prostriedkov, ktoré umožňujú prepojenie pracovného plánu a jeho jednotlivých celkov s časovým plánom. Jeden z najefektívnejších je tzv. „Ganttov graf“. V rámci grafu môžeme vidieť, ako sú jednotlivé úlohy prerozdelené do pracovných celkov podľa rozpätia obdobia ich realizácie. Ich začiatok a ukončenie je jasne vyznačené. V tomto type grafu míľniky projektu môžu byť stanovené a označené ako aj kritické faktory, týkajúce sa hlavne obdobia počas ktorého sa majú úlohy zrealizovať. Výhodou tohto grafu je jasné rozloženie a prehľadnosť faktov [6]. Na Ganttovom grafe môžeme ľahko sledovať priebeh projektu a dôležité termíny (takzvané „deadline“), zlepšuje koordináciu v tíme.

Zostrojenie Ganttovho grafu pritom podobne ako pri matici zodpovednosti nieje nijak technicky náročné. Riadky predstavujú úlohy alebo aktivity a horizontálna os predstavuje čas. Rôzne značky potom môžu reprezentovať dôležité udalosti, míľniky, termíny. **Error! Reference source not found.** zobrazuje príklad Ganttovho grafu.



Obr. 2. Ukážka Ganttovho grafu [6].

Nevýhodou Ganttovho grafu by podľa môjho názoru mohla byť malá detailnosť úloh v grafe, respektíve ak by sme na ganttovom grafe chceli zobraziť podrobne každú dielčiu úlohu, graf by musel mať značné rozmery. Ganttov graf by som teda použil na zobrazenie väčších etáp projektu, ktoré by mali vlastné grafy zobrazujúce časový plán menších úloh vrámci nich. Ďalšou nevýhodou by mohla byť absencia znázornenia vzťahu medzi jednotlivými úlohami - z grafu nemusí byť jasné, či úlohy, ktoré idú po sebe, jedna od druhej závisia, čo je často veľmi podstatné. Ak by totiž úlohy na sebe záviseli, druhá z nich nemôže začať ak sa prvá oneskorí. Tento problém sa dá tiež eliminovať napríklad použitím šípok, ktoré by znázorňovali vzťahy medzi úlohami. Táto dodatočná funkčnosť však podľa mňa znižuje prehľadnosť a ľahkú čitateľnosť tejto metódy zobrazovania.

Bugzilla

Matica zodpovednosti aj Ganttov graf sú podporné techniky pre vedenie projektu. V nasledujúcej časti tejto práce by som chcel spomenúť aj podporné prostriedky pre riadenie softvérového projektu vo forme konkrétnych softvérových aplikácií. Jedným z nich je práve softvér s názvom Bugzilla.

Bugzilla je systém pre sledovanie problémov. Keď Mozilla, veľký projekt web prehliadača na poli otvoreného softvéru, potrebovala systém pre sledovanie problémov, jeho vývojári vytvorili svoj vlastný systém-Bugzillu, tiež ako otvorený softvér. Prvá verejne vydaná verzia bola napísaná v jazyku Perl a komunita otvoreného softvéru pokračovala v jej vylepšovaní. Dnes používa Bugzillu množstvo užívateľov, vrátane projektov otvoreného softvéru (Apache, Linux, Open Office) rovnako ako aj súkromných či verejných organizácií (NASA, IBM). A tak sa zameranie Bugzilly, zmenilo z nástroja pre projekt Mozilla, na jeden z hlavných systémov pre sledovanie problémov [10].

The screenshot shows the Bugzilla search and filter interface. It includes the following sections:

- Search:** Summary: contains all of the words/strings, Search button.
- Product:** Bugzilla, Calendar, Camino, CCK, Composer.
- Component:** Accessibility, Account Manager, Address Book, Addressbook/LDAP (non-UI), Administration.
- Version:** 0.17, 0.3, 0.4, 0.5, 0.6.
- Target:** Future, mozilla1.8alpha6, 3.0, Jan.
- Filters:** A Comment, The URL, Whiteboard, Keywords.
- Status:** UNCONFIRMED, NEW, ASSIGNED, REOPENED, RESOLVED, VERIFIED, CLOSED.
- Resolution:** FIXED, INVALID, WONTFIX, DUPLICATE, WORKSFORME, MOVED.
- Severity:** blocker, critical, major, normal, minor, trivial, enhancement.
- Priority:** -, P1, P2, P3, P4, P5.
- Hardware:** All, DEC, HP, Macintosh, PC, SGI, Sun.
- OS:** All, Windows 3.1, Windows 95, Windows 98, Windows ME, Windows 2000, Windows NT.
- Email and Numbering:** Any of: bug owner, reporter, QA contact, CC list member, commenter.
- Bug Changes:** Only bugs changed between: and Now, where one or more of the following changed: [Bug creation], Alias, Assignee, URL, and the new value was:.
- Sort results by:** Reuse same sort as last time.

Obr. 1. Zobrazenie úlohy v Bugzille [10].

Obr. 1 zobrazuje obrazovku pre špecifický problém (chybu, vylepšenie, úlohu, záplatu). Pre sledovanie konkrétneho problému je potrebné, vedieť ho lokalizovať. Polia *Product*, *Component*, *Version*, *Status* a *Reporter* súvisia so sledovaním, zatiaľčo polia *Summary*, *Status Whiteboard*, *Keywords*, *Severity*, *Attachments* a *Dependencies* súvisia s riešením daného problému. Tieto polia obsahujú dáta, ktoré sa musia vložiť, keď sa zakladá problém a ktoré potom pomáhajú filtrovať pri hľadaní problému. Ďalšia cesta ako nájsť informácie je cez „Bugzilla’s reports“, ktoré popisujú aktuálny stav problémov a grafy, ktoré zobrazujú stav aplikácii v priebehu času [10].

Bugzilla je web-aplikácia, čiže interakcia s užívateľmi prebieha iba cez HTML stránky. Napriek tomu, že konštrukcia Bugzilly naznačuje že by mala podporovať komerčné databázy, pracuje iba s databázou MySQL, populárnou otvorenou databázou [10]. V novších verziách však podporuje aj ďalšiu otvorenú databázu PostgreSQL, navyše je možné podľa potreby dorobiť podporu inej databázy [1].

Podľa môjho názoru podporeného niekoľkoročnými skúsenosťami z praxe, je Bugzilla veľmi kvalitným a profesionálnym prostriedkom pre podporu riadenia projektov. Jej robustnosť poskytuje široké pole pôsobnosti a uplatnenie od malých projektov až po projekty medzinárodných rozmerov (zoznam projektov využívajúcich Bugzillu možno nájsť na [1]).

Aj keď je Bugzilla pôvodne softvér orientovaný na sledovanie problémov a chýb v programoch, jej architektúra a funkčnosť nebráni ju používať ako softvér pre sledovanie akýchkoľvek úloh, teda aj úloh v procese vývoja v softvérovom projekte. Pri novom projekte riadiaca osoba vytvorí novú doménu úloh (pole *Product*), vránci ktorej potom vytvorí úlohy. Úlohy majú popis, prioritu, závislosť na iných úlohách, ďalej môžu mať nastavený dátum dokedy je úlohu potrebné dokončiť („deadline“), stav a ďalšie. Zakladateľ úlohy, potom môže buď úlohu dakomu prideliť, alebo ju nechať ako novú a vývojári si ju podľa vlastného úsudku pridelia. Na úlohe môže pracovať aj viacej ľudí naraz. Pri každej úlohe je možné pridávať komentáre a takýmto spôsobom poskytovať možnosť komunikácie vývojárom, prípadne aj osobám mimo projektu, ako napríklad zákazníkom. V novších verziách je možné pri úlohe zadávať aj počet odpracovaných hodín a časový odhad na dokončenie. Systém potom vie všetky dáta vyhodnocovať a vytvárať tabuľky a grafy či už pre celé projekty, jednotlivé úlohy, alebo vývojárov. Tieto informácie môžu byť podľa môjho názoru užitočné pre prehľad a analýzu vykonávaných prác na projektoch a teda veľmi nápomocné pri riadení projektu.

Keďže Bugzilla je otvorený softvér nieje problém ju upraviť tak, aby spolupracovala s repozitárom zdrojových kódov ako napríklad *Subversion* alebo *Concurrent Versioning System*. Týmto prepojením by Bugzilla umožnila sledovať priebeh prác až na úroveň zmien zdrojových kódov, čo však nemusí byť vždy žiadúce.

Bugzilla podporuje aj export zoznamu úloh do rôznych dátových formátov:

- *iCalendar* - pre zobrazenie úloh v kalendári
- *CSV*, *XML* - pre ďalšie spracovanie, napríklad aj do tabuľkového editora, kde z týchto dát vieme zostrojiť Ganttov graf, alebo maticu zodpovednosti

Záver

Medzi základné podporné techniky pre riadenie softvérového projektu patria matica zodpovednosti a Ganttov graf. Obsahom matice zodpovednosti je pridelenie vzťahov medzi úlohy alebo aktivity a členov tímu. Matica zodpovednosti sa dá vyhotovovať pre rôzne stupne detailnosti. Ganttov graf zobrazuje časový harmonogram-časovú návznosť jednotlivých úloh, aktivít a dôležitých míľnikov v projekte. Môže byť doplnený ďalším značením pre zobrazenie vzťahov medzi úlohami. Je vhodný len pre menší počet zobrazovaných úloh, pri väčšom počte sa stáva neprehľadný. Bugzilla je systém pre sledovanie problémov vyvinutý spoločnosťou Mozilla. Jeho robustnosť poskytuje dobré využitie aj pre vytváranie, sledovanie a riadenie úloh v softvérovom projekte. Jeho otvorenosť ponúka možnosť prepojiť ho s inými technickými prostriedkami používanými pri vývoji softvéru. Pomocou exportu dát je možná ďalšia analýza dát týkajúcich sa projektov, úloh alebo členov tímu. Tieto vlastnosti robia zo systému Bugzilla veľmi výkonný podporný prostriedok pre riadenie softvérového projektu.

Použitá literatúra

1. Bugzilla's Features. [Online] [Dátum: 24. 10 2007.] <http://www.bugzilla.org/features/>.
2. **Collaris Remi-Armand, Dekker Eef, Warmer Jos.** Tailoring RUP made easy: Introducing the Responsibility Matrix and the Artifact Flow. [Online] 15. 9 2006. [Dátum: 24. 10 2007.] http://www.ibm.com/developerworks/rational/library/sep06/collaris_dekker_warmer/.
3. Glossary of Environmental Terms. [Online] 10. 1 2001. [Dátum: 24. 10 2007.] <http://www.dtsc.ca.gov/InformationResources/glossary.cfm>.
4. Glossary of General SE Terms. [Online] [Dátum: 24. 10 2007.] http://www.computer.org/portal/site/seportal/menuitem.092750c1f9509be2169d1205682079ec/index.jsp?&pName=se_level1&path=seportal/subpages&file=glossary.xml&xsl=generic.xsl&.
5. *Designing Task Visualizations to Support the Coordination of Work in Software Development.* **Halverson Ch.A., Ellis J.B., Danis C., Kellogg W.A.** CSCW'06, 2006.
6. **Bienzle, Holger.** *A Survival Kit for European Project Management: Advice for Coordinators of Centralised Socrates Projects.* s.l. : Gröbner Druckgesellschaft m.b.H., 2001.
7. IEEE Std 610. *IEEE Standard Glossary of Software Engineering Terminology (ANSI).*
8. **Jurison, Jaak.** Software Project Management: The Manager's View. *Communications of AIS.* 1999, Zv. 17, 2.

9. **Kahn, Kenneth B.** *The PDMA Handbook of New Product Development*. 2004. ISBN 04-7148-524-1.
 10. **Nicolás Serrano, Ismael Ciordia.** Bugzilla, ITracker, and Other Bug Trackers. *IEEE Software*. 2005.
 11. **Software Engineering Institute.** SEI-CMU-93-TR-25. 1993.
- Použitá literatura

Annotation

Support resources for leading of software project management focused on task monitoring, human resources, matrix of responsibility

Nowadays is leading of software project more and more accentuate. Persons charging to lead such projects but have available numerous support resources and techniques, which can help them to keep order and overview over project. Subject of this paper is not only overview of these resources, but also author's experiences, remarks and his point of view of this issue.