

SLEDOVANIE ÚLOH V SOFTVÉROVOM PROJEKTE

„The trouble with programmers is that you can never tell what a programmer is doing until it's too late.“

Tomáš Háber

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
Tomas.Haber[zavináč]gmail[.]com

Abstrakt. Vývoj softvéru má svoje špecifiká. V máloktoom odvetví sa vyskytuje také vysoké percento neúspešných projektov (štúdie uvádzajú viac ako 50%) a iba pomerne malá časť projektov (približne 16%) je dokončená v dohodnutom termíne a bez prekročenia rozpočtu. Tieto čísla sa môžu zlepšiť aj používaním nástrojov na sledovanie úloh. Umožňujú lepšiu kontrolu stavu práce na projekte a sledovanie vyťaženia ľudských zdrojov a týmto pomáhajú napr. lepšie reagovať na prípadné zmeny v plánoch. V eseji sa zameriavam na porovnanie formálneho sledovania úloh pomocou systému na sledovanie úloh a neformálneho používaním anotácií v zdrojovom kóde. Opisujem aké sú výhody formálneho prístupu k sledovaniu úloh, ktoré výhody ponúka a aké nevýhody prináša. Venujem sa úlohe neformálneho sledovania úloh ako indikátora stavu projektu a vhodný doplnkový spôsob sledovania úloh pre vývojárov. Analyzujem nakoľko je vhodné a nutné sledovanie úloh formalizovať a dôsledky takéhoto nastavenia.

Kľúčové slová: sledovanie úloh, anotácie

Úvod

Vývoj komerčného softvéru je skupinová aktivita. Projektíví manažéri vytvárajú špecifikácie, vývojári implementujú funkcie a tímy zabezpečenia kvality hľadajú a potvrdzujú defekty spolu s im priradenými opravami. Kvôli veľkému množstvu ľudí a

2 Tomáš Háber

pozícií zapojených v tejto forme skupinovej práce, sa komunikácia a kolaborácia stala integrálnou súčasťou procesov vývoja. [2]

Naviac k množstvu úsilia mnohých ľudí ktoré je potrebné vyvinúť, vyžaduje vývoj kvalitného softvérového produktu opatrný manažment mnohých nesúvisiacich častí. Funkcie a konkrétne úlohy pre vývojárov sa rodia zo špecifikácií a defekty a chyby sa nevyhnutne nájdu. Tieto chyby sú často spravované použitím systému na sledovanie úloh. Systém na sledovanie úloh pomáha softvérovým tímom organizovať hlásenie chýb, ich priradenie, sledovanie, vyriešenie a spoľahlivý centrálny, zdieľaný zoznam úloh, ktorý je používaný členmi manažmentu a slúži aj ako archív dokončenej práce.[2]

Ako doplnok používaniu takehoto systému sa môže používať aj menej formálne procesy a nástroje na podporu sledovania úloh. Konkrétne sú textové anotácie v podobe komentárov v zdrojovom kóde.

V prvej časti eseje sa venujem opisu formálneho a neformálneho sledovania úloh a ich použitiu a funkciám. V ďalšej časti sa venujem výhodám a nevýhodám týchto prístupov k sledovaniu úloh. V závere sa venujem zhodnoteniu porovnania formálneho a neformálneho sledovania úloh a ich vzájomného vzťahu.

Sledovanie úloh

Formálne spôsoby sledovania úloh

Systémom na sledovanie úloh rozumieme softvérový nástroj, ktorý umožňuje tímu zodpovedného projektu, zaznamenávať a sledovať stav všetkých neukončených úloh. V skutočnosti ale systémy na sledovanie úloh poskytujú oveľa viac funkcionality, ako napr. sledovanie požiadaviek, chýb, otázok a odpracovaného času.[2]

Systém na sledovanie úloh slúži ako centralizovaná databáza pre sledovanie chýb, funkcionality a otázok, od ich vytvorenia až po uzatvorenie v konečnom stave. [2]

Chybami rozumieme defekty v softvérovom projekte, ktorými sa ma zaoberať vývojový tím a môžu byť nahlásené zákazníkmi, členmi tímu zodpovedných za sledovanie kvality alebo hocikým, kto ma prístup k systému na sledovanie úloh.[2]

Funkcionalitou rozumieme novú funkciu alebo vylepšenie existujúcej funkcie, ktoré sú zvyčajne zadávané projektovými manažérmi alebo vedúcimi tímov. [2]

Otázkami sú myslené otázky prevažne od zákazníkov týkajúce sa širokého spektra tém od technických až po obchodne informácie.[2]

Pri zmene stavov týchto úloh (nová, vyriešená, uzatvorená a pod..) sú k nim ukladané rôzne anotácie. Tieto anotácie zahŕňajú atribúty, ako napr. popis alebo dokedy ma byť úloha vyriešená, spolu s históriou priradených riešiteľov a diskusie.[2]

Systém na sledovanie úloh môže okrem sledovania úloh slúžiť aj ako úložisko informácií. Okrem jednoduchého zoznamu spracovaných chýb, funkcionality a otázok, ukladá systém aj obrovské množstvo informácií dôležitých pre organizáciu, ktorá systém využíva.[2]

Postupom času sa v systéme na sledovanie úloh vybuduje zarážajúce množstvo informácií týkajúce sa podpory zákazníkov, nedokončené nápady pre nové funkcie a komunikácia k

týmto záležitostiam. Toto množstvo informácií je často využívané pre organizáciu aj pre vývojový tím na niekoľkých úrovniach.

Projektoví manažéri pri komunikácii či už so zákazníkmi alebo vývojovým tímom využívajú informácie zo systému na sledovanie úloh na podporenie svojich argumentov konkrétnymi údajmi.

Mnohé systémy na sledovanie úloh poskytujú integráciu s systémami na správu verzií (VCS), vďaka tomu môže vývojový tím lepšie porozumieť logike z zdrojového kódu, keďže je k nemu v systéme dostupné zadanie i dôvody, ktoré viedli ku konkrétnej implementácii.

Systém na sledovanie úloh môže slúžiť aj ako komunikačný a koordinačný uzol. S ohľadom na úlohy, ktoré zohráva systém na sledovanie úloh v rámci vývojového procesu, nie je prekvapivé, že podstatná časť komunikácie medzi členmi tímu je uložená v tomto systéme. Ak sa informácie zo všetkých stretnutí a dohodnuté priority a postup ukladá do systému v podobe komentárov k príslušným úlohám v systéme, poskytuje systém centralizovaný zdroj informácií uvedených v kontexte projektu. Táto vlastnosť býva uvádzaná ako najväčšie pozitívum používa systému.[2]

Excel

Ako doplnok k systémom na sledovanie úloh je často používaný Microsoft Excel. Tento nástroj je využívaný hlavne projektovými manažermi, ktorý takýmto spôsobom kompenzujú nedostatky používaného systému na sledovanie úloh.

Výhodou je pre nich Excel ako nástroj s ktorým denne pracujú. Zároveň je výsledný súbor priamo použiteľný pre ďalšie potreby projektového manažéra, ako napr. informovanie zákazníka o stave projektu.

Nevýhodou, ktorú si často neuvedomia je napr. nemožnosť súčasnej upravovania súboru viacerými používateľmi, chýbajúca história operácií.

Anotácie v zdrojovom kóde

Formálne spôsoby sledovania úloh za použitia systémov na sledovanie úloh, môžu byť dopĺňané aj neformálnymi spôsobmi sledovania úloh slúžiacimi viac pre potrebu vývojárov ako pre potrebu projektového manažéra. Jedným z týchto spôsobov sledovania sú anotácie v komentároch v zdrojovom kóde. [1]

Komentáre v zdrojovom kóde sú všeobecným typom anotácií k úlohám, kde vývojári vkladajú dokumentáciu priamo do zdrojového kódu. Slúžia na doplnenie informácie, ktorú nie je možné jednoduchým spôsobom zistiť zo zdrojového kódu, prípadne vysvetlenie neštandardného správania. Dostatočne okomentovaný kód je výrazne jednoduchší na pochopenie i zmenu. Pre vývojára jednoduchým spôsobom ako priradiť informáciu k časti zdrojového kódu je označenie textovou anotáciou.

Moderné integrované vývojové prostredia ako napr. Eclipse a Visual Studio, podporujú záložky v zdrojovom kóde a navigáciu v rámci anotácií v komentároch, ako sú napr. TODO, FIXME, a XXX. Eclipse navyše umožňuje definovať nové vlastné anotácie a má pre ne vlastnú perspektívu (špecifické usporiadanie editora, stromu projektu a dopĺňujúcich informácií).[1]

V rámci Eclipse existuje projekt Mylyn, ktorý integruje Eclipse so systémami na sledovanie úloh ako napr. Bugzilla. Umožňuje tým prácu so systémom na sledovanie úloh priamo z prostredia Eclipse.

Monitoruje aktivitu používateľa a na základe toho mu zobrazuje relevantné a pre používateľa potrebné informácie ako napr. úlohy v systéme na sledovanie úloh týkajúce práce otvoreného súboru so zdrojovým kódom. Aj pri použití Mylyn sa do kódu vkladajú anotácie, čo poukazuje na potrebu ich existencie a užitočnosť.[4]

Anotácie môže byť použité a sú používané vývojármi na rôzne účely. Používané sú napríklad na označenie jednotlivých častí väčších úloh.

Využívané sú vo väčšine prípadov iba na označenie krátkodobých úloh, a na dlhodobé úlohy sa takmer nevyužívajú. Veľké množstvo anotácií, akou je napr. TODO alebo FIXME je vnímané ako indikátor problémov s kvalitou zdrojového kódu, ktorý ešte vyžaduje veľké množstvo úprav a opráv. Používané sú napr. v prípade ak vývojár vie o prípade kedy sa jeho kód nemusí nutne správať podľa zadania, ale keďže takáto situácia nie je pravdepodobná, označí kód iba anotáciou, a spolieha sa na to, že v budúcnosti ak sa v tomto kóde chyba vyskytne tak mu táto anotácie napomôže opraviť chybu. TODO anotácia je najčastejšie využívaná v situácií, kedy vývojár aktuálne nemá čas sa venovať úlohe, preto ju označí anotáciou a venuje sa jej až na tom má čas. Používa sa aj na odloženie menej významných úloh, ktoré nie sú vyslovené nutné pre projekt, ale bolo by ich vhodné v budúcnosti implementovať.[1]

Anotácie sú používané viac ako napr. záložky v zdrojovom kóde nakoľko tieto po úprave kódu, už nie sú relevantné, pričom anotácie sa pri presunutí kódu presunú spolu s kódom.

Z výskumov vyplýva, že počet TODO anotácií klesá čím viac blíži ukončenie projektu, keďže úlohy odložené na neskôr označené TODO anotáciami sa riešia v závere projektu. Zistené bola skutočnosť, že čím sú úlohy zadané vývojárom menšie, tým menší je počet TODO anotácií.

Porovnanie

Výhody používania systému na sledovanie úloh

Zmyslom formálneho sledovania úloh je získavanie údajov o stave úloh v štruktúrovanej forme. Tieto údaje sú exaktné a sú použiteľné napr. ako referenčné údaje pre odhad a plánovanie. Systém na sledovanie úloh slúži v organizáciách aj ako úložisko histórie komunikácie medzi členmi tímu prípadne zákazníkmi, a informácie uložené v tomto systéme je možné ľahko vyhľadať a sú prístupné sú v historickom kontexte projektu. Pre projektových manažérov sú aj informácie o stave projekte a prípadne vyťažnosti jednotlivý členov tímu veľmi dôležité nakoľko sú potrebné pre správne a informované rozhodovanie sa a riadenie projektu.

Nevýhody používania systému na sledovanie úloh

Ak sa v organizácií používa systém na sledovanie úloh, je v záujme organizácie aby tento systém používali aj zákazníci.

Prvým riešením a aj najjednoduchším je podmieniť riešenie požiadaviek zákazníka zahrnutím tejto požiadavky do zmluvy o technickej podpore. Týmto krokom ale neriešime motiváciu zákazníka. Ak zákazník nerozumie, prečo by mal systém na sledovanie úloh používať, môžeme predpokladať, že aj v prípade, že ho používať bude výsledný efekt nebude taký pozitívny ako v prípade ak by rozumel významu používania systému.

Zo strany realizátora projektu existuje silná motivácia systém používať. Prináša im veľké množstvo informácií a funkcionality, pričom vstupnou investíciou je čas venovaný práci so systémom.

Zákazník ak predtým systém na sledovanie úloh nepoužíval ma z uvedených roli najmenšiu motiváciu tento systém používať. Je motivovaný krátkodobou výhodou priamej telefonickej a emailovej komunikácie, vďaka čomu ušetrí čas a spätná väzba je okamžitá rýchlejšia ako v prípade použitia systému na sledovanie úloh.

Aby sme získali kvalitatívne hodnotné príspevky od zákazníka, je potrebné zákazníka motivovať a presvedčiť o dlhodobých výhodách použitia formálneho sledovania úloh. Silným argumentom je možnosť si pre zákazníka kedykoľvek mať prehľad o sledovaní práce i bez priameho kontaktu s určenou osobou vo firme.

Správne používanie systému na sledovanie úloh zo strany zákazníka je podmienené správnym používaním systému a prípadne poskytnutím školenia zákazníkovi.

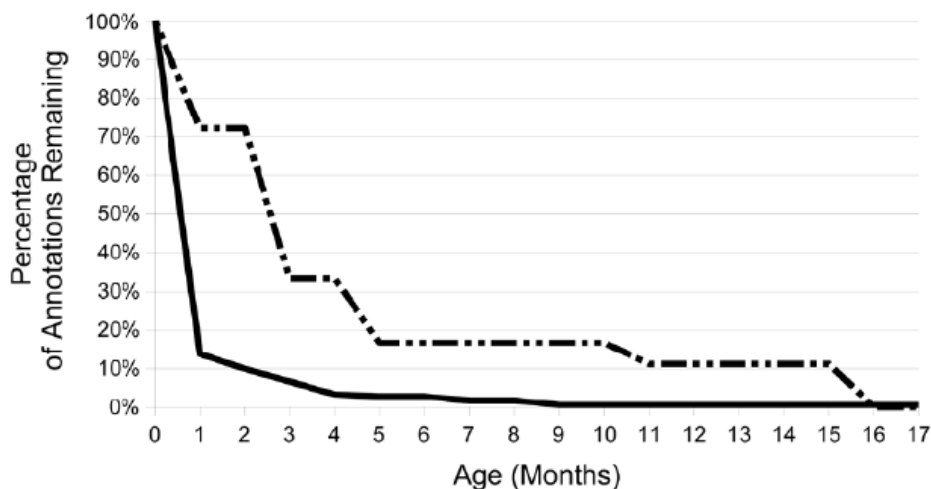
Výhody využívania anotácií

Výhodou je nízka náročnosť vytvorenia anotácie a oproti využívaniu systému na sledovanie úloh. V prvom prípade stačí jedna veta s anotáciou napr. //FIXME: fails with NPE . V druhom prípade zaberie oveľa viac času prihlásenie sa do systému, vyplnenie relevantných informácií a potrebný je aj podrobnejší opis chyby.

Ďalším dôvodom, prečo sú tieto anotácie používané je ich adresnosť. Keďže sú zadávané priamo v zdrojovom kóde, je na prvý pohľad jasné k čomu sa vzťahujú. Tým, že sú v zdrojovom kóde sú aj vo systéme na správu verzii a dostupná je aj ich história.

Nevýhody využívania anotácií

Keďže zadávanie anotácií je na dobrovoľnej báze, nie je možné na ich základe získavať presne informácie. Pre riadenie projektu majú tieto informácie iba orientačnú hodnotu, vieme z nich zhruba určiť v akom stave sa projekt nachádza, Je preukázané, že čím viac sa projekt blíži ukončeniu tým je počet anotácií menší ako to ilustruje graf na obrázku č.1.[1]



Obr. 1: Graf počtu anotácií

Problémom je aj to, že ako uvádzajú vývojári anotácie nie sú vždy aktualizované alebo odstránené po opravení problému. Podľa môjho názoru nie je možné riadiť projekt na základe týchto neformálnych anotácií nakoľko neposkytujú presne ani spoľahlivé informácie.

Záver

V eseji som sa zaoberal používaným formálneho a neformálneho sledovania úloh a dostupnými podpornými nástrojmi. Z vlastného pohľadu som opísal ich výhody a nevýhody. Podľa môjho názoru je formálneho sledovanie úloh nevyhnutné pre potreby riadenia projektu, nakoľko pre projekt je dôležité exaktne sledovať stav úloh nakoľko tieto informácie sú potrebné aj pre iné účely ako je napr. vyúčtovanie odpracovaných hodín, a pre potreby vedenia organizácie.

Neformálne sledovanie úloh použitím textových anotácií v zdrojovom kóde je v prvom rade robené vývojármi pre vývojárov a pre projektových manažérov nie sú potrebné. Môžu ale slúžiť ako kontrolný indikátor stavu projektu, ako príklad môžem uviesť, stav kedy je projekt formálne dokončený ale zdrojové kódy obsahujú veľké množstvo TODO anotácií, ktoré svedčia o neukončenom/nedokončenom vývoji.

Neformálne sledovanie úloh teda má svoje miesto a má aj svojich používateľov pre ktorých je nápomocné, a teda je v záujme organizácie ho podporovať. Neslúži ako náhrada ale ako vhodný doplnok sledovania úloh, ktorý indikuje stav projektu z pohľadu vývojového tímu.

Použitá literatúra

1. Storey, M., Ryall, J., Bull, R., Myers, D., and Singer, J. 2008. TODO or to bug: exploring how task annotations play a role in the work practices of software developers. In Proceedings of the 30th international Conference on Software Engineering (Leipzig, Germany, May 10 - 18, 2008). ICSE '08. ACM, New York, NY, 251-260.
2. Bertram, D., Volda, A., Greenberg, S., and Walker, R. 2010. Communication, collaboration, and bugs: the social nature of issue tracking in small, collocated teams. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (Savannah, Georgia, USA, February 06 - 10, 2010). CSCW '10. ACM, New York, NY, 291-300..
3. Jurison, J. 1999. Software project management: the manager's view. Commun. AIS 2, 3es (Nov. 1999), 2.
4. Mylyn, Project Description. Dostupné na internete. http://www.eclipse.org/projects/project_summary.php?projectid=tools.mylyn [cit:2010-Október]

Annotation

Issue tracking in software projects

Software development has it's specifics. No industry such a low unsuccessful project rate (studies state that the number of unsuccessful projects over 50%) and only a small portion of projects are finish on time and budget. The figures could improve by using issue tracking systems. They support better status updates on development progress, support capacity planning and make plan re-scheduling easier. I'm focusing on comparing the formal task management using issue tracking systems and informal task using text annotations in source code. I analyzed the relationship between formal and informal task tracking