

Podpora riadenia softvérového projektu pomocou skúmania softvérových skladísk

JÁN GREGOR

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
Jan.gregor[at]chello.sk*

Abstrakt. Dátové skladiská, ako napríklad systémy pre správu verzií zdrojových kódov (CVS), sú v dnešnej dobe nevyhnutnou súčasťou vývoja softvérových projektov. Tieto systémy zaznamenávajú informácie súvisiace s históriou vývoja projektov, ako napríklad informácie týkajúce sa refaktorizácie zdrojových kódov. Pri skúmaní týchto údajov v dostatočne veľkom množstve je možno s použitím vhodných nástrojov zúžitkovať tieto informácie pri vývoji projektu, a tak poskytnúť užitočné informácie nielen samotným vývojárom, ktorým môže pomôcť napríklad predvídaním chýb v zdrojových kódoch, ale aj manažérom projektov, ktorí môžu efektívnejšie korigovať smerovanie projektu. Tento dokument sa zaoberá skúmaním nástrojov a metód, ktoré efektívne využívajú tieto informácie a pomocou nich poskytujú užitočný podporný prostriedok pri vývoji softvérového projektu.

Úvod

Dátové skladiská, ako napríklad systémy pre správu verzií zdrojových kódov (CVS) obsahujú množstvo užitočných informácií o historickom vývoji projektu. Veľkosť množiny informácií, ktoré nám tieto systémy poskytujú, dáva príležitosť pre výskum a vývoj nástrojov pre ich spracovanie a využitie v rôznych etapách vývoja softvérového projektu.

Historické informácie získané z týchto systémov obsahujú množstvo poznatkov týkajúcich sa vývoja projektu. Pomocou nich je možné získať prehľad o detailnom vývoji projektu. Využitie týchto informácií je veľmi široké a v súčasnosti existuje viacero nástrojov, ktoré využívajú práve tieto informácie. Tieto nástroje dokážu pomocou vhodnej interpretácie týchto údajov pomáhať nielen vývojárom, ale aj projektovým manažérom pri vývoji projektu. Tento dokument sa zaoberá možnosťami využitia týchto informácií zo systémov CVS a predstavením nástrojov pre podporu softvérových projektov, ktoré tieto informácie využívajú.

Získavanie informácií zo softvérových skladísk

Pred samotnou analýzou možností využitia informácií zo softvérových skladísk je dôležité objasniť spôsob získavania informácií z týchto systémov.

Informácie obsiahnuté v systémoch v CVS sú statického charakteru a v textovej forme. Pre prácu s týmito údajmi je vhodné ich pred samotnou analýzou vhodne extrahovať. Možností extrakcie týchto informácií je viacero, v súčasnosti existuje viacero nástrojov, ktoré túto extrakciu vedú. Ako príklad uvediem nástroj *C-REX* [1], využívaný pri výskume informácií získaných z CVS. Tento nástroj skúma historické záznamy z CVS a zhromažďuje informácie o zmenách v zdrojových kódach až po úroveň zmien v jednotlivých metódach a premenných. Takto získané informácie sú následne pripravené pre ďalšiu analýzu a spracovanie.

Využitie informácií získaných zo softvérových skladísk

Informácie získané zo systémov CVS a nástroje pracujúce s týmito údajmi poskytujú podporu pre rôzne činnosti v rámci softvérového projektu. Z hľadiska cieľového používateľa daného nástroja môžeme preto rozdeliť tieto nástroje na dve hlavné skupiny:

- Podporné nástroje určené pre vývojárov
- Podporné nástroje určené pre manažérov

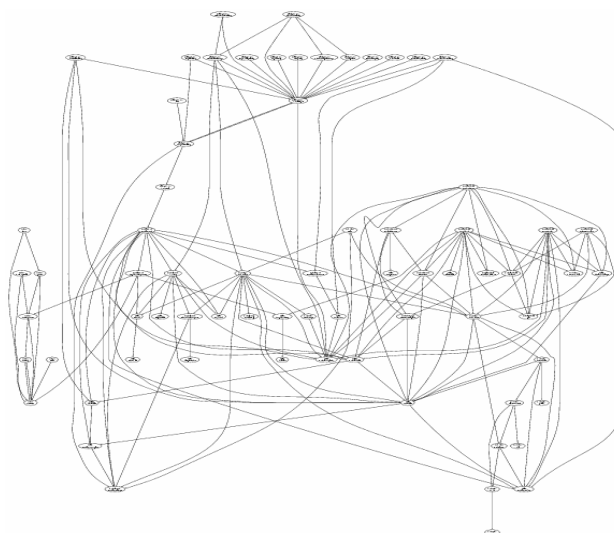
Podporné nástroje určené pre vývojárov

Pri vývoji softvérového systému sa veľmi často stáva, že sa vývojár dostane k časti projektu, ktorej architektúru celkom presne nepozná a z dôvodu časového ohraničenia musí v projekte uskutočniť zmeny aj s minimálnou znalosťou činnosti a štruktúry danej časti. Takéto zmeny často prinášajú výskyt nekorektných zásahov do kódu, a tým pádom poskytujú priestor pre vznik nových chýb.

Pre orientáciu v rozsiahlych projektoch boli vytvorené viaceré techniky, umožňujúce lepšie pochopenie architektúry veľkých systémov.

Základným prostriedkom, ktorý systémy CVS priamo poskytujú je uchovávanie historických verzií jednotlivých zdrojových kódov. Takto je možné sledovať postupný vývoj jednotlivých častí projektu. Pre začínajúcich vývojárov, ktorí sa potrebujú v novom projekte zorientovať je študovanie aktuálnych zdrojových kódov veľmi obtiažne a veľaokrát aj nemožné. Preto je vhodné študovať prvotné verzie zdrojových kódov, z ktorých je možné jednoducho pochopiť základnú architektúru systému

a nezaťažovať sa jednotlivými rozšíreniami, ktoré v systéme pribudli počas vývoja. Ďalším spôsobom, ako pomôcť vývojárom lepšie pochopiť architektúru softvérových systémov sú aj *grafy závislostí*. Tieto zobrazujú vzťahy medzi jednotlivými časťami systémov a súvislosti medzi komponentmi. Príklad takéhoto grafu je možné vidieť na Obr.1[2] , ktorý zobrazuje graf závislosti vygenerovaný systémom *TortoiseCVS*.



Obr. 1. Graf závislosti generovaný systémom TortoiseCVS

Zmieňované grafy ale neumožňujú zobrazovať informácie týkajúce sa jednotlivých komponentov, ktoré by mohli pomôcť lepšie spoznať danú architektúru. Vhodným zdrojom pre doplnenie týchto informácií sú práve softvérové skladiská. Týmto spojením je možné dosiahnuť vhodný prostriedok pre pochopenie architektúry rozsiahlych softvérových systémov. Takýto prostriedok predstavil aj Ahmed E. Hassan. Informácie získané zo systému CVS nazval *Source Sticky Notes*[1]. Zaujímavým faktom je, že daný nástroj autor priamo demonštroval na architektúre operačného systému *NetBSD*.

Použitie spomínaných podporných prostriedkov prináša nepochybne zefektívnenie procesu zoznamovania sa vývojárov s novými projektmi. Ich správnym využitím a vhodnou reprezentáciou dôležitých informácií spätých s projektom, je možné predchádzať chybám, ktoré vývojári môžu spôsobiť nedostatočnou znalosťou systémovej architektúry. Nasadenie takýchto prostriedkov má podľa môjho názoru najväčší význam pri rozsiahlejších projektoch, kedy nie je architektúra celkom jasná z dostupnej dokumentácie a je potreba vizualizovať celý systém. Pri skúmaní projektov menšieho rozsahu by sa mali vývojári sústrediť na prehľadávanie CVS systémov, kde nájdu prvotné verzie vyvíjaných projektov. V týchto verziách sa väčšinou nachádzajú

hlavné kostry programov, ktorých funkcie sa dajú tak veľmi ľahko pochopiť a nie je nutné hľadania jadra systému v spleti zložitých modulov a prídavných funkcií.

Podporné nástroje určené pre manažérov

Úlohou manažérov v softvérových projektoch je efektívne využiť ľudské zdroje, ktoré sú na projekt pridelené, sledovať a riadiť vývoj projektu a snažiť sa o správne smerovanie vývoja s predvídaním možných chýb počas vývoja.

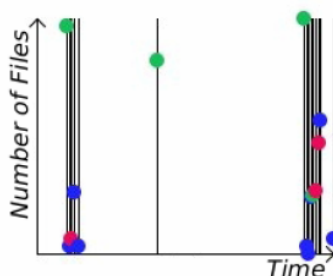
Bežnou súčasťou vývoja softvérového projektu je aj výskyt chýb. Tieto chyby môžu byť rôzneho pôvodu a nachádzať sa v rôznych častiach projektu, aj keď sa jedná o jeden typ chyby. Oprava týchto chýb je často rozsiahla činnosť a veľakrát časovo náročná. Záznamy o oprave týchto chýb sa taktiež zaznamenávajú do systému CVS, a tak je možné získať ich vhodným spracovaním ucelený prehľad o tom, aké typy chýb sa vyskytovali v zdrojových kódach a zároveň ako, kde a kedy boli opravované. Spracovaním informácií týkajúcich sa opráv chýb v zdrojových kódach sa zaoberajú aj Chadd C. Williams a Jeffrey K. Hollingsworth, ktorý vyvinuli nástroj pre detekciu chýb pomocou získavania historických dát zo systémov CVS[3]. Týmto nástrojom sa snažia zlepšiť techniky vyhľadávania chýb, ktoré sú realizované napríklad statickou analýzou zdrojových kódov. Tento nástroj zhromažďuje historické informácie o zmenách vykonaných v zdrojových kódach týkajúcich sa opravy chýb. Jednotlivé chyby, ktoré sú v zdrojových kódach, sú identifikované na základe poznatkov, či sa hodnoty funkcií, resp. referencií testujú predtým, ako sa použijú. Kombináciou týchto dvoch zdrojov informácií je tak možné predvídať chyby, ktoré sa v projekte nachádzajú, resp. môžu sa v budúcnosti nachádzať. Tento nástroj bol nasadený na CVS systémy projektov *Apache Web Server* a *Wine*, kde preukázal veľmi dobré výsledky oproti použitiu statickej analýzy pre vyhľadávanie chýb. Použitie tohto nástroja umožňuje manažérom efektívne predvídať chyby vo vyvíjanom softvérovom projekte a sústrediť dostatok prostriedkov na vývoj takých častí systému, pri ktorých je najväčšia pravdepodobnosť vzniku chýb. Tento poznatok sa dá aplikovať aj na fázu testovania, kde tento nástroj pomôže manažérovi získať predstavu o optimálnom rozdelení ľudských zdrojov pre testovanie na základe výskytu chýb v jednotlivých častiach projektu.

Pri vývoji softvérových projektov je nutné, aby jednotliví vývojári spolu komunikovali, keďže časti projektu, ktoré sú pridelené jednotlivým členom tímu spolu súvisia a kooperujú. Správna komunikácia medzi členmi tímu je nutnou podmienkou pre vývoj projektu, na ktorom sa podieľa viacero vývojárov, pretože iba správnym porozumením súvislosti medzi jednotlivými časťami je možné vyvíjať samostatné časti tak, aby v konečnom dôsledku zapadli do celkovej architektúry a vedeli správne fungovať. Kontrola tímovej práce jednotlivých členov je ďalšou úlohou manažérov, ktorú by mali spĺňať, pokiaľ chcú aby sa vývoj projektu uberal správnym smerom. Manažér potrebuje vedieť, či sa vývoj projektu práca na ňom realizuje podľa očakávaní. Teda napríklad potrebuje vedieť, či pri vývoji existuje hlavný vývojár

a skupina pomocných vývojárov, alebo každý člen tímu pracuje samostatne na svojej časti, alebo je práca členov na projekte spoločná. Na podporu sledovania práce jednotlivých členov tímu na spoločnom projekte existujú opäť podporné nástroje. Tieto nástroje využívajú informácie o autoroch zmien, ktoré sa v zdrojových kódach udiali, a spracovaním týchto údajov poskytujú prehľad o zainteresovanosti jednotlivých členov v jednotlivých častiach projektu. Vhodnou interpretáciou týchto informácií je ich vizualizácia. Existuje viacero techník pre vizualizáciu týchto údajov. Peter Weißgerber, Mathias Pohl a Michael Burch rozlišujú tri základné typy vizualizácií týchto údajov [4]:

- Prehľad transakcií
- Matica Autor - Súbor
- Dynamický graf Autor - Súbor

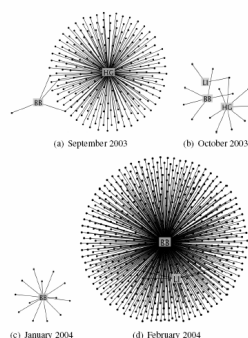
Prehľad transakcií je vizualizačná technika zobrazujúca počet súborov, v ktorých sa udiali zmeny a autorov, ktorí tieto zmeny vykonali. Príklad tejto vizualizačnej techniky je možné vidieť na Obr.2.



Obr. 2. Ukážka vizualizačnej techniky Prehľad transakcií [4].

Matica Autor - Súbor zobrazuje vzťah súborov projektu a množstva zmien na týchto projektoch od jednotlivých vývojárov. V jednotlivých súboroch sú graficky zobrazené množstvá zmien, ktoré autor na daných súboroch vykonal.

Dynamický graf Autor – Súbor zobrazuje informácie o súvislosti autorov a zdrojových súboroch, ktoré boli nimi menené za isté časové obdobie. Tieto súvislosti sú reprezentované grafmi, kde jednotlivé uzly tvoria súbory a ich autori. Príklad použitia takejto vizualizačnej techniky je vidieť na Obr.3.



Obr. 3. Ukážka vizualizačnej techniky Dynamický graf Autor – Súbor [4].

Informácie o práci jednotlivých členov tímu na spoločnom projekte sa dajú zúžitkovať aj pri vývoji *open-source* projektov. Pri vývoji takéhoto typu systémov je situácia mierne odlišná od práce na komerčnom projekte v rámci tímu, ktorý zastrešuje jedna softvérová spoločnosť. Pri vývoji takéhoto softvéru je do projektu zapojených mnoho vývojárov, ktorí sa môžu nachádzať na akomkoľvek mieste na svete, a tak je tímová práca a identifikácia jednotlivých úloh vývojárov veľmi obtiažna. Touto problematikou sa zaoberajú napríklad aj Ligu Yu a Srimi Ramaswamy. V ich práci poukázali na potrebu identifikácie úloh jednotlivých členov tímu [5]. Táto identifikácia sa dá realizovať prehľadávaním systémov CVS a zhromažďovaním informácií o práci jednotlivých členov. Takýmto spôsobom je možné po istom čase vývoja pomerne presne určiť pozície jednotlivých vývojárov a taktiež lokalizovať jadro vývojového tímu a skupiny externých vývojárov.

Pomocou predstavených vizualizácií je možné v prehľadnej forme získať prehľad o jednotlivých členoch tímu a ich práci na projekte. V prípade problémov s vývojom projektu je možné na základe týchto informácií zmeniť pridelenie úloh na projekte, resp. zmeniť spôsob práce členov tímu na samostatnú, resp. tímovú, a tak zefektívniť vývoj projektu a v konečnom dôsledku tak podporiť správne smerovanie projektu do úspešného konca. V prípade vývoja *open-source* projektov predstavujú tieto techniky veľmi efektívny nástroj na identifikáciu pozícií jednotlivých členov, a dopomáhajú k lepšej koordinácii práce. Myslím si, že pri vývoji rozsiahlych projektov takéhoto typu by mala byť existencia takéhoto podporného prostriedku samozrejmosťou, keďže geografické rozloženie jednotlivých členov tímu, resp. skupín vývojárov môže byť skutočne veľmi rozdielne. V takýchto prípadoch je identifikácia pozícií členov a informácia o ich vzájomnej spolupráci neoceniteľná.

Záver

Skúmanie softvérových skladísk má nesporne veľký význam pre podporu softvérového projektu. V tomto dokumente som sa snažil predstaviť možnosti spracovania informácií, získané z týchto skladísk a ich využitie v nástrojoch na podporu riadenia. Či už sa jedná o podporu vývojárov pri štúdiu rozsiahlych projektov, grafickú vizualizáciu komponentov veľkých softvérových systémov, alebo podporu manažérov pri predvídaní chýb v projektoch a rozdeľovaní ľudských zdrojov pre jednotlivé časti projektu, či kontrolu a prehľad práce jednotlivých členov tímu na projekte, poskytujú informácie získané zo softvérových skladísk nenahraditeľné informácie pre podporu každej spomínanej činnosti. Predstavené existujúce príklady využitia sú dostatočnou motiváciou pre vytváranie nových nástrojov, ktoré budú ťažiť z vysokej informačnej hodnoty týchto dát a prinesú tak nové podporné prostriedky pre efektívnejší vývoj softvérových systémov.

Použitá literatúra

1. Hassan, E. Ahmed : Mining Software Repositories to Assist Developers and Support Managers. In: *22nd IEEE International Conference on Software Maintenance (ICSM'06)*, (2007) pp. 339-342.
2. Irving, F. : C/C++ Dependency graphing, (2005)[online]
URL: < <http://flourish.org/cinclude2dot> > [Posledný prístup 13 december 2005]
3. Williams, Chad.C, Williams, C., Hollingsworth : Automatic Mining of Source Code Repositories to Improve Bug Finding Techniques. In : *IEEE Transactions on Software Engineering*, (2005) pp. 466-480
4. Weissberg, P., Pohl, M., Burch, M.: Visual Data Mining in Software Archives to Detect How Developers Work Together. In: *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)* pp. 9
5. Yu, L. , Ramaswamy, S. : Mining CVS Repositories to Understand Open-Source Project Developer Roles. In : *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)* pp. 8

Annotation

Mining software repositories as a software project support system

Software repositories (e.g. CVS) are nowadays indispensable part by developing software projects. These systems are tracking informations regarding the evolutionary history of software projects ,e.g. informations related to refactorizing source codes. By studying sufficiently amount of these informations, with use of suitable tools, is possible to use them for project developement and give useful informations to developers ,e.g by foresight bugs in source codes and also to project managers by helping them to manage project more efficiently. This document presents tools and methods, which use effeciently these informations and offer useful support tools by software project developement.