

Monitorovanie otvorených softvérových projektov

MARIÁN BYSTRICAN

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

Abstrakt. Vývoj počítačového softvéru je nesporne veľmi zložitý a časovo náročný proces, ktorý si vyžaduje kvalitný manažment. Existujú rôzne typy softvérových projektov, tie sa od seba líšia svojou zložitosťou, rozsiahlosťou, zúčastnenými pracovníkmi a samotným charakterom. Bez ohľadu na tieto skutočnosti, ak chceme dospieť k úspešnému ukončeniu a uvedeniu do prevádzky, je potrebné neustále monitorovať prebiehajúce práce počas každej fázy vývoja. Takto je možné sledovať, či napredujeme správnym smerom a ak nie, včas situáciu zachrániť a ušetriť zdroje, ktoré by boli zbytočne premrhané. Špecifickým prípadom je vývoj aplikácií s otvoreným zdrojovým kódom, kde na projektoch pracujú členovia prislúchajúcej komunity, mnohokrát dobrovoľne a na vlastné náklady, pričom môžu byť rozmiestení v rôznych častiach sveta. V takomto prípade nie je možné použiť štandardné metódy riadenia a monitorovania projektu a treba zvoliť špeciálne prístupy, ktorými sa zaoberá táto esej.

Úvod

Ľudstvo dnes žije v 21. storočí, posledné roky zvyknú byť nazývané aj informačným vekom. Ekonomiky štátov sa spájajú, prebiehajú procesy integrácie, automatizácie technologických i netechnologických procesov, majitelia firiem a manažéri čoraz väčšmi apelujú na efektivitu a najmä výsledky práce, na ktorú ich zamestnanci, podriadení a napokon i oni sami vynaložili nemalé úsilie.

Tieto procesy prebiehajú aj vďaka pokroku informačných technológií. Kedysi bol hardvér neporovnateľne drahší oproti softvéru, no teraz je to skôr naopak a pri vývoji rozsiahlych systémov sa často na hardvér hľadí až ako na druhoradú vec. Rôzny špecifický softvér je dnes žiaduci, vzniká obrovské množstvo firiem zaoberajúcich sa jeho vývojom, takže už prakticky hovoríme o softvérovom priemysle.

Vývoj softvéru je veľmi zložitý proces, ktorý si vyžaduje kvalitný manažment. Z pohľadu projektového manažéra je to kompozícia hierarchicky usporiadaných, čiastočne logicky nezávislých a časovo nadväzujúcich vykonávaných úloh, ktoré sa môžu začať iba v prípade, keď je k dispozícii dostatok kvalifikovaných zdrojov.

Manažment projektov softvérových a informačných systémov, október 2008, s. 1-7.

Aby práca prebiehala plynule, aby sme mohli povedať kde v procese napredovania sme a v súvislosti s tým zabezpečiť zdroje na to, aby bola ukončená do určených termínov, v horšom prípade včas odhaliť, že projekt je ohrozený novými rizikami, je neoddeliteľnou súčasťou vývoja aj monitorovanie všetkých aktivít a následné vyhodnocovanie zozbieraných informácií a prípadné úpravy plánu.

Bežné metódy monitorovania

Bežný postup monitorovania a kontrolovania v organizáciách je založený na reportovaní ľuďmi (angl. human-based reporting). V pravidelných časových intervaloch sa uskutočňujú porady vedenia, operatívne tímové porady, kde sa spoločne preberajú jednotlivé úlohy, kontroluje sa ich stav, riešia sa prípadné zistené problémy. Na základe výsledku porád môžeme zistiť, či napredujeme k cieľu podľa plánu a robiť správne strategické rozhodnutia. Aj tu však existuje riziko, že zúčastnení majú tendenciu utajovať niektoré skutočnosti alebo neobjektívne priznať problémy, čo môže v istých prípadoch predstavovať riziko.

Fakt, že nad projektom strávime veľa hodín a investujeme do neho nemalé finančné prostriedky, nemá nič spoločné s odpoveďou na otázku, či v skutočnosti napredujeme ďalej. Zaujímavý výklad výroku Fredericka Brooksa „Nosenie dieťaťa trvá presne deväť mesiacov, bez ohľadu na to, koľko žien je na to určených.“ prenesený na tému napredovania vo vývoji podal Bas de Baar v článku pre portál searchsoftwarequality.com [1]: zmysel výroku by sme podľa neho mohli premietnuť do softvérového projektu na človeko-mesiace (mesiace odpracované jedným človekom). Povedzme, že projekt bude odhadom trvať tri človeko-mesiace. Postupom času sa však tvrdenie skrúti na „tri mesiace“ či už v zápisoch, alebo v myšliach zúčastnených osôb. Keď vedúci projektu na prípadnú požiadavku skoršieho ukončenia zareaguje odpoveďou „v poriadku, zapojíme viac pracovníkov“, vzniká problém: nie všetky úlohy sa dajú rozdeliť, napríklad kvôli postupnej nadväznosti alebo sa to jednoducho nedá kvôli elementárnosti, potom nie je možné pracovať nepretržite, navyše noví ľudia sa musia pred vzhupnutím do projektu aj niečo o ňom naučiť, zvýši sa čas strávený komunikáciou. Preto človeko-mesiace sú tu síce dobrým ukazovateľom finančným (ak poznáme, koľko nás stojí jeden človek, vieme si jednoducho zrátať, koľko nás bude stáť celý projekt), no nehovoria ako ďaleko sme od cieľa.

Na monitorovanie softvérových projektov a meranie progresu sa používajú v súčasnosti rôzne mierky: zarobená hodnota, nákladové rozpočty, Ganttové diagramy, diagramy PERT, registre rizík, sieťové diagramy, body za funkčnosť, počet riadkov zdrojového kódu, počet chýb, body za prípady použitia, zložitosť podmienok (angl. conditional complexity), čas potrebný na opravu (angl. time to repair), čas do zlyhania (angl. time to fail). Ako je každý projekt jedinečný, rovnako ako pre každú zúčastnenú skupinu sú najviac relevantné iné údaje, tak sa i tieto metódy používajú buď samostatne, alebo v rôznych kombináciách, podľa ich vhodnosti v danej situácii.

Otvorený softvér

Špecifickým prípadom je vývoj softvéru s otvoreným zdrojovým kódom. Na takýchto projektoch zvykne pracovať množstvo ľudí, často z vlastnej vôle a bez nároku na honorár, majú k dispozícii rozdielne veľa času. Najväčším problémom však je, že vývojári sú rozmiestnení v rôznych krajinách sveta a rôznych časových pásmach, čo takmer vylučuje akékoľvek stretnutie v reálnom čase. V tomto prípade nemôžeme mnohokrát využiť klasické metódy, ktoré sa bežne používajú v spoločnostiach alebo softvérových domoch.

Pre otvorené projekty je tiež príznačné, že z hľadiska návrhu, plánovania a riadenia sú viac neformálne. Aj keď sa tu môže zdať vývoj neorganizovaný, nie je to celkom tak a hlavne existujú oprávnené výhody. Rápidny vývoj a masívne hodnotenie komunitou, flexibilita v používaní a zmene zdrojového kódu, nízkonákladovosť, preberanie práce druhých a s tým súvisiaca motivácia k vytváraniu štandardov, sú niektorými z mnohých príkladov.

Existujú viaceré organizácie, ktoré podporujú takýto vývoj, niektoré si vybrali len jeden alebo niekoľko menej konkrétnych projektov. Sú tu aj organizácie snažiace sa v globále podporovať a zlepšovať vývoj otvorených projektov, na čele stojí Nadácia otvoreného softvéru (angl. Open Software Foundation, OSF). Jednou z nich je aj Softvérová nadácia Apache (Apache Software Foundation, ASF). Do jej inkubačného systému sa môže zapojiť hocikákoľvek otvorený softvér, ktorý žiada o financie. ASF má záujem vytvárať komunity a ich prácu na takýchto projektoch podporovať, no musí sa jednať o projekt, ktorý má potenciál byť úspešný. Práve na zistenie tejto skutočnosti je vhodné projekt najprv istý čas monitorovať a až potom vykonať kroky vedúce k pomoci. Rovnako aj bežiacie projekty môžu byť počas vývoja ohrozené rôznymi rizikami, napríklad nedostatkom alebo vystúpením kľúčových vývojárov, čo môže mať za následok stratenie viery a motivácie prispievateľov a komunity. Situácia, kedy bol projekt v minulosti úspešný, ale cieľový trh sa zmenil a o špecifickú technológiu nemusí byť ďalej záujem, tiež neprospeva motivácii vývojárov. Otázka znie: ako máme efektívne monitorovať otvorený vývoj a odhaliť rizikové momenty, keď sa bežné metódy použiť nedajú?

Monitorovanie založené na udalostiach

V nasledujúcich riadkoch si povieme niečo o udalostných systémoch a ako by sa dali využiť pri monitorovaní vývoja práve otvoreného softvéru.

Udalostný systém

Komunita okolo distribuovaných udalostných systémov definuje udalostný systém (systém založený na udalostiach, SZU) ako „systém, v ktorom producenti odosielajú udalosti a kde pomocné aplikácie stojace uprostred (angl. messaging middleware) doručujú udalosti konzumentom s ohľadom na ich vopred špecifikované záujmy“ [2].

Typické použitie takýchto systémov predstavuje paradigma „zverejni a odoberaj“ (angl. publish and subscribe), kde sa producenti a konzumenti navzájom nepoznajú.

Konzumenti sa každý samostatne, a nezávisle zaregistrujú s tým, že ich zaujíma špecifická udalosť alebo skupina udalostí. Producenti udalosti potom odosielajú do tém, systém ich prijme a včas spropaguje a doručí záujemcom. Oproti tradičnej paradigme „požiadavka/odpoveď“ (angl. request/reply) je tu výhoda v tom, že komunikácia môže byť anonymná, účastníci nemusia byť dostupný v rovnakom momente a nenastáva tu žiadne blokovanie, takže netreba kontrolovať tok informácií.

Udalostné systémy sú široko používané na integráciu voľne previazaných komponentov, vrátane senzorov, ovládačov zariadení a databáz. Vďaka záujmu sa použitie komerčne rozširuje a zlepšuje sa výkon a jednoduchosť používania takýchto systémov.

Vývoj otvoreného softvéru ako udalostný systém

Proces vývoja v softvérovom projekte s otvoreným zdrojovým kódom môže byť tiež modelovaný ako multi-agentový systém založený na udalostiach: v tomto modeli sú agentmi účastníci projektu a ich interakcie a zmeny stavu sú udalosti.

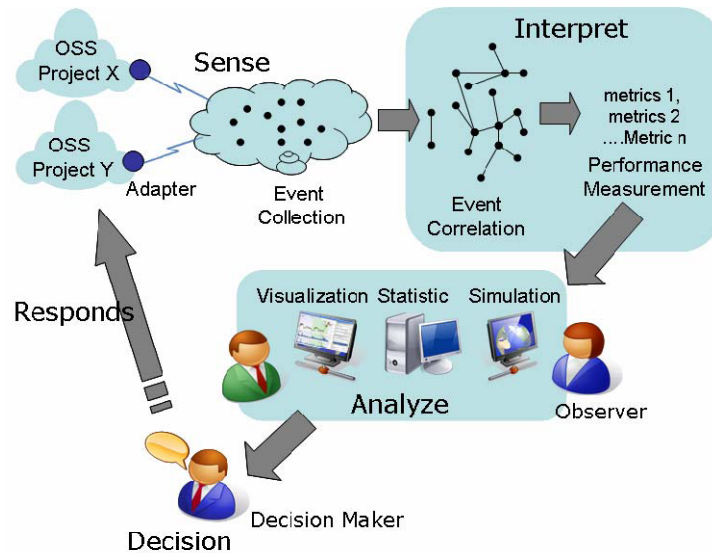
Počas samotného procesu vývoja agenti interagujú a menia stav z jedného na druhý. Môžu byť v roli producentov, ktorí publikujú udalosti prostredníctvom pomocných softvérov (mailing list, bug tracker, SVN), doručujúcich udalosti ostatným agentom. Títo zas vystupujú ako konzumenti/abonentí podľa vopred upresnených záujmov. Keďže väčšina účastníkov komunikácie sa navzájom nepozná, pochádzajú z rôznych častí sveta, časových zón a používajú rôzne technológie a komunikačné rozhrania, správy sú najčastejšie posielané podľa nasledujúcej schémy:

Keď si vezmeme ako príklad systém na zber a manažment informácií o chybách (angl. bug tracker), používateľ alebo vývojár, ktorý nájde v softvéri nedostatok, môže byť považovaný za producenta, ktorý posiela správu o existencii chyby do systému (udalosť: zaevidovanie chyby). Systém po uskutočnení potrebných interných operácií na zaevidovanie a vyhodnotenie vyšle informáciu o novej chybe napríklad prostredníctvom elektronickej pošty (mailing listu, udalosť: notifikácia o chybe). Neskôr niekto z príjemcov notifikačných správ o chybách túto zanalyzuje, vyhodnotí a pošle výsledok systému (udalosť: analýza chyby).

Spomínané dáta sa dajú po vhodnom spracovaní použiť na získanie informácií o tom, ako projekt pokračuje. Monitorovací systém zaznamenáva zmeny stavov a zbiera jednotlivé udalosti, aby ich mohol neskôr zanalyzovať a vyhodnotiť, premietnuť do informácií vypovedajúcich niečo ľuďom, ktorí sú zodpovední za rozhodovanie.

Na Obr. 1 je definovaný model, ktorý rozlišuje dve triedy používateľov. Prvou sú pozorovatelia, ktorí sa zaujímajú o projekt, lebo si napríklad chcú rozšíriť vedomosti o procese vývoja alebo zistiť, či projekt za niečo stojí. Druhou skupinou sú ľudia podobní, no tí navyše zodpovedajú za strategické rozhodnutia, príkladom môžu byť IT manažéri, ktorí práve vyberajú biznis riešenie pre svoju firmu a potrebujú zistiť, či je projekt dostatočne stabilný, živý, či vývojári pružne reagujú na chyby. Tieto informácie sú podobne využiteľné aj pre projektových manažérov v spomínanej nadácii Apache na zistenie, či sa projekt nestáva rizikovým a ak áno, v správnej chvíli

reagovať, aby neupadol a zotavil sa. Na Obr. 1 je prvá skupina nazývaná anglickým názvom observer, druhá zas decision maker.



Obr. 1. Udalostný systém monitorovania projektov [4].

Indikátory stavu

Otvorené projekty poskytujú mnoho udalostí a artefaktov, ktoré možno sledovať počas vývoja. Tieto môžu slúžiť ako indikátory stavu, vďaka čomu vedú vypovedať, či je projekt v dobrom stave alebo hlbokých problémoch. Niektoré z nich, ktoré sa úspešne a rutinne používajú sú nasledovné:

- otvorené problémy, oneskorenie služieb: chyby a nedostatky sú zaznamenané v systéme na zaznamenávanie chýb, ale potrebné opravy nie sú uskutočnené v primeranom čase
- proporcie: počítame proporcie elementov ako množstvo príspevkov v mailing liste, chýb za sledované obdobie, aktualizácií v repozitároch zdrojových kódov; tieto hodnoty použijeme na porovnanie projektov, aby sme sa zistili, aký je vhodný vzťah, pomer medzi týmito ukazovateľmi
- aktivita a sila komunity: indikuje, či má projekt dynamickú komunitu, napríklad počet stiahnutí v porovnaní s počtom príspevkov v mailing liste; počet aktívnych pokročilých používateľov (ktorí pomáhajú ostatným) v mailing liste a intenzita ich aktivity, interakcia vývojárov v (rozdielnych) mailing listoch.

Ako príklad monitorovacieho procesu si uveďme sledovanie prispievania vývojárov v mailing liste z pohľadu projektového manažera, chceme teda zistiť intenzitu komunikácie. Sledovanou veličinou bude priemerný počet emailov za mesiac,

graf z týchto dát pozorovateľovi ukáže trend prispievania vývojárov. Pozitívna tendencia nahor indikuje, že sú aktívni a majú ochotu venovať sa projektu. Ďalej budeme sledovať percento výskytov dolného limitu počtu emailov za mesiac, čo nás včas varuje o znížení aktivity.

Pri monitorovaní založenom na udalostiach je hlavnou úlohou identifikovať relevantné udalosti. V našom prípade je prvou udalosťou prichádzajúci email, kedy sa email zaznamená do zoznamu príspevkov. Na konci mesiaca spustí systém skript, ktorý vytvorí štatistiku počtu poštových správ a vyhodnotí ich počet. Povedzme, že máme tri stavy vývoja: normálny, rizikový a kritický, nech je kritickou hranicou 70 emailov za mesiac. Ak sme dostatočne vysoko nad touto hranicou, považujeme vývoj projektu za normálny, keď počet klesne napríklad 4-krát počas šiestich mesiacov tesne nad hranicu 70, je stav rizikový a po klesnutí pod túto hranicu máme seriózny problém. Aby sme zabezpečili včasné upozornenie, môžeme definovať ďalšiu udalosť: výstražný signál, ktorá bude reakciou na prichádzajúci mail a tá spustí podobný skript, ktorý však pošle upozornenie okamžite s tým, že sa pozrie na posledných 30 dní dozadu oddnes.

Na túto tému bola uskutočnená aj štúdia [3], ktorá porovnávala aktívne projekty s tými, ktoré upadali a zistilo sa, že s tých neaktívnych odišli kľúčoví vývojári, čo spôsobilo demotiváciu tímu a v priebehu šiestich mesiacov postupne klesala ich mailová aktivita. Zistenie svedčí o tom, že tieto ukazovatele môžu priniesť stručnú informáciu o stave projektu a skoro informovať o rizikovej situácii.

Záver

V tejto eseji sme sa zaoberali monitorovaním softvérových projektov a pozreli sme sa bližšie na projekty s otvoreným zdrojovým kódom, kde nie je ľahké sledovať vývoj pomocou bežných metód, ktoré sa používajú v pevne zviazaných formách organizácií (klasické softvérové firmy). V tomto prípade je vhodné použiť alternatívu, ktorá modeluje vývoj otvoreného projektu ako udalostný systém.

Oproti reportovaniu ľuďmi sú pri monitorovaní založenom na udalostiach informácie zbierané a vyhodnocované podpornými nástrojmi, pomocou ktorých sa dá proces priblížiť a priniesť informácie pozorovateľom a záujemcom zvonku. Tí vďaka tomu môžu lepšie porozumieť, v akom stave je v súčasnosti projekt a jeho komunita, a pochopiť, ktoré faktory a riziká ovplyvňujú jeho úspech.

Model založený na udalostiach a dáta zozbierané a vyhodnotené automaticky by sa dali použiť a mohli by byť užitočné i pri neotvorenom vývoji, hlavne tam, kde je u reportovaní ľuďmi podozrenie na nesystematickosť, nekompletnosť alebo nekonzistenciu.

Použitá literatúra

1. Baar, B.: Estimating actual project progress. 1.11.2008
http://searchsoftwarequality.techtarget.com/tip/0,289483,sid92_gci1293291,00.html

2. Final program of 4th International Workshop on Distributed Event-Based Systems (DEBS'05). 10.6.2005
<http://research.cs.queensu.ca/~dingel/debs05/>
3. Wahyudin D., Schatten, A., Mustofa, K., Biffi, S., Tjoa, A.: *Introducing Health Perspective In Open Source Web-Engineering Software Projects, Based On Project Data Analysis. The 8th International Conference on Information Integration and Web-based Applications & Services (IIWAS2006), Yogyakarta, Indonesia.* 2006.
4. Wahyudin, D., Tjoa, A.: *Event-Based Monitoring of Open Source Software Projects.* Second International Conference on Availability, Reliability and Security (ARES'07). 2007.

Annotation

Open source projects monitoring

Software development is admittedly very complicated and time consuming process which requires quality management. There are many types of software projects, they differ from each other by complexity, proportion, workers involved and character by himself. Regardless those facts, if we want to come to successful finish and implementation, it is required to constantly monitor tasks running during every phase of project. It is possible to watch if we move ahead right direction and if not, early retrieve the situation and save resources which may be uselessly wasted. Specific case is development of open source applications involving members of community, often working voluntarily and at own charge, they may be located all around the globe. In this case it is not possible to use standard methods of project controlling and monitoring and we need to choose special approaches.