

MONITOROVANIE SOFTVÉROVÉHO PROJEKTU – MNOŽSTVO MOŽNOSTÍ, NÁPADOV – ČO Z NICH POUŽIŤ?

*Nikto neurobil väčšiu chybu ako ten, kto neurobil nič
iba preto, že mohol urobiť iba trochu. (Edmund Burke)*

Milan Laslop

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
milan33[zavináč]gmail[.]com

Monitorovanie softvérového projektu je dôležitou súčasťou práce na projekte, ktorej predchádza proces analýzy a návrhu samotného spôsobu monitorovania. Okrem explicitného monitorovania, kedy vývojári sami tvoria informácie, ktoré sú hodnotené pri monitorovaní, sa používa aj implicitné, ktoré zahŕňa automatický zber dát, napríklad zo softvérových repozitárov. Bolo navrhnutých mnoho spôsobov monitorovania, ktoré sa väčšinou viažu na systémy s konkrétnou charakteristikou (napríklad systémy založené na komponentoch) alebo sa viažu na konkrétny spôsob vývoja (napríklad open source softvér).

Esej analyzuje obmedzenia niektorých zo študovaných spôsobov monitorovania a prináša zamyslenie sa nad týmito ohraničeniami a možnosťou nájsť širšie možnosti uplatnenia metód, dokonca až hľadaním ich prieniku v nejakom konkrétnom prípade. Snaží sa zistiť, či je jednoduchým spôsobom možné vytvoriť čo najvšeobecnejšie (najširšie použiteľné) zjednotenie nájdených metód.

Kľúčové slová: monitorovanie, repozitár, metrika

Úvod

Každá činnosť si vyžaduje nejaké plánovanie. Aj tie najjednoduchšie, z bežného života, pri ktorých sa len podvedome rozhodneme (naplánujeme si), čo a akým spôsobom vykonáme. Počas vývoja softvéru sa ale vyskytujú zložitejšie činnosti ako tie, na ktoré je ľudstvo zvyknuté a ktoré ľudia vykonávajú každodenne. Sú náročnejšie na navrhnutie a samotné vykonávanie, najmä ak ide o prácu, do ktorej je zapojených viac ľudí a priebeh práce jedného človeka ovplyvňuje prácu iných.

Takouto činnosťou je napríklad monitorovanie softvérových projektov. Počas monitorovania musia viacerí ľudia vykonávať isté činnosti, a to nie chaoticky, ale podľa dohodnutého postupu, ktorý je navrhnutý počas plánovania monitorovania. Práve v čase plánovania zisťujeme, akým spôsobom chceme náš projekt monitorovať.

Môžeme teda premýšľať nad tým, ako také monitorovanie vykonať. Samozrejme, naše rozhodnutia závisia napríklad od veľkosti tímu a toho, čo sa vlastne počas tvorby softvéru vykonáva. Ľudia pracujúci na projekte môžu napríklad písať zdrojové kódy, udržiavať ich v repozitároch, prispievať do diskusií alebo tvoriť dokumentáciu. Keď navrhujeme samotnú činnosť, samozrejme chceme, aby sa potom vykonávala čo najjednoduchšie. Aby celý tím nebol zatažený zbytočnými činnosťami.

Radšej strávime viac času nad plánovaním činnosti, ak nám to môže ušetriť čas pri jej vykonávaní, alebo skvalitniť výsledok vykonávania. Logické je napríklad nahradiť monitorovanie, pri ktorom ľudia počas svojej práce sami poskytujú dáta na vyhodnocovanie (napríklad vyplňaním formulárov v rôznych systémoch na manažment projektu) takým monitorovaním, ktoré vytvára tieto dáta automaticky, na základe činností, ktoré členovia tímu aj tak vykonávajú počas práce na projekte.

Preto sa pokúsime popremýšľať nad vhodným spôsobom monitorovania, pričom použijeme výsledky už existujúcich štúdií a už navrhnuté metodiky.

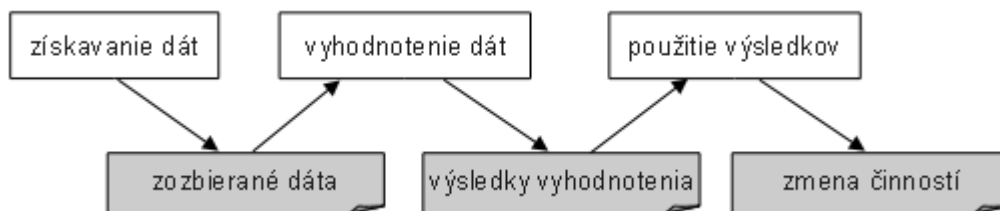
K čomu má monitorovanie viesť?

Na začiatku zisťovania, ako softvérové projekty v tíme monitorovať, si položíme otázku, čo vlastne chceme dosiahnuť. Chceme zisťovať stav projektu. Tento stav sa v čase mení, čo robí úlohu zložitejšou. Ale ako sa dá takýto zistený stav využiť? Tu sa vynorí niekoľko možných odpovedí:

- budeme vedieť lepšie odhadnúť budúci vývoj, alebo vývoj ďalších projektov (dôležité sú napríklad časové odhady alebo odhady typov zamestnancov, ktorých môžeme do projektu zapojiť)
- môžeme zistiť schopnosti jednotlivých zamestnancov a na základe toho mať lepšiu predstavu o štruktúre tímu

Ide teda o monitorovanie celého projektu a tiež jednotlivých členov tímu, ktorí svojou prácou ovplyvňujú smerovanie projektu.

Skúsme vytvoriť jednoduchú postupnosť činností, ktorá vedie k niektorému z týchto spomenutých výsledkov.



Obr. 1. Etapy jedného kroku monitorovania projektu.

Sú identifikované tri činnosti (Obr. 1) a z každej z nich máme výstup, ktorý sa použije v ďalšej činnosti. Teraz sa môžeme zamyslieť nad metódami, ktoré by sa v každom kroku dali použiť.

Tieto tri kroky sú podobné postupu, ktorý je uvedený v [2], kde sú spomenuté tri časti:

- definícia monitorovania (vytvára sa definícia procesu monitorovania, aby sme vedeli, kam monitorovanie smeruje)
- zber dát (pozostáva zo zaznamenávania dát a ich interpretácie tak, aby zodpovedali požadovaným informáciám o projekte)
- analýza (zistenie aktuálnej situácie a nájdenie nových rozhodnutí na základe tejto situácie)

Prvý krok (definícia monitorovania) stanovuje spôsob monitorovania a jeho ciele, takže ovplyvňuje všetky tri naše identifikované etapy procesu. Druhý krok (zber dát) zodpovedá dvom etapám (získavanie dát a vyhodnotenie dát). Tretí krok (analýza) zodpovedá použitiu výsledkov (výstupom je zmena činností v projekte na základe nameraných hodnôt).

Proces navrhovaný v [3] má základné etapy zhodné s vyššie navrhnutými – nachádzajú sa tu časti označené ako zbieranie dát, analýza dát a vykonanie akcie. Týmto etapám navyše predchádza etapa plánovania zberu dát.

Zbieranie dát

Dáta potrebné na vyhodnocovanie môžeme zbierať dvomi základnými spôsobmi.

Prvým z nich je poskytovanie dát samotnými členmi tímu, kedy im pribudne ďalšia práca navyše. Môžu v pravidelných časových intervaloch alebo v stanovených časoch po dokončení nejakej činnosti tvoriť dokumenty s predpísanou štruktúrou, kde zhodnotia svoju prácu a stav časti projektu, ktorá sa ich týka. Tiež sa môže používať dodatočný softvér, ktorý spôsob zápisu dát ešte viac formalizuje.

Druhý spôsob je využitie dát zo systémov, ktoré sa v tíme používajú. Keďže členovia tímu pracujúci na projekte majú veľa práce aj bez toho, aby sme im dávali vyplňať formuláre, ktoré sa budú vyhodnocovať, musíme nájsť spôsob, ako proces získavania dát čo najviac zautomatizovať. Na to využijeme dáta, ktoré sa tak či tak vytvárajú.

Existuje spôsob získavania hodnôt pre metriky, ktorý identifikuje zdroje získavania dát a akcie k nim [1]. K zdrojom, z ktorých môžeme získať dáta, patria repozitáre (kde sú uložené zdrojové kódy a dokumentácia), fóra, databáza chýb, komunikačné kanály alebo

stránky s dokumentáciou. Pri každom z týchto zdrojov sa dajú identifikovať akcie, ktoré členovia tímu môžu vykonávať, napríklad pridanie súborov do repozitára alebo odoslanie odpovede do diskusného fóra. Akcie môžu byť hodnotené váhou a niektoré dokonca môžu byť hodnotené negatívne. Tu sa teda okrem štandardných metrík, ku ktorým patrí napríklad počet vytvorených riadkov zdrojového kódu, používa hodnotenie vykonávania mnohých ďalších akcií, ktoré majú dopad na vývoj projektu.

Okrem príspevku člena každého tímu môžeme používať aj metriky, ktoré súvisia s činnosťou používateľov alebo ľudí, ktorí sa len príležitostne zapájajú do projektu. Podľa [2] môžeme merať napríklad počet chýb nahlásených používateľmi, počet stiahnutí softvéru alebo počet používateľov, ktorí pomáhajú iným, čo je výhodné najmä pri otvorených (open source) projektoch. Tento prístup by sa dal rozšíriť tak, aby zasahoval aj iné ako otvorené projekty. Aj okolo komerčných projektov sa napríklad vytvára skupina používateľov, ktorí si navzájom pomáhajú pri práci s daným softvérom na fórach. Tak isto sú napríklad posielané hlásenia chýb a informácie o vyriešení problémov.

Ak sa vytvára softvér na mieru, môžeme ako používateľov brať zadávateľov projektu a identifikovať aj ich akcie a zahrnúť ich početnosť vo výpočte ukazovateľov.

Nemusíme uvažovať iba nad hodnotami, ktoré sa týkajú jedného konkrétneho človeka, ktorý sa nejakým spôsobom zapája do projektu (programátor, manažér, používateľ, zadávateľ...). Môžeme nájsť aj hodnoty, ktoré sa týkajú projektu ako celku. Ide napríklad o počet dokončených komponentov, vyriešených úloh alebo zostávajúcich nevyriešených chýb.

Prístup navrhnutý v [3], kde sú zadávatelia zapojení do sledovania stavu, určuje metriky, ktoré hovoria o tom, či niektorý z cieľov konkrétnej skupiny ľudí zapojených do projektu (napríklad zadávateľov) bol splnený. Ďalšou navrhnutou metriku je to, ako často a akým spôsobom sú vykonávané postupy, ktoré vedú k dosiahnutiu niektorého zo stanovených cieľov.

Zatiaľ sme sa zaoberali jednoznačne identifikovateľnými akciami, ktoré presne zodpovedajú akciám súvisiacim s priebehom softvérového projektu, ktoré vykonávajú ľudia, ktorí majú v projekte konkrétne úlohy a týmito svojimi akciami sa zapájajú do projektu. Tieto akcie sú vyvolávané explicitne. Dajú sa identifikovať aj akcie vyvolané implicitne na základe časového priebehu nameraných hodnôt tých akcií, ktoré sú vykonávané explicitne. Príkladom je prípad, keď počet e-mailov poslaných nejakým členom tímu za mesiac klesne pod určitú hranicu [2]. Aj napriek tomu takúto metriku môžeme zahrnúť do výpočtov, napríklad do výpočtu, ktorý jednotlivým akciám priraduje váhy určujúce, nakoľko sú akcie dobré alebo zlé pre vývoj projektu.

Akcie (ktoré sú vyvolané explicitne alebo implicitne) zodpovedajú jednotlivým metrikám.

Vyhodnotenie dát

Keď máme zozbierané dáta, treba ich vyhodnotiť štatistickými metódami, aby sme získali hodnoty, ktoré čo najlepšie charakterizujú to, čo chceme hodnotiť.

Pri zozbieraní dát, ktoré vyjadrujú početnosť akcií vykonaných jednotlivými členmi tímu, vieme získať pre každého člena tímu hodnotu, ktorá hovorí o množstve hodnotnej práce vykonanej na projekte [1]. Každý hodnotenej akcii sa priradí váha, ktorá

charakterizuje mieru pozitívneho dopadu akcie na vývoj projektu. Výsledky získané touto metódou nehovoria ale nič o tom, ako sa projekt vyvíja (nepomôže nám napríklad pri časových odhadoch), ide vlastne iba o hodnotenie každého zamestnanca a ich porovnanie medzi sebou. Keďže sme v predchádzajúcej časti zahrnuli aj iné prístupy zbierania dát a nemáme iba dáta týkajúce sa zamestnancov, ale aj napríklad zadávateľov a používateľov, dá sa tento prístup zovšeobecniť na každú úlohu pri vývoji projektu. Samozrejme, napríklad programátor bude mať hodnotu počítanú na základe početností iných akcií a ich váh ako napríklad používateľ.

Je možné nájsť význam aj pre kombinácie, postupnosti akcií [2]. Akcie vykonané v určitom časovom intervale v presne stanovenom poradí môžu hovoriť o plnení nejakého konkrétneho cieľa, preto sa takémuto postupu môže priradiť zvlášť pozitívny dopad na vývoj projektu.

Použitie výsledkov

Vypočítané hodnotenia zamestnancov, získané pomocou metódy [1], nám pomôžu porovnať množstvo práce vykonané zamestnancami. Treba poukázať na to, že vypočítané hodnoty (získané sčítaním početností akcií, ktoré sú násobené váhami akcií), nemajú pre nás až takú hodnotu, ako jednotlivé početnosti akcií. Tieto početnosti sa dajú zohľadniť pri zisťovaní charakteristík zamestnancov, na základe ktorých ich môžeme lepšie priradiť na jednotlivé pozície a lepšie im pridelovať úlohy.

Sledovanie vývoja hodnôt v čase a časových obdobiach, kedy sa hodnoty zásadne menia (napríklad klesnú pod stanovenú hranicu [2]), môže slúžiť na zistenie náhlych zmien, ktoré môžu narušiť chod projektu.

Podľa [3] je vhodné zdieľať stav projektu (výsledky meraní) aj so zadávateľmi projektu, čo im umožňuje priebežne reagovať na stav projektu.

Zjednotenie analyzovaných etáp

V predchádzajúcich častiach boli vybrané viaceré prístupy z nájdených metodí, niektoré boli zovšeobecnené aj na iné typy softvérových projektov alebo bolo nájdené ich širšie použitie, na dosiahnutie ktorého stačili jemné modifikácie.

Teraz sa môžeme pozrieť na zjednotenie celého prístupu k monitorovaniu, teda skúsime nájsť súvis troch etáp, ktoré sme na začiatku identifikovali a ktorými sme sa doteraz zaoberali samostatne.

Ako základ pre celkové zhodnotenie zvolíme rozdelenie na metriky. Ku každej metrike identifikujeme jej dopad na ďalšie fázy – navrhujeme váhy a to, či sa dá použiť na hodnotenie členov, spolupráce celého tímu, odhad zostávajúceho času alebo na prezentovanie stavu zadávateľovi. Príklady metrik a k nim odhadované výsledky sú zhrnuté v Tab. 1. Vplyv na fázu *Použitie výsledkov* je zahrnutý v stĺpcoch, ktoré patria vyhodnoteniu dát (váhy sú uvedené iba pri metrikách, ktoré sú aplikovateľné pri použití výsledkov daným spôsobom).

Čím vyššie číslo dostaneme vo výslednom hodnotení, tým je stav tejto oblasti lepší. Napríklad pri prezentovaní stavu zadávateľovi sa hodnotenie pohybuje vždy v záporných hodnotách a až keď nezostávajú žiadne ďalšie úlohy a chyby, je táto hodnota nulová.

Tab. 1. Súvis metrik, vyhodnotenia a použitia výsledkov (niektoré metriky podľa [1] a [2]).

| <i>zbieranie dát</i> | | <i>vyhodnotenie dát (váhy) a možnosť využitia vo fáze použitie výsledkov</i> | | | |
|----------------------|---|--|------------------------------|-------------------------|---|
| zdroj | metrika | hodn. člena tímu | hodn. spolupráce tímu | odhad zost. času | prezentovanie stavu zadávateľovi |
| repozitár | počet riadkov kódu | 0.1 | - | - | - |
| | počet pridaní zdrojového súboru | 100 | - | - | - |
| | počet pridaní adresára (správne zaradenie do adresárovej štruktúry) | 10 | - | - | - |
| | počet pridaní adresára (nesprávne zaradenie) | -20 | - | - | - |
| | počet pridaní časti kódu, v ktorej sa vyskytuje chyba | -30 | - | -20 | - |
| | počet pridaní časti kódu, ktorá opravuje chybu | 30 | - | 20 | - |
| | počet pridaní kódu, ktorý sa nedá skompilovať | -50 | - | - | - |
| | počet pridaní kódu bez komentárov | -20 | - | - | - |
| fóra | počet pridaní otázky s problémom | 0 | - | -20 | - |
| | počet poslaní odpovede na otázku | 1 | 1 | - | - |
| | počet poslaní odpovede, ktorá prispieva k riešeniu | 10 | 10 | - | - |
| | počet poslaní odpovede, ktorá rieši problém | 20 | 20 | 20 | - |
| | počet období dlhších ako 1 týždeň bez nových príspevkov vo fóre | -10 | -10 | - | - |
| správa chýb | počet správnych pridaní chyby | 5 | 5 | - | - |

| | | | | | |
|--------------------|--|-----|----|------|----|
| | počet nesprávnych pridaní chyby (už je pridaná / chyba neexistuje) | -5 | -5 | - | - |
| | počet aktívnych chýb | - | - | -20 | -5 |
| správa úloh | počet pridaní úlohy | 5 | - | - | - |
| | celkový počet dní meškania s riešením úlohy | -20 | - | -100 | - |
| | počet vyriešení úlohy načas | 20 | - | - | - |
| | počet zostávajúcich hodín na riešenie úloh | - | - | -10 | -2 |

Aplikovanie v tímovom projekte

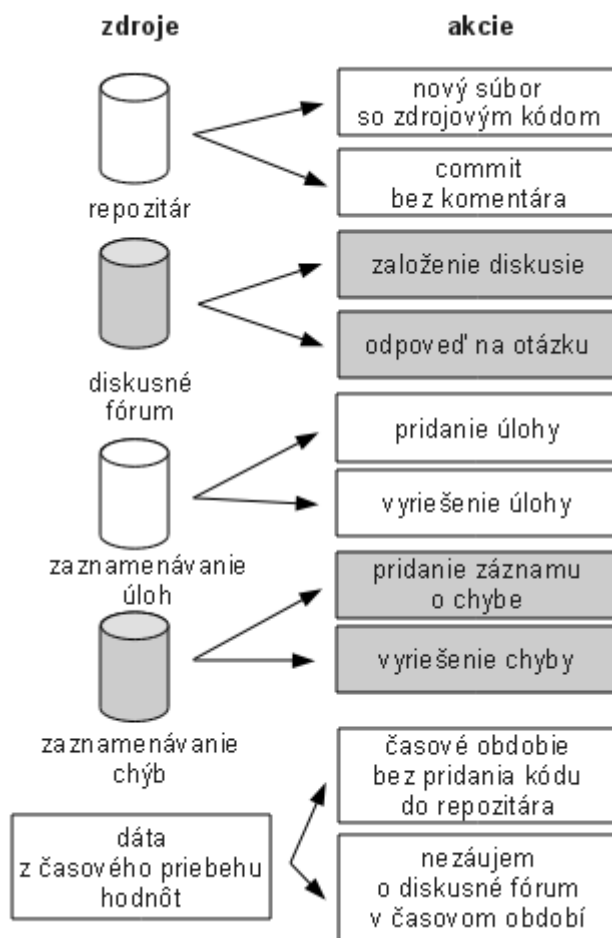
Takúto metodiku zisťovania stavu projektu, ktorá je súhrnom postupov vyňatých z viacerých zdrojov, môžeme použiť ako tím počas riešenia tímového projektu. Okrem činností jednotlivých členov tímu sa dajú identifikovať aj činnosti ľudí, ktorí majú na projekt bezprostredný vplyv, napríklad vedúci projektu alebo tí, ktorí zabezpečujú organizáciu predmetu, v rámci ktorého projekt prebieha.

V tíme pracuje päť ľudí. Sú samozrejme aj omnoho väčšie tímy, ale aj tu sa ukazuje potreba riadenia a monitorovania činností. Dokonca aj v takýchto tímoch treba riadiť komunikáciu medzi členmi a sledovať ich prácu, čo môže viesť napríklad k vyhodnoteniu, či úlohy boli rozdelené optimálne. Takže aj takýto malý tím nám dobre poslúži ako príklad a ako dôkaz toho, že treba činnosť dobre skoordinať. Aj tu sa ukazuje, že aj čo sa zdá byť samozrejmé, pri používaní môže viesť k omylom a nedokonalostiam.

Používame niekoľko systémov, ktoré by sa dali použiť na zber dát, z ktorých by mohlo prebiehať automatické vyhodnocovanie:

- repozitár so zdrojovými kódmi
- zaznamenávanie úloh a chýb
- diskusie

Ďalej môžeme ku každému z týchto systémov nájsť konkrétne akcie, príklady ktorých sú znázornené na Obr. 2.



Obr. 2. Hodnotenie členov v tímovom projekte (niektoré zdroje a akcie podľa [1] a [2]).

Použité boli poznatky z časti *Zbieranie dát*, teda zaznamenávame akcie súvisiace s prácou s úložiskami dát a tiež akcie vyvolané implicitne na základe priebehu nameraných hodnôt.

Hodnotenie sa počíta najjednoduchším spôsobom, teda každému typu akcie sa priradí váha a početnosti sa vynásobia váhami a sčítajú. Tým sa získa hodnotenie člena tímu – čím vyššie číslo, tým je jeho príspevok k napredovaniu projektu väčší. Dôležité je dodať, že niektoré váhy sú záporné, teda že existujú špecifické akcie, ktorých vykonanie „zhorší“ stav projektu – teda vykonanie takýchto akcií (alebo také správanie sa, aby boli splnené podmienky pre implicitné akcie) nie je žiaduce.

Môžeme teda využiť nájdené zjednotenie prístupov tak, ako nám to vyhovuje v tímovom projekte. Vidieť, že metodiky nájdené v zdrojoch, z ktorých sme čerpali, sa dali zovšeobecniť. Napríklad nejde o „pravý“ otvorený projekt (s účasťou používateľov na testovaní a vývoji), ale aj napriek tomu sa dali použiť metriky z [2] – využili sa iba pri činnostiach členov tímu.

Výsledky získané zberom dát a ich vyhodnotením ale nemôžeme vo všeobecnosti aplikovať na porovnanie jednotlivých členov tímu, keďže v takom malom tíme sa nenachádzajú ľudia s rovnakou úlohou v tíme a teda nemôžeme zvoliť pre všetkých

členov tímu rovnakú množinu akcií, ktorých kvantita sa bude merať. Navyše, úlohy v tíme sa časom menia, aby si každý člen vyskúšal viac pozícií, preto sa časové obdobie vyhradené na riešenie projektu delí na menšie časové obdobia, v ktorých je stav pridelenia pozícií členom nemenný. Tieto časové obdobia sú príliš krátke na získanie hodnôt, ktoré sa dajú analyzovať.

Tab. 2. Rozdelenie projektu na časové obdobia a pridelenie funkcií členom tímu.

| | Obdobie 1 | Obdobie 2 | Obdobie 3 | Obdobie 4 | Obdobie 5 |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Člen 1 | vedúci tímu | tvorba dokumentácie | programátor | programátor | architekt |
| Člen 2 | tvorba dokumentácie | programátor | programátor | architekt | vedúci tímu |
| Člen 3 | programátor | programátor | architekt | vedúci tímu | tvorba dokumentácie |
| Člen 4 | programátor | architekt | vedúci tímu | tvorba dokumentácie | programátor |
| Člen 5 | architekt | vedúci tímu | tvorba dokumentácie | programátor | programátor |

Tab. 2 znázorňuje príklad rozdelenia časového obdobia prideleného na riešenie projektu na rovnako dlhé časové obdobia. Každý člen má v každom období pridelenú nejakú pozíciu v tíme a pridelenie musí byť na začiatku navrhnuté tak, aby si každý člen vyskúšal všetky pozície a aby rovnaká pozícia bola pridelená rovnako dlhú dobu každému členovi. Vtedy by sa dal porovnávať aj príspevok jednotlivých členov pre každú pozíciu samostatne, pričom by sa vytvorilo jedno porovnanie pre každú pozíciu a do porovnávaní by sa zahrnuli pre každého člena iba dáta z obdobia, kedy mu bola pridelená príslušná pozícia. Vtedy by sa dalo zistiť, na ktorú pozíciu sa ktorý člen tímu najviac hodí. Tu ale vzniká problém nerovnomerného rozloženia práce na projekte v jednotlivých obdobiach pre jednotlivé pozície, napríklad môžeme predpokladať, že manažérske pozície budú v prvom období zaťaženejšie viac ako programátorské, keďže treba vytvárať plán projektu a navrhovať, čo sa bude počas projektu vykonávať, bez toho, aby sa už niečo programovalo.

Môžeme ale napríklad vyčleniť akcie, ktoré vykonáva každý člen tímu, napríklad prispievanie do diskusie. Tu je ale otázne, či aj na početnosť týchto akcií nemá vplyv aktuálna pozícia člena v tíme, teda či každá pozícia v tíme vytvára rovnaké podmienky na vykonávanie danej akcie a či každá pozícia vyžaduje vykonávanie takej akcie v rovnakom množstve.

Aplikovanie princípov navrhnutých v metodike [3] (zdieľanie stavu so zadávateľom) vyplýva už z povahy projektu, presnejšie z prostredia, v ktorom sa na projekte pracuje. Je samozrejmé, že stav je oznamovaný vedúcemu projektu, ktorého pozíciu môžeme chápať aj ako pozíciu zadávateľa.

Záver

Cieľom eseje bolo pozrieť sa na proces monitorovania rozdelený na etapy, ktoré vedú od zberu dát až po zmeny v riadení projektu vykonané na základe výsledkov monitorovania. V každej fáze bolo identifikované, čo z nájdených metód by sa dalo použiť a v ktorých prípadoch, čím vznikol príklad, ktorý zjednocuje niektoré predtým nájdené metódy používané pri monitorovaní softvérových projektov.

Rozdelenie procesu na fázy dovolilo lepšie identifikovať prienik metodík. Procesy vykonávané v každej metodike sa tým rozdelili do viacerých častí, čo umožňovalo lepšie porovnávanie nájdených metód, ako keby sme porovnávali metodiky ako celok a skúšali v nich nájsť podobnosti.

Po analyzovaní každej fázy samostatne bolo vytvorené zhrnutie, ktoré fázy znovu zjednocuje.

Na záver bola posúdená možnosť aplikovania nájdeného zjednotenia metód na monitorovanie v tímovom projekte. Boli zhrnuté výhody a nedostatky prístupov, ktoré by sa tým používali.

Je teda možné na základe nájdených zdrojov vytvoriť vlastný prístup k monitorovaniu kombináciou používaných prístupov, z ktorých niektoré ani netvoría podstatnú časť pôvodnej metodiky, ale poskytujú myšlienku vhodnú na začlenenie do vznikajúceho spôsobu monitorovania.

Existujú softvérové produkty, ktoré fázu získavania dát a vyhodnocovania spájajú a netreba sa zaoberať možnosťami, ako postupovať v každej fáze samostatne, keďže produkt nám poskytuje už pripravené navrhnuté riešenie. Tu sme preberali každú fázu samostatne, aby sa dali lepšie analyzovať zdroje, ktoré prinášajú navrhované možnosti.

Samozrejme, ak má byť monitorovanie čo najviac automatizované, musia byť používané produkty, ktoré dáta na hodnotenie dokážu poskytnúť. Na monitorovanie má teda vplyv používanie nástrojov na manažment verzií a chýb, diskusných fór a ďalších, z ktorých sa dajú potrebné dáta získavať.

Použitá literatúra

1. Gousios, G., Kalliamvakou, E., Spinellis, D.: Measuring Developer Contribution from Software Repository Data. In: Proceedings of the 2008 international working conference on Mining software repositories. 2008. ISBN 978-1-60558-024-1. s. 129-132.
2. Wahyudin D., Tjoa M.: Event-Based Monitoring of Open Source Software Projects. In: The Second International Conference on Availability, Reliability and Security. 2007. ISBN 0-7695-2775-2. s. 1108-1115.
3. Tsunoda M., Matsumura T., Matsumoto K.: Modeling Software Project Monitoring with Stakeholders. In: Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on. ISBN 978-1-4244-8198-9. 2010. s. 723-728.

Annotation

Software project monitoring – a lot of possibilities, ideas – what to use?

Software project monitoring is an important part of the work on a project, which is preceded by an analysis and design of the monitoring method. In addition to the explicit monitoring, when the developers create information used in the monitoring, the implicit monitoring is used, which involves automatic data collection, for example from the software repositories. A lot of monitoring methods have been proposed, mainly related to systems with a specific characteristics (for example component based systems) or they are bound to the specific development method (for example open source software).

This work analyses the restrictions of the studied monitoring methods and brings some thoughts about their boundaries and the ability to find broader possibilities to use them. There is even an ability to search their intersections in some specific case. We can then determine, if a simple way of creating of an generalized (usable in the most common cases) method exists.