

MANAŽMENT VERZIÍ: POMOC PRI MONITOROVANÍ?

„Ak nevieš kam ideš, ako vieš, že si dorazil?“

(Alka Jarvis, Vern Crandall)

Katarína Valalíková

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

Abstrakt. Monitorovanie softvérového projektu je dôležitá časť, ktorá nám pomáha k jeho úspešnému ukončeniu. Na sledovanie však nestačí len dobrý zrak, ale potrebujeme zvoliť aj vhodné pomocné prostriedky. Správny prístup k monitorovaniu a využitie adekvátnych nástrojov pomáha nielen smerovať projekt k dosiahnutiu vytýčených cieľov, ale takisto v istej miere zaručuje aj jeho kvalitu či dohľad nad jednotlivými členmi tímu. V eseji sa pokúsím priblížiť nástroje, ktoré sa dennodenne využívajú pri práci v tíme a spôsob akým ich môžeme použiť na monitorovanie. Ako základný stavebný kameň je zvolený manažment verzií, ktorého dôležitosť nebude asi nikto popierať. Veď kto sa ešte nedostal do situácie, keď svojimi zmenami zapríčinil nefunkčnosť doteraz funkčného programu a musel sa vrátiť späť?

Kľúčové slová: manažment verzií, verziovací systém, monitorovanie

Úvod

Monitorovanie vývoja softvéru hrá dôležitú rolu pri vytváraní softvérového produktu. Je dôležité, aby sme vedeli, akým smerom sa projekt uberá a aké zásluhy na tom majú jednotliví členovia tímu. Okrem iného nám monitorovanie môže otvoriť oči skôr, ako by sme sa mohli dostať do problémov spojených s nedodržaním stanovených termínov.

Sledovať projekt však nie je jednoduchou úlohou. Keby to mal na starosti jeden človek bez pomoci iných nástrojov a metodík, stálo by ho veľa energie a času, aby dokázal efektívne a účinne predísť rizikám súvisiacim s nesprávnym monitorovaním. Obzvlášť

pri veľkých tímoch by mohol nastať problém pri potrebe sledovať činnosť jednotlivých členov.

Pre monitorovanie vývoja systému je veľmi dôležité zostaviť plán, v ktorom sú opísané jednotlivé úlohy a rovnako aj ich priority. Rovnako je dôležité určiť pravidlá, ktoré pomôžu rozhodnúť sa, či je projekt na správnej ceste alebo nie. Veď predsa cieľom každej spoločnosti je dosiahnuť čo najlepšie výsledky v čo najkratšom čase [5].

Výskumy [5] ukázali, že mnoho projektov sa nevyvíja presne podľa plánu, čo spôsobuje problémy. Projekty nie sú dodávané načas, zvyšuje sa ich cena a rovnako rozpočet, s ktorým sa počítalo, nestačí. Väčšinou za tým stojí nedostatok vedomostí pri plánovaní, vytváranie nových systémov, s ktorými nemajú softvéroví inžinieri skúsenosti a veľká komplexnosť týchto programov spojená s nepredvídateľnými situáciami [5].

Sledovať plány, ktoré boli vytvorené, je možné aj za pomoci podporných prostriedkov. Tie nám uľahčia a zautomatizujú značnú časť, ktorej by sme museli venovať nemalé úsilie. Ako však tieto nástroje vybrať? Aké nástroje nám pomôžu pri monitorovaní nielen jednotlivých členov, ale aj pri sledovaní celkového projektu? Na tieto otázky existuje množstvo odpovedí, nakoľko aj podporných prostriedkov a metodík je čoraz viac. Je len na manažérovi tímu, aby zvážil, čo v skutočnosti jeho zamestnanci a aj on samotný potrebujú a na základe toho sa rozhodol. Jednou z dôležitých úloh manažéra je napríklad aj sledovať činnosti každého zúčastneného na projekte. K tomu mu môže poslúžiť napríklad manažment verzií.

Manažment verzií je dôležitá súčasť každého projektu. To nebude popierať asi nikto. Pomocou systému na verziovanie vieme nielen uchovávať jednotlivé verzie produktu a pri akomkoľvek poškodení súčasnej verzie sa vrátiť o verziu či viac späť, ale poslúži nám aj ako veľká encyklopédia, ktorá nesie záznamy o jednotlivých vykonaných zmenách a aj o tom, kto tieto zmeny spravil. Kedykoľvek sa môžeme spýtať, kto čo pridal, odstránil či zmenil, ako často alebo akým rozsahom daný človek prispieva k vývoju softvéru. Je však systém na verziovanie postačujúcim nástrojom?

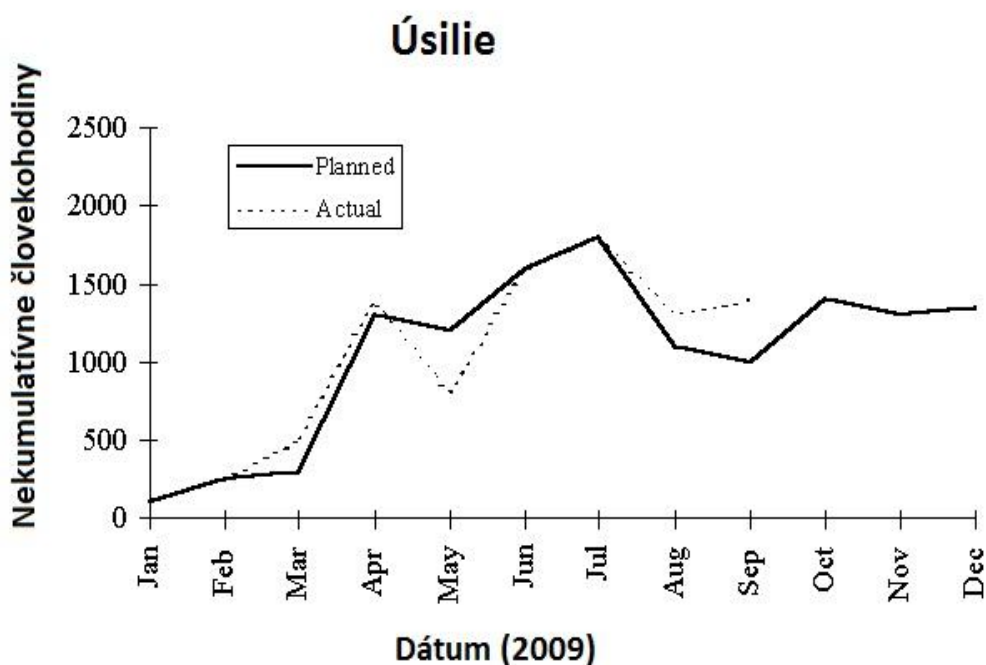
Čo chceme monitorovať?

V prvom rade si musíme uvedomiť, čo je cieľom nášho monitorovania. Existuje množstvo metrík, ktoré sa sústreďujú na kvalitu vyvíjaného softvéru. Líšia sa na základe rôznych pohľadov na vyvíjaný produkt. Ak chceme monitorovať kvalitu softvéru, potrebujeme zvoliť iné metriky ako pri meraní procesu vývoja. V našom prípade sa chceme zaoberať metrikami opisujúcimi proces vývoja softvéru [1]. Podľa [1] medzi ne patria nasledovné:

1. úsilie
2. zmeny požiadaviek
3. náklady a čas

Prvá metrika zo spomínaných, úsilie, sa zaoberá predovšetkým sledovaním ľudských zdrojov. Sleduje náklady spojené so zamestnancami, dodržiavanie plánu, kvalitu

výsledného produktu a úsilie spojené s každou vykonanou aktivitou. Indikátor opisujúci účinnosť môže byť použitý všetkými úrovňami manažmentu, pričom každá z nich si ho vyformuje podľa svojich potrieb. Takto potom môžeme porovnávať aktuálny stav s tým, ktorý je uvedený v pláne. Metrika môže byť použitá aj v rôznych fázach vývoja produktu, od jeho začiatku až po samotný koniec. Obrázok 1 zobrazuje príklad porovnania aktuálneho a plánovaného úsilia v daných mesiacoch [4].



Obr. 1. Obrázok 1: Rozdiel medzi aktuálnym a plánovaným úsilím vynaloženým v jednotlivých mesiacoch [4].

Ďalšou, vyššie spomenutou, metrikou bolo menenie požiadaviek. Tá opisuje stálosť, kompletnosť a pochopenie požiadaviek už na začiatku. Zaznamenáva počet zmien na stanovené požiadavky a množstvo informácií potrebných na ich finálne zadefinovanie (pozri Obrázok 2). Ak máme množstvo požiadaviek, ktoré sa často menia, nasvedčuje to, že s naším produktom niečo nie je v poriadku. Odráža sa to aj na jeho kvalite, náraste ceny a odklonu od pôvodného plánu [4].

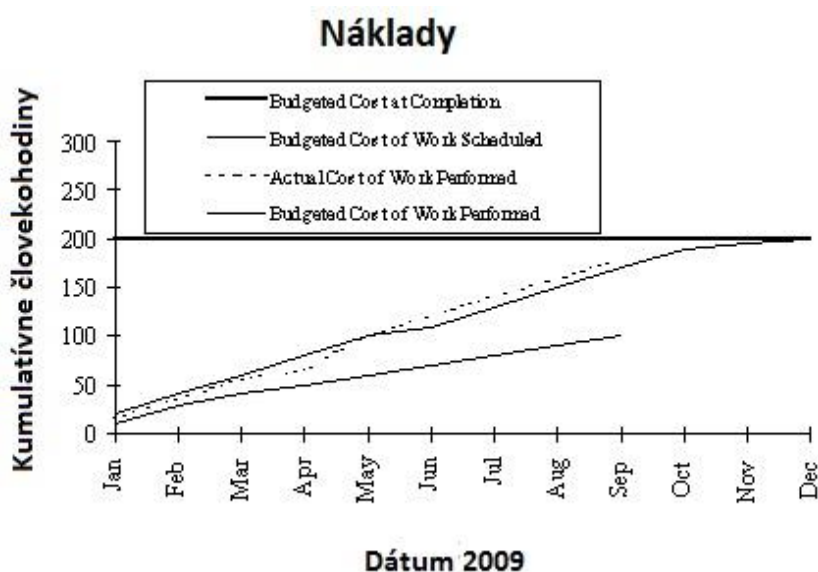
Indikátor stálosti požiadaviek poukazuje na ich množstvo, zmeny a počet nedefinovaných požiadaviek, ktoré sa označujú ako TBD. Táto metrika, podobne ako predchádzajúca, je aplikovateľná vo všetkých fázach, od začiatku vývoja projektu, po jeho koniec. Obzvlášť dôležitý je indikátor požiadaviek vo fázach špecifikácie a návrhu [4].

Obr. 2: Zobrazenie celkového počtu požiadaviek (Total), kumulatívny počet zmien požiadaviek (Cumulative Change) a počet nedefinovaných požiadaviek (TBD) počas daného obdobia [4].

Poslednou spomínanou metrikou sú náklady a čas. Tá je veľmi dôležitá z hľadiska úspešnosti projektu. Primárnym komponentom je cena odhadovaných prác. Manažéri musia správne zaradiť práce, rozoznať úroveň zručnosti na splnenie úloh a využiť odhady a plány na určenie ceny celého projektu. Náklady na projekt sú zvyčajne vyjadrené v človeko-hodinách [4].

Indikátor ceny je úzko spojený s indikátorom úsilia, kde cena je definovaná ako nahromadenie výdavkov úsilia. Na získavanie odhadu nákladov sa používa štruktúra rozpisu prác, ktorá identifikuje odlišné elementy pre požiadavky, návrh, dokumentáciu, programovanie a testovanie, integráciu, overovanie a testovanie systému [4].

Ďalším možným spôsobom rozdelenia nákladov môže byť na základe komponentov, funkcionality alebo nastavovaní. Týmto systémom je manažérom umožnené sledovať aktuálnu cenu a vymeriavať ju voči určenému rozpočtu. Obrázok 3 je príkladom indikátora nákladov [4].



Obr. 3: Príklad indikátora nákladov [4].

Pri monitorovaní vývoja softvérového projektu sa využíva viacero metrík. Spomenuté boli metriky vhodné pri sledovaní procesu vývoja. Nasledovná tabuľka (Tabuľka 1) obsahuje stručný opis metrík, ktoré sú odporúčané použiť pri vývoji softvéru.

Tab. 1: Niektoré z odporúčaných metrík používaných pri vývoji softvéru [4].

Metrika	Stručný opis	Indikátor
Pokrok	Poskytuje informáciu o tom, ako dobre sa projekt vyvíja vzhľadom na plán	Aktuálne vs. plánované ukončenie úlohy
Úsilie	Poskytuje pohľad na prispievania	Aktuálne vs. plánované profily

	zamestnancov na cenu, plán, dodržiavanie cieľa a kvalitu produktu	zamestnancov
Náklady	Poskytuje sledovať aktuálnu cenu voči odhadovanej a predchádza budúcim nákladom	Aktuálne vs. plánované náklady
Výsledky revízie	Poskytuje status akcií z revízií jednotlivých fáz	Status akcií
Chybové hlásenia	Poskytujú náhľad do produktu a kvality procesu a efektívnosť testovania	Status chybových hlásení (počet otvorených, zatvorených, rozpracovaných problémov)
Stálosť požiadaviek	Poskytuje pohľad na rozsah a dosah zmien požiadaviek	Počet zmenených/objasnených požiadaviek

Čo všetko potrebujeme na monitorovanie?

Monitorovanie vývoja projektu pomáha do viesť ho úspešne do konca. Prístup k vývoju sa líši od projektu k projektu. Existuje viacero metód používaných pri vyvíjaní projektu. V súčasnosti medzi veľmi populárnu a obľúbenú metódu patrí agilný spôsob vývoja softvéru, v minulosti, ale v niektorých prípadoch aj v súčasnosti, bol značne využívaný najmä vodopádový model vývoja softvéru. Avšak aj keď používame rozdielne prístupy, každý projekt sa skladá z rovnakých častí, a to analýzy požiadaviek, špecifikácie požiadaviek, návrhu riešenia, implementácie, nasadenia aplikácie a jej testovania. Najmä pri práci vo väčších tímoch je veľmi dôležité, aby sa jednotlivé časti poriadne naplánovali.

Keď manažérovi dodáme dobre vypracovaný plán, jeho úlohou je dohliadnuť na splnenie jednotlivých úloh. Je na ňom, aký spôsob kontrolovania zvolí. Ak chce využiť na sledovanie činnosti jednotlivých členov manažment verzí, tak si musí zodpovedať na otázku: Je manažment verzí skutočne vhodné použiť? Dá sa použiť v jednotlivých fázach vývoja systému? Ako a v čom mi manažment verzí pomôže a aké sú jeho nedostatky?

Manažment konfigurácií podporných prostriedkov

Manažment nastavenia jednotlivých podporných prostriedkov pri vývoji softvéru hrá dôležitú úlohu. Obzvlášť je známe jeho použitie v neskorších fázach vyvíjania produktu, ako je implementácia, integrácia a testovanie. Podporné prostriedky sa dajú však použiť aj v skorších fázach [3]. Otázne však je, akou mierou sa podieľajú na vylepšení alebo lepšie povedané ako efektívne sú v skorších fázach. Z toho vyplýva ďalšia otázka, a to, kedy sa má použiť aký podporný prostriedok?

Pri práci v tímoch sa používa viacero podporných prostriedkov. Jedným s nich je aj manažment verzí. Ten je reprezentovaný verziovacím systémom. Použitie verziovacieho systému je dobre známe takmer na každom projekte. Taktiež je možné vybrať si typ, ktorý je pre daný projekt výhodnejší, a to buď centralizovaný alebo decentralizovaný.

Okrem verziovacieho systému sa v značnej miere používajú ako podporné prostriedky rôzne systémy na pridelovanie úloh. Tieto dva nástroje sú častokrát

previazané, čím zvyšujú prehľad a presnosť pri monitorovaní. Cieľom však je poskytnúť pohľad len na jeden prístup z nich, a to manažment verzíí.

Manažment verzíí

Manažment verzíí slúži najmä na uchovávanie jednotlivých verzíí dokumentov počas vývoja. Použitie verziovacieho systému v ranných fázach, ako je analýza a návrh nie je úplne vyhovujúce. Na začiatku sa totiž môžu meniť často požiadavky a takisto niektoré chyby sa ukážu neskôr, v čase prototypovania. Ak by sme manažment použili už v tomto skorom čase, mohli by sme dospieť k prílišnej komplexnosti už na začiatku vývoja. Navyše verziovacie systémy sú navrhnuté tak, aby zaznamenávali zmenené riadky súboru, či už zapríčinené vymazaním alebo pridaním. Nezaujímajú ich však logická štruktúra dokumentu, čo môže spôsobovať viaceré problémy [3].

Pri dokumentoch, ktoré obsahujú napríklad UML diagramy, v dôsledku hocijakej aj menšej zmeny, by verziovací systém mohol podávať mylné informácie. Navonok by sa zdalo, že sa zmenila značná časť dokumentu, no v skutočnosti by šlo len napríklad o prídanie metódy do triedy [3]. Na druhej strane si však myslím, že je dobré, ak sa uchovávajú aj rôzne druhy návrhu či analýzy. Nemá význam použiť to na automatické zistenie rozsahu zmien a teda monitorovanie vývoja, ale v prípade poškodenia lokálnej verzie, ku ktorej mohlo dôjsť napríklad poškodením disku, si môžeme z úložiska stiahnuť túto verziu znova a znova. Nemusíme sa obávať o stratenie svojich dát.

Vo fázach implementácie, testovania a integrácie je použitie verziovacieho systému takmer nevyhnutnou súčasťou. Tu nám bohaté poslúži aj pri automatizácii sledovania vývoja projektu a jeho jednotlivých členov. Verziovacie systémy, ako napríklad Subversion, poskytujú bohatú funkcionálnu možnosť zistiť zmeny v určitom časovom horizonte. Takisto môžeme sledovať aj činnosť jednotlivých členov. Pomocou rôznych kritérií sa dostaneme k mnohým štatistikám, ktoré môžu slúžiť ako odpoveď k dopytovaniu sa na metriky súvisiace s procesom [2].

Manažment verzíí a rôzne fázy vývoja

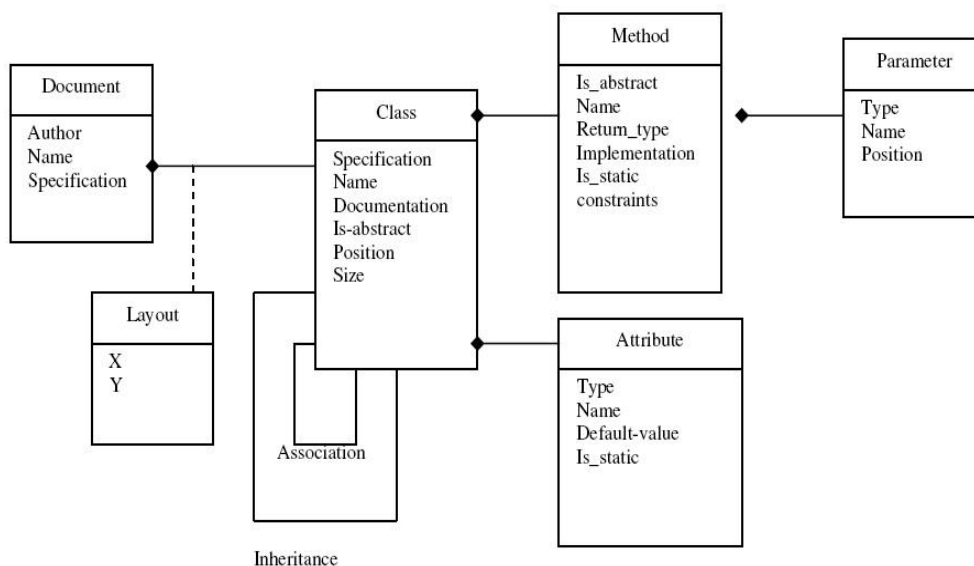
Otázkou, na ktorú sa pokúsime odpovedať v tejto časti je, ako sa dá použiť manažment verzíí v rôznych fázach vývoja softvéru a ako vhodné v nich je jeho využitie. Budú opísané tri fázy, a to fáza analýzy, návrhu a nasadzovania.

Fáza analýzy

Ako bolo povedané už skôr, vo fáze analýzy môže vzniknúť množstvo rozličných dokumentov, súvisiacich s častým menením a prispôbovaním požiadaviek na systém. Pomocou verziovacieho systému môžeme síce zistiť, ako často sa tieto požiadavky menili, no je pravdepodobné, že tieto informácie pre nás nebudú dôležité z hľadiska rozsahu zmien. Dostaneme teda prehľad o tom, že daný súbor bol x-krát zmenený, aj kto ho menil, ale nevieme povedať, čo v ňom bolo zmenené [3].

Fáza návrhu

Fázu návrhu zvyčajne reprezentujú UML diagramy. Ako pri analýze aj tu platí, že sa môžu často meniť v dôsledku zmeny špecifikácie požiadaviek na systém, ale aj prípadným rozširovaním systému. Bežné verziované systémy sa s takto meniacimi sa dokumentami nevedia vysporiadať, čo bolo už viackrát spomenuté. Preto je potrebné použiť verziovací systém podporujúci modelovanie jemne zrnitých dát (pozri Obrázok 4 a Obrázok 5) [3].



Obr. 4: Meta model jemne zrnitého dátového modelu pre UML diagram tried [3].

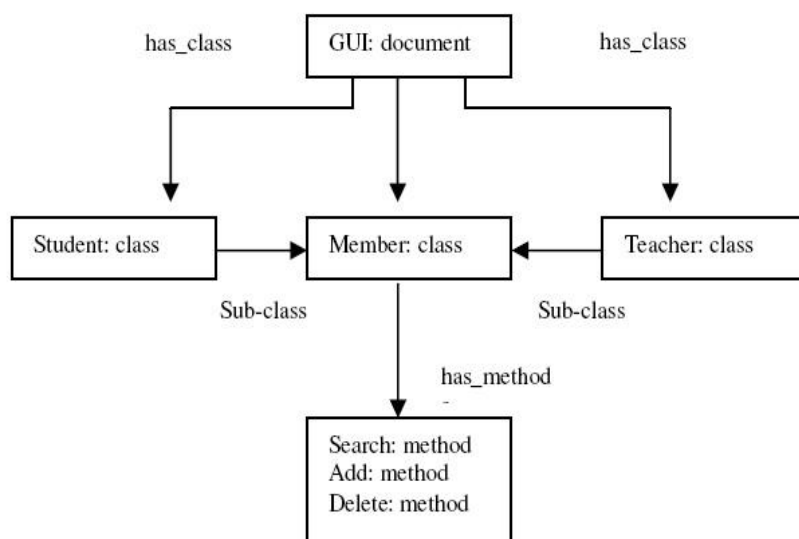
Na obrázku (Obrázok 4.) môžeme vidieť obojsmerné vzťahy medzi objektami, čím celá štruktúra môže byť prechádzaná od koreňa až po koncové body a naopak. Vzťahy medzi typmi objektov sú [3]:

1. Dokument obsahuje triedy, triedy obsahujú metódy a atribúty a metódy závisia od parametrov.
2. Triedy môžu vyjadrovať vzťah dedičnosti, agregácie alebo asociácie medzi sebou.

Pri použití verziovacích systémov, ktoré podporujú modelovanie jemne zrnitých dát sú splnené dve požiadavky [3]:

1. Všetky modifikácie v dokumente sú reprezentované ako operácie nad stromovou štruktúrou syntaxe dokumentu.
2. Objekty v dokumente sú posúvané namiesto toho aby boli vymazávané a znovu vkladané.

Vďaka tomu máme prehľad nielen o tom, kto dokument menil, ale aj čo v ňom zmenil.



Obr. 5: Príklad jemne zrnitej štruktúry objektov [3].

Fáza nasadenia

V prípade fázy nasadenia ide o sledovanie zmien v jednotlivých komponentoch aplikácie. Tie sa pohodlne ukladajú do centrálneho úložiska, kým si niektorý z členov tímu nezmyslí prejsť na novšiu verziu. Vtedy sa zo svojho spánku prebudia všetky nové verzie, aby sa dostali do nového prostredia. Takto sú potom uložené jednotlivé zmeny v daných komponentoch a v prípade, že sa niečo pokazí, vrátíme sa k predchádzajúcej verzii alebo k verzii, ktorá bola posledná stopercentne funkčná a môžeme naše chyby napraviť. Použitie verziovacieho systému v tejto fáze je normálnym javom [3].

Vyhodnocovanie jednotlivých metrik súvisiacich s procesom vývoja

Ako môžeme vyhodnotiť jednotlivé metriky súvisiace s procesom vývoja softvéru pomocou verziovacieho systému? Na túto otázku odpovedá využitie napríklad systému Subversion a programu s ním spojeným, ktorý dokáže robiť viaceré štatistiky. Jednou z možností je napríklad vyhodnotenie množstva pridaného kódu. Príkazom „ukáž zmeny“ si vieme pozrieť zmeny, ktoré boli v úložisku vykonané. Tie sa môžu týkať jedného súboru, ale taktiež je možné prezrieť si rozdiely, ktoré vykonal jeden človek. Navyše, štatistiky sa dajú robiť v určených časových horizontoch. Vieme napríklad zistiť všetky zmeny, ktoré vykonal člen tímu od poslednej zmeny v úložisku [2].

Získavame tak prehľad nad prácou jednotlivých členov. Takisto nám pomáha približne určiť, koľko sa ktorý člen projektu venoval, koľko času približne strávil nad vyriešením konkrétneho problému. Vidíme aj frekvenciu prispievania jednotlivých členov.

Môžeme vyhodnotiť akým dielom jednotliví členovia prispeli k naplneniu istého cieľa [3]. To nám pomáha nielen monitorovať projekt a jeho smerovanie, ale pomocou toho vidíme, kto má s čím lepšie skúsenosti, komu ide práca, ako sa hovorí, viac od ruky a kto sa viac fláka alebo jednoducho povedané, nie je rovnako šikovný.

Vyššie opísané možnosti monitorovania sa sústredia hlavne na úsilie a s ním spojeným časom stráveným na vývoji systému. Ďalšou metrikou, ktorá nás zaujíma pri monitorovaní procesu vývoja softvéru, sú náklady a čas. Môžeme vychádzať z toho, že keď poznáme štatistiky práce jednotlivých členov a vidíme takisto, v ktorej oblasti vynikajú a v ktorej, naopak, nie sú až tak užitoční, vieme lepšie odhadovať aj s tým spojené náklady a čas.

Samozrejme, verziovací systém nie je až natoľko silným nástrojom, aby nám ukázal presný čas v hodinách strávených nad konkrétnym problémom, môžeme však odhadovať, že ak daný člen prispieval svojimi zmenami do verziovacieho systému v pondelok a následne vo štvrtok, vyriešenie problému, ktorým prispel, mu trvalo približne 3 človeko-dni. Na základe tohto vieme potom v budúcnosti vytvárať ďalšie plány, na základe ktorých sa bude monitorovať ďalší vývoj softvéru. Pomáha to aj pri rozdeľovaní úloh a alokovaní zdrojov potrebných pre ne.

Záver

Manažment verzií sa javí ako silný nástroj podporujúci monitorovanie smerovania nielen celého projektu, ale aj smerovania jednotlivých členov. Nevýhodou však zostáva, že ho nie je možné použiť úplne univerzálne, vo všetkých fázach vývoja a pri rôznych druhoch projektoch.

Pravdou však je, že jeho využitie sa môže stať užitočným nielen čo sa týka uchovávanía jednotlivých verzií kódu, ale istým spôsobom môže pozitívne pôsobiť aj na jednotlivých členov. To môže byť vyvolané tým, že každý človek, ktorý je monitorovaný, vykazuje iné výsledky ako človek, ktorý robí „na vlastnú päsť“. Pri práci v tímoch je veľmi dôležitou, ba až neoddeliteľnou, súčasťou vedomosť o jednotlivých členoch.

Pomáha to nielen na lepšie odhadovanie a plánovanie úloh, ale istým spôsobom sa tak znižujú aj riziká. Človek, ktorý je tlačný do dodržiavania termínov je disciplinovanejší a snaží sa viac ako človek, ktorý dostane úlohu a nikto sa nestará o to, kedy a na akej úrovni bude táto úloha splnená.

Dôležité je však podotknúť, že aj prílišná kontrola môže byť v niektorých prípadoch kontraproduktívna. Monitorovanie vývoja softvéru s ohľadom na jeho kvalitu stále bola a bude dôležitou úlohou, bez ktorej sa projekt môže vyskytnúť vo veľkých problémoch. No stále treba mať na pamäti, že je potrebné zvoliť nástroje, metódy a metriky na monitorovanie tak, aby u členov tímu nevyvolávali negatívne pocity.

Použitá literatúra

1. Bieliková, M.: Meranie v softvérovom projekte. Dostupné na <http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/prednasky/msi09-2009.pdf> . Dátum stiahnutia: 15.10.2010
2. Jones, C.: Using subversion as an aid in evaluating individuals working on a group coding project. *Journal of Computing Sciences in Colleges*, Vol. 25, No. 3 (2010) 18-23.
3. Kaur, P., Singh, H.: Version management and composition of software components in different phases of software development life cycle. *ACM SIGSOFT Software Engineering Notes*, Vol. 34, No. 4 (2009) 1-9.
4. Software metrics guide. Dostupné na http://sunset.usc.edu/classes/cs577b_2001/metricsguide/metrics.html#p3. Dátum stiahnutia: 15.10.2010
5. Zhao, F., Peng, X., Zhao, W.: Software Development Process Monitoring Based on Nominal Transformation. In *Proceedings of Eighth IEEE/ACIS International Conference on Computer and Information Science*, (2009) 983-988.

Annotation

Essay title in English

Monitoring of software project is important part, that help us to successfully end project. To monitor we cannot only have good vision, but we also need to choose the right tools. The right approach to monitoring of software project and the use of adequate tools can help to be on right way to aim our goals and also can guarantee the quality and supervision on the team members. In this essay, I would like to describe tools, that are used everyday in the team work and the way, this tools can be helpful for monitoring. As the main unit, version management is chosen. Everyone was in situation, when he something get wrong and he wanted to go back.