

# Interaktívna vizualizácia akcií vývojára softvérového produktu

*Výskumný zámer*

*Autor: Bc. Lukáš Sekerák*

*Vedúci: Alena, Kovárová, Mgr., PhD.*

## Motivácia

Pri vývoji softvérového produktu vzniká mnoho nezodpovedaných otázok, na ktoré sa nie vždy ľahko hľadá odpoveď. Mnohé vizualizačné techniky neponúkajú možnosť zobrazíť vývoj softvérového produktu z pohľadu jednotlivých krokov vývojára. V rámci tohto odkrokovaneho pohľadu je možné vizualizovať rôzne akcie vývojára pri programovaní.

Cieľom je teda použiť model bohatý na množstvo údajov (teda odkrokováných akcií) a vytvoriť preňho vizualizáciu. Vizualizáciu, ktorá by poskytla najlepšiu možnosť zobrazenia týchto akcií vývojára, respektíve výsledky určitých analýz.

Vďaka takejto vizualizácii, by sme dokázali odpovedať na viacero zaujímavých otázok pri vývoji softvérového produktu. Napríklad: "Ak vývojár často používa prehliadač (resp. portál) pri písaní svojho kódu, ako často je potom tento kód prepisovaný? Platí táto tendencia na všetky jeho kódy? Platí to u každého iného vývojára rovnako?"

Odpovede na tieto otázky, nám napríklad identifikujú problémy pri vývoji softvérového produktu. Pričom pri vývoji ďalších produktov nám pomôžu predchádzať týmto problémom, prípadne sa z efektívni vývoj. Zároveň tieto odpovede poskytujú objektívny feedback na jeho prácu.

## Analýza súčasného stavu problému

Problém vizualizácie vývoja softvéru môžeme rozdeliť do 2 častí. Prvá časť je vhodný model údajov, teda model, ktorý obsahuje dostatočné údaje na to, aby nám poskytol kvalitný pohľad na vývoj.

V súčasnom stave sa veľa riešení pozerá na vývoj softvéru analyzovaním informácií z rôznych systémov. Napríklad GEVOL nástroj využíva verziovací systém [1]. Takzvané využívajú ukladanie postupných verzií zdrojového kódu na spätné analyzovanie vývoja. Tieto jednotlivé komity analyzujú a rekonštruujú tak vývoj. V článku H. Gall a M. Lanza o vizualizácii a analyzovaní vývoja softvéru [3] tvrdia, že je dobré mať prepojený verziovací systém spolu so systémom na reportovanie bugov. Kde vďaka takému prepojeniu získame veľa užitočných informácií o vývoji.

O vývoj softvéru sa zaoberá aj výskumný projekt PerConIK (Personalized Conveying of Information and Knowledge, [perconik.fiit.stuba.sk](http://perconik.fiit.stuba.sk))<sup>[9]</sup>. Tento projekt zbiera jednotlivé kroky vývojára presne tak ako ich potrebujeme. Teda zbiera údaje o množstve stlačených klávesov, kliknutí myšou, otvorení prehliadača a mnohé ďalšie. Vďaka týmto údajom dokážeme vybudovať vhodný model údajov.

Tento model údajov je potrebné vhodne zobrazíť, čo je druhým veľkým problémom. S týmto problémom bojuje veľa existujúcich nástrojov, týchto nástrojov je však mnoho. Napríklad medzi všeobecné nástroje patria Gephi<sup>1</sup>, Prefuse<sup>[4]</sup>, Jung<sup>[7]</sup>, atď. Preto je potrebné spraviť dobrú analýzu nástrojov. Vizualizáciou vývoja softvéru sa priamo zaoberá hore spomenutý nástroj GEVOL [1], potom Gossip<sup>2</sup>. Ďalším podobným nástrojom s podporou vizualizácie je Lagrein<sup>[5]</sup>. Posledné dva nástroje zobrazujú vývoj softvéru v podobe grafov.

## Identifikácia problémov

### Veľa údajov

Pri vizualizácii má veľký zmysel zobrazovať najmä podstatné informácie. Projekt PerConIK zachytáva akcie vo vysokej granularite (napríklad kliknutie myšou). Čím tento model obsahuje veľa informácií. Pri veľkom počte informácii to je vážny problém, pretože tie obsahujú veľké množstvo elementov. Preto je potrebné tieto elementy – informácie prefiltrovať a zobrazíť len to podstatné.

### Neporiadok v grafe

Jeden problém súvisí s ľudskou pamäťou a Millerovým zákonom<sup>[6]</sup> a druhý s vizuálnym neporiadkom v grafe, pričom Ellis, Dix a Aris uvádzajú ako hlavný dôvod problému skutočnosť, že sa zobrazuje príliš veľa dát na príliš malej časti displeja [2][8]. Tento problém bude potrebné riešiť kvalitným používateľským prostredím, kde používateľovi zobrazím len to podstatné.

### Nájsť odpovede na otázky

Tretím problémom, pre nás najdôležitejším, je potreba sformulovať ďalšie otázky, podobné tým, ktoré sme si spomenuli v zadaní tak, aby sme dokázali na ne odpovedať. Na to aby sme na ne mohli odpovedať musíme sa najprv pozrieť na údaje v PerConIK-u, vybrať zaujímavé údaje a sformovať ich do modelu údajov. Následne v modeli údajov potrebujeme údaje prefiltrovať, podobné údaje zoskupiť a na základe otázky spraviť analýzu a výsledok.

<sup>1</sup> Domovská stránka Gossipu - <https://gephi.org/>

<sup>2</sup> Nástroj Gossip je časťou projektu PerConIK

## Literatúra a zdroje

- [1] COLLBERG, C. et al. A system for graph-based visualization of the evolution of software. In *Proceedings of the 2003 ACM symposium on Software visualization - SoftVis '03* [online]. New York, New York, USA: ACM Press, 2003. s. 77. [cit. 2013-12-07]. Dostupné na internete: <<http://dl.acm.org/citation.cfm?id=774833.774844>>.
- [2] ELLIS, G. - DIX, A. A Taxonomy of Clutter Reduction for Information Visualisation. In *IEEE Transactions on Visualization and Computer Graphics* [online]. 2007. Vol. 13, no. 6, s. 1216–1223. Dostupné na internete: <<http://eprints.lancs.ac.uk/12933/>>.
- [3] GALL, H. - LANZA, M. Software evolution: analysis and visualization. In ... *of the 28th international conference on Software ...* [online]. 2006. s. 1055–1056. [cit. 2013-12-07]. Dostupné na internete: <<http://dl.acm.org/citation.cfm?id=1134502>>.
- [4] HEER, J. et al. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* [online]. [s.l.]: ACM, 2005. s. 421–430. Dostupné na internete: <<http://portal.acm.org/citation.cfm?id=1055031>>.
- [5] JERMAKOVICS, A. et al. Visualizing software evolution with lagrein. In *Companion to the 23rd ACM SIGPLAN conference on Object oriented programming systems languages and applications - OOPSLA Companion '08* [online]. 2008. s. 749. Dostupné na internete: <<http://portal.acm.org/citation.cfm?doid=1449814.1449843>>.
- [6] MILLER, G.A. The Magical Number Seven. In SALA, S. DELLAEd. *The Psychological Review* [online]. 1956. Dostupné na internete: <<http://www.musanim.com/miller1956/>>.
- [7] O'MADADHAIN, J. et al. The JUNG (Java Universal Network/Graph) Framework. In *University of California Irvine California* [online]. 2003. Vol. 03, no. UCI-ICS 03-17, s. 3–17. Dostupné na internete: <[http://www.datalab.uci.edu/papers/JUNG\\_tech\\_report.html](http://www.datalab.uci.edu/papers/JUNG_tech_report.html)>.
- [8] SHNEIDERMAN, B. - ARIS, A. Network visualization by semantic substrates. In *IEEE Transactions on Visualization and Computer Graphics* [online]. 2006. Vol. 12, no. 5, s. 733–740. Dostupné na internete: <<http://www.ncbi.nlm.nih.gov/pubmed/17080794>>.
- [9] PerConIK. In [online]. [cit. 2013-12-07]. Dostupné na internete: <<http://perconik.fiit.stuba.sk/>>.