

Najčastejšie chyby pri vývoji softvéru*

Adam Melikant

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xmelikant@stuba.sk

12. Október 2020

Abstrakt

...

1 Úvod

V tomto článku sa budeme zaoberať najčastejšími chybami, ktoré môžu nastať pri komplexnom vývoji akéhokoľvek typu softvéru. Celý tento vývoj rozdelíme na jednotlivé celky popisujúce postupy vývoja softvéru. Tieto celky podrobne rozoberieme a poukážeme na jednotlivé nedostatky, ktoré následne môžu spôsobiť závažné problémy v praxi. Začneme od začiatku: Plánovanie potrebných informácií, zdrojov, cielený účel softvéru, rozšírenia atď. Následne rozoberieme štruktúru softvéru, chyby pri navrhovaní, programovaní či celkovej štruktúre, návrhu a funkcionalite. Dostatočne vysvetlíme zaužívané postupy, ktoré vedú k najčastejším chybám a aj to ako sa im efektívne vyhnúť prípadne čo najrýchlejšie opraviť aby sa následne zamedzilo kaskádovému vzniku ďalších problémov. Po úspešnom dokončení softvéru poukážeme na chyby v distribúcii a rôznych iných aplikáciách, ktoré sú častým nedostatkom hlavne z hľadiska dosiahnutia čo najväčšieho dopytu.

2 Zber Informácií

XXXXXXXXXXXXXx

3 Vyhládávanie údajov

::::::::::Djihdskjaflhskja

4 Štrukturovanie softvéru

XXXXXXXXXXXXXXXXXXXXXx

*Najčastejšie chyby pri vývoji softvéru, ak. rok 2020/21, vedenie: Vladimír Mlynarovic

5 Návrh štruktúry

::SCKAKHDKJLFASHFKJCNb

6 Programovanie SW

::SCKAKHDKJLFASHFKJCNb

7 Distribúcia a Propagácia softvéru

:XXXXXXXXXXXXXXXXXXXXXXXXXXXXxxx

8 Iná časť

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť ??), a potom na ešte nejaké (časť ??).¹

Môže sa zdať, že problém vlastne nejestvuje [1], ale bolo dokázané, že to tak nie je [2,3]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [4]. Dôležité veci možno *zdôrazniť kurzívou*.

9 Záver

Na záver zhrnutie:

Literatúra

- [1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [2] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [3] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.
- [4] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. http://www.sei.cmu.edu/productlines/frame_report/.

¹Niekedy môžete potrebovať aj poznámku pod čiarou.