

Princípy tvorby softvéru

Integrácia a nasadzovanie

Robert Lukočka

lukotka@dcs.fmph.uniba.sk

www.dcs.fmph.uniba.sk/~lukotka

M-255

- Integrácia a testovanie - v akom poradí?

- Integrácia a testovanie - v akom poradí?
- Integrácia pomáha odhaľovať defekty.
- Integrácia by mala prebiehať často.
- Časté nasadzovanie nových verzií zákazníkovi = rýchlejší feedback.

Ako dosiahnuť rýchlu integráciu?

- Continuous integration
- Continuous delivery
- Continuous deployment

Ako dosiahnuť rýchle nasadenie? Mimochodom nasadenie bez výpadku systému je architektonicky významná požiadavka.

- Single Source Repository
 - VCS má obsahovať všetko potrebné (v extrémnom prípade aj správnu verziu kompilera) pre zbuildovanie a spustenie projektu (s výnimkou vecí ako OS, JVM, databáza, kompilátor).
 - Minimal branches, stabilná mainline (Reasonable branches are bug fixes of prior production releases and temporary experiments.)
- Automatizovaný build
 - Vie zbuildovať všetko. A robí to rozumne.
 - IDE independent

- Make Your Build Self-Testing
 - Všetky úrovne testovania automatizované
- Everyone Commits To the Mainline Every Day
 - Rýchlejšie nájdenie zlých dohôd a defektov.
- Every Commit Should Build the Mainline on an Integration Machine
- Fix Broken Builds Immediately
 - Takže mainline je udržiavaná v “dobrom” stave
 - V prípade neúspešného testu je povinnosťou commitujúceho rýchla náprava situácie
 - Môžeme mať špeciálny server: CI server. Vizualne pomôcky pre členov tímu (OK / build beží / broken build / broken build, pridlho)

- Keep the Build Fast $< 10\text{min.}$
 - Na rýchlosti buildu treba pracovať.
 - Bottleneck testing
 - Deployment pipeline (alebo staged build),

- prebieha iba vybraná časť testov (commit build)
- commit build zabezpečuje aby sa nad mainlainom dal pomerne bezpečne robiť ďalší vývoj
- ďalšie testy môžu začať buď hneď po commite, alebo v pravidelných intervaloch
- testy, ktoré odhaľujú chyby pri ďalších testoch pridáme do commit buildu
- developeri môžu mať dvoj vlastný build, rýchlejší ako commit build

- Test in a Clone of the Production Environment
- Make it Easy for Anyone to Get the Latest Executable
- Everyone can see what's happening
- Automate Deployment (a prípadne automatický rollback)

- M. Fowler - Continuous Integration
- Continuous integration - Wikipedia