

Metodika na zdieľaný vývoj pomocou gitu

Štruktúra hlavného repozitára

Vytvorený je jeden hlavný repozitár tzv. origin, ktorý obsahuje jednu hlavnú vetvu a vedľajšie vetvy. Hlavnou vetvou je Master vetva a vedľajšie vetvy.

Master Vetva

Je to hlavná vetva. Obsahuje produkčnú verziu aplikácie. Vždy obsahuje stabilnú a funkčnú verziu aplikácie. Do tejto vetvy sa môže pridávať nový kód len z vedľajších vetiev, ktoré boli schválené po code review.

Vedľajšie vetvy

Tieto vetvy slúžia pre developerov na pushovanie nových častí kódu. Táto vetva sa nachádza tiež v repozitári origin a po schválení sa pushne do vetvy Master.

Každá vetva sa bude začínať veľkým písmenom. V prípade viacerých slov budú slová oddelené podčiarkovníkom. Názov branche by mal vyjadrovať jej obsah a jej názov bude písaný v angličtine.

Nová vetva sa vytvorí v nástroji tfs, kde sa dá jednoznačný názov v angličtine a určia sa tasky, ktoré sa do branche budú comitovať, čo je v nástroji umožnené. Na názvy sa používa Camel case.

Príklad na názov vetvy na tvorbu menu:

MenuCreation

Zlúčenie vedľajšej vetvy do Master vetvy

Po daní pull requestu na vedľajšiu vetvu, je potrebné túto vetvu schváliť. Schvaľovať môže iba niekto, kto daný kód nevytvoril. Code review pre vetvu musia urobiť minimálne dvaja ľudia, ktorý tento kód musia schváliť. Po schválení môže byť Pull request schválený a dokončený.

Commitovanie

Commitovať treba ucelené a ideálne funkčné celky kódu.

Commit message

V prvom riadku commit správy by sa mal nachádzať názov commitu. Tento názov by mal jasne vyjadrovať čo obsahuje commit.

Pod nadpisom sa bude nachádzať krátky opis commitu, ktorý by mal byť stručný a jasne vyjadrovať obsah zmien a pridaného kódu.

Správy commitov sa budú písať v anglickom jazyku.

Príklad:

Bootstrap addition

I created css for all forms and add bootstrap.

Revertovanie zmien

Na odstránenie súboru z commitu, bez odstránenia zmien sa použije reset Head názov_súboru.

Ak je potrebné zabudnúť zmeny použije sa príkaz checkout názov_súboru.

V prípade chyby v commite, nie je potrebné vytvoriť ďalší commit, ale treba použiť príkaz commit amend, ktorý prepíše posledný commit.

Push

Po vytvorení commite je potrebné zmeny vložiť do vytvorenej vetvy. Toto sa urobí príkazom Push.

Stiahnutie si novej verzie Master vetvy

Keď sa spojí vedľajšia vetva a hlavná vetva, teda vznikne nová verzia Master vetvy, je potrebné, aby všetci pracujúci na projekte mali najnovšiu verziu Master vetvy. Toto urobia pomocou príkazu Update. V prípade, že medzi Master verziou kódu a verziou kódu, ktorý robí Update nie je konflikt, vetva sa automaticky sama obnoví. V druhom prípade je potrebné vyriešiť konflikty a urobiť merge.

Príkaz Update obsahuje pull request a následne urobí merge projektu. V tomto príkaze je možné nastaviť aj to či sa bude vykonávať Rebase alebo nie. V našom prípade sa bude robiť Rebase. To znamená, že vyberieme možnosť Rebase v okne, ktoré nám zobrazí po výbere príkazu Update project.

Mergeovanie

V prípade, že má niekto rozpracovanú zmenu a do vetvy master bola medzičasom schválená nová verzia, je potrebné svoje zmeny najprv commitnúť a až potom je možné získať novú verziu. Konflikty v kóde si musí vyriešiť autor kódu.

Pull Request

Title -> jasný nadpis pre pull request

Description -> Ako prvé bude opis, čo sa v pull requeste nachádza. Ďalšie budú pridané súbory a následne zmeny, ktoré sa urobili v ostatných už existujúcich súboroch. Tieto opisy nemusia byť dlhé, stačí jasný, krátky opis. Ďalej sa v pull requeste dá určiť, na aké úlohy je daná bracha zameraná. Toto je potrebné tam určiť.

Všetko sa píše v angličtine

Príklad:

Sidebar menu creation and bootstrap addition

I added bootstrap to all forms and create new sidebar menu with css.

Added folder static for css files.

Added file default.css for menu stylesheet.

Modified default.html- added sidebar menu.

Pomoc pri zadávaní príkazov:

<https://www.jetbrains.com/help/pycharm/using-git-integration.html>

<https://www.jetbrains.com/help/pycharm/using-git-integration.html>

Zdroje:

http://labss2.fiit.stuba.sk/TeamProject/2014/team17is-si/metodika_git.pdf

<https://git-scm.com/docs>