

Fakulta informatiky a informačných technológií
Slovenská technická univerzita

Vnímanie neviditeľného [Holographic Eyes]

Metodika pre manažment verzí kódu (angl.)

Číslo tímu: 8

Názov tímu: caneless

Vedúci tímu: Ing. Martin Tamajka

Členovia tímu: Bc. Zsuzsanna Bernáth, Bc. Jakub Domian, Bc. Andrej Hucko, Bc. Dušan Janeček, Bc. Ján Karaffa, Bc. Ľudovít Popelka, Bc. Ľubomíra Trnavská

Študijný program: Inteligentné softvérové systémy

Predmet: Tímový projekt

Akad. rok: 2018/2019

Zodpovedná osoba: Bc. Ľudovít Popelka

Content

Content	1
Introduction	2
Inspiration	2
Branching	2
Model	2
Branches	2
master	2
develop	2
Feature branch	3
Tagging	3
Feature	3
Refactoring	3
Fixing	3
Testing	3
Nomenclature	3
Branch naming	3
Example branch	4
Commit message	4
Example commit	4
Gitlab	4
Pull request	4
Sources	5

Introduction

This document sets version control conventions for team developers. Used version control tool is Git.

Inspiration

Our approach builds heavily on GitFlow branching model for Git.

Branching

Model

We reduce Gitflow model to 2 stable and feature branches.

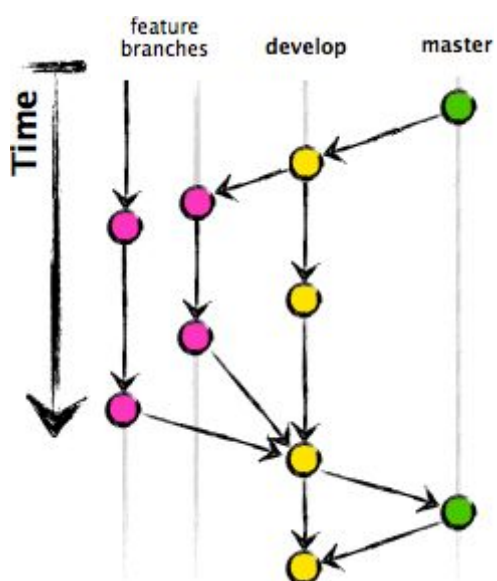


Fig. C1 (taken from GitFlow, edited): Branching model

Branches

master

Contains deployed application. Updated once per Scrum sprint.

develop

Contains master plus new, not deployed features. Updated after a user story has been done.

Feature branch

1 per user story. Contains user story feature.

Tagging

Each task has a single type.

Feature

tag **feat**

Purpose: new feature, edit feature, documentation

Refactoring

tag **ref**

Purpose: code refactor - no functionality change!

Fixing

tag **fix**

Purpose: bug fix, malicious/spaghetti code treatment, performance improvement

Testing

tag **test**

Purpose: add missing tests, edit tests

Nomenclature

No specific separator is mandatory further, can be dash or underscore.

Branch naming

Here we talk about feature branches only.

<developer initials> - <user story id> - <user story name>

User story information from VSTS.

Example branch

John Doe develops a user story with ID 1234 and name Create website:

jd-1234-create-website

Commit message

<developer initials> - <tag> - <task id> - <keywords>

Task information from VSTS. For tags see chapter Tagging. Keywords up to 12 words, inspired by task name.

Example commit

John Doe pushes commit after solving a task with ID 1234 and name Add footer:

jd-feat-1234-add-footer

Gitlab

We use Gitlab within Visual Studio Team Services (VSTS).

Pull request

Requested by branch owner (= developer responsible for the user story) after all associated tasks have been closed. Completed by anybody after all other requirements fulfilled (i.e. code review for example).

After pull request has completed, the requesting feature branch is deleted. Does not apply to master or develop.

Sources

DRIESSEN, V. Introducing GitFlow [online] 2018. Available from:

<https://datasift.github.io/gitflow/IntroducingGitFlow.html>

GAŠPAR, P. Kontinuálna integrácia [online] In: Manažment v tvorbe softvéru 2018/19.

Available from: <http://www2.fiit.stuba.sk/~simko/manazment/p07-qm1-FIIT-PG-2018.pdf>

abralvalheri. Commit Message Guidelines [online] In: GitHubGist. Available from:

<https://gist.github.com/abralvalheri/34aeb7b18d61392251a2>

Microsoft. Visual Studio Team Services [online] Available from:

<https://visualstudio.microsoft.com/team-services/>