

# Kolaborativní vývoj

DELTA - Střední škola informatiky a ekonomie, s.r.o.

Ing. Luboš Zápotočný

15.09.2025

CC BY-NC-SA 4.0

# **Workflow a nástroje spolupráce**

# Git a hostingové platformy

Pro Git existuje celá řada hostingových platform, které vám pomůžou spolupracovat na projektech

Nejznámější jsou například

- GitHub
- GitLab
- Bitbucket
- Gitea

# Role uživatelů v GitLabu

GitLab umožňuje nastavit následující role uživatelům:

- Guest
- Planner
- Reporter
- Developer
- Maintainer
- Owner

# Role uživatelů v GitLabu

**Každá role má jiná** přístupová práva a **oprávnění**

Například Guest až Reporter **nemohou** vytvářet projekty ani podskupiny, Developer a dále už **mohou**

Podskupiny **mohou** vytvářet pouze uživatelé s rolí Maintainer nebo Owner

**Některá oprávnění se týkají větví** a lze například nastavit, že do hlavní větve **mohou mergovat pouze** uživatelé s rolí Maintainer nebo Owner

# Role uživatelů v GitLabu

Více o rolích najdete na <https://docs.gitlab.com/user/permissions>

# Projektový management

GitLab a většina dalších nástrojů podporuje **základní sadu** funkcí **pro projektový management**

- Issues (zadáávání úkolů, evidence práce)
- Issues board (vizuální workflow pro issues)
- Milestones (projektové cíle na nějaké období)
- Wikis (dokumentace projektu)
- Deployments (ucelené informace o releasovaných změnách)

# Projektový management

Často se ale setkáte s externími nástroji pro projektový management, které jsou poté propojené s GitLabem

Nejčastěji se setkáte s nástroji jako například

- Jira
- Trello
- Asana



# Větvení, branch protection, merge requesty

**Branch protection** je metoda nastavení oprávnění a pravidel pro důležité větve jako například již zmíněné pravidlo o tom, kdo může mergovat nebo pushovat do hlavní větve

Systémy ale často umožňují nastavit i **pravidla pro merge requesty**

Jedním z často používaných scénářů je, že změny z merge requestu nemohou být začleněny pokud nejsou splněny následující podmínky

- CI/CD pipeline je zelená = build a testy procházejí
- Změny zrevidoval alespoň jeden kolega

# Merge requesty

Merge requesty jsou nadstavba příkazu `git merge`

Poskytují možnost vývojářům requestovat změny, efektivně konzultovat s vývojáři a hlavně kontrolovat co se do projektu přidává

GitHub i GitLab dokonce umožňují v rámci MR provést **rebase** a nebo **squash merge**

Squash merge je metoda, která místo všech commitů v MR vytvoří pouze jeden commit s popisem všech změn

Merge requests může být explicitně označen jako **draft**

# Merge requesty

Rebase je metoda, která místo všech commitů v MR vytvoří pouze jeden commit s popisem všech změn

# CI/CD pipeline

GitHub i GitLab umožňují spouštět automaticky takzvané **CI/CD pipeline**

Pipeline je sada kroků, které se mají provést na změnách v MR

Kroky mohou být například:

- Build
- Test
- Deploy

CI/CD budeme více řešit na konci pololetí

# Tagy a releasy

Tagy jsou speciální značky v Gitu, podobě jako větve

Fixně a **dlouhodobě** uchovávají odkaz na historii repozitáře = neměly by se měnit nebo mazat, což neplatí pro větve