



VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

FAKULTA
ELEKTROTECHNIKY
A INFORMATIKY

KATEDRA
INFORMATIKY

Základy informačních technologií

Ing. Michal Radecký, Ph.D. MBA

Verzovací systémy, GIT



Co jsou verzovací systémy

- Verzovací systémy jsou nástroje, které slouží k sledování a uchovávání změn v kódu, dokumentech nebo jakýchkoli souborech během vývoje projektu.
- Tyto systémy umožňují vývojářům a ostatním členům týmu spolupracovat nad „stejnými“ daty.

Přínosy verzovacích systémů

- Sledování změn a evidence historie
- Obnova v rámci historie
- Spolupráce při tvorbě (multi-uživatelský přístup)
- Paralelní vývoj
- Snížení rizika chyb a konfliktů

Typy verzovacích systémů

Lokální

- Pracují pouze na jednom počítači.
- Každý vývojář udržuje svou vlastní kopii projektu.
- RCS (Revision Control System)

Centralizované

- Všechny soubory a jejich historie jsou uloženy na centrálním serveru.
- Vývojáři pracují se soubory, které stahují a odesílají (sdílí se serverem).
- Jedno kritické místo v případě výpadku apod.
- CVS (Concurrent Versions System), Subversion (SVN)

Decentralizované

- Každý vývojář má kompletní kopii repozitáře, včetně historie.
- Umožňuje paralelní vývoj.
- Git

Hlavní funkce

Commit (Potvrdit)

Ukládání změn do repozitáře s krátkým popisem provedených změn.

Branches (Větve)

Vytváření větví pro paralelní vývoj bez ovlivnění hlavní větve.

Merge (Sloučení)

Sjednocení změn z jedné větve do jiné.

Pull/Push

Aktualizace lokálního repozitáře o změny z centrálního repozitáře nebo naopak.

Clone

Převod/klonování repozitářů – vytvoření lokálního ve vazbě na vzdálený repozitář.

Git

- Git je distribuovaný/decentralizovaný verzovací systém používaný pro sledování změn v kódu a správu verzí souborů během vývoje softwaru.
- Byl vyvinut Linusem Torvaldsem v roce 2005 a od té doby se stal jedním z nejpoblárnějších nástrojů tohoto typu.
- Git je otevřený software a je zdarma k používání.
- Umožňuje podporu kolaborativní práce a snadné sdílení kódu přes vzdálené repozitáře (GitHub, GitLab, Bitbucket).
- Široká Podpora různých nástrojů a IDE.

<https://code.visualstudio.com/docs/sourcecontrol/overview>



SOURCE CONTROL



The folder currently
repository. You
will enable sou

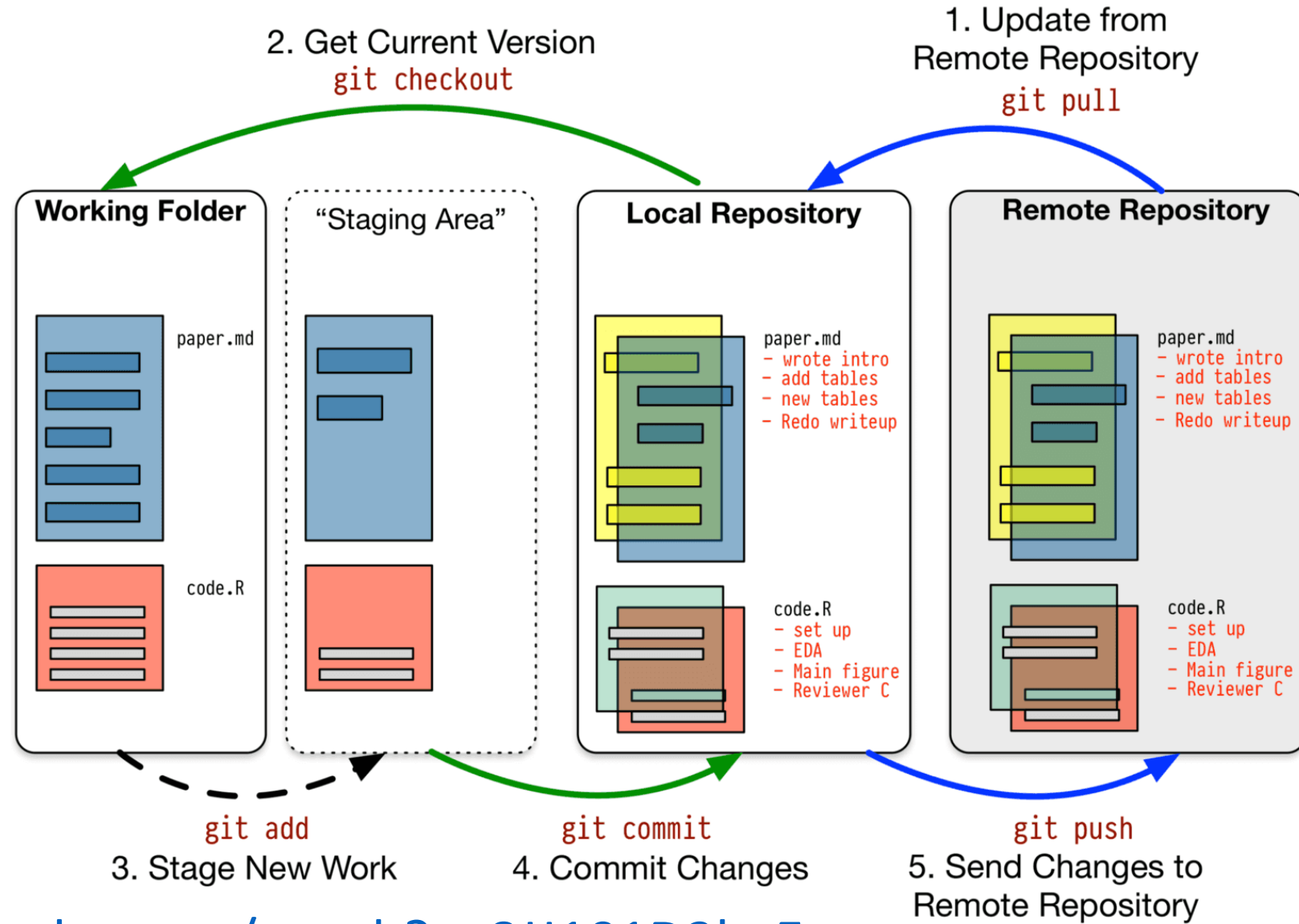


To learn more a
control in VS C



You can also di
GitHub repository
access to source
and GitHub.

Hlavní princip



<https://www.youtube.com/watch?v=8JJ101D3knE>

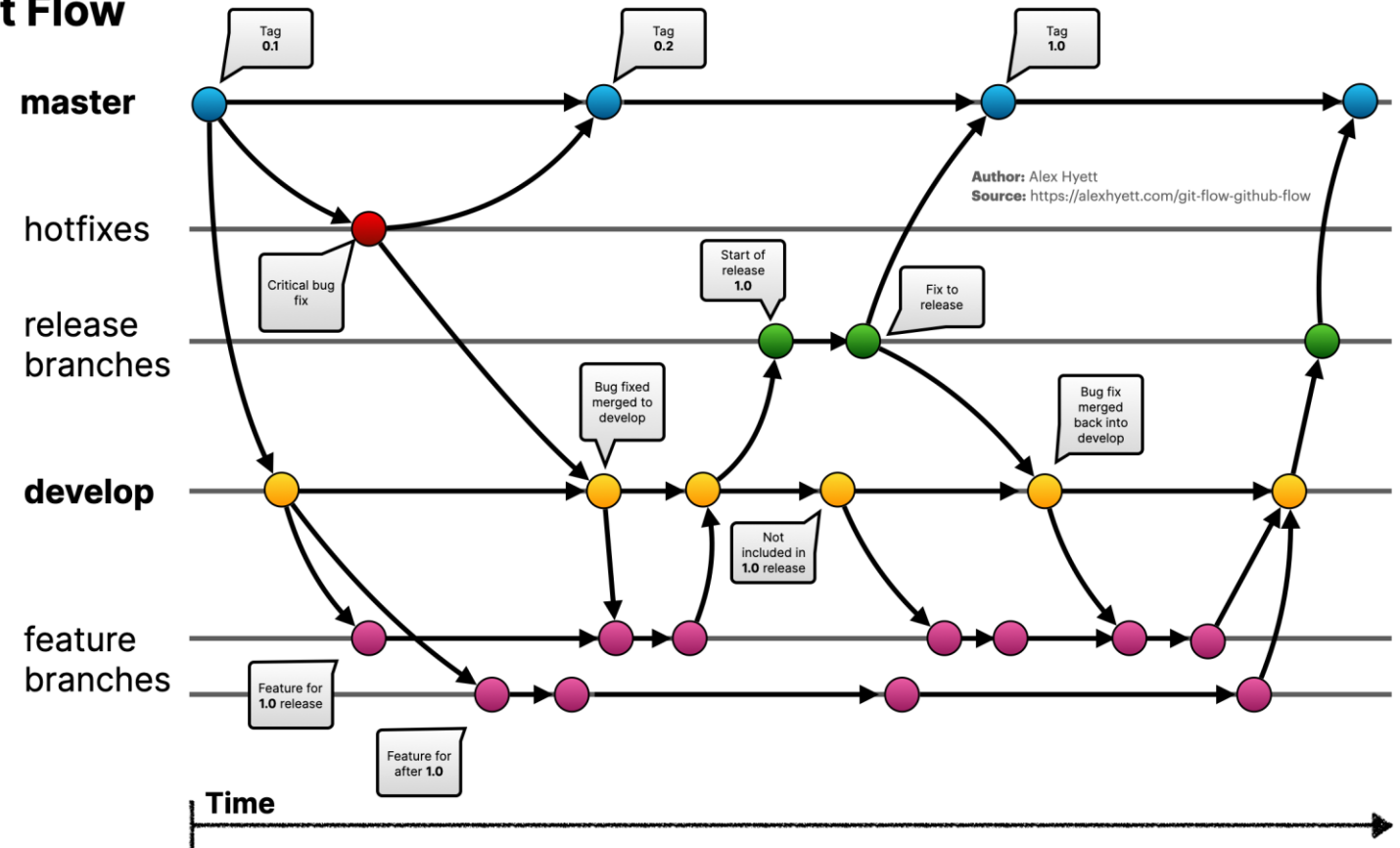
Staging Area

- Index/stavový stroj - klíčový koncept, který umožňuje přesné řízení toho, které změny budou zahrnuty v dalším commitu.
- Funguje jako přípravný prostor pro commit a další fáze procesu.
- Změny, které jsou zde přidány, jsou označeny, ale ještě nebyly zapsány do samotného lokálního repozitáře.
- Označování je možné řídit „add“, což přináší flexibilitu mezi pracovní verzí a commitem (pracovním adresáři x staging area x commit)

Git Flow

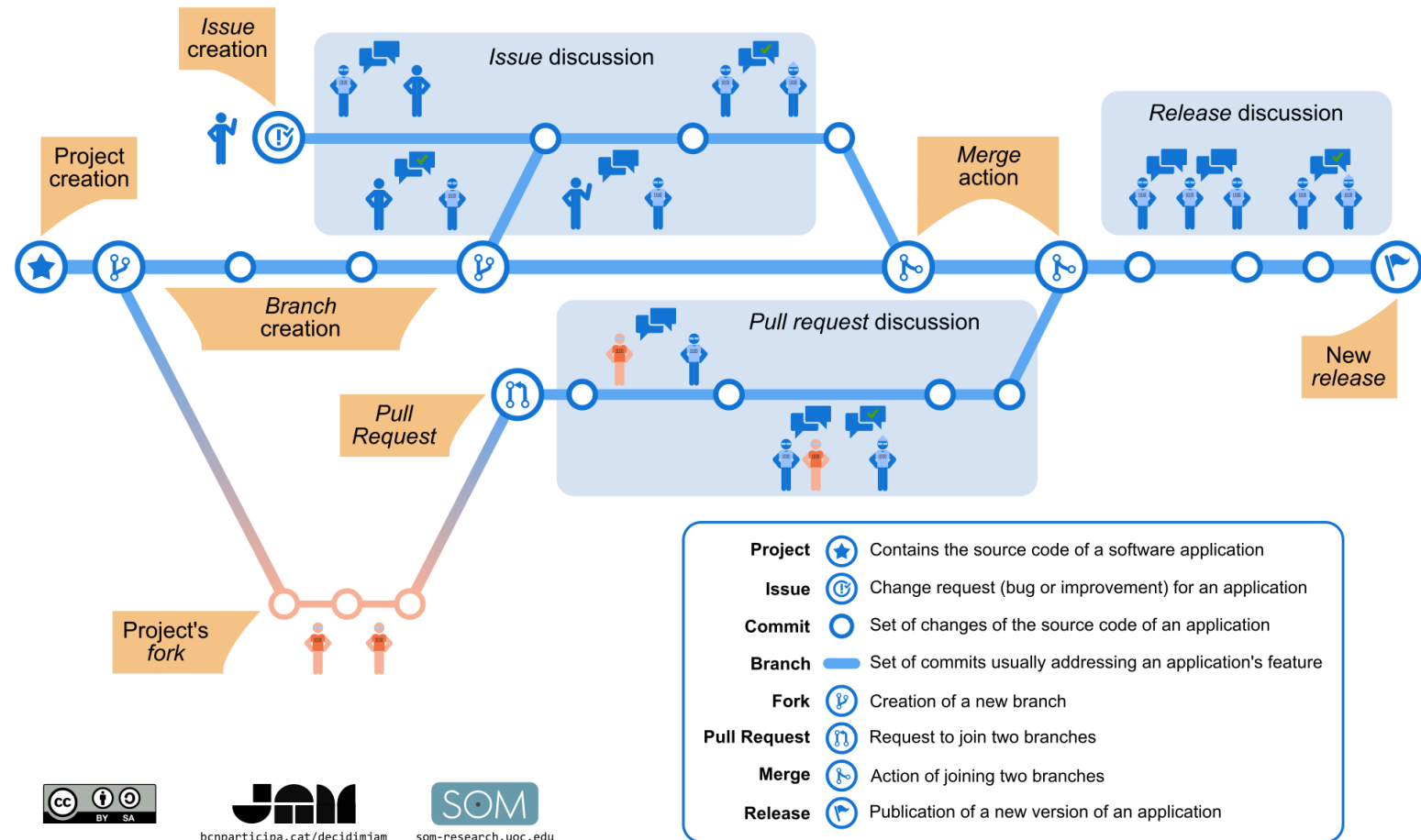
- Efektivní verzování vývoje
- Historie vývoje, dokumentace
- **Branch (větev)** (master/main) způsob, jakým se rozděluje projekt do nezávislých kopií s izolovaným vývojem
- **Merge (spojení)** spojení dvou větví
- **Fetch, Pull, Push** procesy pro synchronizaci mezi typy „úložišť“ s ohledem na definované větve apod.

Git Flow



GitHub Flow

- Efektivní kooperace na vývoji
- Kontrola vytvořeného kódu



- **Pull request**
požadavek na sloučení změn z jedné větve do jiné, typicky v rámci týmové práce
po schválení/kontrole se provede merge mezi větvemi

Instalace Gitu

- Samotný git vs. vizuální „doplňkové nástroje“
 - Git BASH, Git GUI, GitKraken, TortoiseGit, ...
- Instalace
 - Windows - <https://git-scm.com/download/win>
 - macOS – např. `brew install git`
 - Linux – např. `sudo apt update, sudo apt install git`
- Konfigurace
 - `git config --global user.name "Tvé Jméno,,`
 - `git config --global user.email tvuj@email.com`
- Konfigurace
 - `git config --list`
 - `git --version`

Příkazy

git init



Initializes a new Git repository in the current directory.

```
• • •  
$ mkdir railsapp  
$ cd railsapp  
$ git init
```

git add



Stages changes in the current directory and sub directories.

```
• • •  
$ echo "# RAILS APP" > README.md  
$ echo "Demo" >> README.md  
$ git add README.md
```

git commit



Commits staged changes with a commit message.

```
• • •  
$ git add app.css  
$ git commit -m "Initial commit"
```

git push



Pushes local changes to a remote repository.

```
• • •  
$ git add .  
$ git commit -m "Update files"  
$ git push origin main
```

git pull



Pulls changes from a remote repository and merges into the local repository.

```
• • •  
$ git status  
$ git pull origin main  
$ git log --oneline
```

git remote



Add a remote repository, view it, and rename it.

```
• • •  
$ git remote add origin <url>  
$ git remote -v  
$ git remote rename origin upstr
```

Příkazy

clone, diff, revert, log, switch, ...

git branch



List all branches, create a new branch with a specified name, and confirm it's created.



```
$ git branch
$ git branch feature-login
$ git branch
```

git fetch



Retrieves the latest data from a remote repository - but it doesn't integrate any of this new data into your working files.



```
$ git fetch origin
```

git checkout



Switches to the specified branch.



```
$ git branch
$ git checkout feature-login
$ git branch
```

git merge



Merges the specified branch into the current branch (in this case main).



```
$ git checkout main
$ git merge feature-login
$ git log --oneline
```

git status



Displays the status of the repository, including any uncommitted changes



```
$ git status
$ echo "Data Added" >> hello.txt
$ git status
```

git reset

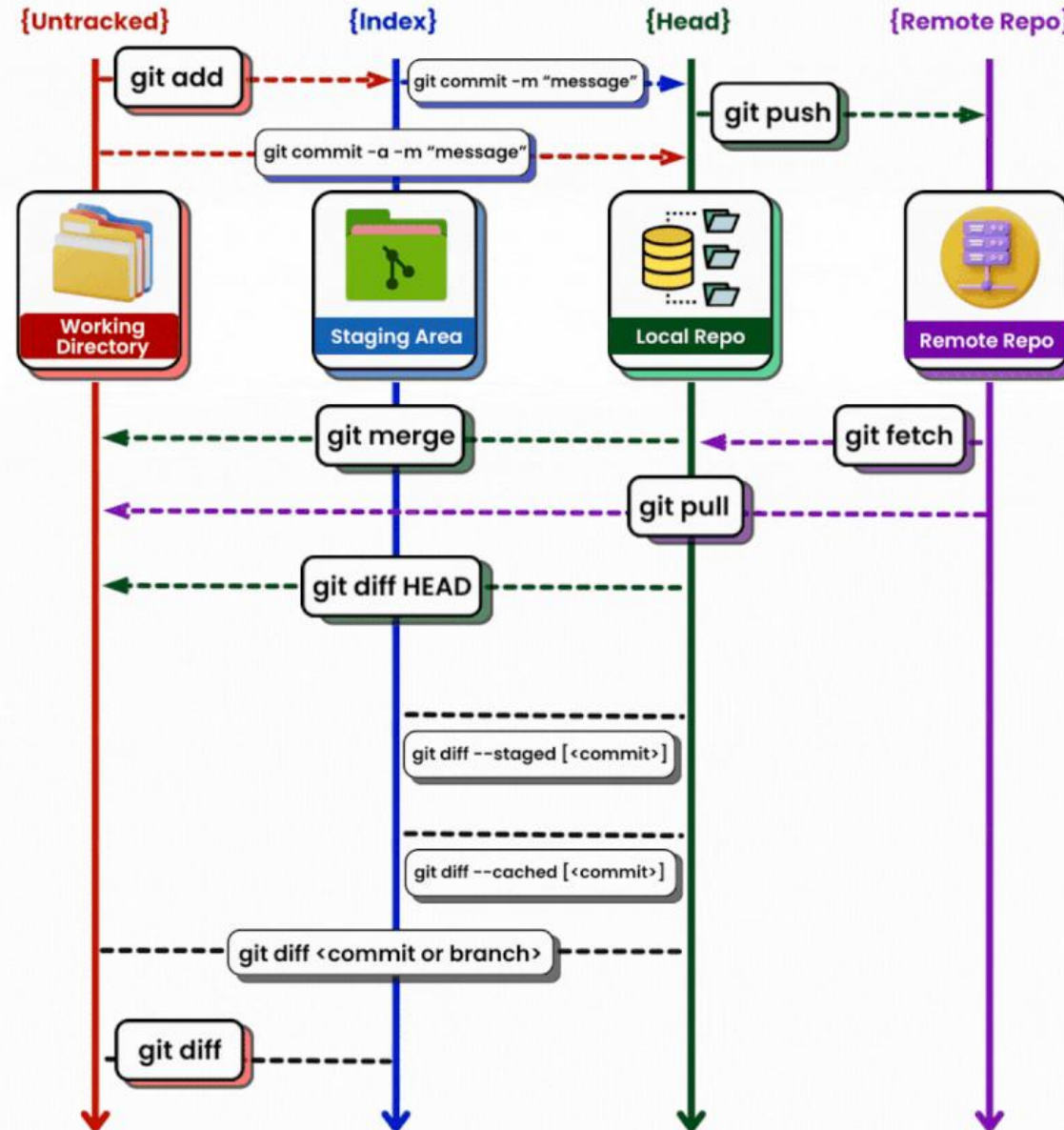


Resets the current branch to the specified commit



```
$ git log --oneline
$ git reset --hard <commit-hash>
$ git status
```

Git Workflow



GitHub

GitHub je webová platforma pro správu verzí a hosting softwarových projektů s využitím platformy Git.

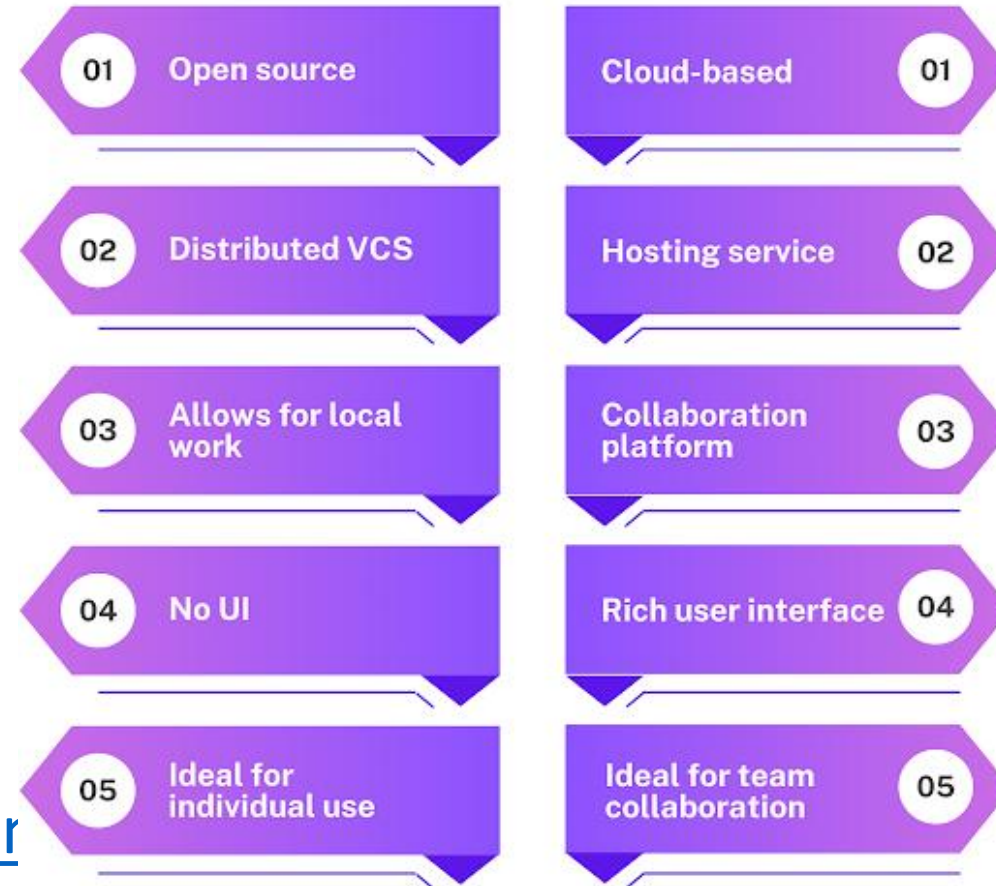
Jedná se o jeden z nejpobulárnějších (420+ mil) repozitářů (Remote Repository) pro vývojáře, kteří chtějí sdílet svůj kód, spolupracovat na projektech a sledovat historii změn.

Alternativy: GitLab, Bitbucket, Azure DevOps

<https://github.blog/developer-skills/programming-frameworks/what-is-git-our-beginners-guide-to->

The Git vs GitHub

What's The Differences



GitHub

Repositáře

- Uživatelé mohou vytvářet repozitáře pro své projekty. Repozitář obsahuje všechny soubory projektu a historii změn.

Stars and Forks

- Uživatelé mohou hodnotit nebo si zakládat vlastní projekty/repositáře z již existujících.

Issues a Pull Requests

- GitHub poskytuje nástroje pro sledování problémů (issues) a návrhů změn (pull requests). To umožňuje efektivní komunikaci a spolupráci v rámci týmu.

Collaborators a Teams

- Uživatelé mohou spolupracovat na projektech přidáváním spolupracovníků nebo vytvářením týmů s přístupovými právy.

Wikis a Projects

- Wikis umožňují tvorbu dokumentace a projektové stránky. Projects nabízí nástroje pro sledování úkolů a organizaci práce.

Gists

- Umožňuje uživatelům sdílet a diskutovat o malých kusech kódu, textu nebo obrázcích.

GitHub Actions

- Poskytuje nástroje pro automatizaci procesů v rámci repozitáře, například sestavení, testování a nasazení.

GitHub Marketplace

- Platforma pro nalezení a integrování nástrojů a aplikací do GitHub repozitářů.

Security a Code Scanning

- Nástroje pro bezpečnostní skenování kódu a zabezpečení repozitářů.

Copilot

- Integrace umožňující využívat prvky AI při vývoji a řízení projektů.