

Vývoj Internetových Aplikací

Nástroje pro vývoj

Ing. Michal Radecký, Ph.D.

www.cs.vsb.cz/radecky

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.

Přínos verzovacích systémů

- Sledování změn a evidence historie
- Obnova v rámci historie
- Spolupráci při tvorbě
- Paralelní vývoj
- Snížení rizika chyb a konfliktů

Typy verzovacích systémů

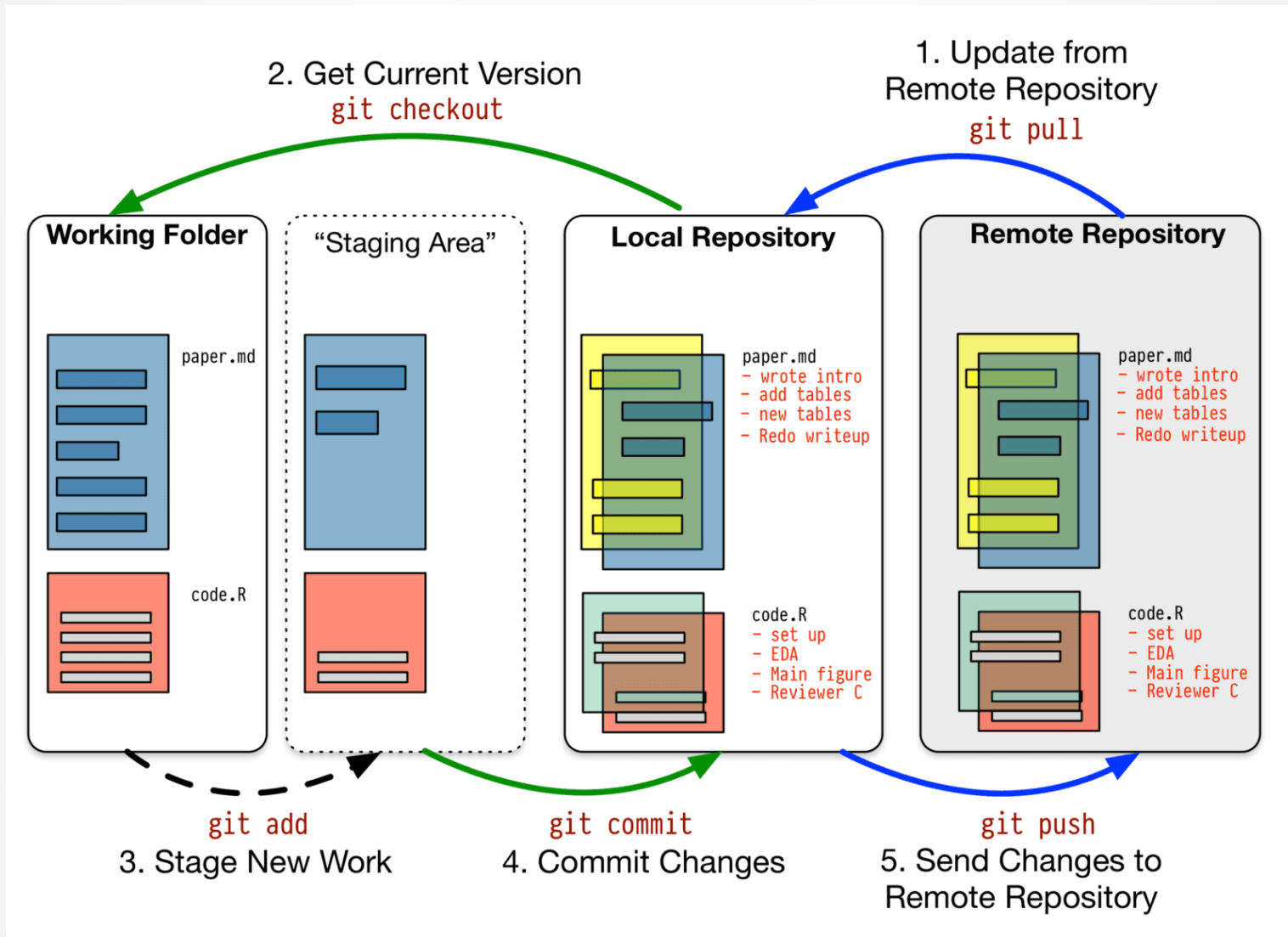
- Lokální
 - Pracují pouze na jednom počítači.
 - Každý vývojář udržuje svou vlastní kopii projektu.
 - RCS (Revision Control System)
- Centralizované
 - Všechny soubory a jejich historie jsou uloženy na centrálním serveru.
 - Vývojáři pracují se soubory, které stahují a odesílají (sdílí se serverem).
 - 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

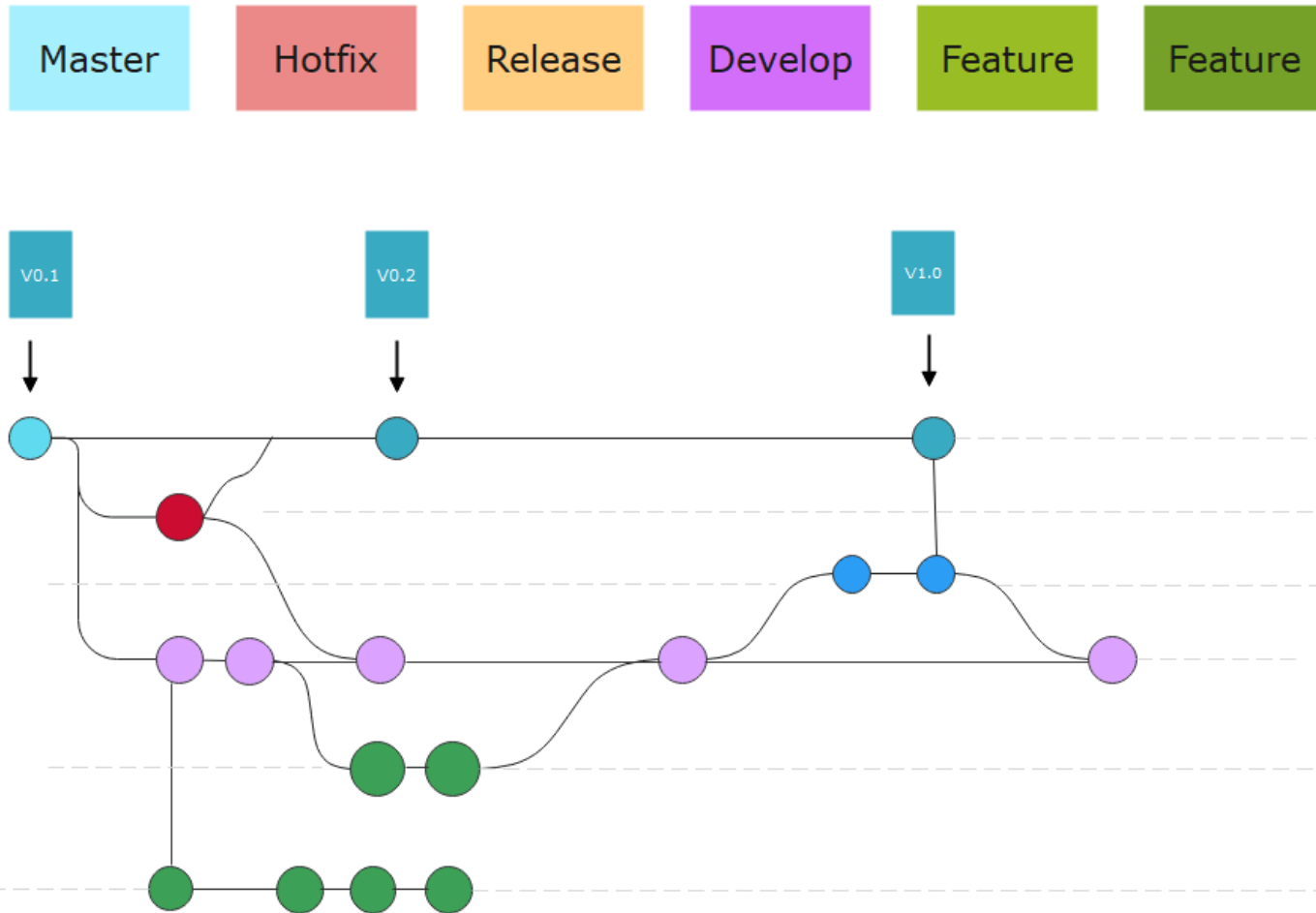
- Git je distribuovaný verzovací systém používaný pro sledování změn v kódu a správu verzí souborů během vývoje softwaru.
- Byl vyvinut Linusem Torvaldsem v roce 2005 a od té doby se stal jedním z nejpoblárnějších nástrojů tohoto typu.
- Git je otevřený software a je zdarma k používání.



Git

- 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.
 - 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.

Git Workflow Diagram



GitHub

- GitHub je webová platforma pro správu verzí a hosting softwarových projektů.
- Jedná se o jeden z nejpobulárnějších repozitářů (Remote Repository) pro vývojáře, kteří chtějí sdílet svůj kód, spolupracovat na projektech a sledovat historii změn.
- https://madnight.github.io/githut/#/pull_requests/2023/3

GitHub

- **Repositáře**
 - Uživatelé mohou vytvářet repositáře pro své projekty. Repositář 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 repositáře, například sestavení, testování a nasazení.
- **GitHub Marketplace**
 - Platforma pro nalezení a integrování nástrojů a aplikací do GitHub repositářů.
- **Security a Code Scanning**
 - Nástroje pro bezpečnostní skenování kódu a zabezpečení repositářů.
- **Copilot**
 - Integrace umožňující využívat prvky AI při vývoji a řízení projektů.

„Balíčkové“ nástroje

- Jedná se o nástroje, které hrají klíčovou roli při vývoji rozsáhlých, typicky webových projektů (javascript).
- Jejich hlavním úkolem je zajistit efektivní správu návazností jednotlivých „komponent“ a podporu vývojářských procesů.
- Vychází z principů ekosystému node.js.

„Balíčkové“ nástroje

- Specifikace závislostí a konfigurace (package.json)
- Instalace závislostí/balíčků
- Lokální vs. Globální kontext (workspace)
- Údržba a správa závislostí a verzí (stavový stroj)
- Hlídaní verzí v rámci vyvíjené aplikace (lock)
- Vazba na veřejné i privátní repozitáře (Registry)

- **npm, yarn, ...**

Workflow nástroje

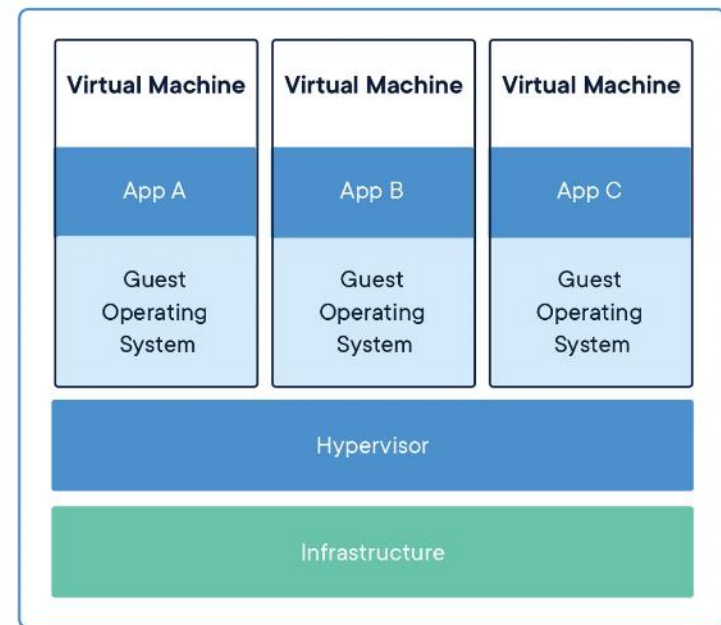
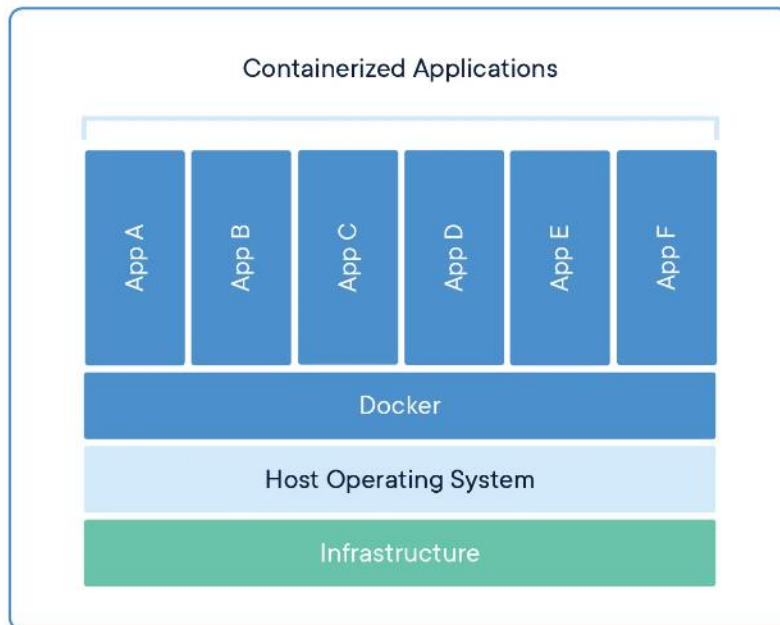
- Nástroje, které umožňují „kompilaci“ a zajištění průběhu „kompilačního“ procesu v rámci vývoje webových aplikací.
- Jedná se v principu o nástroje pro tvorbu balíčků (bundler) a spouštěče úloh (task runner).
- Mohou být navázány na konkrétní technologie (javascript, CSS, TS) nebo obecné.
- Pomocí konfiguračních „skriptů“ se definuje požadovaný výstup a postupy.
- Mohou pokrývat i optimalizace, modulární zpracování, minifikaci, atd.
- **Webpack, Gulp, Grunt, Parcel, ...**

Kontejnerová běhová prostředí

- Platformy pro zajištění běhu ekosystému dané aplikace v definovaném prostředí.
- Izolovaný kontejner obsahující komplexní aplikaci, ale i další nástroje pro její běh (runtime, knihovny, data závislosti, atd.)
- Nejedná se o virtualizaci, ale může být využita vč. infrastruktury.
- Zahrnují zabalení, distribuci a provoz běhového prostředí.
- **Docker, Kubernetes** (orchestralizace)

Docker

- Docker se stal standardem v oblasti kontejnerizace a je široce využíván pro nasazování, testování a správu aplikací v různých prostředích.
- Je oblíbený pro svou lehkost, přenosnost a schopnost rychle a opakovatelně nasazovat aplikace ve standardizovaných kontejnerech.



Docker

- Docker Image
 - Docker Image je základní stavební jednotkou. Jedná se o snímek (snapshot) aplikace, včetně všech potřebných souborů, nastavení a závislostí pro její běh. Images jsou definovány v souborech nazývaných Dockerfile, které popisují postup, jak vytvořit image.
- Docker Compose
 - Nástroj využívající konfigurační soubor (YAML), pomocí kterého zajistí spuštění a běh více-kontejnerového prostředí (služby, infrastruktura, zdroje).
- Docker Container
 - Docker Container je běžící instance Docker Image. Kontejner vytváří izolované prostředí pro aplikaci, ve kterém může běžet nezávisle na hostitelském systému. Kontejnery jsou spustitelné, sdílitelné a opakovatelné, což zajišťuje konzistentní běh aplikace na různých prostředích.
- Docker Registry
 - Docker Registry je úložiště pro sdílení Docker Images. Mohou být interní nebo veřejná (Docker Hub).
- Dockerfile
 - Dockerfile je soubor obsahující instrukce pro vytvoření Docker Image. Popisuje postup instalace a konfigurace potřebných komponent pro běh aplikace v kontejneru.
- https://www.youtube.com/watch?v=_dfLOzulg2o