1. Verzovací systémy
   1. Účel verzovacích systémů
      1. Zachování historie souborů
         1. Možnost vrácení změn
      2. Možnost udržování více verzí najednou
         1. Produkční, vývojová, testovací
         2. Vyzkoušení nové funkcionality bez ztráty funkční verze
      3. Vývoj v týmu: řešení konfliktů
   2. Základní pojmy
      1. Správce verzí
         1. Má na starost automatické číslovaní verzí
         2. Sada nástrojů pro přístup k verzím
      2. Repositář
         1. Místo, kde uloženy verzované soubory
      3. Lokální kopie
         1. Lokální kopie repositáře (nebo je části)
      4. Příkazy
         1. Commit: změna uložená v repositáři
         2. Pull: stáhnout změny ze vzdáleného repositáře
         3. Push: nahrát změny do vzdáleného repositáře
         4. Tag: „Pěkný“ ukazatel na konkrétní commit – nejčastěji se používá pro označení realsů verzí př: v1.5.5
         5. Branch: ucelená vývojová verze repositáře
         6. Merge: slučování větví/ verzí
      5. Neznámější vezrovací systémy:
         1. SVN – subverzion
            1. Centralizovaný: Všechny soubory a jejich historie jsou uloženy na centrálním serveru musí být online aby mohl programovat
            2. Struktura:

Trunk – kmen

Hlavní vývojová větev

Většina změn probíhá zde

Branches – větve

Varianty vývojové větve

Vznik např. Při uvolňování nové verze

Tags – značky

Stejné jako větve

Použití při označování jednotlivých revizí

* + - * 1. Příkazy:

Svn checkout – vytvoření lokální kopie repositáře

Svn commit – nutné připojení k repositáři (Commit se nahrává přímo na vzdálený repositář)

Svn update - aktualizace lokální kopie na novou verzi

* + - * 1. Jednoduší než git
      1. Git:
         1. Distribuovaný: Každý vývojář má svojí verzi udělanou lokálně tudíž může programovat i offline
         2. Vývojový model úzce spojený s větvemi
         3. Příkazy:

Git config --global user.name/mail [xxxx/xxx@xx.xx](mailto:xxxx/xxx@xx.xx)

Git init : vytvoří nový repositář v dané složce

Git clone ProjetUrl: naklonuje kompletní kopii repositáře(včetně historie)

Git status: zobrazí stav necommitnuté práce

Git add FileName: přidá soubor do indexu(stav před commitem)

Git diff FileName: Zobrazí změny mezi pracovní složkou a indexem

Git reset FileName: vrátí změny do posledního známého stavu

Git commit – m „Text“: vytvoří nový commit s textem

Git rm FileName: odstraní soubor z pracovní složky i z indexu

Git stash: uloží necommitnuté změny v pracovní složce do „úschovy“ pro pozdější využití

Git stash pop: nahraje necommitnuté změny do pracovní složky

Git stash drop: smaže uložené změny v „úschovně“

Git log (--oneline/ --graph):zobrazí seznam commitů

Git blame FileName: Zobrazí poslední změnu každého řádku v souboru(kdo a kdy změnu provedl)

Git branch [-a]: vypíše všechny lokální větve [i vzdálené]

Git brach BranchName: vytvoří novou větev

Git checkout BranchName: přepne pracovní adresář do specifické větve

Git branch – d BranchName: smaže větev

Git brachce merge: sloučí větve

Git tag: zobrtazí seznam všech tagů

Git tag TagName: vytvoří tag

Git reset [--hard] CommitSha: přepne pracovní adresář do stavu zvoleného commitu, vrácené změny zůstanou jako necommuitnuté změny

Git revert CommitSha: vytvoří nový opačný commit s opačnými změnami v požadovaném commitu

Git fetch [remote]: stáhne změny ze vzdáleného repositáře, ale nezačlení je do větve

Git pull [remote][branch]: stáhne změny ze vzdáleného repositáře a zároveň udělá merge s aktuální větví

Git push[remote][brach]: nahraje změny do vzdáleného repositáře

* + - 1. Ignorované soubory
         1. Do repositáře patří pouze soubory nezbytné pro kompilaci projektu
         2. Soubory /složky, které nechceme verzovat vypíšeme do souboru „.gitignore“ v kořenové složce adresáře
      2. Verzování v git
         1. Commituje se do lokálního repositáře, tyto commity je pak nutné synchronizovat se vzdáleným repositářem
         2. Na comity leze odkazovat pouze kráceným tvarem první 8 znaků
         3. Pokud jsou na vzdáleném serveru změny, musím první udělat pull a pak push
      3. Zásady pro verzování
         1. Komentáře ke každému commitu
         2. Commitovat pouze zkompilovanou vezi
         3. Každá funkcionalita by měla být commitována zvlášť
      4. Platformy využívající Verzovací systémy: GithHub, GitLab, Azure DevOps, BitBucket