1. DevOps
   1. **Co je DevOps:** je přístup k vývoji a provozu SW, který propojuje vývojáře (Developers) a provozní týmy(Operations), prostřednictvím spolupráce a kultury.
   2. **Kdo je DevOps:** DevOps nejsou jednotlivci, ale spíše týmy a role v rámci organizace. Zahrnuti jsou vývojáři, systémový administrátoři, provozní inženýři, testeři, kteří společně spolupracují na vývoji a nasazení do provozu
   3. **Co DevOps řeší:**
      1. Oddělenost komunikace( tradiční vývoj měl nedostatek komunikace, týmy jsou lépe propojeny),
      2. Pomalé nasazení(při nasazování mohlo docházet k chybám, automatizované procesy zvyšují stabilitu a rychlost)
      3. Nedostatečná agilita(málo agilní při řešení problémů, kontinuální nasazení)
   4. Podmínky Pro vznik DevOps:
      1. Organizační kultura: nutná spolupráce a otevřená kultura
      2. Automatizace a nástroje: Implementaci nástrojů pro automatizaci vývoje, testování
      3. Agilní postupy: Agilní metodiky jako scrum nebo kanban poskytují rámec pro flexibilní a interaktivní vývoj
   5. DevOps feedback loop:
      1. Plan
      2. Code
      3. Build
      4. Test
      5. Release
      6. Deploy
      7. Operate
      8. Monitor
   6. Peklo závislostí: Pokud nemám přesně definováno, které verze knihoven používám nebo můžu pro svůj sw použít
   7. Sémantické verzování je přístup k označování verzí softwaru, který dodává význam jednotlivým verzím. Sémantické zde znamená "nesoucí význam".
      1. Verze softwaru se zapisuje ve formátu MAJOR.MINOR.PATCH (například 1.3.0). Každá část tohoto čísla má svou vlastní významovou roli.
         1. - MAJOR (hlavní) číslo se zvyšuje, když dojde k nezpětně kompatibilní změně oproti předchozí verzi. To znamená, že změna je natolik velká, že může ovlivnit kompatibilitu s existujícím kódem nebo rozhraním.
         2. - MINOR (menší) číslo se zvyšuje, když se přidá nová funkcionalita, která neovlivňuje zpětnou kompatibilitu s předchozí verzí. To znamená, že nová funkcionalita může být přidána bez toho, aby bylo třeba upravovat existující kód nebo rozhraní.
         3. - PATCH (opravný) číslo se zvyšuje, když jsou provedeny opravy chyb, které zachovávají zpětnou kompatibilitu. To znamená, že opravy chyb jsou provedeny tak, aby existující kód a rozhraní stále fungovaly stejně.
      2. Zpětná kompatibilita znamená, že to, co fungovalo ve starší verzi, bude fungovat stejně i v novější verzi. To je důležité pro uživatele softwaru, aby se minimalizovaly problémy s upgradem a aby bylo zachováno očekávané chování.
      3. Sémantické verzování poskytuje jasný a srozumitelný způsob, jak označovat a spravovat verze softwaru. Pomáhá při správném rozpoznávání důležitých změn, které mohou ovlivnit kompatibilitu a funkcionalitu softwaru.
   8. CI/CD
      1. Continuous integration
         1. Jediný repositář
         2. Automatizovaný build
         3. Všichni v týmu kommitují své změny denně
         4. Pokud build neprochází opravit ASAP
      2. Continuos delivery vs Deployment
         1. Continuous Delivery: některé kroky nasazení vyžadují člověka
         2. Continuous Deployment: Proces nasazení je kompletně automatizován