

Vývoj s využitím CI/CD a jeho výhody

Špeciálny projekt

Žiadna aplikácia nie je nikdy taká dobrá, aby nemohla byť ešte lepšia. Ešte prednedávnom sa softvér a jeho aktualizácie dodávali na mesačnej alebo kvartálnej báze. Dnes však moderné princípy vývoja, medzi ktoré patrí aj DevOps, prinášajú výrazné zrýchlenie, ktoré umožňuje dodávať malé časté aktualizácie na dennej báze.

V takom tempe je dôležité zabezpečiť kvalitu softvéru a jeho automatického zavedenia do produkcie. Proces dodávania aktualizácií softvéru je sprevádzaný neustálym automatickým testovaním, posudzovaním zmien ostatnými vývojármi a následným nasadením do produkcie. Jednoducho ideálna situácia vyzerá asi takto:

- 1.) Vývojár upraví kód
- 2.) Všetko ostatné sa deje automaticky

Opísaná situácia môže byť reálna vďaka CI/CD:

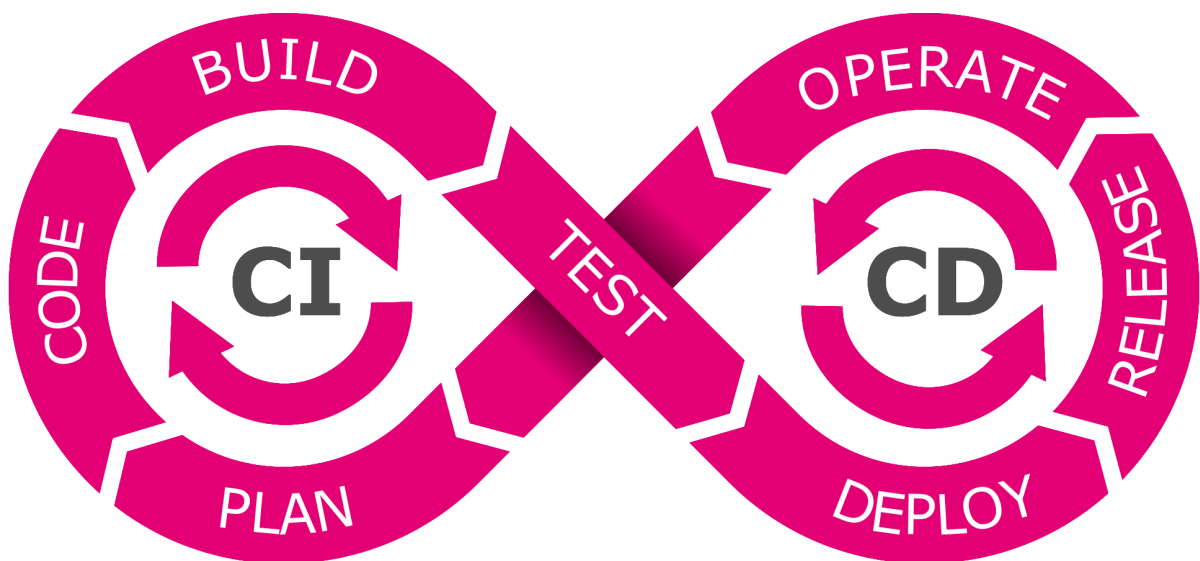
Continuous Integration (CI) – viacero developerov kontinuálne integruje svoje zmeny kódu do produkčnej vetvy spoločného repozitára, niekoľkokrát za deň. Cieľom je čo najjednoduchšia možná integrácia často sa opakujúcich zmien čo najmenších

častí softvéru.

Continuous Delivery – rozšírenie CI o zabezpečenie stavu softvéru, aby ho bolo možné v ktoromkoľvek momente dostať do produkcie. To sa najčastejšie deje pomocou automatického testovania kódu.

Continuous Deployment (CD) – spôsob kontinuálneho zavádzania integrovaného a otestovaného softvéru do produkcie bez zásahu človeka.

T-Systems



Prečo sme sa rozhodli používať CI/CD?

Jedným z projektov [T-Systems Slovakia](#) je SAP Hybrid Cloud (SHC) – možnosť dodania SAP služieb na verejné cloud

infraštruktúry, ako AWS, Azure, OTC, ktoré poskytujú tzv. Infraštruktúru ako Službu (IaaS). Tým sa zabezpečuje flexibilita v alokácii hardvéru, hlavne pre nekonštantne zaťažené aplikácie.

Zákazník platí iba za hardvér, ktorý v danom čase používa – ak ho napríklad v noci nepotrebuje, server, na ktorom SAP služba beží, sa vypne. To výrazne šetrí náklady na prevádzku aplikácie.

CI/CD sme na projekte implementovali hneď po zavedení DevOps kultúry v tíme. Následné zrýchlenie vývoja softvéru viedlo k potrebe častého vykonávania stále sa opakujúceho testovania a nasadzovania. Keďže po úprave kódu vznikala požiadavka okamžitej integrácie, museli sme tieto úlohy zautomatizovať.

Spoločili sme sa na pravidelné automatické testovanie, ktoré sa deje pri každom nahraní zmeny kódu. To, aký dopad má zmena na celý systém, vykonáva počítač automatickými testami.

Ako vyzerá takýto vývoj v praxi?

Pred zavedením DevOps a CI/CD sme pozbierali všetky bugy a požiadavky zákazníka na vylepšenia a naplánovali sme integráciu zmien do aplikácie. Aby nevznikol chaos, vytvárali sme verzie aplikácie. Celý proces bol z pohľadu vývojára pracný a komplikovaný, po aktualizovaní nasledovalo relatívne dlhé obdobie testovania, pozorovania a zavádzania najnovšej aktualizácie.

Vďaka DevOps a CI/CD to dnes vyzerá tak, že vývojár identifikuje problém v kóde, alebo sa zamerá na požiadavku zákazníka, navrhne riešenie v podobe aktualizácie, ktorú nahrá na server repozitára, kde sa spustí CI a testy. Ak niektorý identifikuje chybu, kód sa upraví, opäť nahrá do repozitára a testuje. V prípade úspešného testu sú zmeny kódu posunuté všetkým zainteresovaným vývojárom na ich prehodnotenie a komentovanie.

Po schválení otestovaného kódu sa automaticky spustí CD a zavedie zmenu kódu do produkcie.

Aký je prínos zavedenia CI/CD?

Pred zavedením CI/CD sme dokázali pozbierať všetky požiadavky a nasadiť do produkcie aplikáciu len verzionovane približne raz za mesiac. Dnes prebieha vývoj tak rýchlo, že zmenu verzie už ani nie je možné postrehnúť, pretože viacero vývojárov môže vydať nezávisle od seba niekoľko aktualizácií alebo nových funkcií v priebehu jedného dňa.

Celý kód je zdieľaný so všetkými zainteresovanými kolegami, preto sa k nemu dostanú aj potenciálni vývojári, ktorí ešte reálne neprispeli do kódu, ale majú v rámci DevOps nápad na zlepšenie. Vývojári si môžu zmeny navzájom komentovať a posúvať dopredu aplikáciu aj svoje schopnosti, čo má priamy dopad na atraktivitu projektu.

V neposlednom rade sa nemusíme báť experimentovať s kódom a skúšať ho, keďže fungujúce CI/CD dokáže jednoduchšie odhaliť potenciálne problémy a priniesť ich riešenie ešte v zárodku.

Připravil: Patrik Havrila, Daniel Fečko