

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Teze diplomové práce

**Výběr a nasazení řešení pro řízení vývoje
softwarových projektů**

Bc. Patrik Bláha

© 2018 ČZU v Praze

1. Abstrakt

Současným trendem ve vývoji software je minimalizace technologického dluhu a automatizace procesů pomocí relativně mladé metodologie označované zkratkou DevOps, která zastřešuje životní cyklus softwarového projektu včetně jeho plánování, realizace, testování a nasazení, ale i provozování a reakce na zpětnou vazbu.

Na vývoj software nahlíží agilním pohledem jako na nikdy nekončící proces postupného zdokonalování pomocí velmi rychlé integrace a nasazování velkého množství drobných změn. Tento přístup má na první pohled samé výhody – zákazník bude spokojený s rychlou implementací jeho požadavků, vývojáři se značně usnadní život díky automatizaci neproduktivních činností, software bude bezpečnější díky automatizovanému testování a uživatelé se dočkají skvělé podpory při řešení komplikací. Adopce DevOps je však komplikovaný úkol vzhledem k nutnosti využít při realizaci velké množství nástrojů, jejichž orchestrace je netriviálním úkolem, který trápí velké i malé vývojářské týmy. Roztříštěnost je zde brzdou pokroku.

Tento problém se výrobci vývojářských nástrojů rozhodli řešit vytvářením ekosystémů vzájemně dobře propojených a vyladěných nástrojů, které mají ambici postihnout metodologii DevOps v celé její šíři. Kvůli značné komplexitě však tyto ekosystémy mezi sebou nejsou jednoduše porovnatelné, což znesnadňuje vývojářským subjektům volbu řešení, které by odpovídalo jejich individuálním potřebám. Předmětem této práce je navrhnout model porovnání těchto ekosystémů pomocí vhodné vícekritériální metody a demonstrovat jeho použití na konkrétním vývojářském subjektu.

Klíčová slova: řízení IT projektů, cloud, DevOps, vícekritériální porovnávací metody

2. Cíl práce

Hlavním cílem práce je pomocí vícekriteriálních metod porovnat v současnosti dostupná řešení pro řízení vývoje softwarových projektů a navrhnout konkrétní řešení pro zvolený podnikatelský subjekt. Pro dosažení hlavního cíle jsou stanoveny cíle dílčí.

Prvním dílčím cílem je získat poznatky ze sekundárních zdrojů a definovat teoretický rámec pro další práci jednak v rovině metodik, potřeb a ucelených řešení v současnosti používaných při vývoji software, jednak v rovině dostupných vícekriteriálních metod pro porovnání těchto řešení, včetně volby vhodné metody a definice konkrétních kritérií.

Druhým dílčím cílem je na základě tohoto teoretického rámce vybrat konkrétní v současnosti dostupná řešení jako kandidáty pro porovnání a extrahovat a normalizovat hodnoty zvolených kritérií na základě vlastností těchto konkrétních řešení.

Třetím dílčím cílem je syntéza poznatků z předchozích dílčích cílů, jejich aplikace na konkrétní subjekt, provedení vlastního porovnání a vyhodnocení výsledků.

3. Metodika

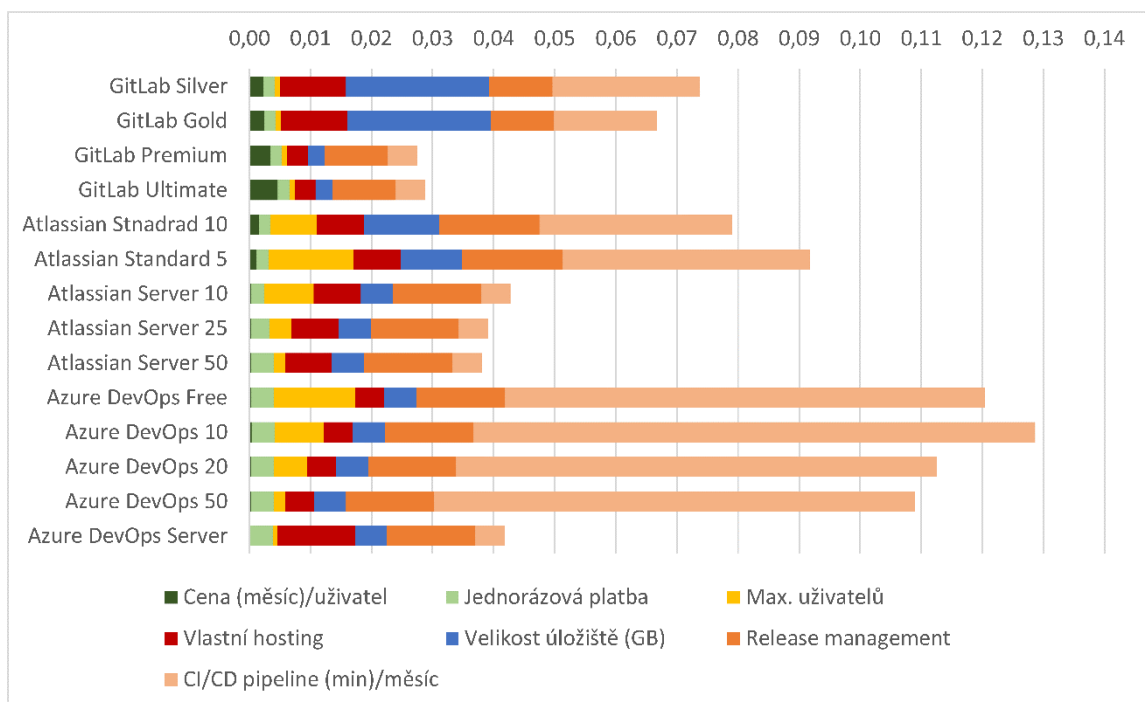
Na základě studia odborné a vědecké literatury a provedené analýzy sekundárních zdrojů, bude vytvořena syntéza poznatků. Budou zvoleny vhodné metody zpracování a bude navržen postup řešení praktické části diplomové práce.

Na základě získaných informací budou stanovena jednotlivá kritéria hodnocení pro porovnání dostupných řešení, bude provedeno vlastní porovnání dostupných řešení pomocí zvolené vícekriteriální metody. V praktické části bude navrženo vhodné řešení pro konkrétní subjekt aplikací vah jednotlivých kritérií vícekriteriální porovnávací metody.

Shrnutí vlastní práce, závěry a doporučení

Na základě provedené analýzy sekundárních zdrojů byla zvolena Saatyho metoda jako vhodný nástroj pro porovnání dostupných řešení. Pro základní orientaci byla provedena obecná kategorizace nástrojů používaných při vývoji software na základě moderní metodologie vývoje DevOps. Na základě této kategorizace byla identifikována konkrétní kritéria zvolené Saatyho porovnávací metody. Na základě tržní analýzy ze sekundárních zdrojů byla vybrána kandidátní řešení pro porovnání a z jejich specifikací byly extrahovány hodnoty jednotlivých kritérií. Na vzniklém vstupním souboru dat byla provedena normalizace a filtrace irelevantních hodnot – v průběhu zpracování se ukázalo, že některá kritéria splňují všechna kandidátní řešení, a tudíž nemělo smysl je porovnávat.

V praktické části práce byly na základě nestrukturovaných rozhovorů se subjektem zjištěny jeho preference, byla sestavena matice vah kritérií. Na základě této matice a očištěných vstupních dat byly sestaveny preferenční tabulky porovnávaných variant pro každé kritérium. Jejich syntézou vznikla tabulka s výsledným hodnocením. Kompletní podrobnosti výpočtu vč. mezikroků jsou vzhledem k velkému rozsahu připojeny jako Příloha A této práce. Výsledné preferenční hodnoty byly transformovány do podoby skládaného pruhového grafu:



Na základě těchto údajů vychází pro potřeby daného subjektu jako velmi vhodné všechny varianty Azure DevOps s tím, že jako optimální vyšla varianta Azure DevOps 20. Zásadní vliv kritéria CI/CD pipeline je očekávatelný vzhledem k udaným preferencím subjektu.

Pozoruhodné však je, proč byly silně penalizovány všechny on-premise varianty které z podstaty mají čas běhu CI/CD pipeline neomezený a bylo by možné očekávat, že právě ony budou nejvhodnějšími kandidáty?

Důvodem pro tento jev je zřejmě skutečnost, že všechny on-premise varianty jsou v porovnání s cloudovými řádově dražší a tudíž nevýhodné, a to i přesto, že subjekt cenové hledisko označil za slabě preferované. Tuto hypotézu však nepodporuje ekosystém GitLab, u nějž jsou ceny cloudové a on premise varianty zcela shodné. V tomto případě však sehrála roli skutečnost, že ekosystém Azure DevOps dává k dispozici neomezený úložný prostor, což je kritérium, které naopak subjekt označil za silně preferované.

Výše uvedený rozbor tedy potvrzuje, že výpočet zvolenou metodou vybral variantu, která je v souladu s požadavky subjektu.

Na základě zjištěných informací je obecně možné konstatovat, že současný stav komplexních řešení pro řízení softwarového vývoje má vůči přístupu DevOps ambivalentní vztah. Na jednu stranu je zřejmá vynikající podpora fází životního cyklu spadající do první poloviny pojmu „Dev“ (tedy *plan, code, build a test*), pro které jsou k dispozici vyztáhlé a prověřené technologie. Na stranu druhou však není ustálena podpora pro fáze z druhé poloviny životního cyklu „Ops“ (tedy *release, deploy, operate a monitor*) - nástroje pro sestavování a nasazení se mnohdy neobejdou bez různých berliček, podpora pro Windows a Mac platformy není zdaleka samozřejmostí, podpora pro monitorování a zpětnou vazbu nezřídka chybí zcela.

Vysvětlením pro tento nepoměr může být jednak skutečnost, že systémový přístup k těmto procesům je stále relativní novinkou, jednak značná diverzita souvisejících technologií, které dosud nedospěli k implementaci jednotných standardů, jako je tomu např. u správy zdrojového kódu, kterému zcela dominuje nástroj Git. Diskutovat příčiny tohoto stavu a možná řešení je spíše námětem pro budoucí práci na toto téma, nicméně zejména při výběru nástroj pro řízení vývoje je nutné mít tuto skutečnost na paměti již nyní vzhledem k očekávanému vývoji a vybírat řešení s ohledem na flexibilitu a dopřednou kompatibilitu

Seznam použitých zdrojů

WEBBER-CROSS, Geoff. *Learning Microsoft Azure*. Birmingham - Mumbai: Packt Publishing, 2014. ISBN 9781782173373.

ŠUBRT, Tomáš. *Ekonomicko-matematické metody*. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, 2011. ISBN 978-80-7380-345-2.

Manifesto for Agile Software Development [online]. 2010 [cit. 201-04-05]. Dostupné z: <http://agilemanifesto.org/>

BASS, Len, Ingo M. WEBER a Liming ZHU. *DevOps: a software architect's perspective*. New York: Addison-Wesley Professional, [2015]. ISBN 978-0134049847.

Periodic Table of DevOps Tools Version 3 – Coming Soon. *Xebialabs: Blog - Continuous Delivery & DevOps Tools for the Enterprise* [online]. 2018 [cit. 2018-12-23]. Dostupné z: <https://blog.xebialabs.com/2018/04/12/periodic-table-of-devops-tools-version-3-coming-summer-2018>

DevOps: Continuous Delivery, Integration, and Deployment with DevOps. Birmingham: Packt Publishing, 2018. ISBN 978-1-78913-299-1.

DUVALL, Paul M., Steve MATYAS a Andrew GLOVER. *Continuous integration: improving software quality and reducing risk*. Upper Saddle River, NJ: Addison-Wesley, c2007. ISBN 978-0-321-33638-5.

GHAHRAI, Amir. *Testing In DevOps - Strategies, Tools and Practices. Testing Excellence — Software Testing for Beginners and Professionals* [online]. 2018, 02.12.2018 [cit. 2019-10-09]. Dostupné z: <https://www.testingexcellence.com/testing-in-devops/>