

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2017/2018
Autoři	Lukáš Kadoch (xkadl17), Marcel Jäger (xjagm04), Robin Jiránek (jirr00)
Téma	Building Autonomous DevOps Capability in Delivery Teams
Datum odevzdání	12. 5. 2018

Abstrakt

Cílem práce je objasnit využití DevOps ve středních a velkých firmách. Dále popsat funkci autonomních týmů DevOps a určit jejich výhody. První část práce se věnuje úvodu, kde je objasněno, co je DevOps a jeho základní charakteristiky. Druhá část se zabývá samotnému využití DevOps ve firmách a výhodami DevOps týmů. Závěrečná část popisuje nejčastější chyby při zavedení DevOps, a také způsob, jak využívat DevOps pomocí teacher model.

Klíčová slova

DevOps, Agile, Autonomous team, Automization

Obsah

1	Úvod	4
2	Co to je DevOps?	5
2.1	Procesy.....	5
2.2	Nástroje.....	5
2.3	Lidé.....	5
2.4	Vlastnosti prostředí.....	6
2.4.1	Spolupráce a komunikace.....	6
2.4.2	Automatice	6
2.4.3	Průběžná integrace	6
2.4.4	Nepřetržité testování	6
2.4.5	Plynulé nasazování	7
2.4.6	Nepřetržitá monitoring a zpětná vazba.....	7
2.4.7	Neustále zlepšování.....	7
3	Využívání DevOps ve firmách.....	7
3.1	Jak se využívá přístup DevOps.....	7
3.2	Snížení nákladů na zdroje.....	8
3.3	Klíčové oblasti DevOps ve firmě	9
3.3.1	Culture	9
3.3.2	Automation.....	9
3.3.3	Lean.....	9
3.3.4	Measurement	10
3.3.5	Sharing	10
4	Výhody DevOps týmů.....	10
4.1	Týmy DevOps musí být autonomní.....	10
5	Nejčastější chyby při používání DevOps	11
5.1	Rozdělování personálu a zdrojů	11

5.2	Vytváření DevOps oddělení	11
5.3	Vysoká očekávání při zavádění modelu DevOps	12
5.4	Očekávání zázraků.....	12
5.5	Nezvládnutí odporu při zavádění.....	13
6	DevOps teacher model jako ideální cesta	13
7	Závěr.....	14
8	Slovník pojmů	15
9	Literatura	16

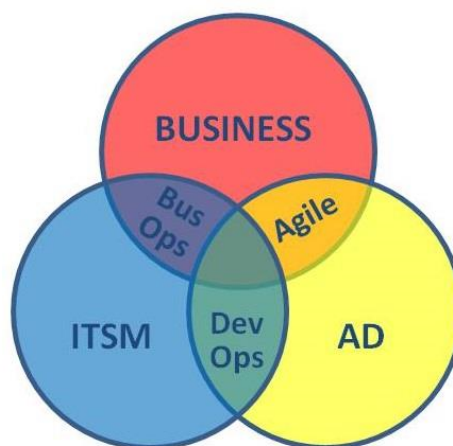
1 Úvod

Byznys neboli jinými slovy zákazník očekává od IT v dnešní době velmi rychlé zavádění nových funkcionalit. Dále také čeká co nejmenší počet chyb a na druhou stranu co nejvyšší spolehlivost.

Aby IT poskytovatel uspokojil všechny tyto požadavky, musí obstarat kvalitní a flexibilní hodnotový řetězec. Ovšem zákazníka zajímá hlavně výsledek. Není pro něj důležité, že celá dodávka služby obsahuje správné plánování a mapování požadavků, analýzy, vývoj kódu, testování, management releasů a nesmí se zapomínat i na následný babysitting, který zahrnuje technickou i netechnickou podporu. Dlouhodobě se ukazuje, že pro výslednou kvalitu dodávané služby nemá vliv pouze fungování a výkonnost jednotlivých složek řetězce, ale velký vliv má také spolupracování jednotlivých složek.

Samotná zkratka DevOps je spojení dvou anglických slov Development (vývoj) a Operation (provoz).

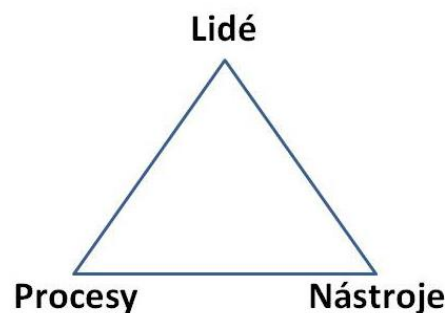
Podle názvu se může zdát, že DevOps je dáno jen týmy Developmentu a týmy Operation. Toto ovšem není zcela pravda. Pod DevOps spadají dodavatelé softwaru, technici, zákazníci, management, ostatní dodavatelé a apod. Jako nejslabší článek tohoto přístupu k IT může být to, že není definováno, co vlastně DevOps obsahuje. DevOps je chápán jako přístup založený na principech Agile a Lean. Tento přístup není definován jako konkrétní soubor postupů či technik. DevOps je spíše koncept, který lze v IT aplikovat a který pomáhá při spolupráci vývoje a provozu. [The Agile Admin, 2018]



Obr. 1 – Rozložení [BestPractice, 2018]

2 Co to je DevOps?

Jak již bylo zmíněno dříve, DevOps je složenina dvou anglických výrazů. Development, což vyjadřuje vývoj a druhé slovo je Operations – to lze volně přeložit do češtiny jako provoz. Celý koncept vlastně vyjadřuje přístup k vývoji software. Přístup si zakládá na komunikaci a spolupráci mezi jednotlivými týmy. Jedná se o týmy vývojové a týmy odborníků, kteří se starají o provoz software.



Obr. 2 – Lidé, proces, nástroje [BestPractice, 2018]

2.1 Procesy

Pro DevOps jsou přesně definované procesy nezbytné. Procesem je definován celý životní cyklus aplikace – návrh kódu, testování, release. Cílem procesů není byrokratické nastavení postupů, ale využití nástrojů, procesů a týmů s cílem dodávat software rychle a kvalitně.

2.2 Nástroje

Nástrojů, které podporují DevOps je obrovské množství. Důležitou podmínku při jejich výběru je provázanost jednotlivých nástrojů. Propojení procesů a nástrojů usnadní práci a umožní plynulé nasazování software.

2.3 Lidé

Rozhodně klíčovým faktorem pro fungování DevOps je prostředí, ve které lidé pracují. Pro prostředí, které podporuje DevOps, je typická dobrá komunikace všech zúčastněných týmů. Jedná se o spolupráci vývoje, testingu, provozu, a i všech ostatních týmů, kteří do procesu zasahují nebo jej ovlivňují. Jestliže prostředí toto vše nepodporuje, je obvykle těžké ho změnit – avšak podpora prostředí je pro DevOps naprosto klíčová.

2.4 Vlastnosti prostředí

Jelikož není přesně definována metodika, kterou se má DevOps řídit, tak i prostředí není přesně určeno. Obecně se ale opakují v praxi následující prvky:

2.4.1 Spolupráce a komunikace

Jednotlivé týmy musí působit jako jeden celek, který chce společně dosáhnout daného cíle. Toto se týká nejen vývoje a provozu, ale celého životního cyklu software. Jedná se o mimo zmíněný vývoj a provoz také o testování, management, podporu a další související týmy, které musí spolupracovat.

S tím souvisí i výborná komunikace napříč všemi týmy. Jedním z hlavních DevOps cílů je schopnost rychle a pružně reagovat na vzniklé změny. Proto je tak důležité, aby byly vytvořeny vhodné komunikační kanály, ke kterým budou mít přístup všichni zainteresovaní a mohli rychle reagovat na podněty ostatních kolegů.

2.4.2 Automatice

Pro zabezpečení rychlosti, kvality a velkého množství požadavků je nutné zvolit vhodný nástroj. Je vhodné automatizovat co nejvíce kroků ve všech možných částech procesu.

2.4.3 Průběžná integrace

Vývojáři mezi sebou velmi často integrují výsledky své práce – tím dokáží odhalit chyby a nedostatky již velmi brzy.

2.4.4 Nepřetržité testování

Nepřetržité testování není náhradou zabezpečení kvality. V DevOps je do testování zapojen téměř každý. Vývoj testuje jím napsaný kód, dále se podílí na integračních testech, doporučují testovací metody, dále mohou připravovat testovací prostředí atd.

Provoz naopak testovací prostředí monitoruje a nastavuje jeho konfiguraci. Tato forma spolupráce dohromady s automatizacemi šetří čas, zvyšují kvalitu a na druhou stranu i snižují náklady, které jsou potřeba na testování.

2.4.5 Plynulé nasazování

Plynulé nasazování je pro DevOps také velmi typické. Charakteristické jsou častější a jednodušší releasy.

2.4.6 Nepřetržitá monitoring a zpětná vazba

Jak bylo zmíněno výše, tak pro DevOps jsou typické častější releasy. Ty s sebou nesou také vyšší riziko pro produkční prostředí. S tímto rizikem se DevOps snaží vypořádat pomocí server monitoring, application performance monitoringu a také continuous customer feedback. Hlavním úkolem je rychle identifikovat, vyhodnotit ale hlavně řešit vzniklé chyby nebo výpadky. Tyto všechny činnosti musí být aplikovány během celého cyklu aplikace.

Důležité je také sbírat zpětnou vazbu od zákazníka, která je nezbytná součástí DevOps. Umožňuje totiž organizaci chovat se pružně a reagovat na zákaznické potřeby.

2.4.7 Neustále zlepšování

Organizace musí mít zaveden proces, který jí umožní neustále se zlepšovat. Zlepšování nemusí být formalizované ani centrálně řízené, důležité je ale zabezpečit zlepšování. [BestPractice, 2018]

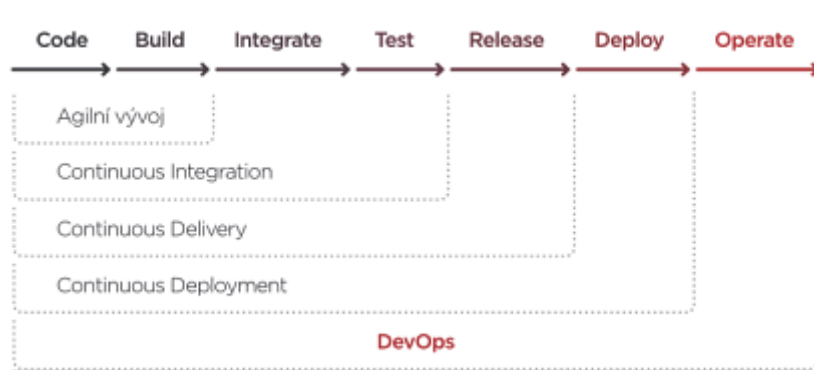
3 Využívání DevOps ve firmách

V dnešní době je přístup DevOps ve firmách často diskutovaným tématem. Při využívání DevOps záleží na jeho formě a uchopení. Popularita DevOps v posledních letech velmi vzrostla a shledala se s úspěšným i neúspěšným nasazením.

Nedávná studie ukázala, že 55 % ze 700 největších IT společností využívá DevOps a dalších 31 % plánuje nasazení v nejbližší době. Příkladem úspěšných společností, které vedou DevOps, jako oblast mezi vývojem a nasazením je Facebook nebo Netflix. [INNOVATIVE TECHNIQUES, 2016]

3.1 Jak se využívá přístup DevOps

Hlavním principem DevOps je snaha o maximální automatizaci jednotlivých kroků životního cyklu dodávky softwaru. Je zapotřebí tyto kroky dostatečně oddělit a udržovat je v menší podobě, což umožní je použít jako znovupoužitelné komponenty s možností rychlé reakce na změny a podporu snadné opravy případných chyb. [IT SYSTEMS, 2016]



Obr. 3 - Ukázka využití DevOps [IT SYSTEMS 11/2016]

Obrázek 3 uvádí, že DevOps můžeme využít v jakémkoliv kroku životního cyklu.

1. **Kódování (vývoj)** – Pomoc při vývoji jednotlivých komponent softwaru a kontrolu verzování.
2. **Build** – Vytvoření buildu softwaru a jeho testování na vývojovém prostředí.
3. **Integrace** – Přesunutí otestovaného buildu na integrační prostředí (prostředí dodané zákazníkem)
4. **Testování** – Provedení jednotkových, integračních a penetračních testů.
5. **Vydání verze** – Vytvoření funkční verze softwaru z připraveného buildu.
6. **Nasazení** – Dodání a nasazení buildu na produkční prostředí.
7. **Monitoring** – Analýza a kontrola produkčního prostředí, hledání chyb v logu a shromažďování zpětné vazby od zákazníka

3.2 Snížení nákladů na zdroje

Běžnou otázkou ve světě byznysu jsou náklady versus přínos. Ačkoliv přístup DevOps může velmi pomoci při vývoji a kontrole softwaru, nejsou jeho náklady vůči přínosům vždy adekvátní.

DevOps také mimo jiné přináší vyšší nároky na řízení kvality, což znamená, že lze očekávat nižší množství chyb, zvýšení rychlosti oprav případných chyb a kontrolu dodání. V důsledku těchto zlepšení tedy dochází ke snížení nákladů na zdroje, jelikož nedochází k častým chybám, a tudíž se zkrátí čas využívání vývojářů a testerů na opravy. Další možností je i vyšší volnost k experimentování, díky rychlé zpětné vazbě a schopnosti operativní dodávky. [IT SYSTEMS, 2016]

Určení přínosů přístupu DevOps se udává čtyřmi základními faktory – důvěra v kvalitu dodávky, náklady na dodávky, trvání dodávkového cyklu a schopnost experimentovat. Cílem DevOps je snížit náklady pomocí těchto faktorů a otočit tedy poměr vysokých nákladů na změny. [IT SYSTEMS, 2016]

3.3 Klíčové oblasti DevOps ve firmě

Princip DevOps figuruje v několika základních oblastech, ve kterých se snaží řešit specifické problémy. Základní model, který se využívá a naznačuje hlavní faktory DevOps se nazývá „CALMS“. Ten definuje, že klíčovými oblastmi jsou: Culture, Automation, Lean, Measurement a Sharing. [The DevOps Handbook, 2016]

3.3.1 Culture

Důležitou částí DevOps je změna organizační kultury do prostředí otevřené spolupráce. To zahrnuje zapojení provozního personálu do návrhu a přechodového procesu aplikace. Dále by se měli účastnit plánovacích meetingů, retrospektiv a ukázek od projektového týmu, aby mohli sdílet své nápady a znalosti v úvodní části procesu.

3.3.2 Automation

Automatizace zahrnuje infrastrukturu, různá prostředí, databázová schémata, referenční data a konfiguraci. Nejčastěji se před nasazením na produkci provádějí automatické testy, které pokrývají širokou škálu potenciálních chyb, konfliktů a nežádoucích jevů. Poté, co jsou testy dokončeny, nastává fáze nasazení. Po správné konfiguraci se ve většině případů jen přesune otestovaný build na produkční prostředí. Díky automatickému přesunu je také možnost rychlého „rollbacku“.

3.3.3 Lean

Další velmi důležitou částí DevOps jsou Lean metodiky. Zde se používají zejména Agilní metodiky vývoje, což znamená, že pokud není ve sprintu naplánovaná daná činnost, tak není vykonána. Pomocí automatickosti se nemusí některé opakující se činnosti znovu vykonávat. Tento způsob je oblíben zejména u vývojářů, jelikož stačí nahrát novou verzi aplikace a nemusí se starat o build nebo nasazení, to se provede automaticky.

3.3.4 Measurement

Každý proces by měl mít své metriky a indikátory pro měření výkonosti. To zahrnuje zejména, jak pracovníci vykonávají svou činnost. Měření by mělo ukazovat, jak si vede projektový tým v jakémkoliv časovém okně. Metriky také občas mívají vliv na objevování nedostatků nebo ústí procesu. Existují také měření výkonosti přímo na produkci, kde se jedná hlavně o monitoring stability aplikace.

3.3.5 Sharing

Poslední oblastí je sdílení znalostí, zkušeností a kapacity. Sdílení je v dnešní době spojeno z velké části nástroji, které umožňují vkládat například hotové komponenty, které jiný tým může převzít. Jednou z klíčových vlastností sdílení je také komunikace. Ve firmách se používají různé aplikace, jako jsou například Skype for Business, Slack nebo Jabber. [Challenges in Adopting, 2016]

4 Výhody DevOps týmů

Zavedení přístupu DevOps má různé výhody v závislosti na jeho rozsahu. Každá komponenta má své aspekty, ve kterých přináší firmě přidanou hodnotu. Hlavním cílem je zejména snížení dodací lhůty a zvýšení počtu nasazených buildů.

S vyšším počtem nasazení přichází i rychlejší a častější odezva od zákazníka, který pravidelně dodávaný software používá. S touto výhodou lze pružněji reagovat na změny a nové funkce, které vedou ke zlepšení aplikace. [Challenges in Adopting, 2016]

4.1 Týmy DevOps musí být autonomní

Jeden z lepších přístupů při zavedení DevOps je zlepšení samosprávy týmů, které mají na starost dodání. Firma by neměla vytvářet izolované DevOps týmy. Tato metoda má své komplikace. Je zapotřebí zlepšit schopnosti softwarových inženýrů, což znamená určitou investici. První transformace také může mít dopad na rychlost dodání. Výhodou je ale snížení externích závislostí, dodání sebevědomí v dalších dodáních a budoucí zvýšení rychlosti. Změny se poté dají vydat bez závislosti na jiném týmu.

V nejlepším případě se DevOps dá využít jaké určitá forma zlepšení v každém dodacím týmu. Pro vylepšení autonomního týmu lze zavést jejich vlastní využití průběžného dodávání z vývoje

do produkce. To znamená, že týmy nebudou převádět zodpovědnost za dodání na jiný tým, ale některé aspekty provedou za sebe.

Zlepšení samosprávy dodacího týmu nemusí nutně znamenat správu všech „kernel patchů“ pokaždé, když budou potřebovat vytvořit build. Jsou zde stále role systémového administrátora a týmů pracujících na infrastruktuře. Ti by se měli soustředit na vytváření frameworků a automatizace, aby se dodací tým mohl koncentrovat na předávání změn na produkci.

Definování frameworku pro nasazení a správa platformy jako služba je dobrý začátek pro DevOps. Správa služeb lze v dnešní době outsourcovat na moderní cloudové systémy, které využívají připravené nástroje pro administraci. [Better SW magazine, 2017]

5 Nejčastější chyby při používání DevOps

Znalost nejčastějších chyb při zavádění DevOps, jejich důsledků a jak jim předejít je pro společnost velmi prospěšné. Níže se nachází několik nejčastějších chyb, které společnosti dělají.

5.1 Rozdělování personálu a zdrojů

Pokud společnost nemá jasný přehled a úplně nerozumí stávajícímu pracovnímu rozdělení a zatížení jednotlivých týmů, jejich silným stránkám a schopnosti plnit úkoly, tak by s nasazením DevOps měla počkat. První krok při zavádění by měl vždy být porozumění pracovním schopnostem týmu a jednotlivců.

Důležité je porozumění silným stránkám jednotlivců. Komunikací s nadřízenými jednotlivých pracovníků lze získat informace, pomocí kterých může společnost lépe rozdělovat jednotlivé úkoly. Pokud toto nenastane, tak budou pracovníci a celé týmy nemotivované a společnost je tak může ztratit.

V neposlední řadě je potřeba vymyslet indikátory/metriky (KPI) podle kterých se bude měřit.

[Ruzovsky, 2017; Earnshaw, 2013]

5.2 Vytváření DevOps oddělení

Společnosti, které začínají s DevOps často udělají chybu, že vytvoří nové oddělení, které má na starosti správu DevOps strategie. Ve většině případů to ale přidá pouze další procesy a naruší

fungování celého modelu. DevOps samozřejmě potřebuje vedoucí tým, ale ne takový, kterého se dosáhne vyčleněním speciálního oddělení.

Firma nemůže od nového oddělení očekávat, že se hladce začlení a začne spolupracovat s ostatními, jen protože to všem bylo nařízeno. Organizace by se tak měla soustředit pouze na vytvoření nových DevOps procesů ve stávajících odděleních a nevytvářet žádná nová. Organizace by také měla mít na paměti, že se jedná o drastickou změnu ve způsobu vývoje, ne o novou funkci podporující business.

[Ruzovsky, 2017; Menzel, 2016]

5.3 Vysoká očekávání při zavádění modelu DevOps

Společnost nesmí čekat, že se zavedením DevOps nějak drasticky změní produktivita a vývojařské týmy budou produkovat 10 nových verzí software za týden místo jednoho. Takto vysoká očekávání vytváří pouze tlak na vývojáře, zmatek a zhoršení pracovních výsledků z jejich strany. To je jeden z častých problémů při zavádění DevOps. S implementací je navíc svázána kulturní revize a tím se liší od ostatních technologických změn. Čím je organizace větší a má hlubší strukturu, tím je zavedení DevOps složitější.

Jednou z možností, jak zjednodušit implementaci DevOps je zavádění po částech. Zavedení se rozdělí na fáze, během kterých by neměl být podceňován výcvik a vzdělávání zaměstnanců.

První měření by měla proběhnout po adekvátní době, aby se neměřilo příliš brzy. Při brzkém měření by naměřené výsledky byly špatné. Počáteční pomalost při zavádění je známou slabou stránkou. Nejlepším řešením tohoto problému je pečlivý výběr týmů, které s DevOps začnou a přidělení pouze menších projektů těmto týmům v počátečních fázích.

[Mavadia, 2017]

5.4 Očekávání zázraků

DevOps se nebude řídit samo. Je potřeba vyčlenit lidi, kteří se budou starat o zdroje, průběh a cíle. Určení priorit a rozdělení projektů na menší časové úseky a milníky je klíč k dosažení pozitivních výsledků od DevOps modelu během zaváděcí fáze.

Dobře určená DevOps strategie je o průběžném učení a zlepšování se. Vybrání řídicího týmu, který bude schopný předvídat a zabezpečit dosáhnoutí těchto plánů je kriticky důležitý faktor pro úspěšnou implementaci v organizaci.

Je potřeba mít na paměti, že DevOps model přinese úspěch pouze tehdy, když je správně zvládnutá jeho implementace.

[Mavadia, 2017]

5.5 Nezvládnutí odporu při zavádění

Při zavádění DevOps do společnosti většinou vznikne odpor od většiny zahrnutých stran. Záleží, jak se osoba, která DevOps navrhuje k tomuto odporu postaví. Pro manažery a zúčastněné strany je většinou přijatelným kompromisem zavedení “hybridního” DevOps modelu, kdy jsou některé části IT modernizovány, zatímco jiné oddělení nebo týmu stále pracují podle starých plánů a změna se jich dotkne až později.

[Singh, 2017]

6 DevOps teacher model jako ideální cesta

Nejdůležitějším úkolem DevOps učitelů je, aby dokázali neustále učit a zvyšovat kvalifikaci jejich týmu. Vždy, když DevOps učitel něco prakticky ukazuje, tak se zároveň ujišťuje, že jiný člen týmu bude schopen v budoucnu zvládnout stejný úkol sám.

Jako ostatní plnohodnotní členové týmu by se i DevOps učitelé měli účastnit agilních setkání spolu se zbytkem týmu a pracovat s jejich backlogem. Díky aktivní účasti v plánovacím stádiu nového projektu, mohou DevOps učitelé pracovat proaktivně a jsou v unikátní pozici, aby ukázali ostatním nejlepší možný způsob testování, nasazení a monitorování. Podobně je tomu i při retrospektivních setkáních, kde jsou učitelé schopni neustále zlepšovat jednotlivé procesy při budování softwaru.

Učitelé by se měli často potkávat mezi sebou, aby mohli diskutovat o překážkách a závislostech jejich individuálních týmů. Bylo zjištěno, že stand-up setkání jednou nebo dvakrát týdně fungují nejlépe. Záleží však na to, jak moc je role učitele vyvinutá. Učitelé by se také měli střídát každých 3 až 6 měsíců. Díky tomu se týmy mohou učit od ostatních a zajistí se tak, že učitelé dokáží rozprostřít své znalosti mezi všechny, které jsou součástí řetězce dodávky software.

Přirozenou otázkou je, jak změnit klasické DevOps týmy na týmy DevOps učitelů. Tato změna může být někdy velmi obtížná, především, když jsou DevOps týmy zvyklé na jejich průběh při vývoji software a jednotlivcům se na tomto způsobu práce nechce nic měnit. V takovém případě je změna z “dělání věcí” na “učení ostatních, jak dělat věci” vždy těžká. Skutečnost je taková, že někteří lidé nejsou schopni projít touto změnou. Pokud se tedy jednotlivci přizpůsobit nechtějí, tak se nepřizpůsobují procesy těmto nepřizpůsobivým jednotlivcům, ale často je potřeba vyměnit zaměstnance za jiné.

[Better SW magazine, 2017]

7 Závěr

DevOps je ve firmách často diskutovaným tématem. Jeho nasazení však není jednoduché a společnost při nasazování musí překonat několik těžkých překážek. Hlavním cílem je při zavádění DevOps je snížení dodací lhůty a zvýšení počtu nasazených buildů. Díky tomuto rychlejšímu vydávání nových verzí tak lze rychleji reagovat na změny, přidávat nové funkce nebo také rychleji nasazovat opravy chyb.

I přes překážky, které při zavádění mohou vzniknout jsou přínosy tohoto modelu jak výrazné, že ho využívá již více než polovina ze 700 největších IT společností a více než třetina o jeho zavedení uvažuje.

8 Slovník pojmů

Pojem	Význam
Autonomy	Nezávislý
Babysiting	Prvotní provoz systému po jeho nasazení do produkčního prostředí
Business	Zákazník požadující produkt od IT
Development	Vývoj
DevOps	Vývoj – provoz
Framework	Softwarová struktura usnadňující programování v daném programovacím jazyce
Operations	Provoz
Release	Nasazování systému do produkčního prostředí
Stand-up	Rychlá schůzka, obvykle účastníci stojí

9 Literatura

1. DevOps | bestpractice.cz . [online]. 2008 [cit. 11.05.2018]. Dostupné z: <https://www.bestpractice.cz/cs/Best-practice/DevOps.alej>
2. What Is DevOps? | the agile admin. *the agile admin | thoughts on agile web operations and devops* by @ernestmueller, @wickett, @iteration1 and @bproverb [online]. Dostupné z: <https://theagileadmin.com/what-is-devops/>
3. EARNSHAW, Aliza, 2013. 5 KPIs that Make the Case for DevOps. *Puppet*[online] [vid. 2018-04-12]. Dostupné z: <https://puppet.com/blog/5-kpis-make-case-for-devops>
4. MAVADIA, Sunil, 2017. Why Enterprises Should Do DevOps Now. *XebiaLabs Blog*[online]. [vid. 2018-05-06]. Dostupné z: <https://blog.xebialabs.com/2017/01/05/enterprises-devops-now/>
5. MENZEL, Gunnar, 2016. DevOps – What is the best organizational structure for me? *Capgemini Worldwide*[online]. [vid. 2018-05-06]. Dostupné z: <https://www.capgemini.com/2016/09/devops-what-is-the-best-organizational-structure-for-me/>
6. RUZOVSKY, Erez, 2017. *Common DevOps Mistakes* » *ADMIN Magazine*[online] [vid. 2018-04-05]. Dostupné z: <http://www.admin-magazine.com/Archive/2017/42/Common-DevOps-Mistakes>
7. SHACKELFORD, David, 2015. Nine Common Ops Mistakes (and How to Prevent Them). *DevOps.com*[online] [vid. 2018-03-31]. Dostupné z: <https://devops.com/nine-common-ops-mistakes-prevent/>
8. SINGH, Ajeet, 2017. 7 Common Mistakes Businesses Must Avoid While Implementing DevOps. *Algoworks*[online]. [vid. 2018-05-06]. Dostupné z: <http://www.algoworks.com/blog/common-mistakes-businesses-must-avoid-while-implementing-devops/>
9. HAMUNEN, Joonas, et al. 2016. Challenges in adopting a Devops approach to software development and operations. Dostupné z: https://aaltoodoc.aalto.fi/bitstream/handle/123456789/20766/hse_thesis_14441.pdf?sequence=1&isAllowed=y
10. NOCERA, Dott Ing Francesco; DI NOIA, Tommaso; GALLITELLI, Davide. 2016. Innovative techniques for agile development: DevOps methodology to improve software production and delivery cycle. Dostupné z: https://www.researchgate.net/profile/Davide_Gallitelli2/publication/309201974_Innovative_Techniques_for_Agile_Development_DevOps_Methodology_to_improve_Software_Production_and_Delivery_Cycle/links/5804de8008aee314f68e08dd/Innovative-Techniques-for-Agile-Development-DevOps-Methodology-to-improve-Software-Production-and-Delivery-Cycle.pdf
11. JUUSO, Miir. 2018. Building Autonomous DevOps Capability in Delivery Teams. *Better Software*. ISSN: 1553-1929

12. KIM, Gene. 2016. *The devops handbook: how to create world-class agility, reliability, & security in technology organizations*. Portland, OR: IT Revolution Press, ISBN 9781942788003.
13. PETŘÍK, Michal, 2016. *Nové pohledy na populární metody projektového řízení*. IT SYSTEMS 11/2016. [online]. [cit. 8.5.2018]. Dostupné z: <https://www.systemonline.cz/rizeni-projektu/zacnete-s-devops.htm>