

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2015/2016
Autoři - Jméno Příjmení, xname	Jakub Mejstřík, xmej07 Petr Krejčí, xkrep33 Michael Friedmann, xfrim17
Téma	ACCELERATING APPLICATION DEPLOYMENT WITH DEVOPS
Datum odevzdání	15. května 2016

Abstrakt

Semestrální práce představuje DevOps, jak v teoretické rovině, tak na konkrétních příkladech z praxe. Cílem teoretické části je přiblížit samotný pojem DevOps, pojednat o historii vzniku a principech. Práce se dále zabývá postupy, využitím DevOps při vývoji softwaru a používanými nástroji. Praktická část představuje konkrétní příklady firem z praxe, které zavedly DevOps, a poukazuje na přínosy zavedení DevOps ve firmě.

Klíčová slova

DevOps, deployment, zavedení, historie, continuous integration, continuous delivery, continuous deployment, Puppet, Chef, Docker, Jenkins, Nordstrom, Texas.gov

Obsah

Abstrakt	1
Klíčová slova	1
1 Úvod.....	4
2 Definice pojmu DevOps	4
3 Původ vzniku DevOps	4
3.1 Rok 2007.....	4
3.2 Rok 2008.....	5
3.3 Rok 2009.....	5
3.4 Rok 2010.....	5
3.5 Rok 2011.....	5
3.6 Rok 2012.....	5
3.7 Rok 2013.....	5
3.8 Rok 2014.....	5
4 Principy a cíle DevOps.....	5
5 Hodnoty DevOps (business, tým, zákazník)	6
5.1 DevOps kultura spolupráce.....	6
5.2 DevOps nástroje.....	6
5.2.1 Úložiště zdrojového kódu (Repository).....	6
5.2.2 Sestavovací server (Build Server)	6
5.2.3 Správa konfigurací	6
5.2.4 Virtuální infrastruktura	6
5.2.5 Automatizace testů.....	6
6 DevOps postupy	6
6.1 Vodopádový vývoj (Waterfall Development).....	7
6.2 Nepřetržitá integrace (Continuous Integration)	7
6.3 Nepřetržitá dodávka (Continuous Delivery).....	8
6.4 Nepřetržitě nasazení (Continuous Deployment)	8
7 DevOps ve vývoji software	8
7.1 Konflikt mezi Dev a Ops.....	8
8 Příklady nástrojů DevOps	9
8.1 Puppet.....	9
8.2 Chef.....	9
8.3 Docker	9
8.4 Jenkins.....	10
8.5 Ansible.....	10
8.6 Skype.....	10
8.7 Slack	10
8.8 Trello	10
9 Zavedení DevOps do praxe	10
9.1 1. Vybudování pocitu naléhavosti	10

9.2	2. Vytvoření vůdčí koalice	10
9.3	3. Rozvoj vize a strategie	11
9.4	4. Komunikace vize	12
9.5	5. Posílení ostatních k jednání o vizi	12
9.6	6. Vytváření krátkodobých úspěchů	12
9.7	7. Sjednocení vylepšení a přizpůsobení DevOps.....	12
9.8	8. Zakotvení nových přístupů v kultuře společnosti	12
10	Příklady z praxe	13
10.1	Nordstrom	13
10.2	Texas.gov.....	13
11	Závěr	13
	Seznam zdrojů.....	14

1 Úvod

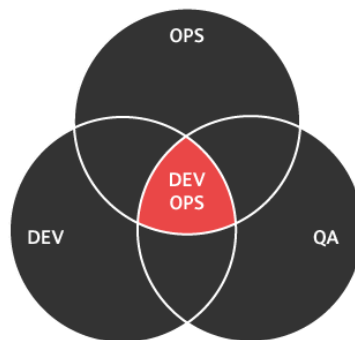
Koncept, kultura a způsob myšlení DevOps je v současné době velmi diskutovaným tématem. Mnoho firem tento koncept zavádí především z důvodů praktické potřeby zrychlit vývoj a dodání nových služeb zákazníkům. Výše uvedené důvody jsou zároveň i podnětem se tomuto tématu podrobněji věnovat nejen v praktické, ale také v teoretické rovině.

Cílem semestrální práce je představit DevOps, jak v teoretické rovině, tak na konkrétních příkladech z praxe. V teoretické části je blíže definován pojem DevOps, pojednáno o historii vzniku a principech. Dále se práce zabývá postupy, využitím DevOps při vývoji softwaru a používanými nástroji. Cílem praktické části je představit konkrétní příklady firem z praxe, které zavedly DevOps, a zhodnotit, jak dobře se jim zavedení podařilo.

Práce čerpá z odborné literatury a dalších zdrojů věnující se a blíže popisující tuto oblast. Ústředním zdrojem je kniha DevOps For Dummies, dále je čerpáno z odborných akademických prací a jako doplňující zdroj informací slouží webové stránky tvůrců jednotlivých nástrojů pro podporu DevOps. Ukázky praktických příkladů jsou čerpány z uceleného zdroje nazvaného DevOps in Practice autora J. Paul Reed.

2 Definice pojmu DevOps

Pojem a koncept DevOps vznikl spojením pojmů vývoj z anglického Development (Dev) a provoz z Operations (Ops) (LBMS, 2015). Jako Development je zpravidla označován tým vývojářů, kteří vytváří nějaký software. Operations je pak tým, který se stará o zavedení a provoz softwaru vyvinutého Development týmem. Dva týmy spolu úzce spolupracují a jejich společným cílem je nabízet zákazníkovi produkty a služby, které požaduje. S narůstajícími nároky na rychlost dodávek nového softwaru vznikla i potřeba zrychlit a zjednodušit celý proces vývoje a nasazení. Z toho důvodu vznikl koncept DevOps, jež má pomoci rozvoji aktivní spolupráce dvou doposud oddělených týmů.



Obrázek 1 Vennův diagram (Carlos, 2015)

3 Původ vzniku DevOps

Myšlenka propojení týmů Developmentu a Operations je relativně nová, její vznik lze datovat do roku 2007. Původcem myšlenky tohoto spojení je Patrick Debois.

3.1 Rok 2007

Patrick Debois si dal za cíl získat praktickou zkušenost z různých oblastí informatiky. Během 15 let vystřídal různé role v IT. Pracoval např. jako vývojář, síťový specialista, systémový administrátor, tester a projektový manažer (Ehle, 2016). Debois strávil mnoho času mezi Dev a Ops. Byl frustrován problémy řízení práce mezi dvěma týmy, které by měly vzájemně úzce spolupracovat. Z jeho pohledu však tato spolupráce nebyla tak úzká, jak by mohla a měla být.

3.2 Rok 2008

Patrick Debois byl inspirován příspěvkem “birds and feather” zveřejněným Andrew Shaferem při příležitosti Agile 2008 Conference. Tato inspirace a vědomí, že i někdo jiný než samotný Debois řeší podobný problém, vedla k založení Google skupiny nazvané Agile System Administration (Ehle, 2016).

3.3 Rok 2009

Patrick Debois odstartoval vlastní konferenci nazvanou DevOpsDays v belgickém Ghentu. O konferenci se psalo na Twitteru pod hashtagem #DevOpsDays, ale po čase byl hashtag přirozeně zkrácen na #DevOps (mj. kvůli omezení počtu znaků jedné zprávy na Twitteru) (Ehle, 2016).

3.4 Rok 2010

V tomto roce se konference DevOpsDays rozšířily do širšího světa. Konference se konaly v Austrálii i Spojených státech amerických. Myšlenka DevOps postupně získávala větší a větší počet zastánců, čímž vzniklo hnutí DevOps (Ehle, 2016).

3.5 Rok 2011

Hnutí DevOps bylo z počátku poháněno několika jedinci a open source nástroji. Později si hnutí získalo pozornost větších firem a stalo se mainstreamem. Tomuto rozšíření mimo jiné napomohla pozornost analytiků z 451 Research (Ehle, 2016).

3.6 Rok 2012

Z DevOps se stal všeobecně známý pojem a konference DevOpsDays navštěvovalo čím dál tím více zájemců (Ehle, 2016).

3.7 Rok 2013

Veřejnost byla hladová po informacích ohledně DevOps a to inspirovalo mnoho autorů k napsání knih na toto téma. Příkladem může být kniha *The Phoenix Project* od Gene Kima, Kevina Behra a George Spafforda nebo kniha *Implementing Lean Software Development* autorů Maryho a Toma Poppendieků (Ehle, 2016).

3.8 Rok 2014

Velké společnosti jako Target, Nordstrom a LEGO se staly prvními, které DevOps zavedly do své podnikové praxe (Ehle, 2016).

4 Principy a cíle DevOps

Hlavním cílem DevOps je zlepšení spolupráce všech zúčastněných od plánování až po dodávání a jeho automatizaci. Cíle lze shrnout v následujících bodech:

- Zvýšení frekvence zavádění
- Zkrácení času od vzniku business požadavku po jeho doručení zákazníkovi (tzv. time to market)
- Snížení míry selhání nových releasů
- Zkrácení času nutného pro opravu chyb
- Zkrácení času na obnovení

Dle State of DevOps Report z roku 2015 jsou IT organizace schopny zavádět 30krát častěji s 200krát kratším časem dodání. Při nasazení pak dochází k 60krát menšímu výskytu selhání a v případě, že selhání nastane, daří se systém obnovit 168krát rychleji (Puppet Labs, 2015).

5 Hodnoty DevOps (business, tým, zákazník)

DevOps silně zdůrazňuje dvě hlavní oblasti, jsou jimi kultura spolupráce a automatizace s použitím speciálních DevOps nástrojů. Zatím co někdo upřednostňuje kultura spolupráce nad používanými podpůrnými nástroji, někdo jiný vidí situaci naopak a staví do popředí nástroje před kultura. DevOps se snaží tyto dvě oblasti vyvážit a nabídnout jejich synergické spojení.

5.1 DevOps kultura spolupráce

Kultura DevOps lze charakterizovat jako kultura, ve které je kladen důraz na zvýšenou úroveň spolupráce, sdílenou zodpovědnost, autonomní týmy, zvyšování kvality, oceňování zpětné vazby a konečně zvýšení úrovně automatizace. Tento výčet není úplný a lze ho doplnit o další hodnoty, které vychází z faktu, že DevOps je rozšířením agilního přístupu.

5.2 DevOps nástroje

Nástroje DevOps zejména pomáhají se správou konfigurací, testováním, sestavováním, zaváděním, správou verzí a monitorováním. Některé z těchto nástrojů jsou podrobněji rozebrány v následujícím textu.

5.2.1 Úložiště zdrojového kódu (Repository)

Úložiště je místo, kam vývojáři ukládají zdrojový kód. Kód si stahují do svých lokálních stanic a po úpravě ho nahrávají zpět do centrálního úložiště. Změny jsou verzovány a je možné se ke starším verzím v případě potřeby vracet.

5.2.2 Sestavovací server (Build Server)

Sestavovací server je nástroj, který umožňuje automatizovanou kompilaci zdrojového kódu do spustitelné podoby. Mezi známé a rozšířené nástroje patří např. Jenkins, SonarQube a Artifactory (Ehle, 2016).

5.2.3 Správa konfigurací

Nástroj pro správu konfigurace udržuje konfiguraci nastavení prostředí. Mezi populární nástroje patří Puppet a Chef (Ehle, 2016).

5.2.4 Virtuální infrastruktura

Virtuální infrastruktura neboli cloud lze získat jako platform as a service (PaaS) od nejrůznějších poskytovatelů, např. od Amazonu nebo Microsoftu. Tato řešení nabízí aplikační programové rozhraní, přes které je možné programově vytvářet nové virtuální stroje v závislosti na aktuální potřebě. Spojení virtuální infrastruktury s nástroji pro automatizaci tvoří silnou stránku konceptu DevOps. Celý proces zavedení nového softwaru se tím nesmírně zjednodušuje a automatizuje.

5.2.5 Automatizace testů

Nástroje pro automatizaci testů existovaly již před vznikem konceptu DevOps, přesto tyto nástroje výrazně zrychlují a zjednodušují celý proces testování. Zároveň použitím těchto nástrojů lze předejít mnoha chybám, které jsou způsobeny lidskou nepozorností.

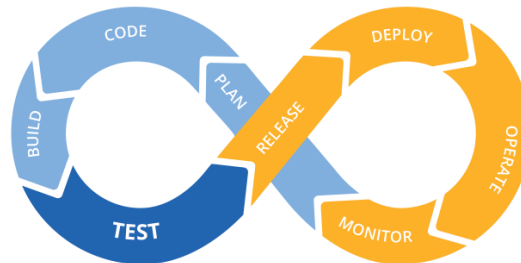
6 DevOps postupy

DevOps rozděluje velké projekty do malých celků, které jsou snadněji spravovatelné od návrhu přes nasazení až po provoz. Opakované a časté nasazování může plynule přejít z jedné skupiny do další, dokud nejsou převedeny do produkce s minimálním rizikem narušení. Malé nasazení jsou snadněji laditelná v průběhu vývojového procesu a jsou stabilní po jejich uvedení do provozu. DevOps zahrnuje všechny fáze životního cyklu vývoje softwaru:

- Definici požadavků
- Vývoj

- Testování
- Vydání
- Nasazení
- Provoz a monitoring

Avšak namísto předání na konci každé fázi DevOps navrhuje, že týmy průběžně spolupracují a komunikují, což je rozdíl oproti agilnímu přístupu, kde komunikace probíhá v pravidelných intervalech.

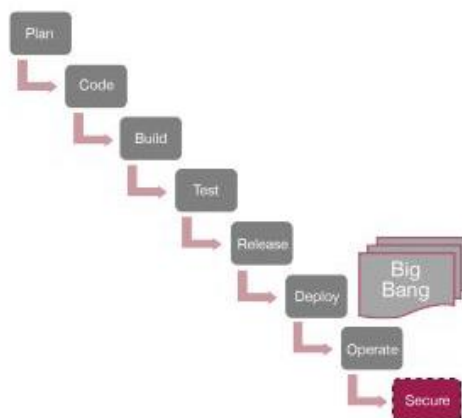


Obrázek 2 Smyčka Dev (vlevo) a Ops (vpravo) (TRICENTIS, 2016)

Kromě spolupráce a komunikace, DevOps doporučuje automatizaci všech úkolů, které jinak zvyšují pravděpodobnost selhání člověka a snižují efektivitu v celém vývojovém cyklu softwaru. Opakující se úkoly a procesy, které jsou automatizovány, snižují čas a úsilí, zvyšují kvalitu a konzistenci výstupů. Dlouhodobou vizí DevOps je dosáhnout úrovně účinnosti a automatizace, která umožní nepřetržité dodávání.

6.1 Vodopádový vývoj (Waterfall Development)

Před nepřetržitou integrací vývojářské týmy psaly kód v rozmezí 3-4 měsíců. Následně měly vývojářské týmy integrovat svůj kód za účelem finálního nasazení. Různé verze kódu většinou však byly natolik odlišné, že by úprava kódu a následná integrace mohla trvat měsíce. Tento proces je velmi neefektivní.



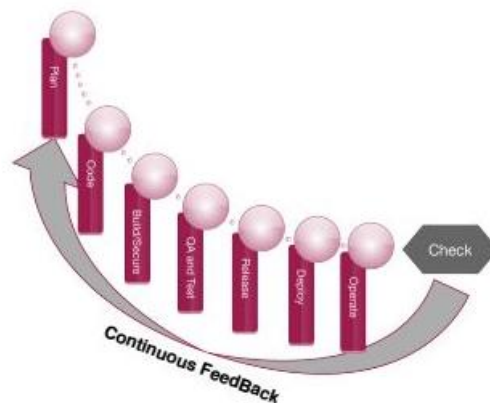
Obrázek 3 Tradiční vodopádový model (ISACA, 2016)

6.2 Nepřetržitá integrace (Continuous Integration)

Nepřetržitá integrace je nový způsob, jak rychle integrovat nově vyvinutý kód do již existujícího kódu. Tento způsob ušetří spoustu času a tým je tak dříve připraven k dalšímu uvedení.

Tento způsob vývoj nepřišel spolu s DevOps. Nepřetržitá integrace je agilním inženýrským postupem pocházejícím z metodiky Extrémního programování. DevOps přijal tento termín, protože automatizace potřebovala nepřetržitou integraci. Tento způsob je často prvním krokem k DevOps zralosti.

Nepřetržitý integrační proces z hlediska DevOps vyžaduje kontrolu kódu, kompilaci (často binární spustitelný kód) použitelný kód a následně chod některých základních ověřovacích testů.



Obrázek 4 Nepřetržitá zpětná vazba (ISACA, 2016)

6.3 Nepřetržitá dodávka (Continuous Delivery)

Jedná se o způsob dodávky kódu rozšiřující nepřetržitou integraci. Je na vrcholu nepřetržité integrace. Při provádění nepřetržité dodávky, lze přidat další automatizaci a testování, čímž lze dosáhnout téměř připraveného kódu k nasazení. Výhodou toho způsobu je absence lidského zásahu. Jedná se o způsob, který má připravit kód do stavu nasazení.

Carl Caum z PuppetLabs definuje nepřetržitou dodávku jako sérii praktik navržených, tak aby bylo zajištěno, že kód může být rychle a bezpečně nasazen do produkce. Toho lze dosáhnout dodáním každé změny kódu do produkčního prostředí a díky automatizovanému testování. (ISACA, 2016)

6.4 Nepřetržitě nasazení (Continuous Deployment)

Nepřetržitě nasazení je nejpokročilejším stupně evoluce nepřetržité dodávky. Jedná se o způsob nasazení všemi cestami do produkce bez nutnosti lidského zásahu.

Týmy, které využívají nepřetržitou dodávku, nenasazují netestovaný kód. Místo toho nově vytvořený kód podléhá automatizovaným testům, předtím než se dostane do produkce. Uvolněný kód jde obvykle jen k malému procentu uživatelů, kde je automatizovaná zpětná smyčka, která monitoruje kvalitu a použití předtím než je kód šířen dále.

7 DevOps ve vývoji software

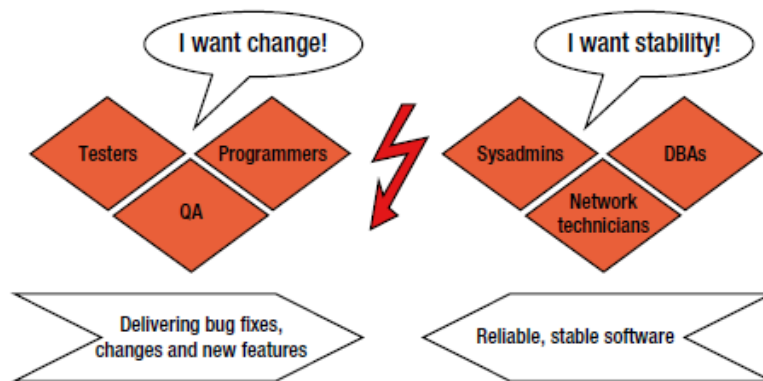
Tradičně jsou do vývoje softwaru zapojeny dva týmy ve firmě. Každý tým se zaměřuje na vlastní cíle, kterých chce dosáhnout. Vývojáři se zaměřují na samotné vytváření či úpravu kódu, takže svůj výkon měří v rychlosti vytváření nové funkcionality. Naproti tomu tým, který se stará o zavedení a provoz softwaru, svůj výkon hodnotí dle funkčnosti serverů nebo dobou odezvy aplikace.

Mezi funkce vývoje řadíme, jak vývoj, tak zajištění kvality. Dohlíží se na to, aby software splňoval požadovanou kvalitu, architekturu databází, uživatelské rozhraní. Tým Ops se pak zaměřuje na síť, servery, úložště, provoz databází a bezpečnost.

7.1 Konflikt mezi Dev a Ops

Napříč týmy vzniká konflikt, který je způsobem samotným zaměřením jednotlivých týmů. Jedná se o potřebu změny a strach ze změny. Vývojáři vytvářejí změny (nové funkce, opravy chyb) a chtějí změny pro-

mítnout na produkci. Zatímco provozní tým se snaží vyvarovat změnám, které mohou změnit podmínky potřebné pro produkční systémy, což je jejich hlavním cílem.



Obrázek 5 Konflikt Dev a Ops (Hüttermann, 2012)

Často dochází k situaci, že probíhá optimalizace v každém týmu zvlášť, namísto toho, aby se optimalizoval celý proces vývoje a provozu softwaru.

Oddělené týmy a odlišné cíle nejsou jedinými bariérami mezi týmy. Mezi další patří strach, který je charakterizován obavou z práce ostatních týmů, jejichž práce je automaticky považována za špatnou nebo odlišný terminologický jazyk neumožňující komunikaci napříč týmy.

8 Příklady nástrojů DevOps

DevOps nepřinese kýžené výhody, pokud se v týmech nepoužijí správné nástroje, které jsou zaměřené na sdílení informací a úkolů, automatickou kompilaci kódu, správu konfigurací anebo automatizaci procesů.

8.1 Puppet

Jedná se o software pro podporu správy konfigurací. Puppet je dostupný ve dvou verzích, jak v open-source řešení, tak ve variantě Puppet Enterprise. Enterprise verze automatizuje konfigurace hardwaru i softwaru. Umožňuje provádět rychlé a automatizované změny napříč fyzickými i virtuálními stroji s ohledem na konzistenci. Zkracuje dobu nasazení softwaru, umožňuje opakovaně využít definovanou konfiguraci a poskytuje detailní pohled na konfiguraci zařízení a softwaru.

8.2 Chef

Je nástroj správy konfigurací, který slouží k efektivní správě konfigurací a údržbě serverů společností. Nástroj operuje s takzvanými předpisy, které představují configurační elementy vyjadřující základní pravidla konfigurace a distribuce. Umožňuje integraci s platformami založenými na cloud řešení jako je Microsoft Azure, Amazon EC2 a další. Chef je dostupný ve třech verzích, od verze free až po enterprise řešení.

8.3 Docker

Software umožňující automatizaci nasazování aplikací do jednotlivě izolovaných kontejnerů, které obsahují veškeré kódy, systémové nástroje a knihovny pro běh aplikace. Poskytuje tak další vrstvu abstrakce a virtualizace, což zaručuje chod aplikace bez ohledu na prostředí, ve kterém je spuštěna.

8.4 Jenkins

Open-source nástroj nepřetržité integrace pro testování a reportování změn při vývoji. Umožňuje rychle detekovat chyby a provádět automatizované testování. Samotné spuštění může být inicializováno aplikováním změny kódu v systému pro správu verzí, naplánováním spuštění v nástroji Cron či dotazem ze specifické URL adresy.

8.5 Ansible

Jedná se o jednoduchý systém pro automatizaci, který pokrývá oblasti správy konfigurací a nepřetržitého nasazení. Obsahuje i sledovací systém prostředí k provádění auditu, procházení změn, vzdálené provádění příkazů a plánované spuštění činností.

8.6 Skype

Jedná se o peer-to-peer program zprostředkovávající audio a videohovory. Služba umožňuje zasílat mezi uživateli také zprávy a soubory. Pomocí Skype Video Calling lze vytvářet videokonference. Ke Skypu je možné se připojit přes webový prohlížeč nebo přes klienta pro desktop či mobilní zařízení.

8.7 Slack

Program určený ke komunikaci napříč týmy, který poskytuje uživatelům možnost chatování v tematicky oddělených kanálech a podporuje také soukromé zprávy. Slack umožňuje integrovat služby jako Google Drive, Trello, Dropbox nebo GitHub.

8.8 Trello

Webová služba určená k řízení projektu založená na systému Kanban. Pro jednotlivé projekty se tvoří takzvané boardy, které v sobě obsahují seznamy s úkoly. Úkoly je možné mezi jednotlivými seznamy přesouvat a tím vyjadřovat jejich stav. Dále je možné úkolům přiřadit tagy nebo odpovědného uživatele.

9 Zavedení DevOps do praxe

Změna kultury a procesů v jakékoliv vyspělé organizaci je složitá, ale právě to je to, co DevOps vyžaduje při zavedení do společnosti. Pro začátek je vhodné postavit proměnu společnosti na nějakém ověřeném modelu změny, který se osvědčil v minulosti. Michael Schmidt z Automoc doporučuje 8 kroků k úspěšnému zavedení DevOps do společnosti, které jsou postavené na modelu změn Johna Kottera z jeho proslulého díla Leading Change.

9.1 1. Vybudování pocitu naléhavosti

Doručit rychleji a levněji. To jsou dva nejběžnější požadavky v businessu. A je jedno jestli se jedná o dodání digitální transformace, hledání pružnější a nákladově efektivnější alternativy k internímu IT nebo jiné požadavky.

IT organizace si musí být plně vědoma naléhavosti změny, tak aby zůstala relevantní pro business. V souvislosti s vývojem a provozem je tato naléhavost měřena rychlostí a dobou uvedení nové funkce a dalšími jinými ukazateli.

9.2 2. Vytvoření vůdčí koalice

Pro úspěšnou DevOps transformaci je potřeba vyčlenit tým, který bude schopen podporovat změny a zároveň komunikovat pokrok. Tým by měl obsahovat i členy z vedení provozu a vývoje plus klíčové zainteresované osoby do DevOps procesů.

9.3 3. Rozvoj vize a strategie

DevOps vize pro nepřetržitou dodávku by měla být v souladu s business vizí. Ta by měla být dostatečně široká a měla by stačit na pokrytí všech aspektů DevOps, ale zároveň dostatečně úzká pro rozhodování v následném procesu změn.

Gartner používá termín „mezní objekty“ pro označení věcí, které lze vykládat různě v různých společnostech, ale mají dost společného k porozumění, aby byly užitečnými mechanismy pro komunikaci a zlepšení znalostí. Mezní objekty jsou považovány za možný způsob, jak sladit hlavní koncepce, objekty jsou flexibilní místním požadavkům, ale dostatečně robustní pro zachování společné identity napříč lokalitami. Nepřetržitý plán nasazení je nutností, jež zahrnuje zestručnění důležitých témat a jejich priorit na přibližné časové ose. Zabránění podrobnému plánování projektu na základě toho, že DevOps by se měla řídit agilním přístupem a nepřetržitým zlepšováním.

Měla by celá organizace přijmout DevOps nebo by měl DevOps zůstat omezen jen na oblasti, v nichž je potřeba agilní přístup?

Na to neexistuje žádná definitivní odpověď, ale je důležité mít na paměti:

- DevOps je hodně o kultuře společnosti. V dlouhodobém horizontu je obtížné udržet jednotnou společnost postavenou na dvou odlišných kulturách
- IT systémy jsou často vzájemně závislé, jednoduché změny musí být provedeny v celé řadě systémů

9.4 4. Komunikace vize

Neustálá komunikace vize, strategie a plánů, znovu a znovu. Mimo to si zažít kulturu nepřetržité dodávky jako například vyhýbat se obviňování, stavět jednotlivce před procesy a prosazovat transparentnost.

9.5 5. Posílení ostatních k jednání o vizi

Mnoho aspektů vize, jako je například snížení složitosti procesů a revidované pochopení spolupráce, by již mělo proniknout do firemní kultury, i když ještě není DevOps plně zavedeno.

Vůdčí koalice potřebuje odstranit překážky (např. příliš mnoho procesních kroků nebo schválení rigidních přístupových práv k systému). Tyto překážky omezují schopnost experimentovat s novými nástroji či metodami. Ve skutečnosti by měly být jakékoliv systémy nebo struktury, které vážně narušují vizi, nahrazeny. Rovněž riskování, netradiční nápady, aktivity a akce by měly být podporovány.

9.6 6. Vytváření krátkodobých úspěchů

Pro úspěšné zavedení DevOps a transformaci je důležité viditelné zlepšení výkonu, nejlépe na několika pilotních projektech, které využívají zřízený DevOps tým.

Při výběru pilotního týmu zajistit:

- Důvěryhodný tým schopný nepřetržité dodávky
- Vhodnou aplikaci z hlediska architektury a technologií
- Vysvětlení přínosu DevOps pro business

9.7 7. Sjednocení vylepšení a přizpůsobení DevOps

Využití důvěryhodnosti odvozené z úspěšných pilotních projektů změnových systémů, struktury a politiky, které se nevejdou do nepřetržité dodávky vize.

Různé týmy si přizpůsobují DevOps svému specifickému prostředí. Tyto týmy mohou využít svých zkušeností, procesů a nástrojů vyvinutých skrz pilotní projekt, ale nejsou povinni tak učinit, dokud jejich implementace nebude v souladu s nepřetržitým dodáváním vize. Je vhodné se vyvarovat rámcové správě, která je v rozporu s agilním přístupem.

9.8 8. Zakotvení nových přístupů v kultuře společnosti

Formulovat spojení mezi novým chováním a úspěchy společnosti s nepřetržitou dodávkou. K tomu je potřeba správných metrik spojených s nejvyšší úrovní klíčových výkonnostních indikátorů. V této fázi musí být přijat celý systém do nové kultury, tak aby se změna stala součástí běžného chodu společnosti.

Zakotvení nových přístupů musí být též podpořeno metrikami, podnětnými plány a vhodným stylem vedení.

10 Příklady z praxe

10.1 Nordstrom

Nordstrom je americký řetězec obchodních domů, který nabízí produkty online na webu nordstrom.com.

V roce 2004 byly týmy, které měly na starost webovou prezentaci společnosti, rozděleny striktně na vývoj a provoz. Samotný eshop byl velmi komplexní a nasazování změn znamenalo pro týmy velkou zátěž. V počátku implementace nové koncepce byl vytvořen DevOps tým, jehož úkolem bylo připravit změny ve firemní kultuře a připravení společnosti na změnu. DevOps tým si zvolil nástroj Chef a začal s implementací *Infrastructure as a code*.

Prvním projektem realizovaným pomocí DevOps byla změna a nasazení systému zprostředkovávajícího platby. Podle zkušeností se celková doba provedení a nasazení změn odhadovala na 18 hodin, proto se tým rozhodl při konfiguraci serverů využít nástroje Chef. Pro zabezpečení úspěšné realizace byly spojeny týmy vývojářů, databázových specialistů a další zainteresované skupiny do jednoho velkého týmu. Výsledkem několika týdnů spolupráce bylo snížení potřebné doby z 18 hodin na 4 hodiny a zcela automatizovaný proces nasazování změn na servery. (REED, 2014)

10.2 Texas.gov

Oficiální stránky státu Texas poskytující základní informace a služby týkající se například žádostí o řidičské a zbrojní průkazy, potvrzování a vydávání dokumentů, placení daní a jiné. Softwarový vývoj zajišťuje pro stránky texas.gov firma Texas NICUSA, LLC.

V dané oblasti existují specifika, zejména se jedná o nutnost vyvíjet a nasazovat software určitým způsobem a v souladu s platnou legislativou. K využití DevOps se firma rozhodla poté, co byl zadán projekt, který nemohl být uskutečněn jiným způsobem. Stejně jako v případě první firmy Nordstrom, byl vytvořen tým s názvem *Continual Service Improvement*, který měl na starost zavedení DevOps a změnu firemní kultury.

Specifikem nebylo pouze legislativní omezení, ale také zvýšená potřeba zabezpečení dat uživatelů. Proto byl součástí každého agilního týmu expert v oblasti bezpečnosti. Do procesu tvorby produktu byla z hlediska bezpečnost zahrnuta tak, že v každé iteraci byl uskutečněn bezpečnostní audit. Pomocí DevOps docházelo ke kooperaci vývojářů, bezpečnostních expertů a provozního týmu. Koncept DevOps byl použit k zajištění bezpečnosti, jakož to jedné vlastnosti, vytvářeného produktu. (REED, 2014)

11 Závěr

Definované cíle se podařilo naplnit v jejich plném rozsahu. V první části práce byl představen koncept DevOps a stručná historie. Dále byly popsány principy, kultura a jednotlivé nástroje, které jsou definovány konceptem DevOps. Byl kladen důraz především na ústřední hodnoty DevOps, které jsou základními kameny konceptu DevOps. V druhé polovině teoretické části jsou popsány postupy, konflikty mezi Dev a Ops a příklady nástrojů. V závěru teoretické části následoval popis zavedení DevOps do praxe. Před samotným závěrem práce byly popsány dva praktické příklady zavedení DevOps do praxe, konkrétně ve společnostech Nordstrom a Texas.gov. V průběhu vypracovávání práce nevznikly žádné problémy ani zásadní omezení, pro téma DevOps existuje dostatek podkladů a zdrojů.

Seznam zdrojů

- SHARMA, Sanjeev. DevOps For Dummies. 2014, Hoboken: John Wiley & Sons., ISBN 978-1-118-73378-3
- KRISTINSSON, Rögnvaldur, 2015. Software Development with DevOps. Haaga-Helia, University of Applied Sciences. Bakalářská práce. Haaga-Helia.
- SLATINSKÝ, Karel. Integrace agilních a DevOps přístupů do ITIL. Brno. Diplomová práce. Masarykova univerzita. Fakulta informatiky. Vedoucí práce Barbora BÜHNOVÁ.
- TUPÁ, Monika. Aplikace ITIL na servisní projekty v IT. Brno. Diplomová práce. Masarykova univerzita. Fakulta informatiky. Vedoucí práce Jaroslav RÁČEK.
- ISACA, 2015. DevOps Overview. Rolling Meadows: ISACA.
- LBMS, 2015. DevOps. [online]. [cit. 12. 3. 2016]. Dostupné z: <http://www.lbms.cz/reseni/devops/>
- NEW RELIC. DevOps: Collaborative Software Development. [online]. [cit. 12. 3. 2016]. Dostupné z: <http://newrelic.com/devops>
- EHLE, Dennis, 2016. What Is DevOps? Versionone [online]. [cit. 12. 3. 2016]. Dostupné z: <https://www.versionone.com/devops-101/what-is-devops/>
- LOUKIDES, Mike, 2012. What is DevOps? In: O'Reilly Radar. [online]. [cit. 12. 3. 2016]. Dostupné z: <http://radar.oreilly.com/2012/06/what-is-devops.html>
- BESTPRACTICES.CZ. DevOps. [online]. [cit. 12. 3. 2016]. Dostupné z: <https://www.bestpractice.cz/cs/Best-practice/DevOps.alej>
- SKYTAP INC., 2015. Accelerate Software Development with DevOps and Hybrid Cloud. [online]. Místo: Seattle, 6. 2016. [cit. 11. 3. 2016]. Dostupné z: https://www.skytap.com/wp-content/uploads/2015/06/skytap_whitepaper_accel_software_dev_devops.pdf
- MEHROTRA, Sumit, 2014. How to accelerate software development by harnessing the power of DevOps, agile and hybrid cloud. In: NetworkWorld. [online]. [cit. 11. 3. 2016]. Dostupné z: <http://www.network-world.com/article/2176850/tech-primers/how-to-accelerate-software-development-by-harnessing-the-power-of-devops--agile-and-hyb.html>
- HEWLETT PACKARD ENTERPRISE DEVELOPMENT, 2016. DevOps. [online]. [cit. 11. 3. 2016]. Dostupné z: <http://www8.hp.com/us/en/software-solutions/devops-solutions/>
- FORRESTER. Application Delivery Speed Drives Success. [online]. [cit. 12. 3. 2016]. Dostupné z: <https://ssl.www8.hp.com/ww/en/secure/pdf/4aa6-3329enw.pdf>
- IBM. IBM DevOps. [online]. [cit. 12. 3. 2016]. Dostupné z: <http://www.ibm.com/ibm/devops/us/en/>
- SEARCHSOFTWAREQUALITY. Accelerating Application Development With DevOps.[online]. [cit. 12. 3. 2016]. Dostupné z: http://tngconsultores.com/kw/pluginfile.php/53/mod_glossary/attachment/893/DevOps_eGuide.pdf
- CA TECHNOLOGIES, 2016. So You Think You Can DevOps? In: SlideShare.net [online]. [cit. 12. 3. 2016]. Dostupné z: <http://www.slideshare.net/CAinc/so-you-think-you-can-devops>
- JAVA PORTÁL, 2010. Operations a DevOps. [online zvukový záznam ve formátu MP3]. Dostupné z: <http://www.java.cz/article/czpodcast-39-operations-devops>
- PUPPET LABS, 2015. State of DevOps Report. Dostupné z: <https://puppetlabs.com/sites/default/files/2015-state-of-devops-report.pdf>
- PUPPET LABS, 2016. Puppet enterprise [online]. [cit. 30. 3. 2016]. Dostupné z:

<https://puppetlabs.com/puppet/puppet-enterprise>

CHEF SOFTWARE, INC., 2016. Get started with Chef [online]. [cit. 30. 3. 2016]. Dostupné z: <https://learn-chef.io/tutorials/>

DOCKER., 2016. What is Docker [online]. [cit. 30. 3. 2016]. Dostupné z:

<https://www.docker.com/what-docker>

JENKINS, 2016. Jenkins documentation [online]. [cit. 30. 3. 2016]. Dostupné z:

<https://jenkins.io/doc/>

ANSIBLE, 2016. Ansible [online]. [cit. 30. 3. 2016]. Dostupné z: <https://www.ansible.com/>

SLACK, 2016. Slack Guides [online]. [cit. 1. 4. 2016]. Dostupné z: <https://slack.com/>

TRELLO, 2016. Trello Tour [online]. [cit. 1. 4. 2016]. Dostupné z: <https://trello.com/tour>

MICROSOFT, 2016. About Skype [online]. [cit. 1. 4. 2016]. Dostupné z: <https://www.skype.com/en/about/>

SCHMIDT, Michael, 2015. Eight steps to a sustainable DevOps transformation [online]. [cit. 2. 4. 2016]. Dostupné z: <http://devops.com/2015/06/29/eight-steps-sustainable-devops-transformation/>

REED Paul, 2014. DevOps in practice. 1. vydání. Sebastopol: O'Reilly Media, Inc. 1. edice, ISBN: 978-1-491-91306-2.

HÜTTERMANN Michael, 2012. DEVOPS FOR DEVELOPERS [online]. Dostupné z: <http://huettermann.net/devops/mismatch.png>

TRICENTIS. TOSCA TESTSUITE: TESTING THE MISSING PIECE IN DEVOPS Tosca Testsuite: Testing The Missing Piece in DevOps Tosca. [online]. [cit. 9. 4. 2016]. Dostupné z: <http://www.tricentis.com/solutions/devops/>