

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Tailoring your DevOps to organisational structure

Semestrální práce

Autoři:

Natalie Zacharova (xzakn00)

Filip Krejčí (xkref08)

Petr Dvorský (xdvop41)

Datům: 16.12.2017

Abstrakt

Tato semestrální práce pojednává o hnutí DevOps neboli Development and Operations což znamená přístup k vývoji softwaru zdůrazňující důležitost kooperace vývojářů a odborníků na informační technologie z provozu. Cílem je vysvětlit proč přizpůsobení organizační kultury je důležitým faktorem pro správné fungování metodiky DevOps v organizacích. Pro dosažení tohoto cíle jsou stanovené 2 dílčí cíle:

- Vysvětlit základní principy fungování metodiky DevOps.
- Vymežit pojem organizační kultura.

Klíčová slova

DevOps, Development and Operations, Organisational Culture, Agile.

Obsah

1	Úvod	4
1.1	Definice DevOps	5
1.2	Pilíře DevOps	5
1.3	Důvody použití DevOps	6
1.4	Osvědčené postupy při adaptaci DevOps	7
2	Seznam literatury	9

1 Úvod

Procesy, funkce, dokumentace a role tvoří základní stavební prvky modelů a metodik, které se zabývají řízením informačních systémů a technologií. Mají za úkol zajistit co možná nejefektivnější provoz a rozvoj informatiky v rámci podniku. Tyto modely a metodiky jsou většinou těsně navázané na podnikové řízení a poskytují pro něj nástroje (Pour, 2006).

Mezi metodiky patří i metodiky agilní, jedná se o různé metodiky, které vznikaly od druhé poloviny 90. let a které prosazují myšlenku, že jedinou cestou, jak prověřit správnost navrženého systému, je vyvinout jej (nebo jeho část) co nejrychleji, předložit zákazníkovi a na základě zpětné vazby jej upravit. Každá z agilních metodik je svým způsobem specifická, ale všechny jsou postaveny na stejných principech a hodnotách (Buchalceková, 2005).

Agilní metodiky zažívají v současné době značný rozvoj a jejich seznam se neustále rozšiřuje. Mezi ně se řadí také metodika Scrum a hnutí DevOps.

Metodika Scrum je zaměřena především na řízení projektu. Vývoj software probíhá v rámci 30 denních iterací nazývaných Sprint, na jejichž konci je dodána vybraná množina užitečných vlastností. (Buchalceková, 2005).

DevOps stejně jako každá nová technologie nebo populární hnutí, se stalo jakýmsi „buzzwordem“ - každý o tom hovoří, ne všichni ví, o co jde. DevOps dá se říci, je jakýsi akcelerátor agilního řízení projektů SCRUM v celém životním cyklu softwarů.

„Manifest agilního vývoje software“ uvádí čtyři hodnoty, přičemž prvky na levé straně mají větší relativní význam než prvky na pravé straně (Fowler, a další, 2001). Dávají přednost:

- jednotlivcům a interakci před procesy a nástroji;
- fungujícímu softwaru před obsažnou dokumentací;
- spolupráci se zákazníkem před sjednáváním kontraktu;
- reakci na změnu před následováním plánu.

Z výše uvedených hodnot je zřejmé, že cílem agilních metodik je, aby se vývoj více orientoval na zákazníka.

1.1 Definice DevOps

V současné době neexistuje jednotná definice DevOps a v odborné literatuře tak můžeme najít různé formulace, z nichž každá nahlíží na pojem DevOps z odlišného hlediska.

Podle Agile Admin (The Agile Admin, 2010) znamená DevOps hnutí vyplývající ze dvou spřízněných trendů. Prvním je agilní vývoj a druhým je rozšířené chápání hodnoty spolupráce mezi vývojem a operativním řízením během všech fází vývojového cyklu softwarů.

Patrick Debois, zakladatel DevOps hnutí, pak hovoří o DevOps jako o konceptu, který vznikl kvůli rostoucímu nepochopení a nízké spolupráce mezi týmy vývoje a provozu ve velkých softwarových společnostech (Debois, 2009). Autor dále uvádí, že DevOps je lidský problém. Tím je tedy myšleno, že každá organizace by měla mít svou unikátní kulturu DevOps.

Wikipedie (2016) sjednocením různých zdrojů definuje DevOps jako „přístup k vývoji software, který zdůrazňuje komunikaci, spolupráci a integraci mezi vývojářem a odborníky na informační technologie z provozu. DevOps je reakcí na vzájemnou závislost vývoje softwaru (Dev) a IT provozu (Ops). Jeho cílem je pomoci organizaci rychle produkovat softwarové aplikace a služby.“

1.2 Pilíře DevOps

Zatím co neexistuje žádný skutečný způsob „jak dělat DevOps“ a spousta definic se jen pokouší vystihnout komplexitu DevOps. Lepším způsobem pro pochopení podstaty tohoto hnutí jsou čtyři pilíře DevOps:

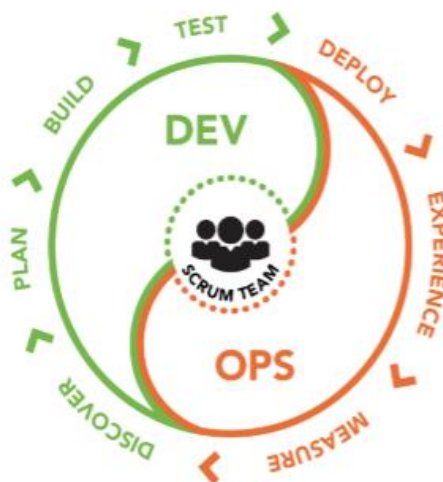
- **Kultura** – DevOps vyžaduje změnu kultury společnosti tak, aby akceptovala společná rozhodnutí vedoucí k dodání vysoce kvalitního softwaru koncovému uživateli. Znamená to, že programový kód už nemůže být jednoduše předán operačním činnostem bez vzájemné komunikace.
- **Automatizace** – DevOps se spoléhá na plnou automatizaci tvorby, nasazení a testování tak, aby bylo dosaženo co nejkratších časů nasazení a následně rychlého dodání a zpětné vazby od konečného uživatele.
- **Měření** – přispívá k lepšímu porozumění současného stavu a k nastavení cílů. K otestování trhu a času potřebného k nasazení nové verze softwaru mohou být použity běžné obchodní metriky, např. zisk.
- **Sdílení** – Sdílení se odehrává na různých úrovních, od sdílení znalostí, např. o nové funkcionalitě v nové verzi, po sdílení nástrojů a infrastruktury (Humble and Molesky, 2011).

1.3 Důvody použití DevOps

Mezi hlavní důvody použití DevOps patří:

- Lepší zkušenost zákazníka – dodat takovou službu, která zákazníkovi umožní zvýšit zisk a zlepšit konkurenceschopnost.
- Zvýšená kapacita pro inovace – moderní organizace používají štlhlé principy uvažování pro zvýšení jejich kapacit na inovace. Cílem je odstranit odpad a přepracovávání a přesunout zdroje k aktivitám s vyšší hodnotou. Zde lze využít tzv. AB testování, kdy jsou malé skupině zákazníků předány dvě verze softwaru s jinou sadou schopností. Ta verze, která je úspěšnější, je pak dodána všem zákazníkům. To je možné pouze s automatickými mechanismy, které DevOps poskytuje.
- Rychlejší čas vytvoření hodnoty – zkrácení času potřebného k dodání hodnoty vyžaduje takovou kulturu, automatizaci a praktiky, které umožňují rychlou, efektivní a spolehlivou dodávku softwaru napříč celým procesem vývoje (Sharma, 2014).

Neoddělitelnou součástí výsledku tak jak ho vnímá zákazník, je služba jako celek, a to v případě softwaru zahrnuje veškerý cyklus: správné naplánování, zmapování požadavků, vývoj kódu, testování, řízení releasu, nasazení do produkčního prostředí a následnou podporu. (viz Obr. č.1).



Obrázek č.1 DevOps cyklus (DevOps Institute,2017)

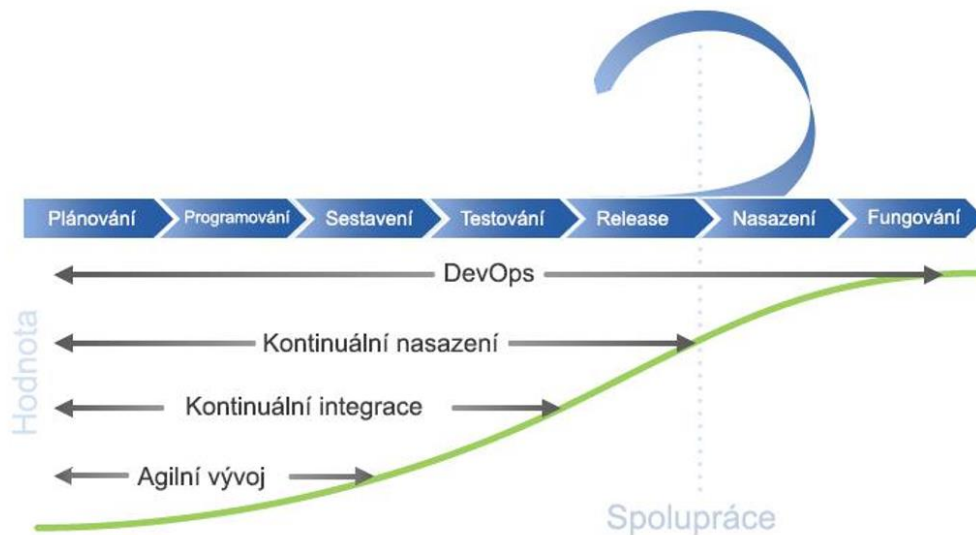
1.4 Osvědčené postupy při adaptaci DevOps

Sharma (2014) doporučuje 5 osvědčených postupů, které je třeba vzít v úvahu při adaptaci DevOps:

- Plánování předání produktu uživatelům (Release planning) – plánování nasazení jedné nebo více změn služby IT, které jsou sestaveny, testovány a nasazeny najednou. Většina společností dnes plní tento úkol pomocí tabulek a schůzek (často dlouhých) se všemi zainteresovanými stranami v celé společnosti. Dobře definované procesy a automatizace však eliminují potřebu těchto tabulek a zdlouhavých schůzek a umožňují efektivnější release plánování.
- Kontinuální integrace (Continuous integration) – kontinuální integrace umožňuje neustálé testování a kontrolu kódu. Ujistěte, že hotový a integrovaný kód je v souladu s kódem jiných vývojářů a dalších aplikačních komponent a chová se tak, jak bylo navrženo. Cílem je vyhnout se možnostem integračních problémů, které přicházejí z velkých, nepravidelných sloučení kódu.
- Kontinuální nasazení (Continuous delivery) – kontinuální nasazení je přirozeným výsledkem kontinuální integrace. V procesu automatického nasazování softwaru pro testování, systému testování a produkčního prostředí umožňuje DevOps využívat stejné automatické procesy ve všech prostředích pro zlepšení efektivity a redukovat rizika vyplývající z nekonzistentních procesů. V testovacím prostředí urychluje automatická konfigurace obnovování testovacích dat a tím nasazování softwaru do testovacího prostředí.
- Kontinuální testování (Continuous testing) - do kontinuálního testování patří: řízení testovacích dat, integrační testování, funkční testy, bezpečnostní testy a další.

- Kontinuální monitorování a zpětná vazba zákazníka (Continuous monitoring and feedback) - zpětná vazba zákazníků přichází v různých formách, jako jsou žádosti změn (Change Request) nebo neformální stížností. Podniky potřebují dobře definované procesy, aby dokázali reagovat na zpětnou vazbu zákazníků, tyto procesy by měli být dostatečně agilní. Zpětná vazba také pochází z monitorování serveru, na kterých je aplikace spuštěna.

Oblasti pokrytí DevOps umožňují obchodu maximalizovat rychlost dodání IT služby, a to od počátečního nápadu přes vydání softwaru, zpětnou vazbu od zákazníka až po vylepšení založená na této zpětné vazbě. Na obrázku níže je vidět, jak tyto principy fungují společně s agilním vývojem:



Obrázek 6: Oblasti pokrytí DevOps. Zdroj: (CollabNet, 2016), upraveno autorem

1.5 Organizační kultura

Pro správné pochopení principů DevOps a důležitosti kultury při jeho využívání je důležité si přiblížit pojem organizační kulturu. Organizační kultura byla vždy důležitým prvkem, nýbrž důležitou a často rozebíranou se stala v posledních dvou desetiletích. Jedná se o nehmotný koncept a je těžké přímo definovat její dopady a přínosy, ale jelikož ovlivňuje fungování celé organizace, je podstatné se tématem organizační kultury do hloubky zabývat.

Neexistuje jedna jasná definice organizační kultury, ale zkráceně se dá říct, že se jedná o soubor základních přesvědčení, postojů, hodnot a norem chování, které jsou sdíleny v organizaci.

Následující prvky můžeme tedy označit jako základní stavební kameny kultury (jak hmotné, tak nehmotné):

- Základní představy
- Hodnoty
- Artefakty
- Postoje

- Normy

Tyto prvky, nebo také faktory dále ovlivňují fungování celé organizace díky projevům myšlení, citění a chování zaměstnanců. To má samozřejmě přímý vliv na směřování a vývoj organizace.

Z tohoto důvodu je důležité aby se zaměstnanci co nejvíce ztotožňovali s organizační kulturou dané organizace. Kultura je ovlivňována různými faktory napříč organizací.

Mezi takové faktory můžeme zařadit například:

- Odvětví, ve kterém se organizace pohybuje
- Geografická poloha sídla organizace
- Základní vize organizace
- Velikost organizace
- Historie vývoje organizace
- Management a jeho způsoby řízení

Největší vliv na organizační kulturu mají její zaměstnanci. Je taktéž důležité aby byla kultura jasně čitelná a dominantní. To zapříčiní lepší čitelnost z pohledu zaměstnanců a zaměstnanci se tak můžou s kulturou lépe ztotožnit.

Každá organizace stojí na zaměstnancích a pokud jsou schopni akceptovat kulturu, tak z toho plyne pro organizaci nespočet výhod jako jsou například: příjemné pracovní prostředí, zvýšená důvěra, efektivní prostředí, týmová práce, sdílení informací, důvěra v organizaci, přijímání nových zaměstnanců.

1.6 Vliv organizační kultury na fungování DevOps

Především kultura a splynutí dvou různých oddělení, tedy development a operations, je stěžejní pro fungování principů DevOps. Proto je důležité se zabývat vlivem organizační kultury na používání principů DevOps.

Častým problémem při zavádění DevOps je právě přehnaná důvěra v nástroje a věnování pouze malé pozornosti kultuře. DevOps kultura je chápána jako soubor několika nejdůležitějších faktorů, které jsou stěžejní pro správné fungování DevOps.

1.6.1 Komunikace

Základním stavebním kamenem nejsou nástroje ani procesy či best practices, ale především schopnost zaměstnanců developmentu a operations mezi sebou komunikovat. Pokud se vytvoří blok v komunikaci mezi těmito celky není možné aby DevOps správně fungovalo. Právě bezproblémový přechod informací a z development na operations je jedním z hlavních cílů DevOps.

1.6.2 Důvěra ve změnu

Je prací především vedoucích pracovníků aby zaměstnance připravili na přicházející změny. Tato změna může být sebelepší, ale pokud je prosazována bez důvěry zaměstnanců, tak má jenom malou šanci na úspěch. Jedním z fungujících postupů přesvědčení většinu zaměstnanců je začít v menším měřítku. Lépe se aplikuje změna na jeden menší projekt než na celou organizaci. Tato skupina poté může jít příkladem úspěšné změny.

1.6.3 Upřímnost, otevřenost a transparentnost

Dotýká se především práce a výsledků developmentu a operations. Pokud jsou rozhodnutí jak developmentu a operations jasně vidět a můžou je jednotlivé teamy konzultovat, pomůže jim to lépe se dohodnout na jednotlivých řešeních. Každá z těchto stran dělá různé kompromisy, a tyto kompromisy se

nemusí vždy shodovat, dokonce můžou být v naprostém kontrastu. Proto je důležité aby každý mohl nahlížet z různých úhlů pohledu a k tomu dopomůže právě transparentnost mezi jednotlivými celky.

1.6.4 Důvěra

Důvěra v druhé je důležitým faktorem, protože development a operations si neustále navzájem předávají práci. A pokud mezi nimi nefunguje jakési pouto důvěry, tak nemůžou správně fungovat. Většinou mezi nimi vznikají nedorozumění a nepochopení. Je to především kvůli tomu, že development a operations mají každý naprosto jiný krátkodobý cíl. Proto lepší propojení pomocí vzájemné důvěry a transparentnosti mělo přinést vzájemné pochopení a odbourat časté nedorozumění.

1.6.5 Spolupráce

Tento faktor chápeme jako schopnost jednotlivých zaměstnanců spolupracovat na všech možných úrovních a po dobu celého cyklu DevOps. Nestačí pokud jsou vývojáři schopni produkovat kvalitní kód, pokud ho nejsou navazující celky schopni dále otestovat, správně nasadit, monitorovat, plánovat a na základě monitorování znovu vyvíjet.

1.6.6 Žádné obviňování

Při špatně nastavených zodpovědnostech a nejasně definovaných úkolech dochází k obviňování a svádění chyb na ostatní. Ukazování prstem nemá v DevOps kultuře žádné místo, jelikož se jedná o nekonstruktivní, většinou silně emotivní jednání, které nijak nepřispívá k řešení problému. Naopak nabourává koncepty důvěry a transparency, které jsou pro DevOps nadmíru důležité.

1.6.7 Motivace

Správná motivace by měla fungovat v každé organizaci. Zaměstnanci, kteří nedůvěřují vlastní práci a nevidí v ní hlubší smysl nejsou pro organizaci velkým přínosem.

1.6.8 Společné cíle

Je důležité, aby všichni, koho se DevOps dotkne chápali a měli jasně definované společné cíle. Pomůže to pracovníkům důsledněji pochopit jednotlivé procesy v organizaci a zároveň podpoří motivaci. Zaměstnanci jsou lépe schopni pochopit proč plní jednotlivé úkoly a jaký to má důsledek. Měli bychom tedy zaměstnancům pomoci nahlížet na jejich úkoly z většího měřítka. Jedním z hlavních společných cílů je především cílový zákazník.

1.6.9 Umění přijmout selhání

V mnoha organizacích je chyba vnímána jako koncový stav, který poškozuje organizaci. Proto je k chybám přistupováno velice negativně. DevOps kultura naopak říká, že chyba je příležitost ke zlepšení. Pokud jsou chyby kvalitně řízeny a odbourá se přístup averze, můžou se v organizaci lépe aplikovat změny. Rovněž roste důvěra zaměstnanců, jelikož nemají nadále pocit, že musí být neomylní. Je tedy dobré vytvořit prostředí, které je schopné rychle reagovat na chyby, ale nikoli za cenu omezení růstu organizace.

1.6.10 Nástroje

Kulturu samozřejmě tvoří i používané DevOps nástroje. Tyto nástroje jsou obsluhovány lidmi, kteří musí chápat principy a cíle DevOps. Nástroje sloužící k využívání DevOps jsou pouze poslední třešničkou na dortu, jelikož vyžadují uživatele, kteří jsou schopni akceptovat DevOps kulturu a jejich vnímání a chování tomu odpovídá. Proto se k DevOps nepřístupuje jako k nástroji či procesům, ale především jako ke kultuře, která je těmito nástroji podpořena.

1.7 Adaptace organizační kultury

Pokud se povede vytvořit v organizaci mezi development a operations kulturu, která se ztotožňuje s uvedenými faktory, tak je organizace schopna na plno využít všeho co DevOps nabízí. Pokud ovšem takovou kulturu nemá, je důležité se zaměřit na její transformaci. Tato transformace vyžaduje trpělivost a nepřichází ze dne na den. Především kvůli faktu, že kultura má hluboké kořeny a závisí jen a jen na zaměstnancích a přesvědčovat lidi o správnosti změn je vždy dlouhý a obtížný proces. Pokud se tento proces úspěšá a výsledky jsou očekávány příliš brzy může na to organizace doplatit a synergie mezi developmentem a operations tak vůbec nenastane.

Právě adaptace organizační kultury je často podceňována a díky tomu DevOps nefunguje správně. S tím souvisí špatné přístupy k DevOps a to především díky špatnému pochopení základních principů DevOps a toho, jak je aplikovat.

2 Mylné představy o DevOps

V minulých kapitolách bylo představeno DevOps, jeho důvody použití a osvědčené postupy při adaptaci DevOps. Dále jsme představili organizační kulturu a její důležitost pro DevOps.

Pojďme se před samotným závěrem práce podívat na DevOps z trochu jiného pohledu. A to, jak DevOps neaplikovat a jakých chyb je třeba se vyvarovat.

V současné době je velice populární tvrzení firem, o tom, že si osvojily DevOps nebo jinou metodiku či best practice. Takové tvrzení může sloužit jako výborná propagace pro přilákání nových talentů do týmů. Samozřejmě taková prohlášení mohou přilákat i potenciální zákazníky, kteří v takových tvrzeních vidí něco jako garanci dodávky jimi požadovaného softwaru.

Dále je potřeba uvést, že společnosti si rády upravují metodiky podle sebe nebo využívají jenom některé jejich části. To vede k tomu, že o sobě mohou říci, že využívají danou metodiku, ale využití jenom části metodiky nevede k 100% optimalizaci, kterou metodika nabízí. DevOps není titul nebo ocenění, které je možné získat. DevOps je kultura, filozofie a způsob jak přistupovat k dodávání kódu z vývoje do produkce.

V následujících podkapitolách jsou zmíněné, podle nás, nejčastější a nejzávažnější mylné představy o implementaci DevOps.

2.1 „Potřebujeme koupit DevOps“

IT oddělení potřebují ke své práci hardware: servery, počítače, či síťové prvky. Potřebují také software: OS, antivirové systémy, monitorovací systémy, různé aplikace pro fungování celé firmy atd.

Všechny tyto věci se musí nakoupit. Pro CIOs je tak přirozené, že si myslí, že mohou koupit další věc, která jim pomůže k lepšímu fungování firmy. A to je důvod, proč některé firmy nabývají mylné představy, že DevOps je něco co mohou koupit. Jak již bylo řečeno, DevOps je filozofie, je to kultura. Kultura, tedy lidé, je jedním ze základních stavebních pilířů DevOps.

To však neznamená, že DevOps je zadarmo. Je možné zaplatit si školení o DevOps, či nějaký nástroj, který DevOps v organizaci podpoří. Nákup těchto věcí je ale jednou z mnoha aktivit, které je ve

společnosti třeba udělat aby se dosáhlo pravého účinku DevOps. Rozhodně ale nepatří mezi aktivity, které by měly být uskutečněny mezi prvníma.

2.2 „Chceme DevOps, hned!“

Tlak shora na manažery IT není žádnou novinkou. Business požaduje spolehlivé, user-friendly prostředí, které zaměstnancům umožní pracovat rychleji a efektivněji a tím společnost vydělá více peněz.

Ano, IT je ve firmách od toho, aby podporovalo business. Faktem je, že pro správný chod IT je potřeba určité metodiky, určitého přístupu, či způsobu myšlení. Nic z toho však nelze implementovat přes noc. Jelikož je kultura organizace jedním z hlavních pilířů DevOps, a jelikož se jedná o lidský faktor, je velice obtížné kulturu přizpůsobit pro potřeby nejen DevOps ale jakékoliv jiné metodiky. Zaměstnanci firmy jsou zvyklí na pracovní postupy, které využívají denně po dobu několika let. Každá velká změna je doprovázena řadou námitek a přesvědčení zaměstnanců o tom, že nový postup je lepší je *běh na dlouhou trať*. Proto DevOps nelze aplikovat *přes noc*.

2.3 DevOps není jenom sada nástrojů

Dokážeme si představit, že CIO řekne „Koupíme Chef nebo Puppet a ten nám vyřeší náš problém. A zároveň budeme mít zavedené DevOps“. O tom ale DevOps není. Samozřejmě, pokud organizace nemá software, který by DevOps podpořil, je dobré nějaký obstarat.

Automatizace je také jedním ze základních pilířů a pro automatizaci je potřeba software. Takový software si může firma koupit, nebo vytvořit vlastní, který sedí přesně pro potřeby organizace. Nesmí se ale stát, že organizace začne ignorovat ostatní oblasti DevOps. Automatizace a sní spojené nástroje jsou opět jen jednou oblastí DevOps.

Skvělý potenciál DevOps nelze získat bez nástrojů. Pokud ale firma ignoruje ostatní aspekty DevOps a soustředí se pouze na nástroje, nemůže nikdy dosáhnout plného potenciálu DevOps.

2.4 „Nemáme čas zavádět automatizaci“

Tohoto tématu jsme se dotkli již v minulé podkapitole. Nyní rozvedeme, proč je dobré zavést automatizaci v souvislosti s DevOps.

„Nemáme čas zavádět automatizaci“ – opět se dostáváme k tlaku na zaměstnance shora. Pokud zaměstnanec nebo manažer IT nedokáže odůvodnit přínosy automatizace a tím přesvědčit CIO a top management k automatizaci v souvislosti s DevOps, tak firma nebude nikdy automatizovat a tím nedosáhne potenciálu, který DevOps nabízí.

DevOps říká, že pro dosažení co největšího potenciálu je potřeba automatizovat co největší počet manuálních procesů. DevOps doporučuje automatizaci testování, nasazení, serverových nastavení atd.

Věta „Nemáme čas zavádět automatizaci“ také naznačuje časově náročný proces zavedení automatizace, na který bohužel zaměstnanci nemají čas z důvodu pracovní vytíženosti. Ano, zavedení automatizace je časově náročný proces, ale podle DevOps je výhodné investovat trochu více času do automatizace, protože se tato investice vrátí v krátkém časovém horizontu.

Pokud firma automatizuje doporučené procesy, tak se tato časová investice za krátkou dobu vrátí, protože každý automatizovaný proces bude rychleji a předpokládá se, že i spolehlivěji zpracovaný. Organizace, která se rozhodne neautomatizovat, nemůže dosáhnout potenciálu DevOps.

2.5 Kontinuální vydávání kódu k nasazení

Přesuňme se tedy od automatizace k frekvenci nasazování. Účelem DevOps je rychleji opravit chyby a rychleji produkovat nové funkce podnikových aplikací. Velké množství organizací praktikuje rigorózní metod jako je například *Waterfall model*. Tento typ metod se bohužel pro DevOps nehodí. Pro DevOps je potřeba agilních metod. S tím souvisí i kontinuální vydávání a nasazování.

Ve své podstatě jsou agilní metody tvořeny rychlým vydáváním malých změn. Dá se říci, že DevOps je vlastně synonymem pro koncepty *kontinuální integrace a kontinuální nasazování*. Zde je klíčovým slovem slovo *kontinuální*. Pokud firma vydává změny kódu méně často než jednou za rok, nelze o tom mluvit jako o kontinuálním vydávání/nasazování. V této oblasti je zde DevOps proto, aby se zkrátil interval vydávání kódu do produkce ze dnů/měsíců na hodiny, popř. minuty. K tomu samozřejmě pomáhá kultura organizace a také automatizace procesů, které dříve společnost prováděla manuálně. Pokud firma vydává změny méně často než jednou ročně, praktikuje DevOps špatně.

2.6 „Máme rozpočet pro DevOps“

Vraťme se ještě ke kapitole 2.1 „*Potřebujeme koupit DevOps*“. Jak jsem již uvedl, DevOps není něco co firma nakoupí, tudíž nelze v prvních aktivitách zavedení DevOps plánovat rozpočet na DevOps. Pokud toto firma udělá, je to krok špatným směrem a bohužel také jeden z prvních kroků zavedení DevOps. Jak již bylo zmíněno, základní aktivitou je změna organizační kultury, poté je možné přemýšlet nad tím jak zafinancovat DevOps.

Rozpočet pro DevOps je sekundární aktivitou. Hlavní při implementaci DevOps je přizpůsobit organizační kulturu a tím připravit vhodné prostředí pro implementaci DevOps.

3 Závěr

V práci jsme definovali pojem DevOps a jeho hlavní pilíře, kterými jsou:

- Kultura
- Automatizace
- Měření
- Sdílení

Dále jsme pro tuto práci objasnili pojem *organizační kultura* a její vliv na fungování DevOps. Jelikož je kultura jedním z pilířů DevOps, je její vliv na fungování této filosofie zásadní. V neposlední řadě jsme naznačili adaptaci organizační kultury.

Jako poslední část práce jsme zvolili téma *Mylné představy o DevOps*, kde jsme zmínili, podle nás, nejmarkantnější chyby při zavádění DevOps.

Kultura se často považuje za měkký aspekt IT, ale pro devops má nejvyšší možnou důležitost. Představuje to místo, kde společnosti často nezvládnou dosáhnout příslibu devops. Mohou zvládat automatizaci s optimální kombinací nástrojů, mohou průběžně aktualizovat svůj kód, ale pokud přehlídí kulturu tak zavedení DevOps hnutí jim nepřinese žádný úspěch.

4 Seznam literatury

Sharma, S. (2017) *The Devops adoption playbook: a guide to adopting devops in a multi-speed it enterprise*. Indianapolis, IN: John Wiley and Sons.

Davis, J. and Daniels, K. (2016) *Effective devOps: building a culture of collaboration, affinity, and tooling at scale*. First edition. Beijing ; Boston: O'Reilly.

International Conference on Advances in Computing, C.& A. et al. (2016) *2016 International Conference on Advances in Computing, Communication and Automation (Spring): ICACCA 2016 : proceedings : 8-9 Apr 2016*, Tula's Institute, Dehradun, India.

Erich, F., Amrit, C. and Daneva, M. (2014) *A Mapping Study on Cooperation between Information System Development and Operations*. In: A. Jedlitschka et al., (eds.) *Product-Focused Software Process Improvement*. Cham: Springer International Publishing, pp. 277–280.