

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS2019
Autoři - jméno, příjmení, xname	Tomáš Vrzák, vrzt01 Michal Mach, macm06
Téma	Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities

Abstrakt

Continuous Delivery a Deployment se soustředí na rychlejší dodávku produktových změn na produkci za předpokladu, že kvalita roste nebo se nesnižuje.

Cílem je rozebrat přiřazený článek a analyzovat průzkum, který je součástí článku Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities.

Metodou analýzy projdeme článek a vyhodnotíme kritické faktory z oblasti týmové struktury, spolupráce a odpovědnosti při zavádění CD.

Výsledkem jsou 4 týmové struktury, které se objevují při zavádění CD:

- Dev a Ops týmy odděleny
- Dev a Ops komunikace zajištěna prostředníkem
- Menší Ops týmy působící s větší odpovědností pro Dev tým
- Ops členové integrovány do Dev týmu

Obecně platí, že při zavádění CD roste potřeba rychleji komunikovat v rámci všech stakeholderů projektu. Nejčastější jevy jsou:

- Umístování týmů blízko u sebe
- Rychlý feedback v rámci životního cyklu
- Propojené úkoly, sdílená odpovědnost
- Nárůst používání kolaboračních nástrojů (pro komunikaci, dokumentaci)

Většinu členů týmu se po zavedení CD mění nebo jen částečně mění jejich odpovědnost a náplň práce. Nejčastěji změny odpovědností jsou:

- Rozšíření znalostí členů týmů
- Adaptace návrhu nových řešení v rámci CD

- Prioritizace úkolů

Klíčová slova

Continuous delivery, deployment, DevOps, kolaborace, týmová struktura

Obsah

Úvod	4
Vymezení základních pojmů	4
DevOps	4
Continuous Delivery	4
Continuous Integration	5
Continuous Deployment	5
Propojení pojmů Continuous Delivery a Continuous Integration	6
Dopad na týmovou strukturu	6
Dev a Ops týmy odděleny, zvýšená kolaborace	6
Integrace Ops a Dev týmů za pomoci prostředníka (DevOps tým).....	6
Menší Ops týmy s větší odpovědností pro Dev tým.....	7
Žádné Ops týmy (integrované s Dev týmy).....	7
Procentuální rozdělení respondentů do skupin	7
Dopad na spolupráci	8
Nástroje podporující spolupráci v týmu.....	8
Pohled na týmovou spolupráci.....	10
Umístění týmů Dev a Ops blízko sebe.....	10
Rychlý feedback	10
Propojené úkoly a sdílení odpovědnosti	10
Častější používání kolaboračního nástroje	10
Zvýšená transparentnost a informovanost	11
Větší míra zapojení Ops členů	11
Dopad na zodpovědnost	11
Rozšíření zkušeností	12
Přijímání nových řešení v rámci CD.....	12
Prioritizace úkolů.....	12
Závěr.....	12
Seznam zdrojů	13
Literární a internetové zdroje.....	13
Zdroje k obrázkům.....	13

Úvod

Continuous Delivery a DevOps obecně jsou vnímány jako budoucnost implementace softwaru ve společnostech – od startupů až po enterprise sféru. Management si od změny organizační struktury na DevOps a osvojení Continuous Delivery procesu slibují zefektivnění práce IT. Tedy snížení chybovosti, zvýšení kvality řešení a v neposlední řadě i snížení nákladů. Pokud bychom byli schopni naplnit všechny tři body, měli bychom téměř dokonalé IT oddělení, které by efektivně vyvíjelo za poměrně nízké náklady. V rámci této práce bychom se rádi zaměřili hlavně na překážky a benefity co se týkají týmové struktury, týmové spolupráce a zodpovědnosti jednotlivých členů týmu.

Cílem práce je nalezení odpovědi na tři otázky, a to, jak nejlépe zorganizovat tým, aby dokázal co nejefektivněji přijmout Continuous Delivery proces? Jak se změní spolupráce v týmu a napříč ostatními týmy? Změní se zodpovědnosti jednotlivých členů v týmu?

Na začátku si vydefinujeme, co je to vlastně DevOps a Continuous Delivery, včetně Continuous Integration, které je nedílnou součástí samotného Continuous Delivery.

V průběhu práce rozebereme 3 základní dopady, a to na týmovou strukturu, zodpovědnost a na spolupráci. Seznámíme se s jednotlivými možnými dopady, se kterými se setkáváme v různých dotázaných firmách v článku Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities (Mojtaba Shahin, Mansooreh Zahedi, Muhammad Ali Babar, Liming Zhu).

Závěrem zhodnotíme jednotlivé dopady a zodpovíme naše 3 kladené otázky.

Vymezení základních pojmů

DevOps

DevOps je výraz spojující Development (Dev) a Operations (Ops) s cílem zkrácení doby od zavázání se k product-inkrementu do doby nasazení tohoto inkrementu na produkci. Shlukuje v sobě postupy nepřetržitého dodávání a nasazování (Continuous Delivery – CD), automatizovaného testování a přijímání postupu Infrastructure as Code.

Continuous Delivery

Z překladu již vyplývá že jde o průběžné zásobování (continuous = průběžný, delivery = dodání). V našem článku a práci je význam tohoto spojení vnímán jako průběžné dodávání produkt-inkrementu na produkci. Jde o průběžnou činnost, která trvá stále. Standardně se jedná o neustálé dodávání změn produktu na různá prostředí společnosti, kde je daná změna produktu vyžadována. Tato cesta má jediný cíl, a to pravidelné a otestované dodávky změn produktu na produkci. Zajímá se především o to, co se děje po nasazení balíčku na produkci. Continuous Delivery je tak rozšířením k pojmu Continuous Integration.

Continuous Integration

Continuous Integration má podobný cíl jako Continuous Delivery (CD) s tím rozdílem, že se zabývá spíše tím, jak balíček změn na produkci dostat, resp. jak tento balíček změn dostávat přes jednotlivá prostředí společnosti a jak na tento balíček získat co nejrychlejší feedback.

Rád bych zde vymezil 2 důležité pojmy, které se objevují:

- **Prostředí (Environment)**
 - Jendá se o prostředí, kde se daný produkt (včetně jeho změn) nachází. Existují různé druhy prostředí, ve kterých se produkt může nacházet a má různé verze – většinou podle účelu zpracování. Příklady:
 - DEV – vývojové prostředí (produkt se zde vyvíjí)
 - TEST – testovací prostředí (produkt je zde obvykle testován)
 - PROD – zde se nachází verze produktu, která je používaná veřejností. Často také označováno jako „live“ prostředí.
- **Nasazení (Build)**
 - Jelikož je potřeba jednotlivé změny přenést z jednoho prostředí na druhé, je k tomu zapotřebí změny převést (např. z DEV → TEST). Tyto změny jsou často přenášeny automaticky nebo za pomoci nástrojů. Nasazení změn na cílové prostředí je označováno právě jako Build. Může být úspěšný nebo neúspěšný, podle toho, zda došlo k chybám či nějakým varováním.

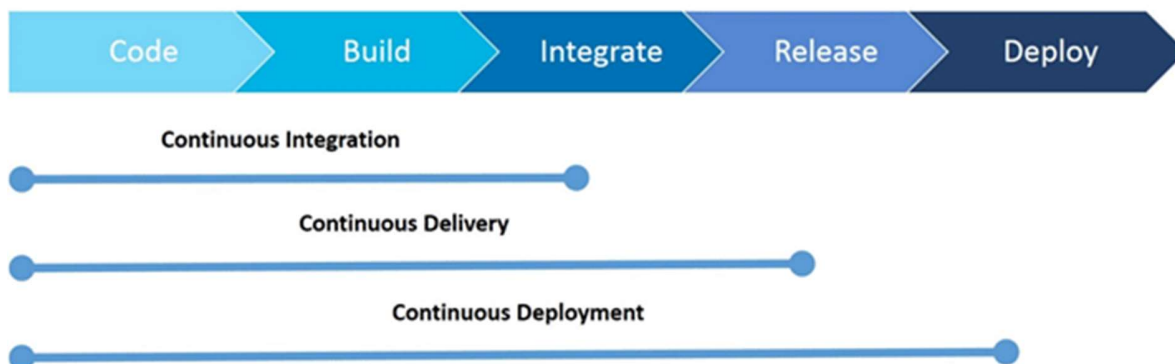
Continuous Deployment

Jelikož cílem CD je snižování doby nasazení změn produktu na produkci je zapotřebí mít také rychlý (automatický) deployment proces, který zajišťuje probublávání product-inkrementu z vývojového prostředí přes testovací, až po produkci. Z překladu vyplývá, že jde o průběžné nasazování (continuous = průběžný, deploy = nasadit). Continuous deployment je prakticky součástí CD, jelikož zajišťuje pravidelné nasazování nových změn např. k testování nebo na prostředí určené pro trénování uživatelů.

Propojení pojmů Continuous Delivery a Continuous Integration

Pro znázornění propojení těchto dvou pojmů jsme si vybrali následující zobrazení (Obrázek 1 – CI and CD pipeline):

Obrázek 1 - CI and CD pipeline



Saviant Consulting. *Continuous Integration Vs Continuous Delivery Vs Continuous Deployment*. [online]. [cit. 07. 05. 2019]. Dostupné z: <https://medium.com/@SaviantConsulting/continuous-integration-vs-continuous-delivery-vs-continuous-deployment-60e96e9a7a08>

CD je nadstavbou CI a obsahuje v sobě Continuous Deployment jako nedílnou součást nových změn produktu pro všechny prostředí. Po všech testech následuje finální nasazení na produkci (tzv. Going Live). (codeship, 2019)

Dopad na týmovou strukturu

V této sekci popíšeme základní skupiny členění vývojových a operačních týmů při implementaci a zavádění CD. Na závěr této sekce přidáme i procentuální zastoupení dotázaných firem, které reprezentují danou týmovou strukturu.

Dev a Ops týmy odděleny, zvýšená kolaborace

V některých případech ve společnosti nebylo nutné příliš zasahovat do týmových struktur nebo je nikterak měnit. Cílem v této struktuře je pouze zajistit, ať už za pomoci nástroje či jiných prostředků, úzkou komunikaci vývojového (Dev) a operačního (Ops) týmu.

Přesným příkladem této změny je například umístění týmu Ops blízko Dev týmu a tím se podpoří a urychlí nutná komunikace mezi těmito týmy.

Tato struktura byla detekována zejména ve velkých společnostech korporátního typu.

Integrace Ops a Dev týmů za pomoci prostředníka (DevOps tým)

Další variantou je vytvořením prostředníka, který komunikuje jak s Dev týmem, tak s týmem Ops. Tohoto prostředníka nazýváme facilitator (často také DevOps tým). Tento tým má za úkol

komunikovat s Dev členy týmu a zároveň také s členy Ops týmu. Mají také za úkol konsolidovat samotnou komunikaci a zajišťovat plynulý přechod informací Dev <-> Ops.

Menší Ops týmy s větší odpovědností pro Dev tým

Tato varianta obsahuje menší Ops týmy, které supportují členy Dev týmu, kteří jsou odpovědní za svůj kód na produkci. Členové Ops mají za úkol pomáhat developerům s operačními úkony a zajišťují to, aby kód, který vývojáři produkují, byl při celkovém buildu stabilnější a nezpůsoboval tolik problémů. Ops tým se podílí na úkonech v rámci Dev týmu, není však jeho součástí, spíše zajišťuje hladší průchodnost a stabilitu a jsou také často nasazováni při build-úkonech, jako je například nasazování na produkci.

Žádné Ops týmy (integrované s Dev týmy)

V pár případech se nacházelo, že členové Ops týmu byly přímo integrovány do Dev týmu. Všichni členové jednoho týmu měli sdílenou a rozšířenou odpovědnost nad operačními úkony jako jsou průběžné nasazování (continuous deployment), optimalizace aplikace na produkčním prostředí, ...

Tato struktura byla zjištěna u menších společností, rychle se rozvíjejících firem nebo startupů, kde tým prakticky obsahoval veškeré role: vývojář, business analytik, QA, Ops člen.

Procentuální rozdělení respondentů do skupin

Největším zastoupením je varianta, kde Ops a Dev týmy jsou odděleny, je pouze udržována vyšší úroveň komunikace (viz. Obrázek 2 - Survey results on patterns of organizing Dev and Ops teams for initiating and adopting CD practices). (Shahin a kol., 2017)

Obrázek 2 - Survey results on patterns of organizing Dev and Ops teams for initiating and adopting CD practices

Dev and Ops teams patterns	#	%
Separate Dev and Ops teams with higher collaboration	33	35.4
Separate Dev and Ops teams with facilitator(s) in middle	16	17.2
Small Ops team with more responsibilities for Dev team	27	29
No visible Ops team	16	17.2
Not Available	1	2.2

Mojtaba Shahin a spol (2017): *Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities*. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities

Dopad na spolupráci

V této kapitole se věnujeme dopadům na spolupráci v týmu implementujícím Continuous Delivery do organizace. Rozebereme si některé nástroje, které pomáhají při spolupráci v týmu. V další fázi rozebíráme výsledky ze zdrojového článku.

Nástroje podporující spolupráci v týmu

Slovo „spolupráce“ se opakovaně používá v DevOps, ale co to vlastně znamená v jeho kontextu?

Odborníci se domnívají, že kdykoli organizace oznámí, že bude implementovat DevOps. Tak první věc, která by měla přijít na mysl, tak je spolupráce.

Každá organizace by se na ni měla zaměřit již od zahájení projektu. Když Dev a Ops pracují společně, nejenže šetří čas, ale také pomáhají při rychlém hledání a opravě problémů. Pokud je nedostatek spolupráce, může se stát, že se problém Ops dotkne, a Ops by si mohl myslet, že je to problém Dev.

Díky správné spolupráci mohou týmy Dev and Ops, monitorováním a zaznamenáváním výkonu a co nejdříve řešit problémy, s ohledem na zákazníka.

Atlassian Jira

JIRA od Atlassian je skvělý software pro řízení projektů a zároveň může být také velmi silným nástrojem pro spolupráci, který lze použít v organizaci. Jedná se o softwarový nástroj, který byl vytvořen speciálně pro zachycení, přiřazení a upřednostnění úkolů pro zlepšení provádění projektu. Má jednoduché a intuitivní rozhraní, která pomáhají celému týmu.

JIRA je nejdůvěryhodnějším nástrojem, který se používá v oblastech spolupráce a také Project Management od společností Spotify, eBay, Cisco a LinkedIn. Může být jednoduše použit pro plánování, sledování a přiřazování úkolů pro lepší sprinty a dále pro lepší deployment. Jak již bylo řečeno, společnost JIRA také nabízí vizuální přehledy dat v reálném čase pro zvýšení výkonu týmu.

Atlassian Confluence

Confluence, cloudová služba od společnosti Atlassian je software pro obsahovou a týmovou spolupráci, který je novým způsobem, jak mají moderní týmy spolupracovat. Můžete vytvářet, sdílet a dokonce spolupracovat na projektech a dělat to vše na jednom místě - Confluence. Informace o společnosti můžete snadno publikovat a organizovat na centrálním místě. To může být dokonce využito ke zvýšení a také sdílení znalostí, které mají kolegové o procesech, na kterých pracují.

Obrázek 3 - Kolaborační nástroje používané v Dev-Ops



Ravindra Savaram (2019): *10 Tools For Effective DevOps Collaboration*. [online]. [cit. 07. 05. 2019]. Dostupné z: <https://mindmajix.com/10-tools-for-effective-devops-collaboration>

Bitbucket Server

Nabídka společnosti Atlassian ve formě Bitbucket Serveru je kombinací GIT serveru a webového rozhraní. Obvykle se jedná o webhostingové řešení, které je určeno pro projekty, které používají buď systémy řízení verzí Mercurial nebo GIT. Tento nástroj je speciálně vyvinut pro profesionální týmy, protože nejen uživatelům umožňuje vyvíjet, ale také spravovat a spolupracovat na projektech GIT.

Slack

Jedná se o kolaborační nástroj, který se využívá ve všech druzích organizací, ať už se jedná o startup nebo velkou nadnárodní společnost. Tomu napomáhají dva druhy subscripcí free verze a premium. Nástroj je velmi dobře integrovatelný do dalších systémů.

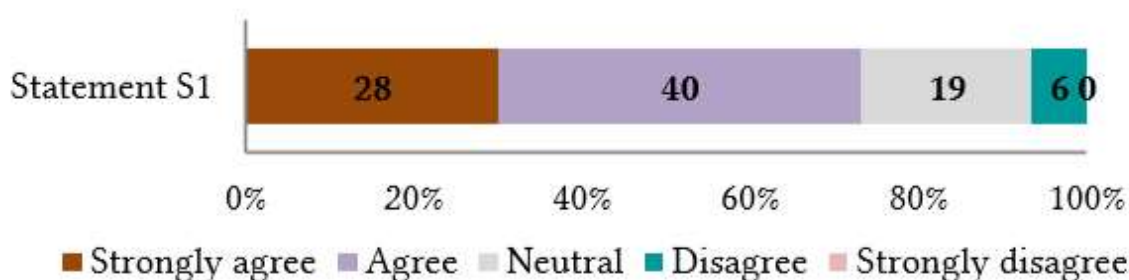
Microsoft Teams

Je to v podstatě pracovní prostor vybudovaný v Office 365 založený na chatu, který umožňuje kolegům z organizace udržovat kontakt. Týmy poskytují jedinečnou funkci poskytování soukromých, skupinových a kanálových rozhovorů mezi kolegy. Toto napomáhá k tomu, aby udržely krok s důvěrností, bezpečností informací. Členové projektového týmu nebo kolegové se mohou vrátit a podívat se také na tyto rozhovory, jak projekt pokračuje.

Pohled na týmovou spolupráci

Při nasazování a implementaci CD nedochází pouze ke změnám týmové struktury, ale také ke změně způsobu spolupráce mezi jednotlivými členy týmů. Většina dotázaných respondentů souhlasí s tím, že při aplikaci CD dochází k výrazně většímu nárůstu kooperace mezi jednotlivými členy týmů (Obrázek 4).

Obrázek 4 - The collaboration between team members has increased in my organization since the adoption of CD



Mojtaba Shahin a spol. (2017): *Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities*. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities

Umístění týmů Dev a Ops blízko sebe

Největší zastoupení při zvyšování spolupráce má strategie umístování týmů blízko sebe. Tím je urychlena a zefektivněna komunikace mezi jednotlivými členy. Většina komunikace probíhá především in-person (z očí do očí).

Rychlý feedback

Někteří z dotázaných uvádí, že dostávají feedback k aplikaci za kratší dobu při CD procesu a že to přináší kvalitnější software, pokud se členové týmu spojí a lépe spolu komunikují.

Propojené úkoly a sdílení odpovědnosti

S nasazováním DevOps a CD je vyžadována rychlost a častá frekvence. Těmito nároky také ale roste potřeba zahrnout do úkolu více členů týmu. Členové adoptují sdílenou odpovědnost a snaží se pracovat společně do maximální možné míry.

Několik respondentů průzkumu vysvětluje, že režie spolupráce byla v jejich organizacích snížena například začleněním testerů jako nedílné součásti Dev týmu místo toho, aby byli testeři pouze „hodnotiteli“.

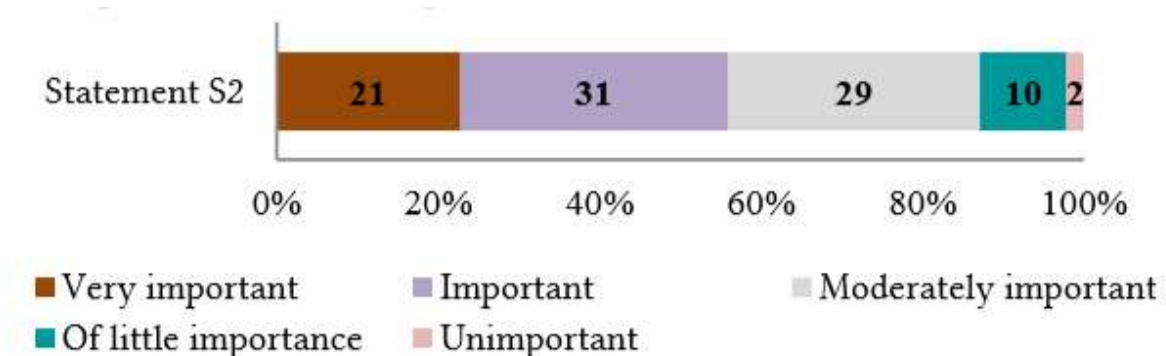
Častější používání kolaboračního nástroje

Při zavádění CD také dochází k nárůstu využívání kolaboračních nástrojů. Toto je důsledkem vyžadované častější komunikace a většina respondentů vnímá navýšené využívání kolaboračních nástrojů pro rychlou a operativní komunikaci (Slack, Skype for Business). Používáním kolaboračních nástrojů se ulehčuje komunikace a členům z různých týmů.

Zvýšená transparentnost a informovanost

S rostoucím tempem projektů při zavádění CD byl zjištěn také nedostatek na celkový přehled nad stavem projektu – build, release status, verze produktu na jednotlivých prostředích, dokumentace problému s deploy procesem. Respondenti byli dotázáni na důležitost mít dokumentaci k dispozici a čerpat z ní potřebné informace (Obrázek 5).

Obrázek 5 - How important is “lack of suitable awareness on status of project among team members” in adopting CD



Mojtaba Shahin a spol. (2017): Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities

Větší míra zapojení Ops členů

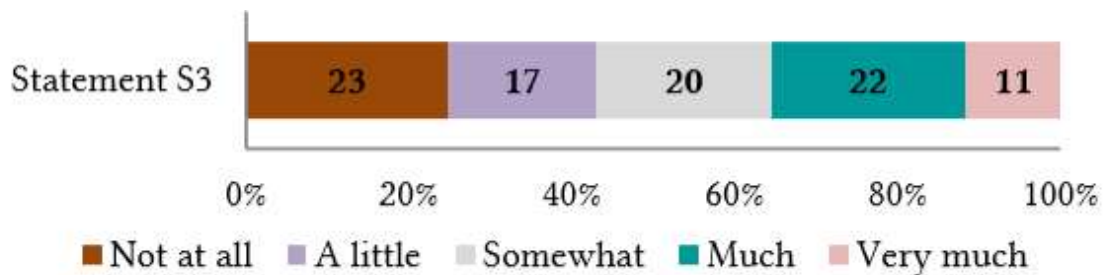
V rámci průzkumu bylo také zjištěno, že komunikace mezi Dev a Ops týmem je rapidně snížena, pokud je některý člen z Ops týmu zahrnut přímo do Dev týmu. To ulehčuje potřebnou komunikaci. Bylo také zjištěno, že se častěji Ops členové zapojují do celého životního cyklu a ne pouze pro fázi nasazování na produkci, což zlepšuje celkovou průchodnost a hladkost projektu. (Shahin a kol., 2017)

Dopad na zodpovědnost

Při adopci CD také často dochází ke změně v jedné důležité oblasti a tou je právě rozdělování zodpovědnosti pro některé členy týmů. Tím je myšleno, že změny v odpovědnosti se nemusí týkat pouze Dev týmu, ale také ostatních stakeholderů.

Zde je procentuální přehled respondentů, kteří uvedli, jak se jejich odpovědnost změnila (Obrázek 6).

Obrázek 6 - My responsibility has changed after our organization adopted CD practices



Mojtaba Shahin a spol. (rok): Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities

V rámci této kapitoly přiblížíme 3 skupiny odpovědností, které se pro dané respondenty změnilly.

Rozšíření zkušeností

Většina respondentů uvádí, že nejčastější změnou pro ně byla nutnost učit se novým věcem souvisejících se zaváděním CD (deployment nástroje). Také vývojáři uvádí, že i pro ně bylo nutností naučit se pracovat s některými operačními úkony, které před CD nemuseli řešit. Také přichází větší důraz na monitoringové dovednosti a rychlejší reakce na vady v produkčním prostředí. Navíc dochází k rozšíření v oblasti automatizovaných testů a schopnost je chápat a pracovat s nimi.

Přijímání nových řešení v rámci CD

Další velká změna přichází v oblasti architektury. Téměř naprostá většina architektů uvádí, že se jejich role změnila po zavedení CD. Nově architekti musí počítat s novými procesy souvisejícími se zavedením CD. Resp. jak navrhovat systém, tak aby zapadal do CD.

Dvěma největšími změnami jsou porozumění a používání microservices a návrh systému pro jiný deployment model, který přineslo CD.

Prioritizace úkolů

Zavádění CD přináší pro jednotlivé členy týmů potřebu zabývat se více hodnotnými úkoly a ty upřednostňovat před těmi méně důležitými. Jelikož se vývojáři také nemohou vracet ke starším verzím a vytvářejí stále novější verze, musí více přemýšlet nad tím, jak svůj kód a implementaci řeší a jak k daným úkolům přistupují. (Shahin a kol., 2017)

Závěr

Na závěr se poohlédneme zpět za kladenými otázkami a pokusíme se na ně odpovědět.

1. jak nejlépe zorganizovat tým, aby dokázal co nejefektivněji přijmout Continuous Delivery proces?

Jelikož větší firmy před zavedením CD preferovali mít týmy odděleně, uplatňují více se zaměřit na podporu komunikace mezi Dev a Ops týmy. Pro menší firmy je jednodušší naopak zahrnout členy Ops do Dev týmu, ať už úplně nebo pouze odpovědnostně.

2. Jak se změní spolupráce v týmu a napříč ostatními týmy?

Při zavádění CD z výzkumu vyplývá, že je vyžadována vyšší potřeba vzájemné komunikace napříč různými týmy, nejčastěji Dev a Ops. Tím pádem roste potřeba využívat častěji kolaborační nástroje pro komunikaci a dokumentaci. Řešením tohoto dopadu je i umisťovat jednotlivé týmy blízko u sebe, aby mohli jednoduše komunikovat mezi sebou osobně.

3. Změní se zodpovědnosti jednotlivých členů v týmu?

Se zavedením CD se většině členů týmu změní jejich zodpovědnost nebo náplň práce. Přichází změny jako pravidelný a automatický build. Vyvíjí se stále dopředu, a proto se musí více myslet na řešení úkolu. Roste znalostní potřeba v oblastech automatického testování a deploymentu.

Většina změn reflektuje samotný cíl CD a to zkracování doby nasazení nových změn na produkci. Rostou s tím náklady na častější komunikaci, lepší týmovou spolupráci a také znalostní růst jednotlivých členů týmu. Životní cyklus se stává agilním a schopnost rychle reagovat na změny je jedním z předpokladů, který by každý člen týmu měl zvládat.

Seznam zdrojů

Literární a internetové zdroje

1. Mojtaba Shahin a spol. (2017): *Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities*. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities
2. Codeship (2019), *CONTINUOUS INTEGRATION ESSENTIALS*. [online]. [cit. 07.05.2019] <https://codeship.com/continuous-integration-essentials>

Zdroje k obrázkům

1. Obrázek 1 - CI and CD pipeline. Saviant Consulting. Continuous Integration Vs Continuous Delivery Vs Continuous Deployment. [online]. [cit. 07. 05. 2019]. Dostupné z: <https://medium.com/@SaviantConsulting/continuous-integration-vs-continuous-delivery-vs-continuous-deployment-60e96e9a7a08>
2. Obrázek 2 – Survey results on patterns of organizing Dev and Ops teams for initiating and adopting CD practices. Mojtaba Shahin a spol (2017). *Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities*. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities

3. Obrázek 3 - Kolaborační nástroje používané v Dev-Ops. Ravindra Savaram. 10 Tools For Effective DevOps Collaboration. [online]. [cit. 07. 05. 2019]. Dostupné z: <https://mindmajix.com/10-tools-for-effective-devops-collaboration>
4. Obrázek 4 – The collaboration between team members has increased in my organization since the adoption of CD. Mojtaba Shahin a spol (2017). Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities
5. Obrázek 5 - How important is “lack of suitable awareness on status of project among team members” in adopting CD. Mojtaba Shahin a kol. (2017). Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities
6. Obrázek 6 - My responsibility has changed after our organization adopted CD practices. Mojtaba Shahin a spol (2017). Adopting Continuous Delivery and Deployment: Impacts on Team Structures, Collaboration and Responsibilities. [online]. [cit. 07. 05. 2019]. Dostupné z: https://www.researchgate.net/publication/315381999_Adopting_Continuous_Delivery_and_Deployment_Impacts_on_Team_Structures_Collaboration_and_Responsibilities