

riganti

 Update Conference
Prague 2019



Azure DevOps Deep Dive

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

tomas.herceg@riganti.cz

[@hercegtomas](https://twitter.com/hercegtomas)

www.tomasherceg.com/blog



Agenda

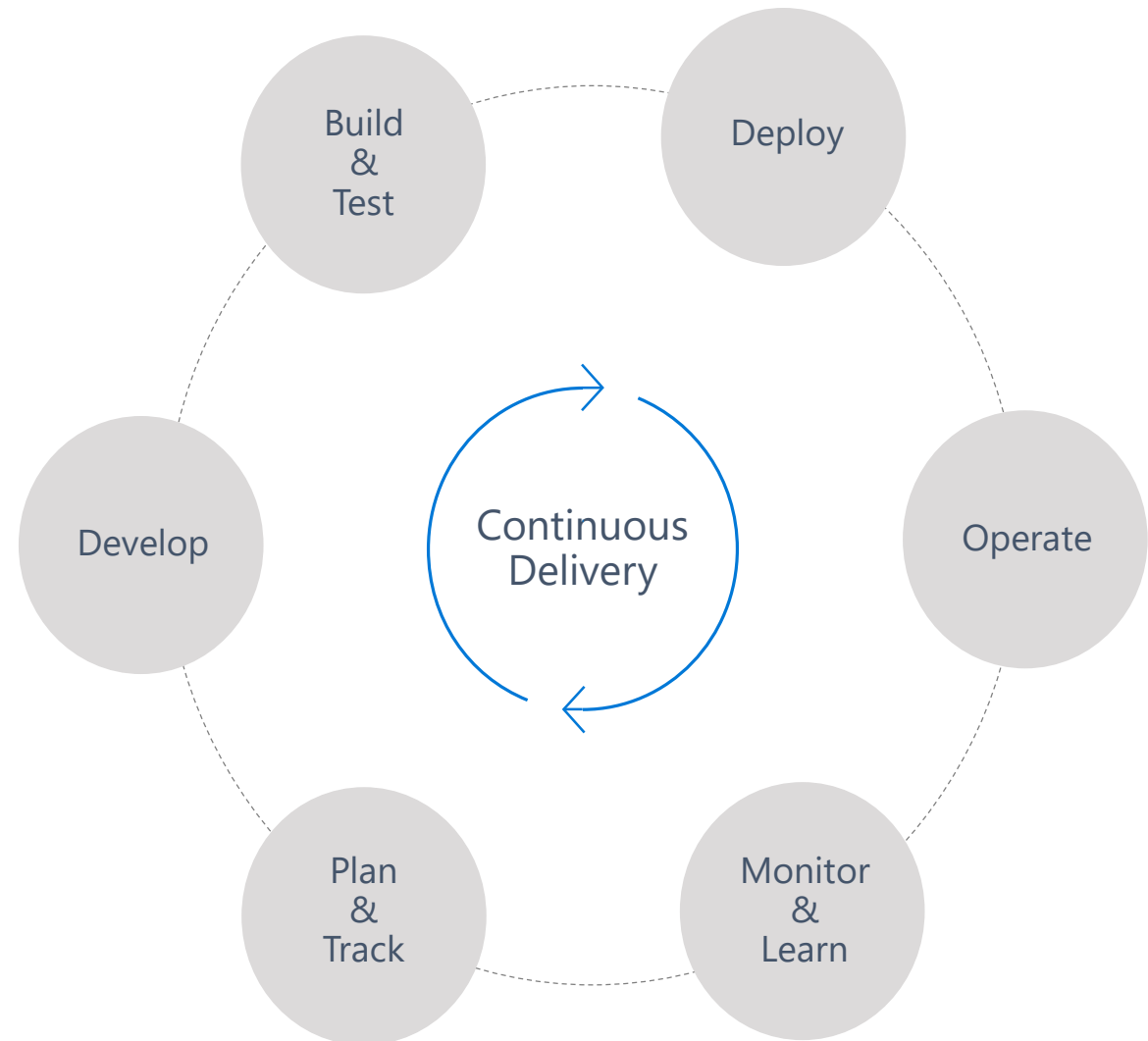
- **Úvod do Azure DevOps**
- Automatické buildy
 - .NET Framework
 - .NET Core
- Release management
 - Nasazování na vlastní IIS
 - Azure App Service
 - Desktopové aplikace



Co je DevOps?

- Lidé
- Procesy
- Nástroje

- „Continuous Delivery“
- Zkrácení času k uvedení nové funkcionality





K čemu DevOps?

- Lepší feedback od uživatelů a z produkčního prostředí
- Boří se hranice mezi vývojáři a IT správci
- Snížení chybovosti release
- "Dokumentace release procesu"
- Víte, co se s aplikací děje (monitoring)



Součásti Azure DevOps



Azure Boards



Azure Pipelines



Azure Repos



Azure Test Plans



Azure Artifacts



Azure Pipelines

- Kontinuální integrace a nasazení
- Podpora vlastních i cloudových build agentů
- Build navázaný na každý commit
- Release management s podporou více prostředí



Azure Boards

- Kanban boardy, backlog, týmové dashboard
- Podpora různých metodologií
 - Agile
 - Scrum
 - ...
- Možnosti customizace



Azure Repos

- Správa verzí zdrojového kódu
 - Git
 - TFVC (Team Foundation Server)
- Pokročilé nástroje pro pull requesty a integrace



Azure Test Plans

- Definice testovacích plánů a scénářů
- Spouštění testů v různých prostředích
- Podpora desktopových i webových aplikací



Azure Artifacts

- Správa interních balíčků pro vývojáře
 - NuGet
 - Npm
 - Maven
- Snadné sdílení kódu napříč projekty



Jak začít?

- Nemusíte vzít hned vše
- Máte Git hostovaný na vlastním serveru?
 - Nevadí!
- Používáte pro task management Jiru?
 - Nevadí!
- Buildujete přes TeamCity?
 - Nevadí!



Kolik to stojí?

- Azure DevOps
 - Prvních 5 uživatelů zdarma
 - Vlastníci Visual Studio Subscription mají licenci už ceně
 - Basic plán
 - 6\$ / měsíc



Kolik to stojí?

- Build agenti
 - Hostování
 - 1800 minut zdarma (max. 1 paralelní úloha)
 - Každá další paralelní úloha (neomezený počet minut): 40\$ / měsíc
 - Vlastní infrastruktura
 - 1 paralelní úloha zdarma
 - + 1 další za každé Visual Studio Subscription
 - Každá další stojí 15\$ / měsíc



Kolik to stojí?

- Další možnosti
 - Stakeholder access
 - Zdarma, ale jen omezený přístup do Azure Boards
 - Pro zákazníky nebo „neprogramátory“
 - Plán Basic + Test
 - Zahrnuje testovací nástroje
 - 52\$ / měsíc

Jak Azure DevOps používáme v RIGANTI



- Azure DevOps pro buildy a releases
 - Každý projekt má interní *dev* prostředí
 - Každý projekt má automatické buildy a releases
 - Většinou releasujeme i na *staging* a do *produkce*
- Zdrojové kódy na vlastním GitLab serveru
 - Postupně přecházíme do Azure DevOps
 - Lepší UI pro pull requesty
 - Lepší napojení na work items



Jak Azure DevOps používáme v RIGANTI

- NuGet balíčky publikujeme na vlastní NuGet server
 - Licenční politika Artifacts v Azure DevOps se nám donedávna nelíbila
 - Dříve byla potřeba další licence
 - Nyní je to už v ceně Basic plánu nebo VS Subscription a platí se za obsazené GB
- Work Items spravujeme různě
 - Pokud si můžeme vybrat, tak v Azure DevOps
 - Často musíme používat nástroj vybraný zákazníkem (Jira, Trello...)



Co nám přinesly automatické buildy?

- Donutilo nás to mít v projektech pořádek
 - Clone a build na první dobrou
 - Hodně v tom pomohl i .NET Core
- Víc prostředí není problém
 - Počítá se s nimi by design
- Máme vždy aktuální dokumentaci k build a release procesu
 - Stačí se podívat do Azure DevOps, co je třeba pro nasazení



Co nám přinesly automatické buildy?

- Nasazujeme často a s méně chybami
 - Nebojíme se toho, protože se deployment stokrát předtím povedl
 - Projekty, které se nasazují často, jsou obecně velmi spolehlivé
- Víme přesně, co je kde nasazeno
- Lepší zastupitelnost
 - Nasadím kolegův projekt, i když je zrovna na dovolené na Bali



Jak pracné to je?

- Počítejte s klasickými porodními bolístkami
 - Čekání na výsledek buildu je otravné
- Na jednoduché ASP.NET Core projekty je hezká template
 - Není potřeba skoro nic měnit
- Čas strávený automatizací se nám mnohonásobně vrátil



Agenda

- Úvod do Azure DevOps
- **Automatické buildy**
 - **.NET Framework**
 - **.NET Core**
- Release management
 - Nasazování na vlastní IIS
 - Azure App Service
 - Desktopové aplikace



Typická build pipeline

- Stáhnout zdrojové kódy
- NuGet restore
 - Balíčky by neměly být commitnuté v source controlu
- Kompilace
- Spuštění testů
- Publish
 - Ideálně balíček pro nasazení nebo k instalaci
- Výsledek buildu = Build Artifacts



Dva typy build pipelines

- Klasické pipelines
 - UI pro definici jednotlivých kroků
 - Snadná „discoverability“
 - Verzování si musíte řešit sami
- YAML pipelines
 - Buildy se definují v YAMLu a commitují do source control
 - Výhoda
 - verzují se s projektem
 - fungují tam větve
 - UI není tak dokonalé a přívětivé



DEMO

Build .NET Frameworkové aplikace



.NET Framework

- Tasky v klasické pipeline
 - NuGet Installer
 - Visual Studio Build
 - Visual Studio Test
- Publish webové aplikace
 - Nejjednodušší je WebDeploy do adresáře ve filesystemu
 - Parametry pro MSBuild
 - `/p:DeployOnBuild=true /p:WebPublishMethod=FileSystem /p:publishUrl="$(Build.ArtifactStagingDirectory)" /p:DeployDefaultTarget=WebPublish`



DEMO

Build .NET Core aplikace



.NET Core

- Tasky
 - dotnet restore
 - dotnet build
 - dotnet test
 - dotnet publish
- Pro přesnou verzi .NET Core existuje task *Use .NET Core*



Entity Framework Core Migrace

- Nainstalovat .NET Tool pro Entity Framework Core
 - `dotnet new tool-manifest`
 - `dotnet tool install dotnet-ef`
 - Vytvoří se manifest, který commitněte do source controlu
- V rámci buildu udělat restore toolu
 - `dotnet tool restore`



Entity Framework Core Migrate

- Lze vyexportovat všechny migrace
 - Idempotentní – skript lze spustit opakovaně
 - Jsou tam ještě nějaké buggy u vlastních SQL skriptů v migracích, ale lze to použít
 - `dotnet ef migrations script`
 - `--project MigrationsProjectName`
 - `--context DbContextName`
 - `--startup-project StartupProjectName`
 - `--output $(Build.ArtifactStagingDirectory)/migrations.sql`
 - `--idempotent`



DEMO

Build desktopových aplikací s MSIX balíčkem



Desktopové aplikace

- V případě MSIX balíčků je třeba mít Windows 10 SDK
- Tasky
 - Powershell pro úpravu package manifestu
 - Jednoduchá XML transformace
 - Visual Studio Build
- Publish balíčku přes MSBuild
 - `/p:AppInstallerUri=$(MsixInstallUrl)/preview`
`/p:UapAppxPackageBuildMode=SideLoadOnly`
`/p:GenerateAppInstallerFile=true`



Jak je to s build agenty

- Hostování build agenti
 - Připravené prostředí pro kompilace všeho možného
 - .NET, Java, PHP...
 - <https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/hosted?view=azure-devops>
 - Pro Xamarin vývojáře mohou být zajímaví Mac OS X agenti
- Nevýhody
 - Jsou pomalejší
 - Nemají nacheované obvyklé NuGet balíčky a věci, se kterými pracujete



Jak je to s build agenty

- Vlastní build agenti
 - Fyzický stroj nebo virtuální mašina
 - Windows / Windows Server
 - Nejjednodušší je nainstalovat celé Visual Studio
 - Nainstalovat Azure DevOps Build Agenta
 - Linux, Mac OS X
 - Nainstalovat .NET Core
 - Nainstalovat Azure DevOps Build Agenta
 - Lze i jako Docker kontejner



Instalace Azure DevOps Build Agenta

- Organization Settings > Agent Pools
- Vytvořit nový pool
 - Podle projektu nebo účelu
- Přidat agenty
 - Může jich být více
 - Wizard s návodem pro instalaci
- Je třeba v nastavení účtu vytvořit Personal Access Token
 - Agent bude mít stejná práva jako vy



Agenda

- Úvod do Azure DevOps
- Automatické buildy
 - .NET Framework
 - .NET Core
- **Release management**
 - **Nasazování na vlastní IIS**
 - **Azure App Service**
 - **Desktopové aplikace**



Jak fungují releases

- Na vstupu je Build Artifact
 - Soubory / balíčky k nasazení
- Release může mít více prostředí
 - Dev, Staging, Production
 - Různá konfigurace
 - Secrets se dosazují až při releasu
 - Lze konfigurovat vstupní podmínky
 - Do některých se dá nasazovat automaticky
 - Některá mohou vyžadovat schválení



On-premises IIS

- Release agent
 - Nainstaluje se na server s IIS
 - Když dostane pokyn od DevOps, stáhne balíček a nasadí jej na IIS
 - Běží pod konkrétním uživatelským účtem
 - Má tedy práva jen tam, kam ho pustíte
- Agent funguje v roli klienta
 - Drží si HTTPS spojení s Azure DevOps
 - Není třeba otevírat mu vstupní port



DEMO

Nasazení na on-premises IIS



On-premises IIS

- Tasky
 - IIS Web App Manage
 - Umožňuje vytvořit web site a aplikační pool
 - Umožňuje nastavit vše potřebné
 - Režim „Create or Update“
 - IIS Web App Deploy
 - Nakopíruje soubory do složky webové aplikace
 - Umí provést nahrazení hodnot
 - XML / JSON variable substitution
 - Web Config Transforms (nepraktické, nechcete mít secrets v gitu)



DEMO

Nasazení do Azure App Service



Azure App Service

- Nepotřebuje release agenta v Azure
- Lze provést z normálního build agenta

- Tasky
 - Azure App Service Deploy
 - Provede XML / JSON nahrazení
 - Alternativně můžete secrets nastavit v Azure portálu
 - Nakopíruje soubory přes Web Deploy
 - Pozor na režimy ZipDeploy a Run From Package (aplikace nemůže psát do filesystemu)



DEMO

Nasazení desktopové aplikace s MSIX balíčkem



Desktopové aplikace s MSIX balíčky

- Balíček je třeba podepsat
 - Pro každé prostředí může být jiný certifikát
- Tasky
 - Code Signing
 - Je třeba nainstalovat z Marketplace
 - Pracuje s konceptem „Secure Files“
 - Soubor se stáhne jen pro potřeby daného buildu / releasu
 - Výsledný balíček lze vy publikovat na UNC share, do IIS nebo do Azure
 - Stejně jako u předchozích metod



Složitější scénáře s DotVVM

- V rámci vývoje frameworku potřebujeme
 - Spustit unit a integrační testy
 - Zkompilovat framework
 - Nasadit několik sample aplikací (s DLL z předchozího kroku) do IIS
 - Spustit UI testy
- V rámci releasů potřebujeme
 - Vypublikovat interní NuGet balíčky na náš privátní NuGet feed
 - Vypublikovat veřejné NuGet balíčky na oficiální NuGet
 - Vypublikovat licencované NuGet balíčky na speciální DotVVM feed



Co využíváme

- Task Groups
 - Opakující se skupiny tasků
- Vlastní PowerShell pro publish
 - Sestavení balíčků a jejich publikování
 - Úprava verze assembly ve zdrojácích a commit do gitu
 - Vytvoření tagu s publikovanou verzí v gitu
- Kouzlení se Selenium testy
 - Killování browserů a Selenium driverů
 - Nasazování sample aplikací do IIS (msbuild a publish profily)



Propojení s GitHubem

- Projekt je vyvíjen na GitHubu
 - Zdrojové kódy jsou tam
- Pěkné propojení s pull requesty
 - Testy se spouští automaticky
 - Výsledky se propisují zpět na GitHub



U DotVVM 3.0 přejdeme na YAML pipelines

- Build / release proces se v průběhu času mění
- Aktivně se vyvíjí v několika větvích
- Úpravy pipeline pro potřeby jedné větve rozbijí větve ostatní

- YAML umožňuje mít definici commitnutou v konkrétní větvi
- Při pull requestu se dá udělat merge



DEMO

Jak buildujeme a releasujeme DotVVM



Nechete přijít o další akce?

- Sledujte Twitter nebo Facebook
 - https://twitter.com/update_conf
 - <https://facebook.com/updateconference>
- *Update Conference Prague*
 - **12. – 13. listopadu 2020**
 - Vstupenky začneme prodávat v červnu
- A samozřejmě WUG

riganti

 Update Conference
Prague 2019



Q&A

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP, Microsoft RD

tomas.herceg@riganti.cz

[@hercegtomas](https://twitter.com/hercegtomas)

Microsoft takes .NET
Open Source

November 12th 2014