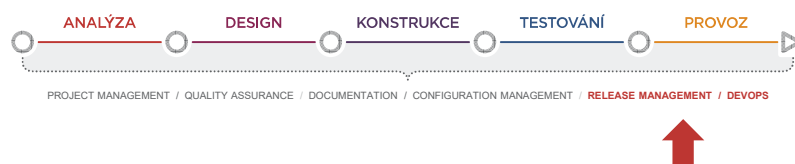




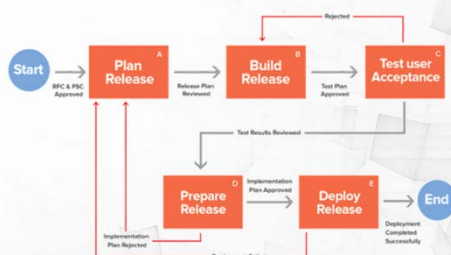
Téma dnešní přednášky

1. Release management
2. Continuous integration / delivery / deployment / DevOps
3. Ukázky z praxe
4. Diskuze



PROFINIT

3



Release management

Motivace

Než se váš systém/změna dostane do produkce:

- › musí ho někdo vyvíjet a testovat,
- › musí být *někde* před nasazením do produkce akceptován
- › může existovat více produkčních prostředí
- › ...

Typicky existují další prostředí mimo cílové !



PROFINIT

5

Sladění terminologie ČS

- › V ČS je pojem Release managementu chápán mírně odlišně vzhledem k dnešní prezentaci
- › *Oddělení RM má zajistit, že se stovky systémů dostanou do produkce včas, jsou integrovány, ...*



- › Aby mohl fungovat celek, musí fungovat i každá dílčí část
- › → o tom bude dnešní přednáška

PROFINIT

6

Motivace

Co musíme umět pro každý systém, než se dostane do produkce:

- › **Vyrobít** dodávku
- › Připravit dodávku pro **instalaci zadavatelem**
- › **Nainstalovat** dodávku

- › Dodat **systém jako celek**
- › **Opravit** malou drobnost
...a opravit ji **rychle a ekonomicky** ...
- › Poradit si s **různými typy prostředí**
 - Aplikační server, databázový a replikační server, operační systém, ...

- › Release **není** „jen o nasazení nové verze“

PROFINIT

7

Motivace

Prostředí mohou být různého typu (různé pohledy):

- › Typicky virtualizovaná



- › Nově „kontejnerizovaná“



- › On-premise x Cloud
(SaaS, PaaS, ...)



- › Jednotky prostředí x tisíce



PROFINIT

8

Motivace

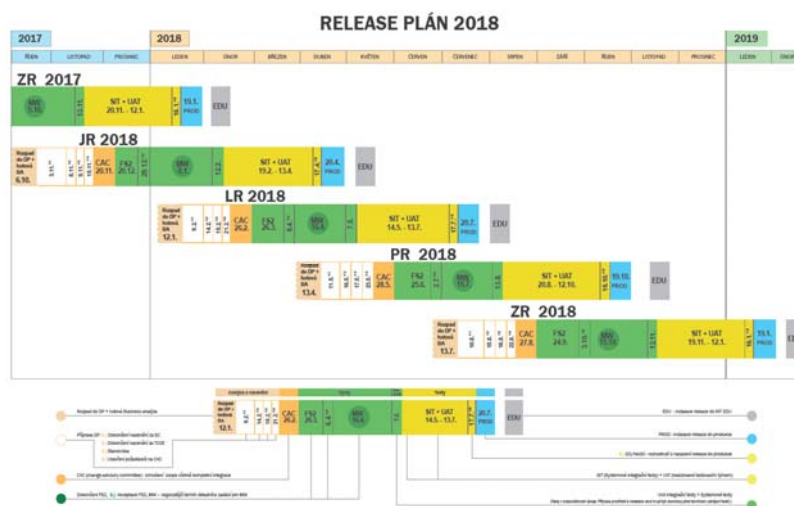
- › Integrovaní se aplikace a jejich závislosti
 - Budou další systémy připraveny?
 - Bude možná integrace?
 - ...
- › Rozdílné vývojové cykly aplikací
 - Vodopád x agile
 - Front-end x back-end
 - Mobilní aplikace x web aplikace x „tlustý klient“
 - ...
- › ... zpoždění v rozsahu dodávek
- › Něco jiného se testuje, něco jiného je na produkci
 - Potřebuji testovat opravu produkce pro verzi X, ale na akceptačním prostředí mám nyní verzi X+1...

JE POTŘEBA ŘÁD A PLÁN

PROFINIT

9

Typické řešení



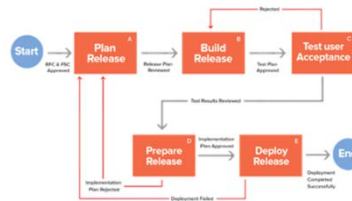
PROFINIT

10

Typické řešení

Je nutné postihnout minimálně:

- › Celkový proces
 - Odpovědnosti, styčné osoby
 - Komunikační matice
 - Formální náležitosti
 - ...
- › Klíčové milníky
- › Dopady na okolní systémy
- › Prerekvizity
- › Kvalitativní nároky
- › ... a to vše pro všechny systémy se zanesením jejich vazeb, ...

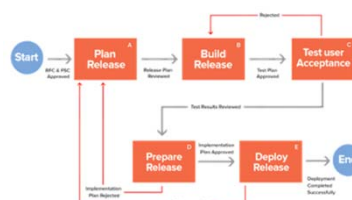


PROFINIT

11

Klíčové pojmy

- › Release (Build)
- › Oprava buildu (patch)
- › Instalační set
 - En-bloc
 - Inkrementální
- › ... slyšeli jste již o multi-speed / Bi-modal IT?



Bimodal IT

vs.



Zdroj: <http://www.devops.com>

PROFINIT

12



Continuous integration / delivery / deployment / DevOps

O čem se vlastně bavíme?

- › Vytvářený software má typicky **tisíce/miliony** řádků kódu
- › Všechny artefakty musí někdo:
tzv. sestavit (build) → otestovat → nasadit → znovu otestovat ...

- › Lze toto realizovat ručně?

Ano → ALE...

- › Člověk se může „uklepnout“ (...a pak je průšvih...)
- › Ruční práce se nemůže měřit s výhodami automatizace
 - Rychlost zpětné vazby
 - Opakovatelnost
 - Bezpečnost



PROFINIT

14

O čem se vlastně bavíme?

- › Komplexita systémů a technologií vede na **nutnost automatizace**
→ jsme jen lidé a děláme chyby
- › Vše, co lze automatizovat, by mělo být automatizováno
- › Některé technologie již ani nepředpokládají ruční zásah
- › Co lze automatizovat?
 - Generování kódu
 - Generování dokumentace
 - Testy
 - Sestavení aplikace
 - Nasazení aplikace
 - ... **téměř vše** ...



Co je DevOps?

- › **Development & Operations**

By **Daniel Johnson**, FORMULA ONE CORRESPONDENT

19 JUNE 2016 • 6:26PM

After 23 hours and 55 minutes, having covered more than 3,200 miles, Anthony Davidson's dream of winning the Le Mans 24 Hours for the first time seemed certain to finally come true. But in the cruellest of denouements, his Toyota broke down on the final lap, losing him the lead and giving victory to Porsche with just three minutes to go.

Davidson and the other two drivers in his car, Sébastien Buemi and Kazuki Nakajima, were having to come to terms last night with one of the most heartbreaking defeats in the history of the famous, gruelling race.

Co je DevOps?

› Development & Operations



PROFINIT

17

Co je DevOps?

› Development & Operations

Často pracujeme například na analýze/vývoji a ani netušíme,
kdo bude naše výstupy nasazovat do produkce

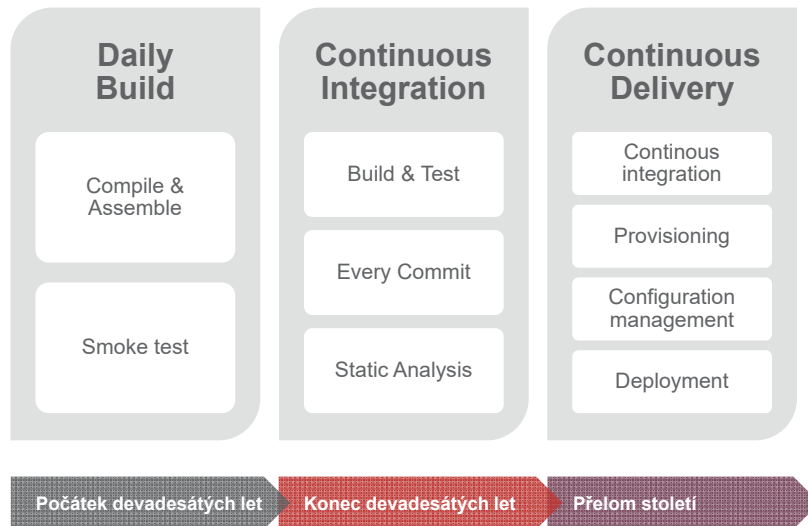
Představte si ale, že jste najednou součástí jednoho týmu,
který pracuje dohromady za jediným cílem → funkční software



PROFINIT

18

Co je DevOps?



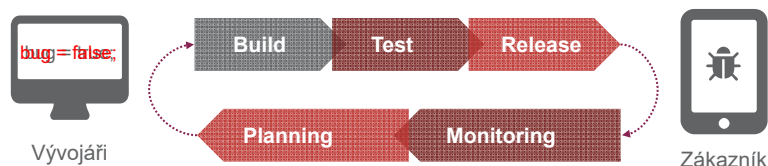
PROFINIT

19

Co je DevOps?

› Forma vývojového cyklu

- Každý krok je maximálně automatizován
- Vše je verzováno a testováno (nejen kód, ale i model databáze, data, ...)
- Na všechna prostředí se používá jeden unifikovaný proces
- Celý cyklus řešen formou malých kroků → Deployment Pipeline
- Rychlá a maximální zpětná vazba
- → podporuje Agilní vývoj (Agile bez DevOps lze jen obtížně realizovat)



› Je nutná odpovídající kultura ve vývojovém týmu

Inspirováno <http://www.slideshare.net/AmazonWebServices/dvo202-devops-at-amazon-a-look-at-our-tools-processes>

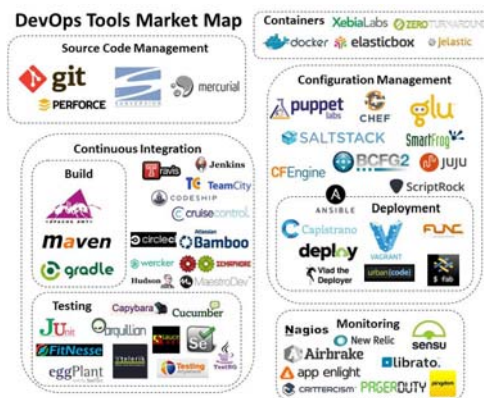
PROFINIT

20

Co je DevOps?

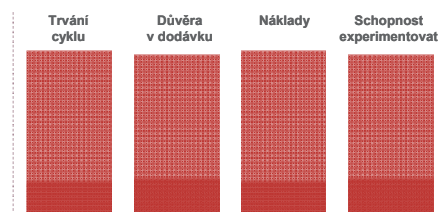
› Efektivní využití nástrojů

- Version Control
- Provisioning
- Configuration Management
- Build Automation
- Artifact Repository
- Static Analysis
- Automated Testing
- Test Data
- Continuous Integration and Delivery
- System Monitoring & Analytics
- ... a **vůle zlepšovat**



Hlavní přínosy

- › Snížení TTM business požadavků
- › Snížení množství chyb se současným zvýšením rychlosti jejich oprav
- › Snížení nákladů na zdroje (development i operations)
- › Jednodušší zapojení nových lidí do týmu
 - silná zpětná vazba
 - tzv. Self-service
- › Možnost „bezpečného experimentování“



PROFINIT



DevOps a architektura

Velmi často pracujeme s tímto: ...A good old monolith

PROFINIT

24

A slide with a white header containing the title 'DevOps a architektura' in bold red text. Below the header, there are two text boxes: a red one on the left saying 'Velmi často pracujeme s tímto:' and a dark gray one on the right saying '...A good old monolith'. Below these is a large image of a black monolith on a white pedestal in a room filled with stacks of papers. On the right side of the slide, there is a vertical red bar with the word 'PROFINIT' in white, and the number '24' at the bottom.

DevOps a architektura

Raději bychom možná ale pracovali s tímto:

... brand new
cool Microservices

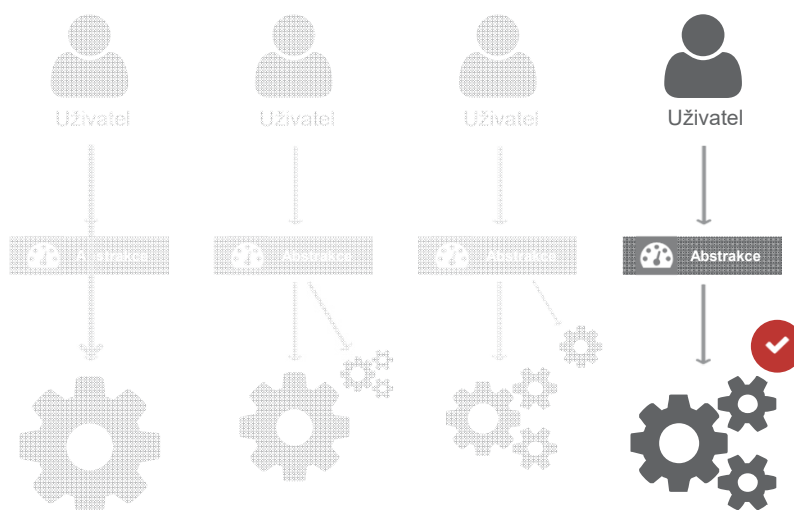


PROFINIT

25

DevOps a architektura

Ve spojení s DevOps lze aplikovat tzv. „Strangler Pattern“

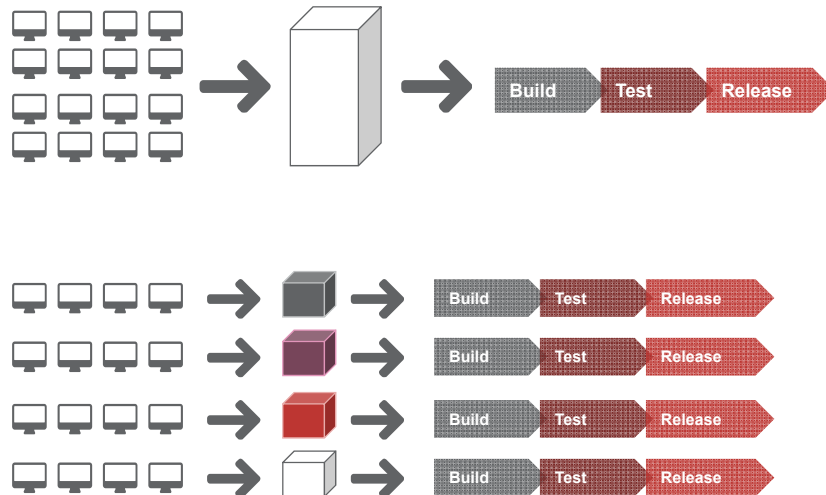


PROFINIT

26

DevOps a architektura

Změna architektury jde ruku v ruce s release cyklem



PROFINIT

27

DevOps a architektura

- › Stejně jako u změny architektury lze DevOps zavádět postupně
- › Postupovat lze „z obou stran“
 - Ze strany vývoje například ve formě automatického buildu, unit testů, ...
 - Ze strany automatizace nasazení na prostředí, apod.
- › Je vhodné zvolit / identifikovat oblasti, ve kterých změna nejvíce prospěje
 - Předpokládá sběr a vyhodnocování správných metrik, například zdroje chyb, důvody odstávek → monitoring

PROFINIT

28

DevOps a architektura



PROFINIT

29

Ideální podoba

Evoluce automatizace



Agilní vývoj

Continuous Integration

Continuous Delivery

Continuous Deployment

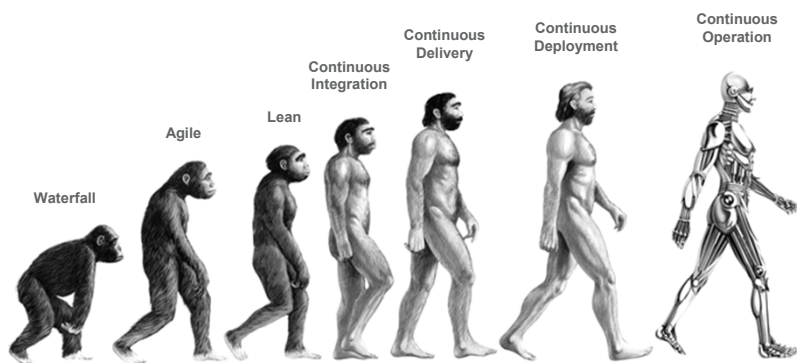
DevOps

PROFINIT

Evoluce automatizace

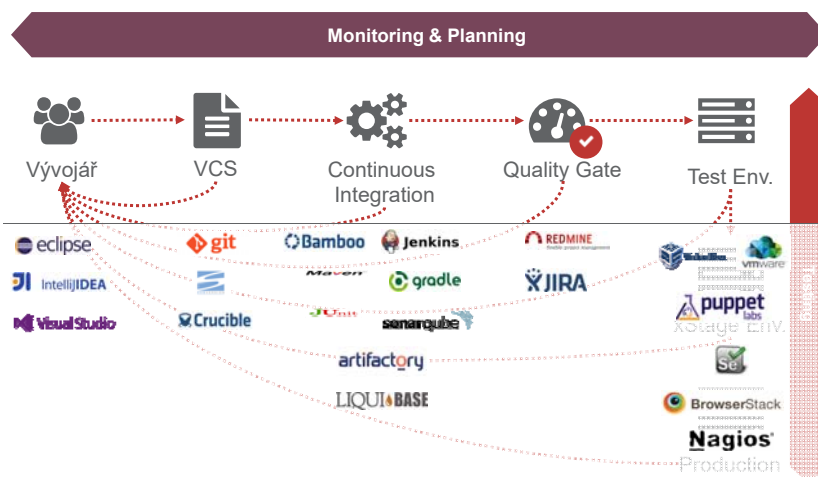


DevOps Movement



PROFINIT

Ideální podoba cesty jednoho řádku kódu



PROFINIT

Bezodstávkové nasazování

- › ... všichni známe odstávky systému vzhledem k nutnosti upgrade
- › ... a s tím spojené nervy a občas probdělé noci

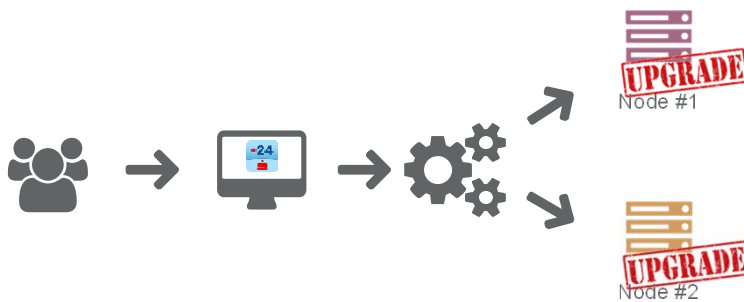


PROFINIT

34

Bezodstávkové nasazování

- › Princip je vlastně jednoduchý....



- › ... nezbytnou prerekvizitou je však vespělá automatizace
→ tzv. „**ONE CLICK DEPLOYMENT**“

PROFINIT

35



Ukázky z praxe

CIC, technologie

- Internetové bankovníctví pro 1,8 mil. uživatelů, 5M+ transakcí měsíčně, 400+ business procesů, 1000+ obrazovek



- › Integrace na všechny klíčové systémy v bance
- › Více než 4,5 mil. řádků kódu v Javě, SQL, PL-SQL, Šablony xhtml, Android aplikace, iOS aplikace (Swift, Objective C)



PROFINIT

37

CIC - nástroje

[illegible]

PROFINIT

38

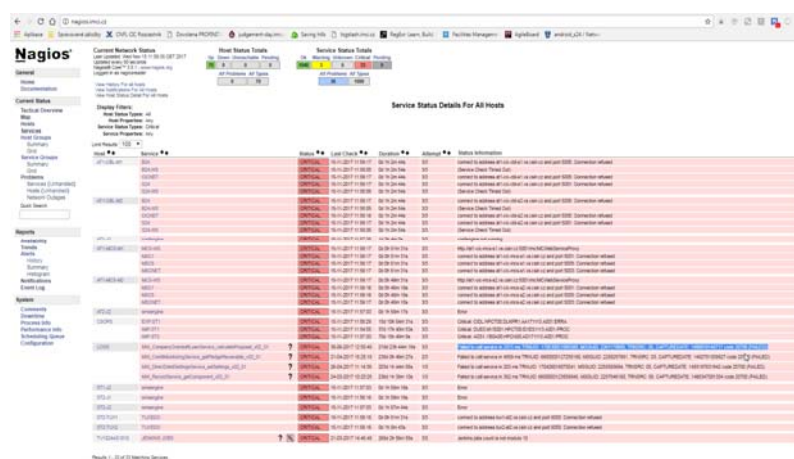
CIC - nástroje



PROFINIT

39

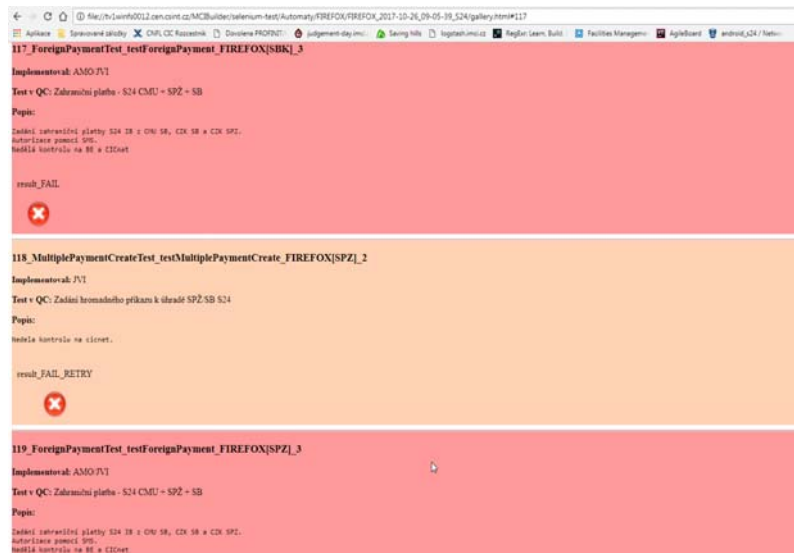
CIC - nástroje



PROFINIT

40

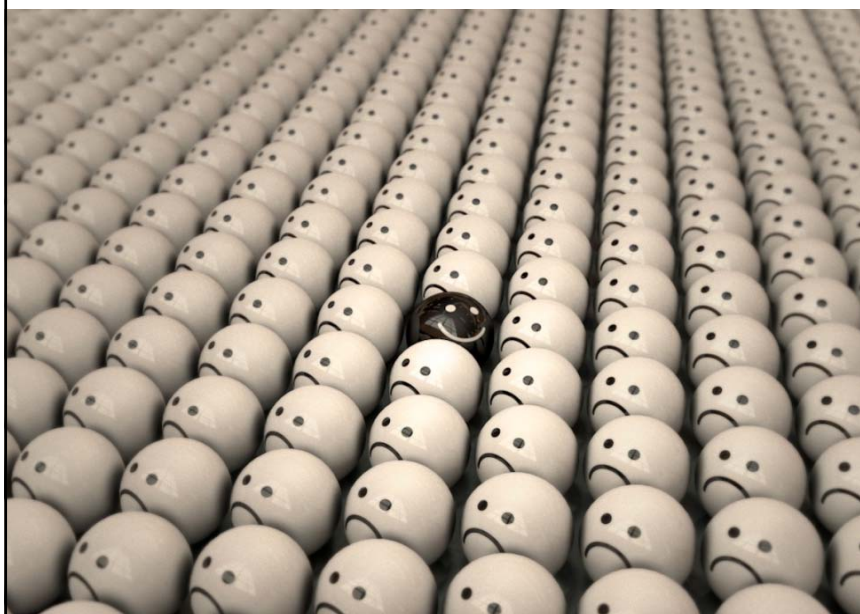
CIC - nástroje



Poznatky z praxe

- › Maximálně **věrné prostředí** vývojové, testovací, akceptační, ...
- › Denní build
- › Proces dodávek
 - jednoduchý
 - **automatizovaný**
- › Kontrolované
 - zálohovací logy
 - reporty automatických testů
- › Když není k dispozici hotové řešení → vlastní microskripty/pluginy
- › Nic ale není černobílé
- › Velmi těžko dosažitelné bez „týmového nadšení“

Poznatky z praxe



PROFINIT

43



Diskuze

PROFINIT

44

**Děkujeme
za pozornost**

PROFINIT
NÁSKOK DÍKY ZNALOSTEM

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6

 Telefon
+ 420 224 316 016

 Web
www.profinit.eu

 LinkedIn
linkedin.com/company/profinit

 Twitter
twitter.com/Profinit_EU

O čem se vlastně bavíme?

Přestanou-li existovat sdílené
disky, repository, PC lidí na projektu,
vývojové prostředí, pak

...

**ze záloh je možno postavit
vývojové prostředí**

...

a to ekonomicky

PROFINIT

46