Automate all the things!

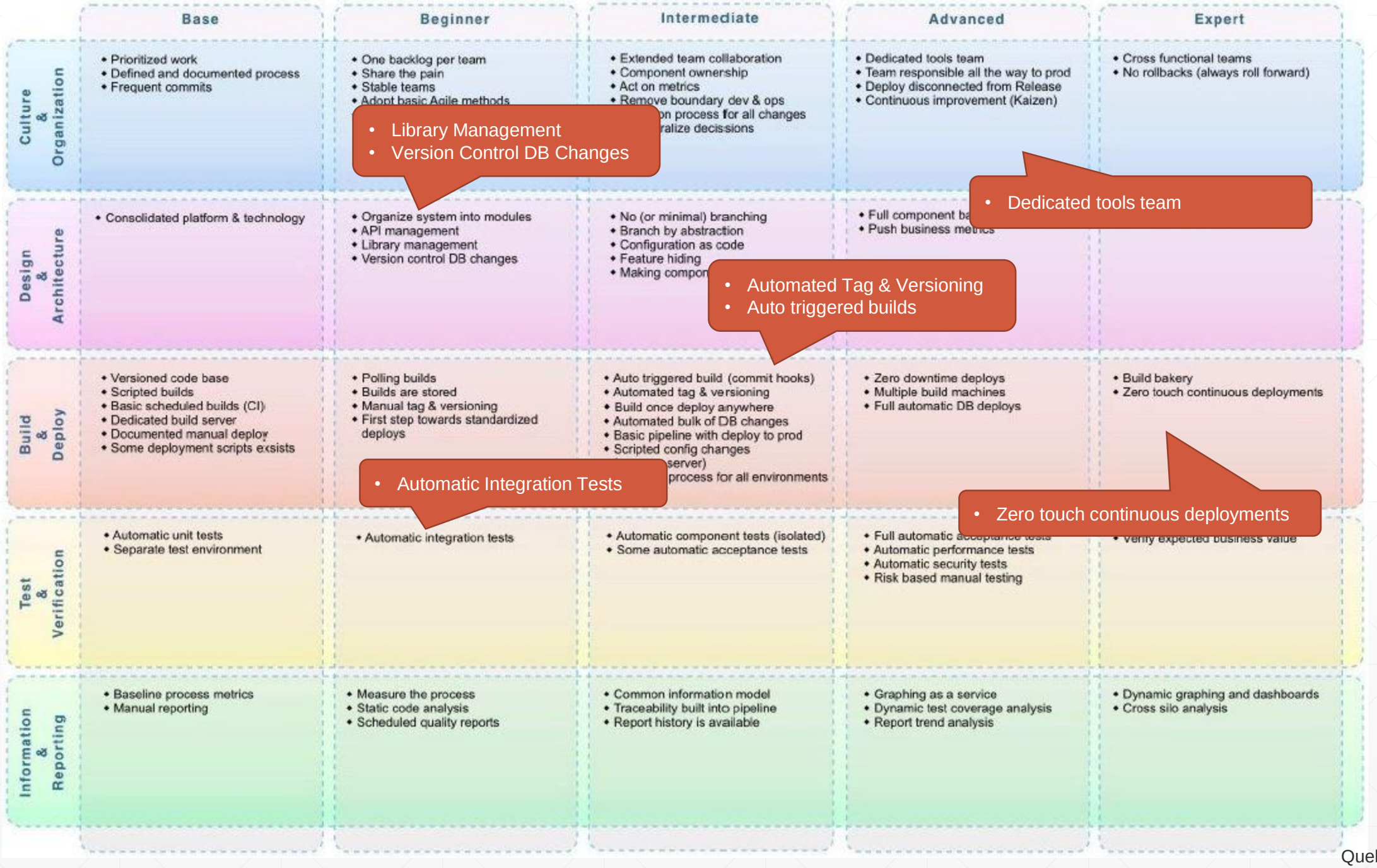
Abenteuer auf dem Weg zu Continous Deployment mit Sitecore

Agenda

- Ein Überblick
- Was nicht funktioniert
- Zweiter Versuch: Grundlagen
- Herausforderungen im Kontext Sitecore

Ein Überblick

Maturlitätsmodelle zu Continous Deployment



Quelle: infoq.com

«Start Small & Continuously Improve»

Was nicht funktioniert

Neuland, Glatteis und Lessons Learned

Automatisieren ohne Erfahrung im Prozess

- Manuelle Abläufe, die nur für das Deployment notwendig sind,
  - gehen vergessen,
  - sind zu wenig getestet,
  - brauchen viel Zeit.

- Unerprobte Abläufe
  - sind aufwändig umzusetzen (fehlendes Know-how),
  - bedeuten ständige Anpassungen (oder schränken ein),
  - machen die Automatisierung komplex & teuer in der Wartung.

Die falsche Technologie einsetzen

- MSBUILD
  - Schlechte Entwicklungsumgebung
  - Schwer zu debuggen
  - Schwerfällig
- Jenkins
  - Man verzichtet auf zu viele Funktionen
- Monolithische Architekturen
  - Schaffen von ungewünschten Abhängigkeiten innerhalb des Tools
  - Schwerfällig bei Änderungen

„Wer als Werkzeug nur einen Hammer hat, sieht in jedem Problem einen Nagel“

Paul Watzlawick

Lessons Learned

- Standard Software Features nutzen
  - Build Steps von Build Server nutzen, nicht mehrere Steps in einem «scripten»
  - VS Publish nutzen, nicht mit selbstgebautem (MSBUILD) Script Files kopieren

- Kleine modulare Komponenten als Bindeglieder im Prozess
  - «günstig» ersetzbar
  - schützen vor gefährlichen Abhängigkeiten im eigenen Werkzeugkasten

- Alles kann automatisiert werden, wenn man ein klares Bild vom Prozess hat.

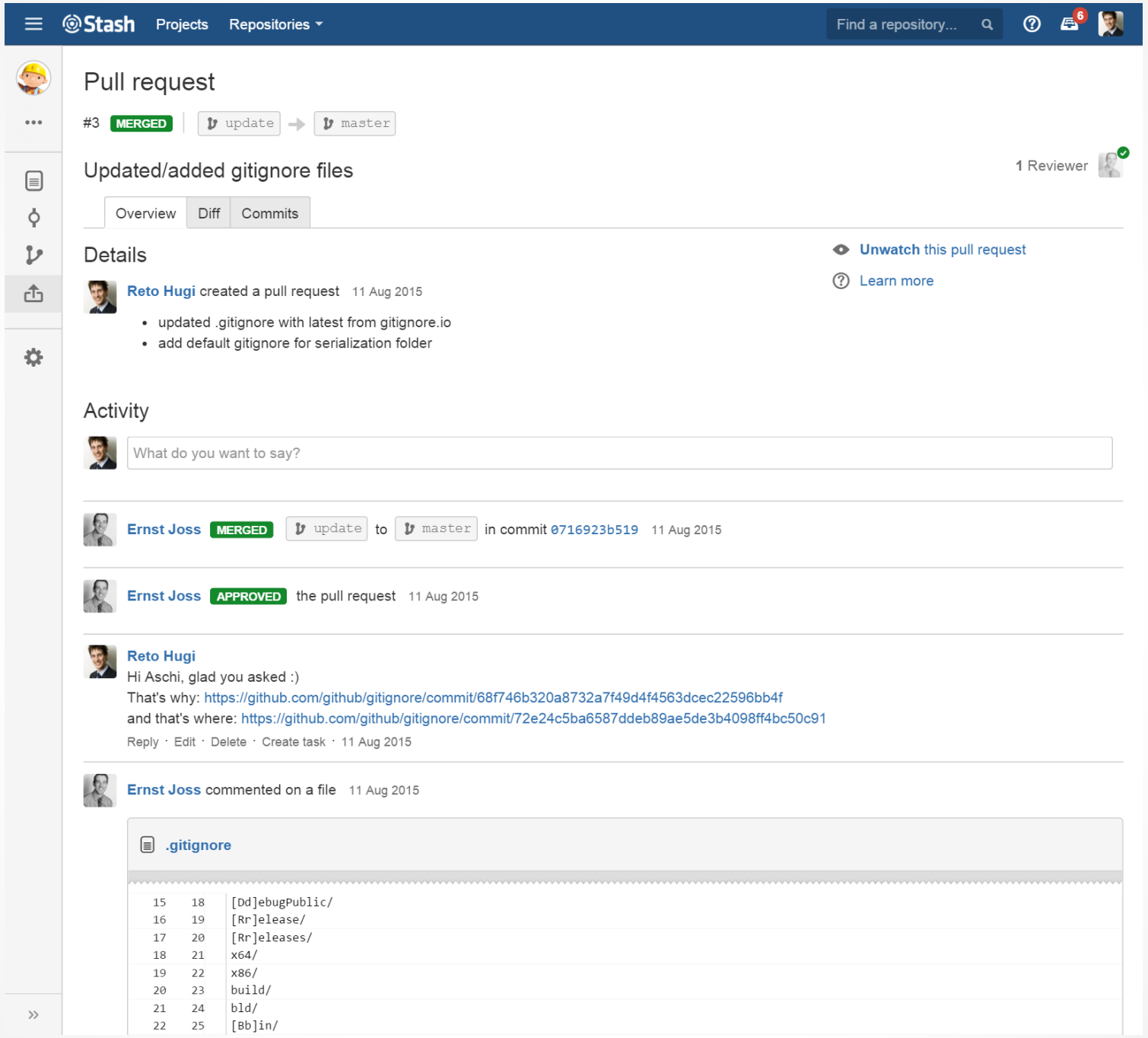
2. Versuch: Grundlagen

Erfahrung ist ein wichtiger Erfolgsfaktor für gelungene Automatisierung

Konzepte und Methoden

- Gitflow
  - Flexibel und dennoch mit klaren Regeln
- Semantic Versioning (SemVer)
  - konsistente Versionierung
  - dokumentiert sich in wesentlichen Teilen selbst («semantisch»)
  - Passt bestens zum Gitflow Branching Modell
- Sitecore
  - Solution «ausserhalb des Webroots»
  - Nur was wir erstellen, gehört in die Versionskontrolle
  - Item Serialisierung: VCS ist führend für Änderungen an Items

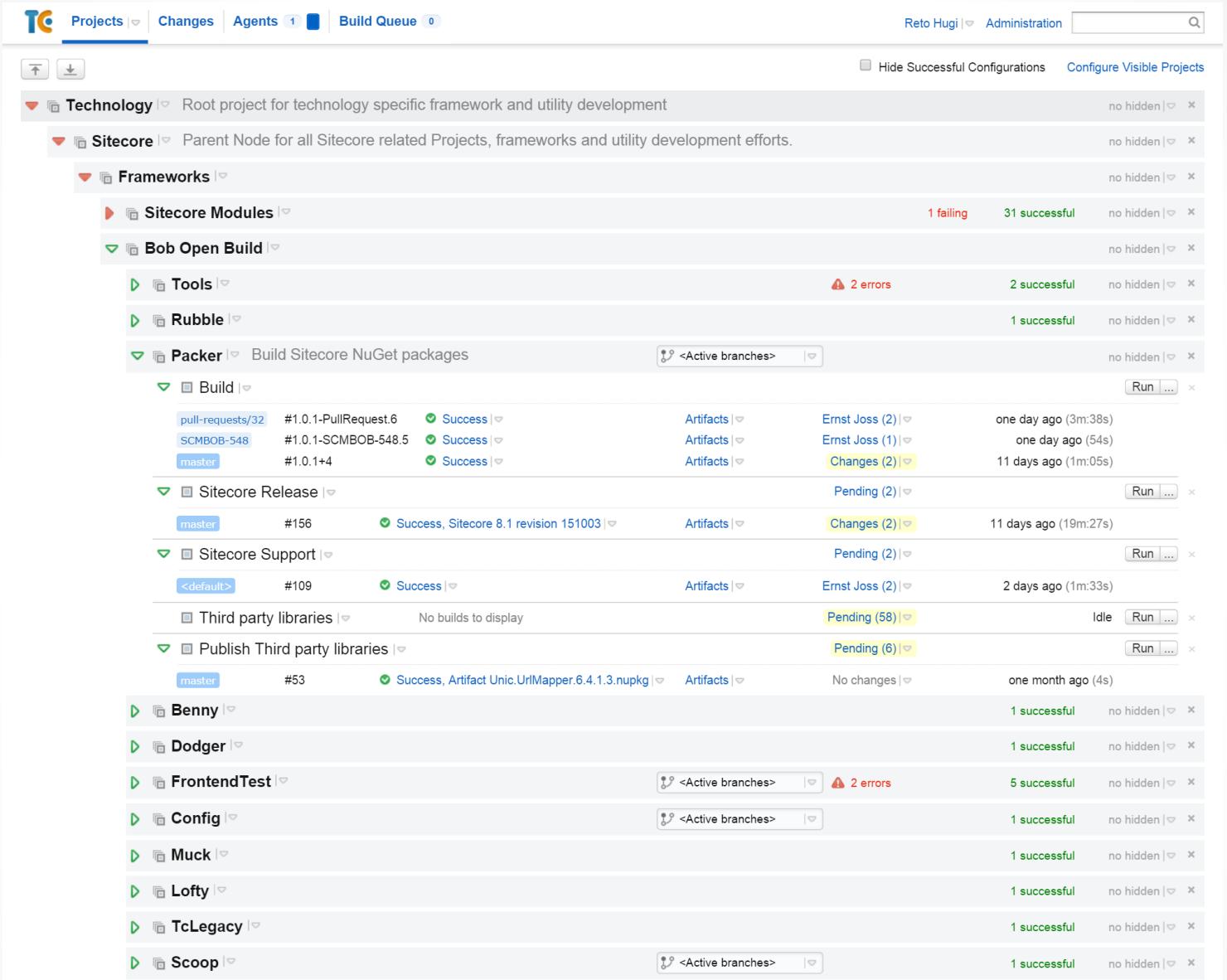
Zentrale Software & Technologie



Stash

Pull Requests / Reviews

Berechtigungen (Projekt, Repo, Branch)



TeamCity

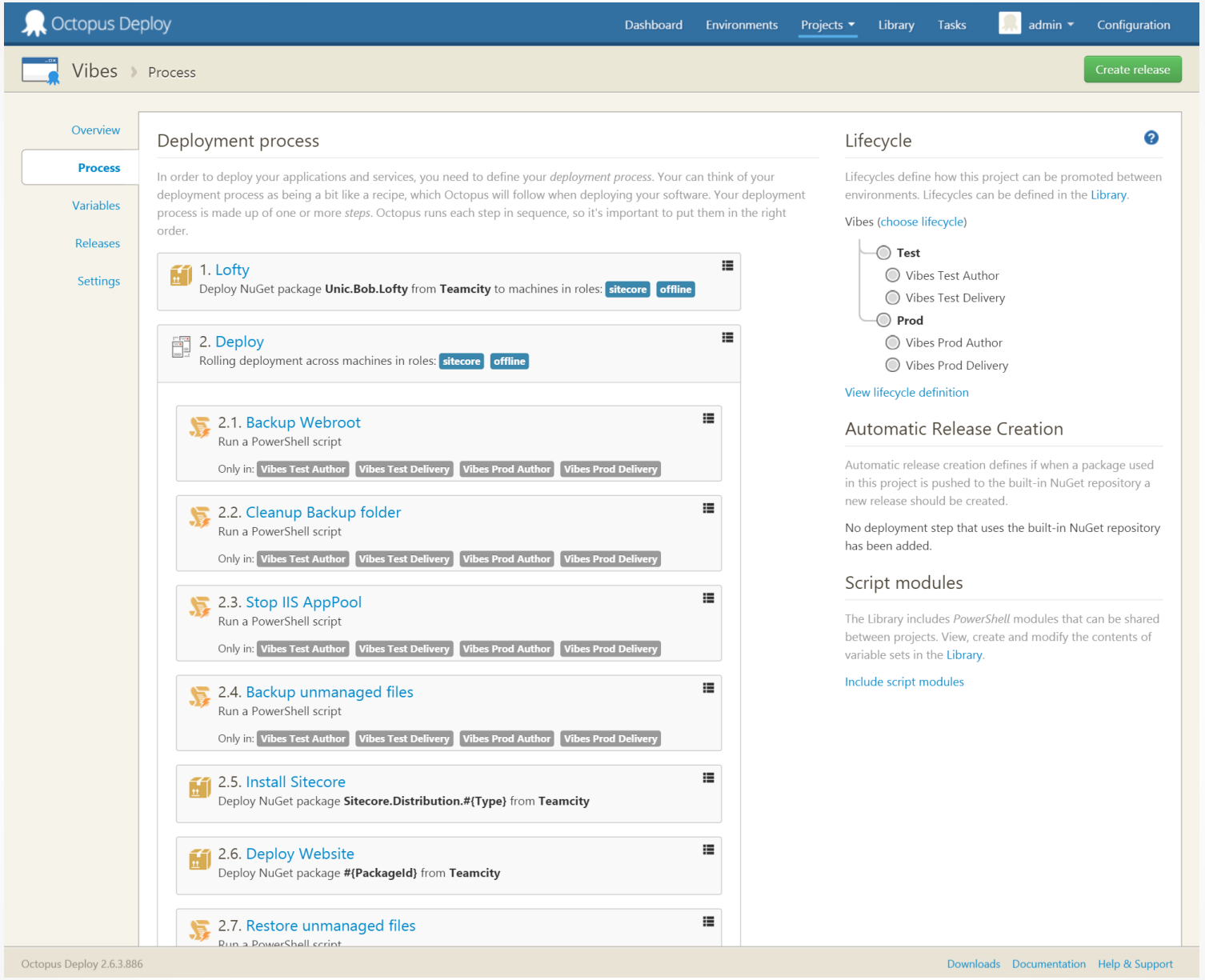
Build Server

Amazon EC2 Build Agents

Vererbbare Build Konfigurationen

NuGet Feed (inkl. Access Control)

14 Build Steps für «Build» Template



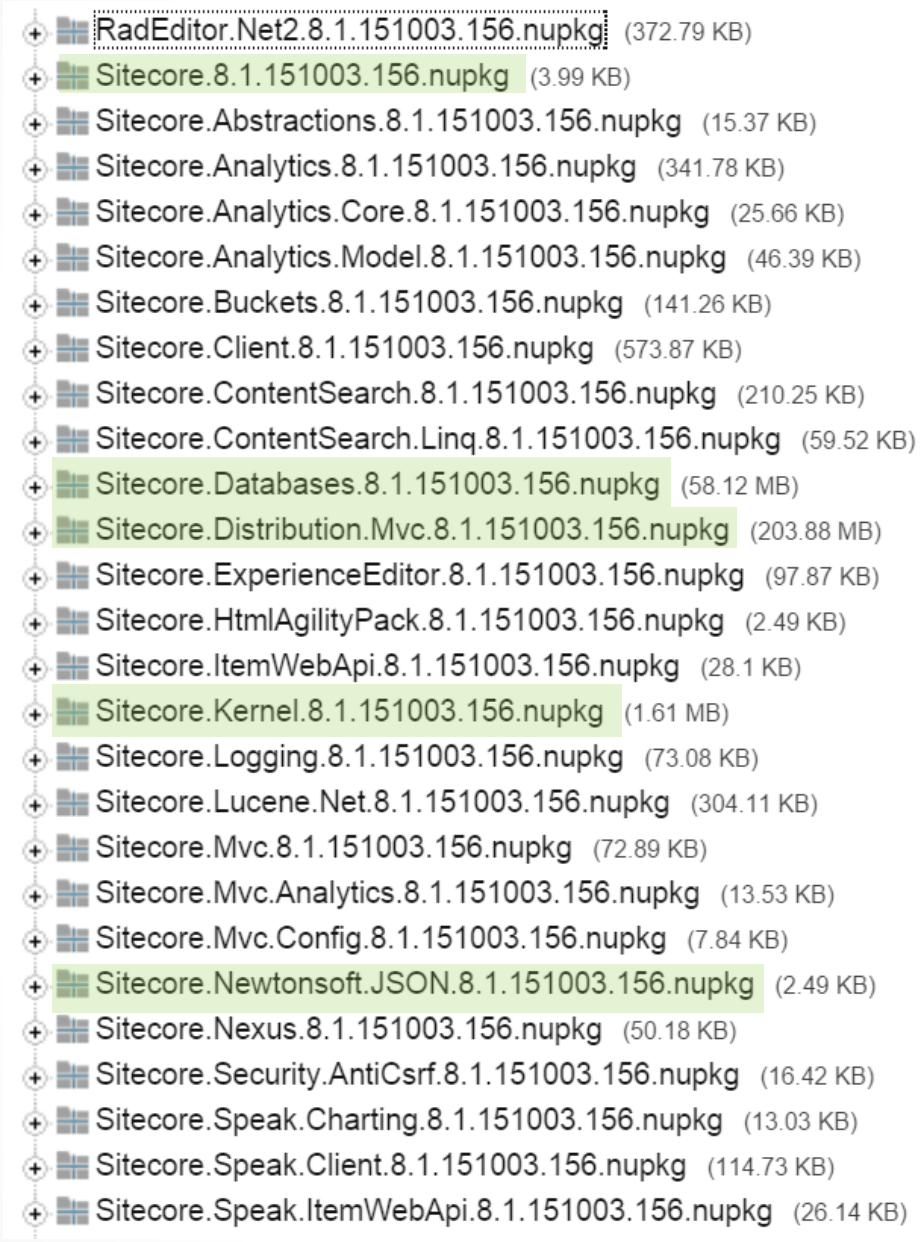
OctopusDeploy

Deployment Server

Online & Offline Deployments

Nightlies von TC getriggered

15 Deployment Steps



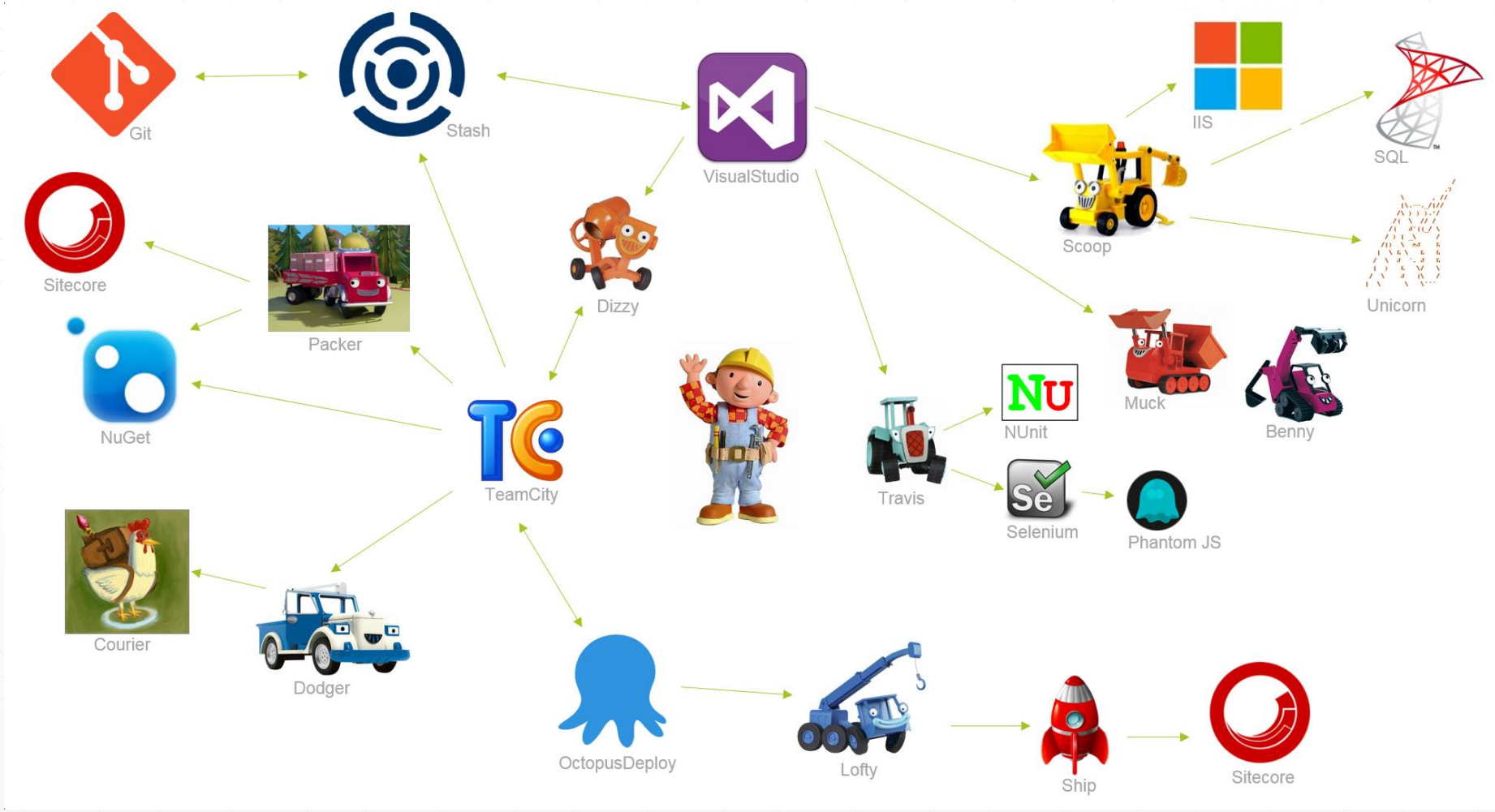
NuGet

Dependency Management

Versionierte Artefakte

Artefakt-Transport

Der Rest: Powershell und Tools

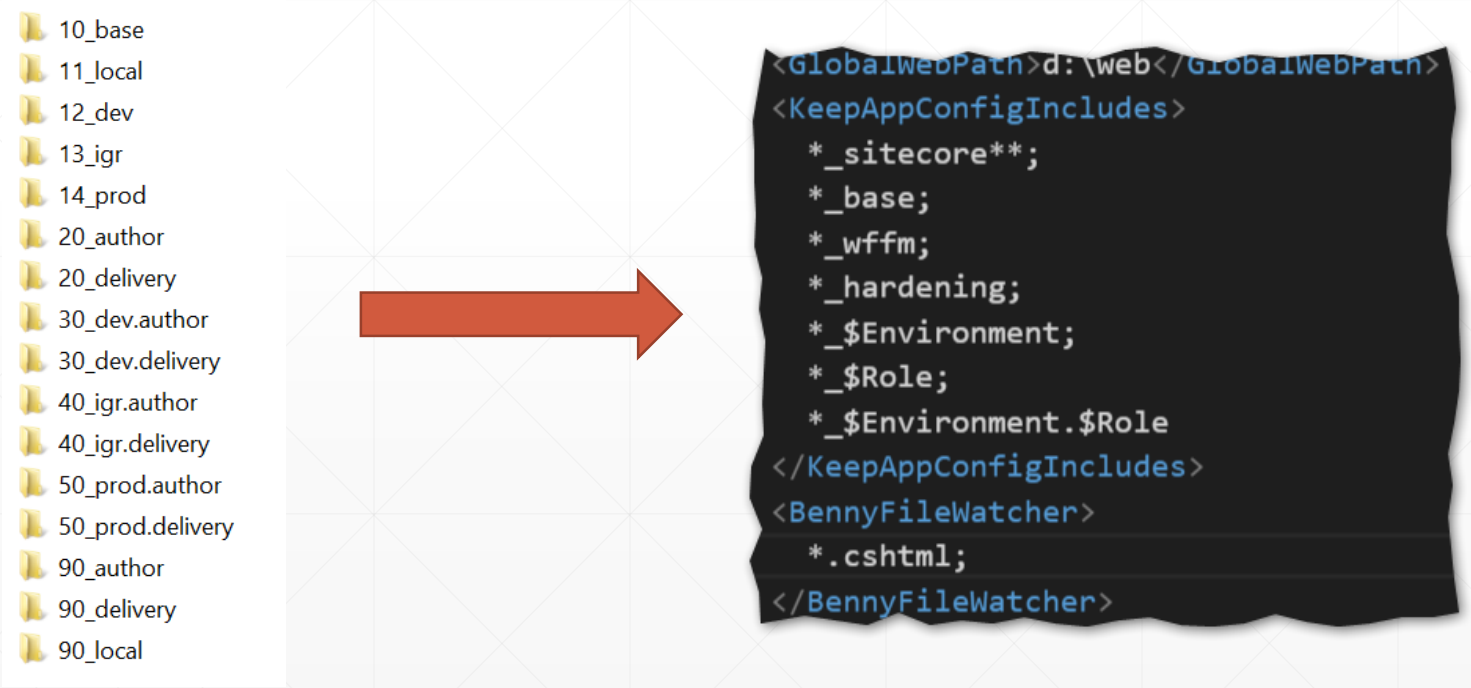


Herausforderungen & Lösungen

Eine Auswahl von Continous Deployment Hürden

Configuration Management

- «include»-Configs
- Zwei Rollen (Management & Delivery)
- N Umgebungen (Dev, Preview, Test, Integration, Produktion usw.)



Versionierung

- Ein Release manuell versionieren scheitert
- Ziel:
  - Automatisch (CI Server)
  - Gemäss SemVer
  - Im Kontext von Gitflow
  - Kompatibel mit NuGet

GitVersion

master	#1.6.0
release/1.6	#1.6.0-beta.1+0
develop	#1.6.0-unstable.26+26
develop	#1.6.0-unstable.25+25
develop	#1.6.0-unstable.24+24
develop	#1.6.0-unstable.23+23

Sitecore Item Management

- Serialization mit Unicorn 2
  - Serialization/app
  - Serialization/appDefault
  - Serialization/testData
- Sitecore Courier
  - Erstellt item.update packages (inkrementel)
  - TeamCity Integration, Vergleich mit letztem Release
- Sitecore Ship
  - Importiert item.update packages, die via OctopusDeploy, cURL & Powershell Script installiert werden

Sitecore Item Management

- Serialization mi rn 2
  - Serialization/
  - Serialization/appD
  - Serialization/testData
- Sitecore Courier
  - Erstellt item.update packa
  - TeamCity Integrati ztem Re
- Sitecore Ship
  - Importiert item. packages, die via OctopusDeploy, cU wershell Script installiert werde

Sitecore Item Management

- Serialization mit Unicorn 3
  - Serialization/app
  - Serialization/appDefault
  - Serialization/testData
- TeamCity
  - NuGet Package der Website enthält auch die serialisierten Items aus «app» und «appDefault»
  - Unicorn Sync via OctopusDeploy & PowerShell Script auf Management Server

Merci!

Reto Hugi

Sitecore MVP 2013 - 2015

twitter.com/retohugi
github.com/retohugi

Was sonst noch geschah

Kleine Highlights

- Automatisierte «Installation» und Konfiguration einer Sitecore Lösung
- Vereinfachtes VS Publishing & Filewatcher
- Automatische Integrationstests
- Trennung von Frontend und Backend
  - «Install-Frontend»
- Installer für Offline Deployments