

Erweitertes Applikations- und Deployment-Management

unter Einsatz des Magento-Composer-Installers

Agenda

1. Brandung – wer wir sind
2. Deployment-Herausforderungen
3. Setup – Composer Deployment
4. Automatisierung von Prozessen beim Deployment

1 brandung – wer wir sind



1 brandung – wer wir sind

GRÜNDUNG



2. STANDORT



IN BEWEGUNG



66 Mitarbeiter **70** bestehende Kunden **>700** realisierte Projekte

1

Kunden



Amir Tchavoshinia

- 5 Jahre Magento Erfahrung
- Teamleiter - 6 Magento Backend Entwickler
- Spezialisierung in Optimierung von Entwicklungs- & Deployment-Strukturen



2 Deployment – Herausforderungen



2

FTP - Deployment



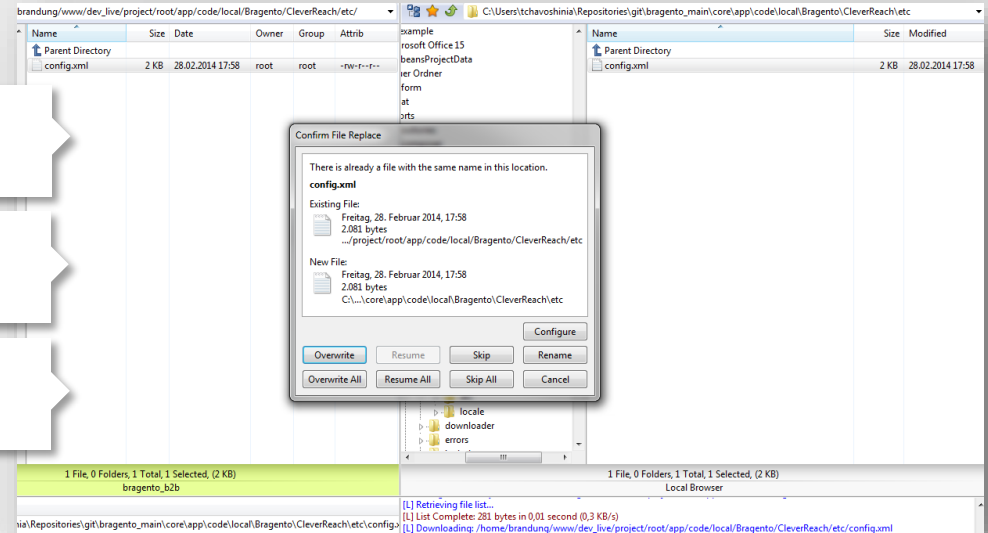
Langsam



Versionsprüfung sehr schwer



Unübersichtlich



2

SVN/GIT - Deployment



Selektive Updates schwierig



Bei Komplett-Versionierung von Projekten
→ große/träge Repos



Modulare Struktur möglich
→ Hoher manueller Aufwand

```
index.php      js      LICENSE.  
index.php.sample  lib      LICENSE.  
install.php    LICENSE_AFL.txt  image  
svn up app/code/local/Bragento/CleverReach/
```

```
cd app/code/local/Bragento/CleverReach/  
app/code/local/Bragento/CleverReach# git pull origin master
```

2

Modman - Deployment



Saubere modulare Struktur



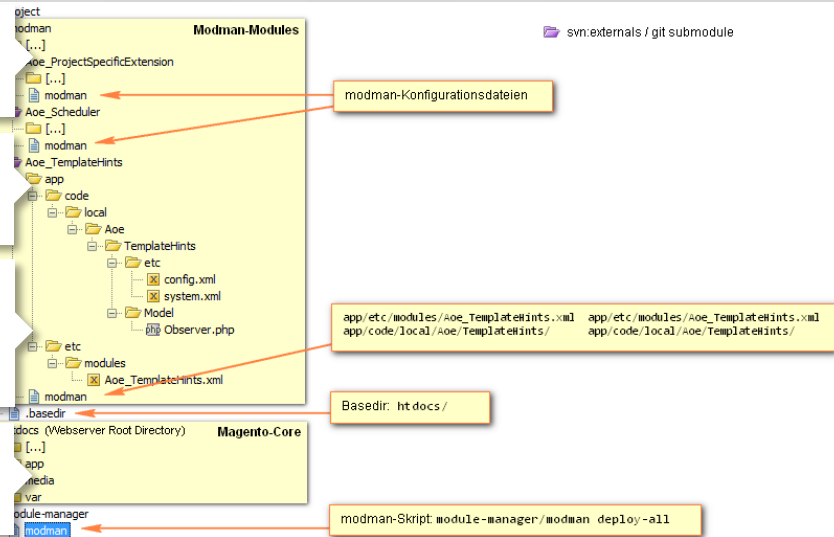
Versions- & Stabilitätskontrolle unübersichtlich



Schwer auf unterschiedliche Umgebungen übertragbar



Kein Dependency-Management

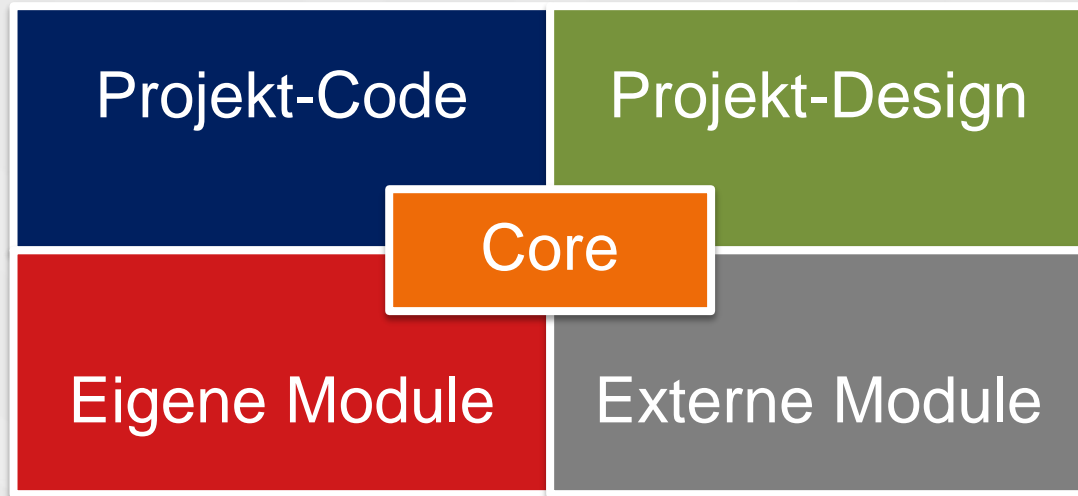


3 Setup – Composer Deployment



3

Das Projekt als Modulsammlung



3

Das Projektkern-Modul

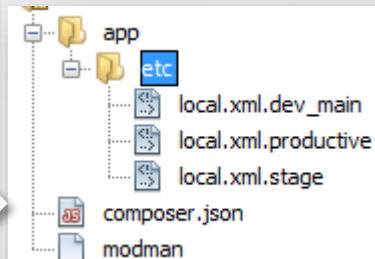
```
{
  "name": "bragento/project_b2b",
  "license": "proprietary",
  "type": "magento-module",
  "description": "Bragento b2b project",
  "require": {
    "magento-hackathon/magento-composer-installer": "1.*",
    "magento-hackathon/composer-command-integrator": "*",

    "bragento/core": "1.8.1.0",
    "bragento/code": "dev-productive",
    "bragento/theme": "dev-productive",

    "bragento/limititerator": "1.0.2.*",
    "bragento/fcm": "2.0.3.0",
    "bragento/logincontrol": "1.3.1.0",
    "bragento/cleverreach": "0.9.2.1",

    "b2b/theme": "1.4.6.1",

    "connect20/locale_mage_community_de_de": "*",
    "magentix/solr": "*",
    "fbrnc/aoe_scheduler": "*",
    "magento-hackathon/hackathon_magemonitoring": "*",
    "firegento/debug": "*"
  },
  "require-dev": {
    "limesoda/sampledtagenerator": "*",
    "fbrnc/aoe_profiler": "*",
    "fbrnc/aoe_templatehints": "*"
  }
}
```



3

Magento-Composer-Installer - Deployment



Saubere modulare Struktur



Versions- & Stabilitätskontrolle



Dependency Management



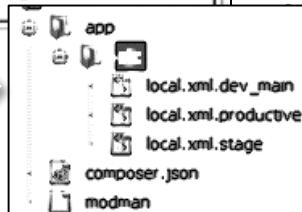
Umgebungs-spezifische Konfiguration möglich



Automatisierung von Prozessen über Scripte & Installer möglich

Projekt-Code

Eigene Module



```
"name": "bragento/project_b2b",
"license": "proprietary",
"type": "magento-module",
"description": "Bragento b2b project",
"require": {
    "magento-hackathon/magento-composer-installer": "1.*",
    "magento-hackathon/composer-command-integrator": "*",
    "bragento/core": "1.8.1.0",
    "bragento/code": "dev-productive",
    "bragento/theme": "dev-productive",
    "bragento/limititerator": "1.0.2.*",
    "bragento/fcm": "2.0.3.0",
    "bragento/logincontrol": "1.3.1.0",
    "bragento/cleverreach": "0.9.2.1",
    "b2b/theme": "1.4.6.1",
    "locale_mage_community_de_de": "*",
    "cli": "*",
    "scheduler": "*",
    "skathon/hackathon_magenmonitoring": "*",
    "debug": "*"
},
{
    "ampdatagenerator": "*",
    "profiler": "*",
    "templatehints": "*"
}
```

4 Automatisierung von Prozessen beim Deployment



4

Die Projekt-Installation

Composer initialisieren

- Magento root dir erstellen
- composer.json Basisdefinition

Projektspezifika definieren

- Repository angeben
- Requirement definieren
→ Projektkern Paket

Projekt Installation durchführen (mit custom installer & scripts)

- Projekt-Installation durchführen – \$ composer install
- (mit custom installer & scripts)
 - Status- & Event- logging
 - Installation aller Pakete
 - Magento-Ordnerrechte setzen
 - Environment Individualisierung
 - Environment suffix entfernen
 - Falls keine env Abhängige local.xml vorhanden
 - ☐ Magento Installer auf CLI oder im Browser
 - Magento DB mit Grundkonfiguration einspielen
 - Cronjob anlegen
 - Initialimport anstoßen (optional)

Der Shop ist komplett installiert und eingerichtet

4

Das Projekt-Update

Projekt-Update durchführen – \$ composer update
(mit custom installer & scripts)

Status- &
Event-logging

Ankündigung
& Countdown
für geplante
Wartung im
Frontend

Maintenance-
mode
aktivieren

Updates
installieren

Environment
Anpassungen
vornehmen

Cache leeren

CDN purgen
(optional)

Maintenance-
mode
deaktivieren

5

Voraussetzungen

- Theoretische Voraussetzungen:
 - Grundverständnis für Modman
→ <https://github.com/colinmollenhour/modman>
 - Grundverständnis für Composer (erweitertes Verständnis → Custom-Installer & -Scripts)
→ <http://getcomposer.org/>
 - Magento Composer Installer
→ <https://github.com/magento-hackathon/magento-composer-installer>
- Infrastrukturelle Voraussetzungen:
 - Modulare Arbeitsweise mit SCM (idealerweise Git/Git Flow)
 - <http://packages.firegento.com> für Community Module
 - Private VCS Quellen werden über eigenes Repository (Satis/Packagist) verwaltet
 - SSH Zugang auf Webserver

Kreativität ist der Weg, **Wirtschaftlichkeit** das Ziel!

Amir Tchavoshinia

Teamleiter Magento-Entwicklung

brandung GmbH & Co. KG

Hansaring 78

50670 Köln

Tel.: +49 221 913920-130

E-Mail: amir.tchavoshinia@brandung.de

Github

bragento