

Entwicklungs- Qualitätssicherung- & Live-Server

Qualitätsgesicherte Entwicklung von TYPO3-Sites

Modelle, Methoden & Erfahrungen mit
TYPO3, Subversion & np_subversion

TUGA 16

Martin Kutschker
Georg Kühnberger

plan2net



Vortragende & Thema

- Vortragende
 - (masi) Martin Kutschker, TYPO3-Core-Dev, TYPO3 Certified T3-Integrator, mk@plan2.net
 - (georg) Georg Kühnberger, TYPO3 Certified T3-Integrator, plan2net GF, gk@plan2.net
- Thema
 - NICHT: Staging- / Production-Environments
 - SCHON: Qualitäts-Sicherung, Team-Developement

 - NICHT: Subversion-Server, redmine usw. aufsetzen
 - SCHON: verteilt & kontrolliert arbeiten mit SVN

Qualitätssicherung (QA) – Warum ?

- TYPO3-Core: Updates, Upgrades
 - TYPO3-Extensions: Updates, Upgrades, Konfigurations-Änderungen
 - TYPO3-Konfiguration: TS, TSconfig, Berechtigungen
 - OS / *amp: Updates, Upgrades...
-
- kleine Änderung, große Wirkungen
 - **Wer** hat **Wann**, **Was**, woran geändert?
 - **Warum** ist dieses/jenes jetzt anders oder gar defekt?

QA-Aufgaben & -Herausforderungen

- « TYPO3-spezifisch » ?
NEIN. - Softwareprojektspezifische Herausforderung-
- Zusammenspiel von OS, Environment, Basis-Applikation, zusätzl. Code, Konfiguration & Inhalt
- verstreut über
 - Configuration: OS
 - Code: App-Core & Erweiterungen
 - Templates: TS, xhtml, css, js ..
 - DB(s) - Inhalte
 - externe Applikationen
- Entwicklung im Team

Lösungen / Methoden

- Separate Instanzen (dev, qa, prod)
- Definierter UND gelebter Freigabe- & Test-Ablauf
- Source-Code Versionierung
- Testen, Testen, Testen!
automatische (Selenium) & manuelle Tests
- Vorteile:
 - Keine ungetesteter / undokumentierter Code
 - (halbautomatische) Dokumentation (SVN commit message)
 - Sourcen sind kontrolliert & dokumentiert
 - Verbessertes Teamwork

Instanzen / Ablauf

- LOCALHOST: Entwicklungsumgebung Workstations
- FULL DEBUG websrf, php, typo3 – in Debug-Modus)
- DEV-SERVER: shared Entwicklungs-Server – Debug-Output & Logging
- QA-SERVER: Qualitäts-Sicherung (Testing & Abnahmen)
kein Debug-Output, aber Logging (Warnings)
- PRODUCTION-SERVER: NO DEBUG

Software-Versionierung

- Warum ?
 - Dokumentiert
 - Nachvollziehbar
 - Rückgängigmachbar
 - Kollaboration
 - Automatisierung
- Versionierung – Welche ?
 - CVS
 - SVN-Subversion
 - GIT (Linux Kernel)
 - ... Mercurial, Bazaar

Subversion – Intro

- **Subversion (SVN)** ist eine Open-Source-Software zur Versionsverwaltung von Dateien und Verzeichnissen.
- Konzepte
 - „Repository“ (gemeinsames zentrales Archiv)
 - „Revisions“ - (Zähler der Änderungen)
 - „working copy“ lokale Arbeitskopie
 - Aus-checken (checkout) / Ein-checken (commit)
 - Semi-automatische Konflikt-Auflösung
- Vorteile
 - Anzeige der lokalen Änderungen erfolgen ohne Netzwerkzugriff
 - nur geänderten Teile werden übertragen
 - Atomare Commits (ganz oder garnicht)

Arbeiten mit Subversion – I

- SVN-Server Installation
- SVN Clients
 - Shell Clients für alle OS: LINUX, OSX, Windows, *NIX ...
 - Gui-Client Windows: Tortoise
 - SVN-Integrationen in IDEs (zb. Eclipse)
- Befehle I
 - checkout (Erstellen einer lokalen Kopie)
 - update (Aktualisieren d. lokalen Kopie)
 - commit (Übertragen d. lokalen Kopie an den Server)
 - diff (gg. beliebige Versionen)
 - revert

Arbeiten mit Subversion - II

- Befehle II
 - log / history
 - Import
 - Export
 - rename & delete (Achtung niemals OS-Befehle verwenden)
- tags - Nomenklatura
 - trunk
 - branches (Kopien von trunk)
 - tags (Kopien von trunk)

Webbased Tools

- Screenshots:
- Redmine
- Trac
- ????

OK - Wie nun TYPO3 versionieren?

- DB-Records (TSconfig, TypoScript Setup/Constants) in Include-Files auslagern, zb:

```
<INCLUDE_TYPOSCRIPT: source="FILE: fileadmin/.../ts_contants.txt">  
<INCLUDE_TYPOSCRIPT: source="FILE: fileadmin/.../ts_setup.txt">
```

```
<INCLUDE_TYPOSCRIPT: source="FILE: fileadmin/.../tsconfig.txt">
```

- Templates >> Filesystem
- Extensions >> Filesystem

Beispiel TYPO3 Setup - versioniert

- Filestruktur – Beispiel
 - fileadmin/templ/xhtml/startseite.html ... css ...
 - fileadmin/templ/setup/pageTSconfig.txt
 - fileadmin/templ/setup/ts_setup.txt
 - fileadmin/templ/setup/ts_constants.txt
 - fileadmin/templ/setup/auto/ - EXT: p2_tsfilesys
 - typo3conf/ext/meineEXT-1
 - typo3conf/ext/meineEXT-2

Software vs. Konfiguration

- Im Repository versioniert (Repository Struktur)
 - /templates/* Templating: js, xhtml, css
 - /extensions/EXT-1 Software: TYPO3-Extensions
 - /configs/dev/ optional Konfiguration: PIDs, UUIDs
 - /configs/qa/
 - /configs/prod/
- nicht versioniert
 - fileadmin/Bilder/... Dokumente (content-files)
 - typo3conf/localconf.php (ABER dokumentieren – zb. wiki)
 - "standard" TYPO3-Extensions aus dem TER
 - DB-Inhalte (content-DB)
 - Konfiguration: PIDs, UUIDs

Tools

- Subversion-Server
- SVN shellclient, oder GUI-Client (Tortoise SVN)
- EXT np_subversion
- Optional:
 - Redmine, trac,
 - oder Ähnliche

Arbeits-Methoden & Ablauf - IDEAL

- Entwicklung & Tests auf lokalen workstations
- Commit ins SVN
- Checkout am Dev-Server (2 clicks mit np_subversion)
- EXPORT auf die QA-Instanz; (opt. Revision „taggen“)
- Tests & Abnahme >> oder zurück an den Start
- danach Export als T3X, Import in Produktion
- oder besser: EXPORT der qualitätsgesicherten Revision auf die Produktion

- optional: Pre- & Post-Commit Hooks

Arbeits-Methoden & Ablauf - PRAXIS

- Leider zu oft gesehen
 - keine definierten QA-Prozesse, keine Server, keine Methode
 - zu wenig Willen oder Zeit zum Testen
 - zu oberflächliche Tests
 - keine QA Instanz
- Mögliche, noch sinnvolle, Reduktion auf:
 - local DEV
 - zentral DEV als QA
 - Production

Arbeitsablauf (lokal >> SVN >> Server) bleibt gleich

Hinweise / Warnungen

DONT 's:

- auf Produktions-Instanzen „schnell mal was im Code geändert“
- aufgrund von Zeitmangel „Ausnahmen“ vom Ablauf machen

DO 's:

- Qualitätssicherungs-Abläufe WOHL definieren & LEBEN
- SVN-Commit oft – mind. 1x am Tag
- Commits MIT Kommentaren
- Testen, Testen, Testen
- nachvollziehbare OS-, Code oder Configurations-Änderungen auf der Produktion erfolgen

Fragen & Antworten / Links

- Bücher

- DAS Subversion Buch

<http://svnbook.red-bean.com/>

- Pragmatic version control

[http://www.amazon.de/Pragmatic-Version-Control-Subversion-](http://www.amazon.de/Pragmatic-Version-Control-Subversion-Programmers/dp/0977616657/ref=sr_1_6?ie=UTF8&s=books-intl-de&qid=1248271703&sr=8-6)

[Programmers/dp/0977616657/ref=sr_1_6?ie=UTF8&s=books-intl-de&qid=1248271703&sr=8-6](http://www.amazon.de/Pragmatic-Version-Control-Subversion-Programmers/dp/0977616657/ref=sr_1_6?ie=UTF8&s=books-intl-de&qid=1248271703&sr=8-6)

- Sourcen

- np_subversion im TER: http://typo3.org/extensions/repository/view/np_subversion/current/
- np_subversion Bugtracker http://forge.typo3.org/projects/extension-np_subversion/issues
- np_subversion Sources https://svn.typo3.org/TYPO3v4/Extensions/np_subversion/trunk

- Videos

T3 podcast - Extension "np_subversion" Bastian Waidelich & Kasper

<http://castor.t3o.punkt.de/files/subversion.m4v>