

Integration von Moodle-Erweiterungen mittels Git

1. Warum mit Git

Wann ist die Aktualisierung über die Moodle-Oberfläche problematisch

- Administratoren sollen die nur Einstellungen aber nicht den Code ändern können, z.B. wenn es Administratoren gibt, die aber keine Serverseitigen Änderungen durchführen können sollen.
Strikte Trennung zwischen Server- und Moodle-Verwaltung
- Mehrere Moodle-Systeme werden zentral verwaltet bei gleicher Codebasis
- Der Code wird zentral verwaltet, da es verschiedene Anpassungen gibt
- Allgemeine Sicherheitsbedenken wegen der Schreibrechte für den Webserver

2. Verschiedene Situationen

Einzelne Moodle-Installation

- Änderungen werden i.d.R. direkt im Git-Repository der Installation gepflegt

mehrere Moodle-Installationen auf demselben Server

- zentrales lokales oder entferntes Git-Repository

Viele verschiedene Moodle-Systeme auf verschiedenen Servern (Moodle-Hosting)

- zentraler Git-Server

3. Möglichkeiten der Integration mittels Git

- submodule
- subtree

4. Übungen

Vorbereitung

Dummy-Moodle:

Branches: MOODLE_34_STABLE, master

Git-URL: github.com/moodle/moodle.git

Dummy-Plugin:

local/example

Branches: master

Git-URL: github.com/grabs/local_example.git

4.1. Integration mit "submodule"

Wann geeignet:

- Bei Lizenzproblemen
- Wenn nur ein einzelnes Repository innerhalb einer einzelnen Installation gepflegt wird
- In Entwicklungsumgebungen, da leicht ins Zielrepository gepusht werden kann.

Wann nicht geeignet:

- wenn zentrales Git für mehrere Moodle-Installation genutzt wird.
- wenn ein Clone zuverlässig konsistent sein soll

Probleme:

- Aktualisieren ist nicht immer einfach
- Entfernen ist kompliziert
- Inkonsistenzen durch vergessene Commits im Hauptrepository

Submodule anlegen:

```
git submodule add -b master /github.com/grabs/local_example.git
    local/example
git commit -m "added submodule local example into local/example"
```

Submodule aktualisieren

```
cd local/example
git pull
cd ../../
git add local/example
git commit -m "Update submodule local/example"
```

Submodule neuen Branch auschecken

```
cd local/example
git fetch origin
git checkout <new branch>
cd ../../
git add local/example
git commit -m "Update submodule local/example to branch <branch>"
```

Submodule entfernen

(leider etwas aufwändig!)

```
git submodule deinit local/example
git rm local/example
rm -rf .git/modules/local/example (or just ../local/)
git commit -m "removed submodule local/example"
```

4.2. Integration mit "subtree"

Plugin hinzufügen

```
git remote add -f local_example /github.com/grabs/local_example.git
git subtree add --prefix=local/example local_example/master -m "added..."
```

Plugin aktualisieren

```
git fetch local_example
git subtree merge --prefix=local/example local_example/master -m "update..."
```

Plugin entfernen

Easy as pie

```
git rm -rf local/example
git commit -m "removed plugin mod/plugin"
git tag removed_local_example
git remote rm local_example #wenn nicht mehr benoetigt
```

Plugin erneut einfügen, evtl. gleich aktualisieren

```
git fetch local_example
git subtree add --prefix=local/example local_example/master -m "added..."
```