

Umstellung von SVN/Trac auf git/GitLab

Was wurde getan, wie nutzen wir es und wo kann und soll es hingehen?

Jan-Hendrik Willms (Uni Oldenburg)

David Siegfried (Uni Vechta)

Roadmap

- Gründe für einen Umstieg
- Mögliche Migrationspfade
- Was möchten wir mitnehmen?
- Zeitpunkt der Umstellung
- Aktivitäten nach der Umstellung
- Aussicht

Gründe für einen Umstieg

- Pluginentwicklung oft schon in GitLab
- Möglichkeiten des Reviews
- Branchverwaltung
- "Zeitgemäß"
- ...

Mögliche Migrationspfade

- Der Stud.IP-Weg:

Wir migrieren alles und schreiben dafür alles selbst.

<https://github.com/teilax/studip-gitlab-migrate>

- Mühsam
- Entscheidung:

Wir machen einen Cut zur 5.0 und sehen dann weiter.

Was möchten wir mitnehmen?

- Idealvorstellung:

*Wir erhalten die komplette Historie inkl.
Verweisen, Kommentaren und Dateianhängen.
Auch die Autorenezuordnung...*

- Es muss nicht immer ideal sein
- Trac und SVN bleiben, keine aktive Entwicklung mehr
- Bugfixes für alte Releasesbranches

Zeitpunkt der Umstellung

- Nägel mit Köpfen Ende Juli 2021

git und gitlab

Liebe Community!

Stud.IP hat jetzt auch git! So in echt und richtig. Also seit gestern wird Stud.IP in unserem gitlab unter folgender Adresse weiter entwickelt: <https://gitlab.studip.de>

- Stud.IP 5.0 erste Version im GitLab

Aktivitäten nach der Umstellung

Authentifizierung

- Mühsames Erstellen von Konten im GitLab
- Weitere Hürde für neue Entwickler:innen
- Authentifizierung im Trac über das DevBoard
- Lösung: JWT

Plugin: [JWTAuth](#)

Aktivitäten nach der Umstellung

Alte Tickets

- Plugin für die Konvertierung von Trac-Tickets
 - Übernahme aller Kommentare
 - Übernahme aller Anhänge
 - Ticket wird mit Hinweis „migrated“ geschlossen
 - Link zum GitLab wird erzeugt

Aktivitäten nach der Umstellung

Integration im DevBoard

- Plugin für Trac war vorhanden
- Angepasst für GitLab:
 - Issue: `gitlab:#12345`
 - Commit: `gitlab:abcd123`
 - Merge Request: `gitlab:!123`
 - Snippet: `gitlab:$123`

Aktivitäten nach der Umstellung

QM und Dashboard

- Trac: Dynamic Fields > Labels in GitLab
- Dashboard fürs Qualitätsmanagement
 - Filter nach Typ (StEP/TIC)
 - Filter nach Versionen
 - Filter nach Status
 - Anzeige Status Merge Request

Aktivitäten nach der Umstellung

Übersetzungen

- Synchronisierung mit weblate
- Aktuell noch manuelles Extrahieren
- Weblate holt sich Neuerungen
- Weblate schreibt Übersetzungen automatisch

Aktivitäten nach der Umstellung

Pipelines

- Trac: Linting und Unit Tests durch Jenkins
- GitLab: Zuerst Jenkins genauso übernommen
- Mittels GitLab CI:
 - Linting PHP, CSS, JS und vue.js
 - Unit Tests
 - Functional Tests
 - JSONAPI Tests
 - PHPStan Level 0 (optional)

Aktivitäten nach der Umstellung

Releases

- Automatische Erstellung von Releases
- Bei Tagerzeugung „vx.y“
 - Installation der PHP-Bibliotheken mittels composer
 - Installation der JS-Bibliotheken mittels npm
 - Bundling der Assets mittels Webpack
 - Packen der Releases
 - Hochladen der Releases zu GitLab

Und nun?

Ausblick und Planung

- Automatische Extrahierung von Übersetzungen
- Automatische Erzeugung von Testsystemen
- Weitere Integration von Codequalitätssicherungen
 - PHPStan
 - PHP Mess Detector
 - Unit Tests
 - Code Coverage

Fragen?
Anregungen?