WWU
MÜNSTER

Course Life Cycle und Lehrenden Self-Service

MoodleMoot 2018

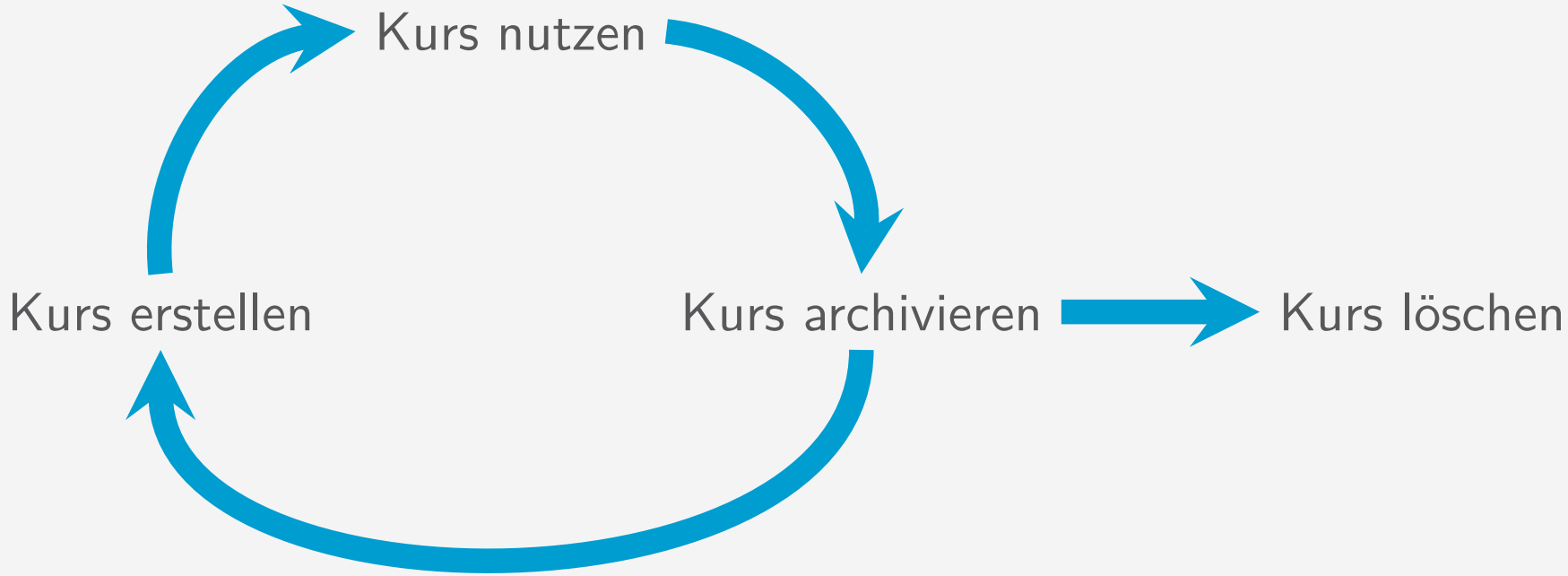
Tobias Reischmann

14. März 2018

wissen.leben

moodlemoot

Phasen eines Kurs Lebenszyklus



Motivation

WWU Münster - Learnweb

- Anzahl Kurse im Wintersemester 2017/18: **2740**
- Anzahl Kurse gesamt: **29440**
- Ältester Kurs von 2007

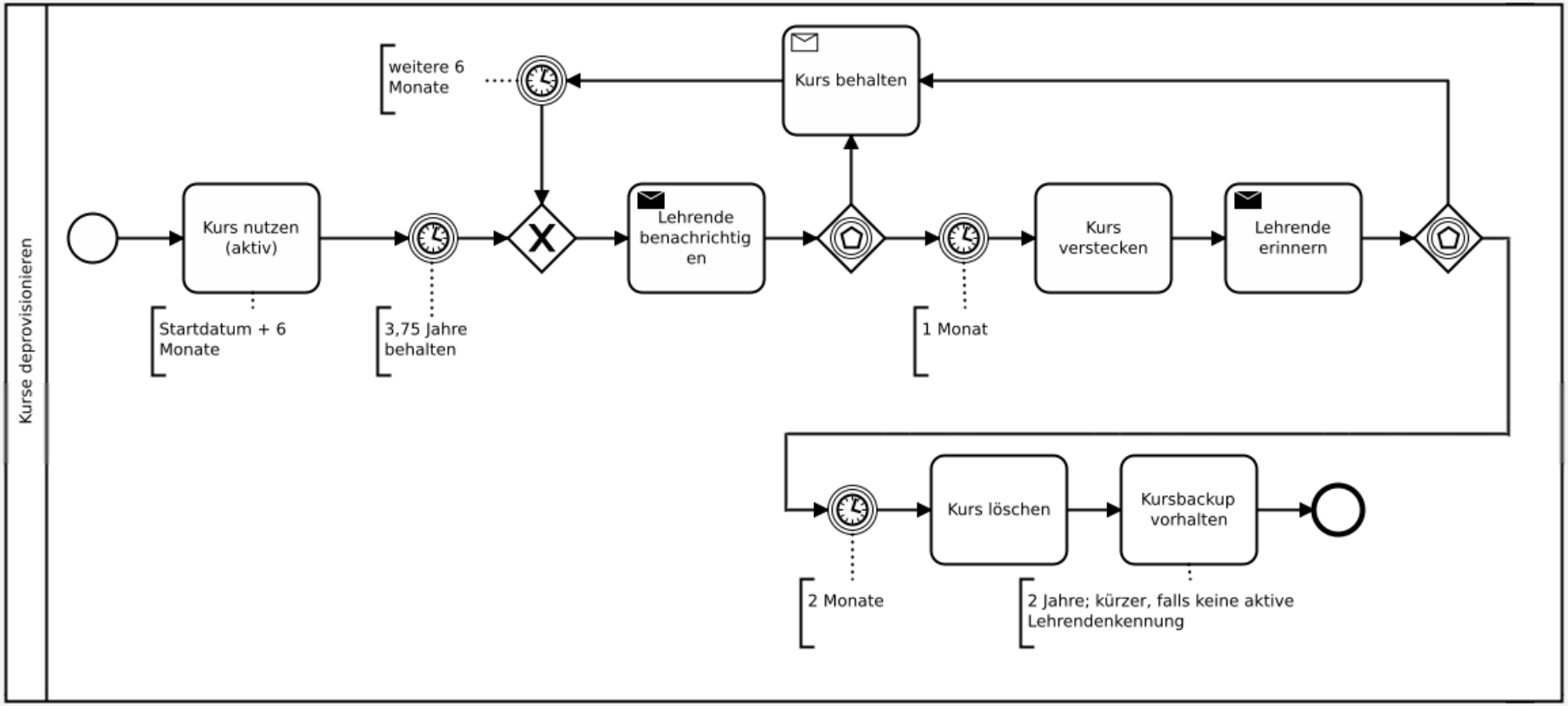
Anforderungen:

- Automatische Abläufe zu den einzelnen Schritten
- Einbinden der Lehrenden in den Ablauf

Agenda

1. Anforderungen und Konzeptions
2. Live Demo
3. Wie können die Anforderungen von euren Plattformen einfließen?

Beispiel Workflow



Anforderungen

Unterschiedliche Organisationen haben unterschiedliche Anforderungen:

- Unterschiedliche Wartezeiten
- Opt-In vs. Opt-out
- Backup von Kursnoten
- Mehrere Erinnerungen
- Andere Trigger
  - X Tage nach Kursende
  - Kein Lehrender mehr im Kurs
- Andere Workflows
  - Kurse ins nächste Semester kopieren
  - Kurse "leeren"
- ...

Subplugins als Bausteine

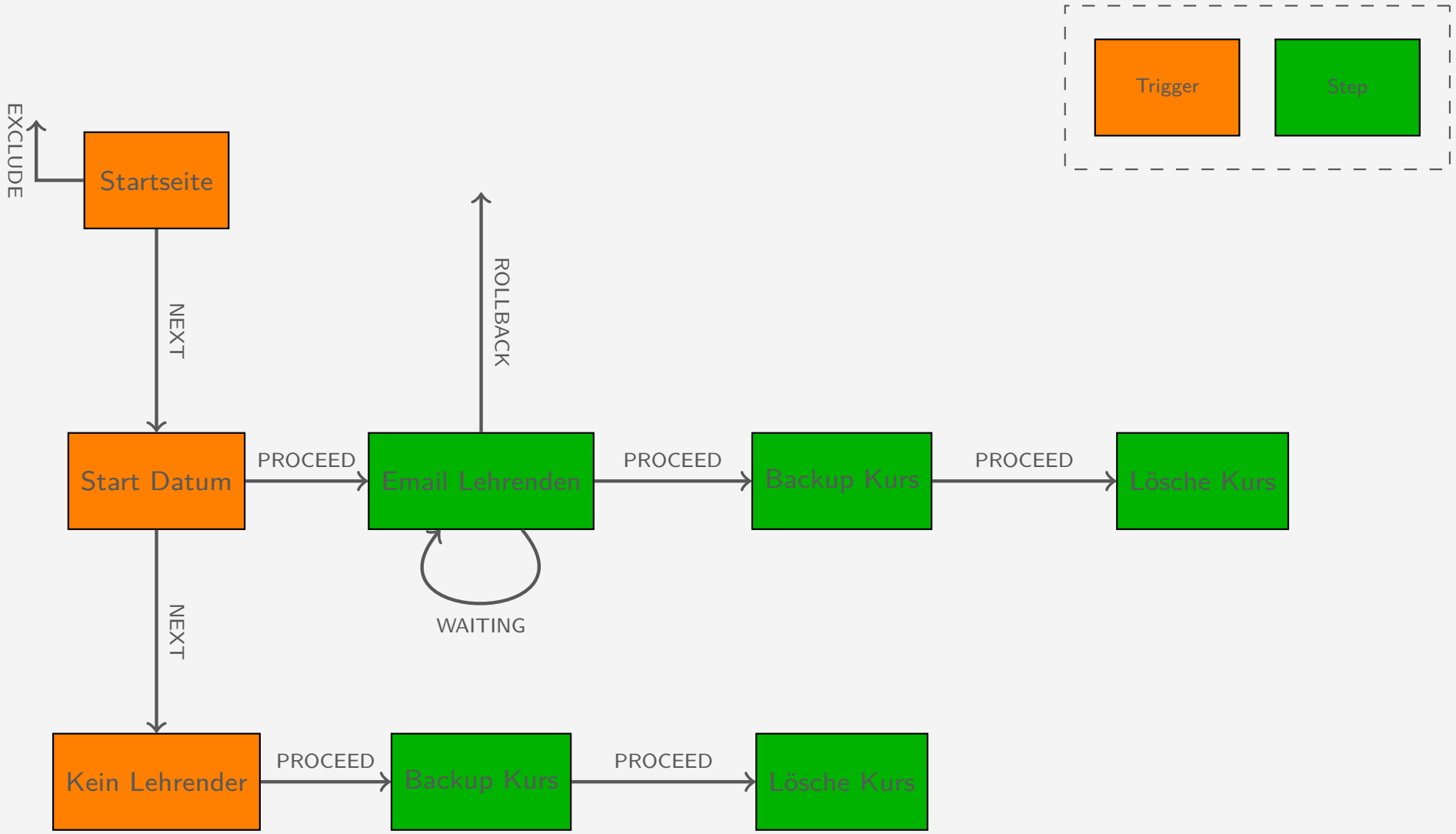
Trigger

- Über den Cron gestartet
- Startet einen bestimmten Prozess für einen Kurs
- Gibt einen von drei Zuständen zurück:
  - TRIGGER
  - NEXT
  - EXCLUDE

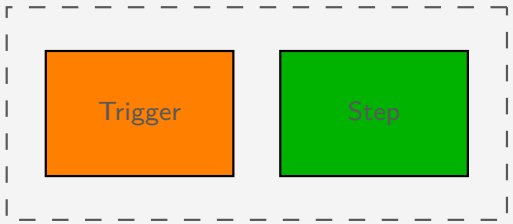
Step

- Schritt eines Prozesses
- Für einen einfachen Arbeitsschritt durch
- Gibt einen von drei Zuständen zurück:
  - PROCEED
  - WAITING
  - ROLLBACK

Mögliche Workflow Definition



Manuelle Workflows



Live Demo

Wie können die Anforderungen von euren Plattformen einfließen?

- Einstellungsmöglichkeiten in existierenden Subplugins
- Eigene Trigger implementieren
- Eigene Steps implementieren
- Teilen
- Andere geteilte Subplugins installieren

Die Subplugin API

- Einfach benutzbare API
- Definiert Interfaces, welche programmiert werden müssen
- Kern Funktionalitäten werden vom Core Plugin zur Verfügung gestellt
  - Speicherung von Einstellungen für Subplugin
  - Speicherung von Daten im Laufe des Prozesses
  - Interface zur Interaktion mit den Lehrenden

Beispielhafte Trigger Implementierung

```
/**
 * Excludes the site course from cleanup.
 * @param $course object to be processed.
 * @return trigger_response
 */
public function check_course($course) {
        if ($course->id == SITEID) {
            return trigger_response::exclude();
        }
        return trigger_response::next();
}
```

Beispielhafte Step Implementierung

```php
/**
 * Deletes the course.
 * @param int $processid of the respective process.
 * @param int $instanceid of the step instance.
 * @param mixed $course to be processed.
 * @return step_response
 */
public function process_course($processid,
  $instanceid, $course) {
    delete_course($course->id, true);
    return step_response::proceed();
}
```

Bei Interesse

Tobias Reischmann

t.re@wwu.de
http://erc.is/p/t.re
http://keybase.io/tobiasreischmann