

Softwaretechnikpraktikum
SS 2015
Datum: 01.06.2015

Gruppe: swp15.aae
Betreuer: Prof. Gräbe
Tutor: Klemens Schöllhorn

Projektteam:
Felix Albroscheit
Dorian Dahms
Paul Eisenhuth
Martin Lechner
Christian Seidemann
Ruth von Borell
Franz Wendt

Qualitätssicherungskonzept

Keine Änderung zum vorherigen Qualitätssicherungskonzept!

Inhaltsverzeichnis

1	Dokumentationskonzept	2
1.1	Sprache	2
1.2	Coding Standard	2
1.3	Code-Dokumentation	2
1.4	sonstige Dokumentation	2
2	Testkonzept	3
2.1	Komponententests	3
2.2	Integrationstests	3
2.3	Systemtests	3
2.4	Abnahmetest	3
3	Organisatorisches	4
4	Quellen	5

1 Dokumentationskonzept

Die Einigung auf ein einheitliches Dokumentationskonzept ist für die erfolgreiche und termingerechte Entwicklung der Stadtteilplattform unerlässlich. Dadurch wird ein gemeinsames Arbeiten erleichtert und garantiert, dass auch bei personellen Veränderungen ein weiters Voranschreiten ohne Probleme möglich ist.

1.1 Sprache

Unsere Standarddokumentationssprache ist deutsch. In deutsch werden die internen Quelltextkommentare verfasst, sowie alle externen Protokolle und Berichte. Ausnahmen hierbei bilden Fachbegriffe wie Contentmanagement System oder Semantic Web, die im Englischen geläufiger sind, sowie Programmstrukturen des Quelltexts, also Klassen, Variablen, Funktionen etc.

1.2 Coding Standard

Wir halten uns beim Programmieren an die Coding Standards(<https://www.drupal.org/coding-standards>) von Drupal, die wiederum entfernt auf den „PEAR Coding standards“ basieren. Durch eine einheitliche Verwendung dieser globalen Standards kann sich jeder Entwickler weltweit besser im fremden Code zurechtfinden. Wie bereits erwähnt stellt die Dokumentation eine Ausnahme dar, da diese auf Deutsch erfolgt. Grund dafür ist, dass es sich um ein lokales Projekt handelt.

1.3 Code-Dokumentation

Eine externe Dokumentation ermöglicht eine schnelle und unkomplizierte Einarbeitung in ein bestehendes Projekt. Ausserdem bietet eine solche Dokumentation die Möglichkeit explizite Probleme nachzuschlagen.

Für die automatische Code-Dokumentation verwenden wir phpDocumentor (<http://www.phpdoc.org/>). Durch Verwendung von Tags in der Quelltextdokumentation kann der phpDocumentor zusammengehörige Blöcke, wie z. B. Klassen, Funktionen, ect., ausmachen und diese formatiert darstellen mitsamt ihren zugehörigen Attributen (Variablen, Rückgabewerte, ect.).

1.4 sonstige Dokumentation

Es ist wichtig, nicht nur den Quelltext zu dokumentieren, sondern auch alle anderen Ereignisse, die das Projekt betreffen zu protokollieren. Dazu zählen interne Gruppenmeetings, wie auch jegliche Treffen mit externen Stakeholdern. Darin sind alle wesentlichen Details zu den Treffen festzuhalten, so dass bestimmte Entwicklungen und Entscheidungen auch nach längerer Zeit nachvollzogen werden können. Diese Protokolle werden in das Wiki in OLAT hochgeladen und sind dort unter dem Punkt „Protokolle“ für jedes Teammitglied einsehbar.

Desweiteren soll auf der späteren Drupalplattform selbst eine Hilfeseite eingerichtet werden mit Anleitungen und FAQs, damit auch technisch unversierte Benutzer das Portal bedienen können.

2 Testkonzept

Bei einem Projekt dieser Grösse ist es von Vorteil, so viel wie möglich automatisch testen zu lassen. Viele Entwicklungsumgebungen und Sprachen stellen Tests und Testumgebungen zur Verfügung, so auch Drupal. Diese automatischen Test berücksichtigen aber keine individuellen Fehler, so dass auch manuelle Tests von Zeit zu Zeit nötig sind. Das Testkonzept unterteilt sich in vier Testabschnitte.

2.1 Komponententests

Als Komponententest bezeichnet man Testverfahren, die nur einzelne Bereiche der Software auf ihre Funktionalität prüfen, ohne Berücksichtigung der Kompatibilität mit anderen Modulen. Für unsere Komponententests benutzen wir das integrierte Drupalmodul Testing. Komponententests haben für uns keine grosse Bedeutung, da wir vermutlich vor allem mit vorgefertigten Modulen arbeiten werden, die bereits getestet sind.

2.2 Integrationstests

Integrationstest prüfen das Zusammenspiel der einzelnen Komponenten, d.h. ob die einzelnen Drupalmodule und die Datenbank harmonieren und richtig dargestellt werden. Für unsere Integrationstests benutzen wir das integrierte Drupalmodul Testing.

2.3 Systemtests

Der Unterschied zu den Integrationstest ist, dass nun auch die Gesamtfunktionalität geprüft wird. Ausserdem wird getestet, ob das Produkt innerhalb der späteren Nutzungsumgebung funktioniert.

Für immer gleiche Anfragen benutzen wir das Tool „Selenium“. Dieses automatisiert Browseranfragen und gestattet auch Stresstest, da beliebig viele Anfragen auf einmal gestellt werden können.

2.4 Abnahmetest

Das Produkt wird dem Auftraggeber vorgeführt und auf Vollständigkeit geprüft. Bei Zufriedenheit des Kunden, kann das Produkt übergeben und abgeschlossen werden.

3 Organisatorisches

Für eine erfolgreiche Zusammenarbeit trifft sich das Team zweimal in der Woche. Dadurch können Fragen schnell geklärt werden und jedes Teammitglied weiss, wie der aktuelle Stand der Dinge ist. Zusätzlich können und sollten sich bei Bedarf Arbeitsgruppen bilden, die konkrete Aufgaben erledigen und sich u. U. separat treffen. Die Ergebnisse dieser Treffen werden beim nächsten Teammeeting vorgestellt. Weiterhin erfolgt die Kommunikation bei teamrelevanten Themen über das OLAT-Forum, sowie bei arbeitsgruppenspezifischen Absprachen über andere private Kanäle. Für gemeinsames Programmieren haben wir ein Git-Repository eingerichtet. Jede Änderung ist dort zu beschreiben. Dabei ist sich an die vorab genannten Standards zu halten, um unnötige Umformatierungsänderungen zu vermeiden.

4 Quellen

<https://www.drupal.org/coding-standards>

<http://www.seleniumhq.org/>

<http://www.phpdoc.org/>