Richtlinie zum „Software-Sprint“

frab – Erweiterung

Schlussbericht

Zuwendungsempfänger:

Mario Manno

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS17S07 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

frab ist ein web basiertes Konferenz Planungswerkzeug. Es hilft dabei Vortrags-Einreichungen zu sammeln, zu bewerten und schließlich anzunehmen. Es organisiert die Kommunikation mit den Beitragenden und unterstützt bei der Erstellung sowie Veröffentlichung des Programmplans. Es werden mehrere Teams und Konferenzen pro Installation unterstützt.

Das Hauptziel des Projektes war der Erhalt der Zukunftsfähigkeit der Software. Dies sollte durch ein Update des Frameworks und der Bibliotheken erreicht werden. Darüber hinaus war geplant zentrale Bereiche der Software zu überarbeiten. Um die Fehlerrate in der zukünftigen Entwicklung zu senken, sollte die automatische Testbarkeit und die Testabdeckung der Anwendung verbessert werden.

Zwei weitere Bereiche, die für viele Probleme gesorgt hatten, sollten verbessert werden: Der erste, die Darstellung von vielen Räumen im Programmplan, hatte immer wieder zu inkompatiblen Einzellösungen auf Quellcode-Ebene von einzelnen Konferenzen geführt, die dann aufwendig bei jedem Update angepasst werden mussten. Der zweite betraf die Struktur der Anwendung. Statt von einer gemeinsamen Übersichtsseite wurde immer von der Existenz einer Konferenz ausgegangen, aber nach einer Neuinstallation existiert keine Konferenz und in bestimmten Grenzfällen lässt sich nicht die richtige ermitteln. Daher war der Zugriff auf wichtige Funktionen wie die Benutzerregistrierung und das Bearbeiten von Einstellungen nur mit einer Konferenz möglich. Der Code war durchsetzt mit Prüfungen auf das Vorhandensein einer Konferenz, wodurch in der Praxis indirekt und direkt zahlreiche Fehler entstanden.

Um auch weiterhin den Nutzerkreis erweitern zu können, war zudem die Erstellung einer Dokumentation vorgesehen.

Daraus ergaben sich die folgenden Meilensteine:

* Rails 5 Framework Update
* Wartung: Rechte-Management, Programm Export, Ruby 2.3.3 Syntax, Tests
* Programmdarstellung bei Veranstaltungen mit vielen Räumen
* Übersichtsseite mit Link zum Einreichungs-Interface und zum Programm
* Benutzer-Dokumentation

Die Reihenfolge der Meilensteine ergab sich durch die Notwendigkeit, zuerst das Update auf die neue Framework Version durchführen zu müssen. Die Dokumentation sollte für die neue frab-Version erstellt werden und wurde daher als letztes eingeplant.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

frab hilft Akteuren der Zivilgesellschaft Konferenzveranstaltungen zu planen und durchzuführen. Zu den Benutzern gehören Konferenzen wie die FrOSCon für OpenSource Software und der Chaos Communication Congress des CCC. Da frab auf einem selbst betriebenen Server installiert werden kann, wird es von vielen kleinen Konferenzen benutzt. Es eignet sich, auch durch die Rechteverwaltung, besonders für Konferenzen mit einem Team von Freiwilligen und Community-Beteiligung.

Ausführliche Darstellung der Ergebnisse

Es wurden alle Projektziele erreicht. Die Änderungen sind veröffentlicht und wurden von der letzten FrOSCon verwendet. Die Dokumentation ist im Wiki auf github veröffentlicht.

Überraschend war, wie kompliziert die alte Rechteverwaltung implementiert war. Aufwendig wurden bisher verschiedene Datenbank-Objekte geprüft; dabei hingen fast alle Rechte an der Rolle des Benutzers in der Konferenz, und nicht an einzelnen Entitäten, wie z.B. einem Vortrag. Das alte Vorgehen wurde durch die verwendete Bibliothek suggeriert und erwies sich im Nachhinein als Designfehler. Nach dem Wechsel der Bibliothek ist der Bereich Rechteverwaltung deutlich lesbarer und deckt mehr Fälle ab als zuvor.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die vielen Gespräche über frab und das positive Feedback auf den Veranstaltungen haben mir gezeigt wie bedeutsam frab für Kulturschaffende ist. Obwohl frab von vielen Konferenzen eingesetzt wird, gibt es allerdings keine öffentliche, gehostete Lösung. Potentielle Benutzer berichten immer wieder von Problemen bei der Installation. Eine Idee wäre es einen Service aufzubauen, bei dem die Kosten für die initiale Erstellung der Service Lösung, die Betriebskosten und die damit einhergehende Wartung durch ein Geschäftsmodel oder einen Träger aufgefangen werden.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ich hatte die Hoffnung eine fertige JavaScript Bibliothek zu finden, die die Darstellung großer Tabellen erleichtert, etwa durch Scrollen und Ziehen mit der Maus. Leider konnte ich keine geeignete Bibliothek finden, die sich einfach integrieren lässt und dabei barrierefrei sowie ästhetisch ansprechend ist. Eine solche Funktionalität für alle gängigen

Browser und Mobile selbst zu entwickeln ist viel zu zeitaufwendig im Rahmen des Projektes. Ich habe mich daher für den Ansatz mit Buttons, die zu dem zweiten Tabellenteil springen, entschieden und zwei weitere Ansichten, ‘Timeline’ und ‘Booklet’ eingeführt, die es dem Benutzer erleichtern Veranstaltungen aufzufinden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

* Homepage http://frab.github.com/frab/
* Release Notizen https://github.com/frab/frab/releases
* Github https://github.com/frab/frab
* Wiki https://github.com/frab/frab/wiki
* Anleitung https://github.com/frab/frab/wiki/Manual

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ursprünglich wollte ich die Rechteverwaltung direkt nach dem Upgrade auf Rails 5 umstellen. Allerdings funktionierte der statische Programmexport nach dem Update nicht mehr und war nicht trivial zu reparieren. Ich musste die Umstellung des Exports daher vorziehen. Da die Reihenfolge keine Rolle spielte, entstanden aber keine Mehraufwände. Tatsächlich stellte sich heraus, dass sowohl der Meilenstein Programmexport wie auch die Rechteverwaltung enger mit dem Bereich Wartung verknüpft waren, als ich ursprünglich erwartet hatte.

Das Ausmaß der Schwierigkeiten bei der Umstellung der Rechteverwaltung war unerwartet und stellte einen deutlichen Mehraufwand dar. Die Überarbeitung des Programmplans war jedoch schneller beendet als erwartet. Der eingeplante Puffer hat daher ausgereicht um die Mehraufwände aufzufangen.

Zu Beginn des Projektes habe ich mich dagegen entschieden erst die Testabdeckung zu erhöhen. In die alte Version wären die Tests nur schwer einzubauen gewesen und hätten mit dem Update ebenfalls überarbeitet werden müssen. Da die Software nicht testgetrieben entwickelt wurde und die Testabdeckung nicht sehr hoch war, ist es aufwendig die Tests nachträglich zu entwickeln. Die so gesparte Zeit bringt man, wie Studien zeigen, leider nach Projektende für Fehleranalyse und Behebung auf. So kam es in der Dokumentationsphase immer wieder zu nicht unerwarteten Unterbrechungen durch Bug-Reports.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Wie bereits erwähnt, gab es Bug Reports auf Github. Außerdem fiel die FrOSCon Konferenz in die Projektphase, so dass ich auf deren Infrastruktur den ersten Testserver in Betrieb nehmen konnte. Die Live-Daten von der FrOSCon halfen beim Testen und bei der Korrektur der Datenmigrations-Skripte. Schließlich konnte ich die FrOSCon-Server auf die neue Version updaten und die so aufgedeckten, neu eingeführten Programmier-Fehler beheben. Ich hoffe in Zukunft mehr Fehler durch Feature-Tests abzufangen und weniger im Produktionsbetrieb testen zu müssen.

Das Gespräch über UX mit Ame Elliott hat gute Impulse gegeben und zu einigen kleineren Änderungen an den Buttons und der Navigationsleiste geführt. Durch die vorangegangen Erneuerungsarbeiten, z.B. die Übersichtsseite und die damit verbundenen Layoutverbesserungen waren diese Änderungen schnell umsetzbar.