Richtlinie zum „Software-Sprint“

Akronym – foodsharing

Schlussbericht

Zuwendungsempfänger:

Raphael Wintrich

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS17S13 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Die Online-Plattform foodsharing.de ist ein sehr Umfangreiches Software-Projekt, welches mit den Jahren immer mehr gewachsen ist. Da zu Begin keine Langfristige Planung für dieses Projekt vorhanden war, war der Quellcode sehr unstrukturiert. Es wurden mit der Zeit immer weitere Software-Bausteine hinzugefügt, ohne von vornherein festsetzen zu können welche Funktionen genau benötigt werden. Das Projekt umfasste zu Begin der Förderphase knapp eine halbe Millionen Zeilen Quellcode, vor allem geschrieben in der Scriptsprache PHP. Es Lag dem Projekt kein Framework zugrunde, weder im PHP Backend noch im Frontend.

Alle Mysql Abfragen werden Roh an den Server gesendet und mit den bereits über 300.000 angemeldeted Benutzern waren modifikationen an der Datenbank ohne ORM sehr Aufwendig.

Durch den von mir geschriebene und ins Unvorhersehbare gewachsene Quellcode war der Einstieg für freiwillige Entwickler sehr schwierig, da es immer eine lange Einarbeitungszeit mit sich gebracht hat. Außerdem wurde der Code noch nicht veröffentlicht, da mit erheblichen Sicherheitslücken zu rechnen war und die Sensiblen Daten der Benutzer geschützt bleiben sollen.

Der Größte Meilenstein war also zunächst alle auffindbaren Sicherheitslücken zu beheben und den Quellcode unter einer Open-Source Lizenz zu veröffentlichen (MIT)

Um den Einstieg für freiwillige Entwickler weiter zu vereinfachen sollte der Quellcode „Aufgeräumt“ d.h. in sich neu Strukturiert, Modular aufgebaut und in einzelne Komponenten unterteilt werden.

Außerdem sollte für verbesserte Wartbarkeit des Projekt die Datenbankbefehle über ein ORM gesteuert werden um Änderungen leichter zu implementieren.

Zuletzt war es wichtig alle hard eincodierten Strings in Sprachvariablen und flat files zu extrahieren um eine Mehrsprachigkeit zu ermöglichen.

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

Letzendlich Profitieren alle Nutzer der foodsharing.de Plattform bzw. der dahinter stehenden Initiative gegen Lebensmittelverschwendung. Alle Menschen die gewillt sind einen Beitrag gegen unnötige Lebensmittelverschwendung beizutragen, haben die Möglichkeit schnell und unkompliziert in ein Netzwerk aufgenommen werden zu können und sich zu engagieren.

Neben dem wachsenden Netzwerk in deutschland von bisher ca. 300.000 privat engagierten und ca. 35.000 ehrenamtlichen die auch in Unternehmen Lebensmittel vor der Vernichtung bewahren, profitieren auch vor allem Menschen aus EU-Ländern von der nun Quelloffenen Software. Bereits zu Beginn hat sich Das Netzwerk nach England, Frankreich, Spanien, Italien, Österreich und in die Schweiz ausgebreitet. Bis auf die 2 letzteren konnte die Software in den Ländern aber noch nicht genutzt werden, da es nicht die Möglichkeit zur Übersetzung gab.

Das Projekt passt zum Themenfeld Civic-Tech, da es sich direkt an die Bevölkerung richtet und Handlungsräume ermöglicht.

Ausführliche Darstellung der Ergebnisse

Es konnten alle großen Meilensteine erreicht werden.

Einzelne Elemente wurden abstrahiert und somit auch erhebliche Mengen an Quellcode reduziert. Bibliotheken wurden in Composer Abhängigkeiten ausgelagert. Somit wurde allein der Quellcode von 500.000 Zeilen auf ca. 300.000 reduziert und eine klare Modulstruktur erkennbar gemacht. Folgende Module wurden extrahiert:

basket

Modul zur verwaltung von sog. Essenskörben, Nutzer aller Benutzerlevel können privat Essen anbieten und somit Essenskörbe anlegen, die anderer Benutzer findern, bz.w darüber informiert werden.

bcard

Bestimmte Benutzer sind berechtigt andere Benutzer für das Abholen von Lebensmitteln in Unternehmen freizuschalten. Dazu werden Ausweise generiert, mit denen sich per QR-Code ausgewiesen werden kann.

bell

Über ein Nachrichten System, werden Benutzer auf wichtige Ereignisse hingewiesen, dieses Modul ist so abstrahiert, das aus jeden anderen Modul Nachrichen in verschiedenen wichtigkeitsstufen an die Benutzer gesendet werden können. Der Benutzer wird über eine Info-Meldung direkt darauf Aufmerksam.

betrieb

Modul zu verwaltung von Unternehmen/Organisation die Lebensmittelüberschüsse anbieten. Bestimmte Benutzergruppen sind zur verwaltung berechtigt, andere wiederum können sich für die Teilnahme in Teams für die Unternehmen bewerben. Abholungen werden über Time-Slots organisiert in die sich Teammitglieder selbst eintragen

bezirk

Foodsharing Gruppen organisieren sich dezentral in regionalen Gruppen. Das Bezirks Modul organisiert diese Regionen in einer Hierarchie. Bestimmte Benutzergruppen, können diese Regionen verwalten.

blog

Einfaches Blog Modul zur veröffentlichung von Neuigkeiten. Bestimmte Benutzergruppen haben zugriff auf die veröffentlichung.

buddy

Jeder Nutzer hat eine privat geführe Liste an Kontakten auf die er über Tools, gezielt und schnell zugreifen kann.

content

Abstraktes Modul zur Verwaltung von Inhalten in verschiedenen Sprachen. Z.b. Statische Seiten Info-Boxen etc.

dashboard

Auf der persönlichen Übersichtsseite eines Jeden Benutzer werden alle ihm zugeordneten Neuigkeiten chronologisch angeordnet aufgelistet. Der Benutzer kann bestimmen welche Inhalte angezeigt werden sollen.

event

Ein Kalender-Modul worüber Regions bezogene oder öffentliche Veranstaltung verwaltet werden. Gruppen und Benutzer können zu den Events je nach einstellung eingeladen werden.

fairteiler

Das Fairteiler Modul organisiert Stellen zu denen Menschen gehen können um Lebensmittel abzugeben bzw. abzuholen. Diese werden intern sowie auf der öffentlichen Karte angezeigt. Benutzer können angaben zu hinterlegten

Lebensmitteln machen, diese Infomationen gehen über verschiedene Kanäle an Benutzer die dem Fairteiler folgen.

groups

Benutzer organisieren sich regional bezogen zu den Regionen & überregional in von bestimmten Benutzergruppen verwaltbaren Gruppen.

mailbox

Bestimmte Benutzergruppen, die auch persönlichen Kontakt zu Organisationen pflegen, bekommen foodsharing spezifische E-Mail Adressen. Die E-Mail können in einem sehr vereinfachtem E-Mail Client verwaltet werden. Alle gängigen E-Mail Funktionen sind vorhanden. (HTML-Mails, Anhänge, mehrere Empfänger..)

technisch werden die E-Mails über ein catchall E-Mail Postfach in die einzelnen Postfächer der Benutzer in die MySQL Datenbank übertragen.

map

Abstraktes Modul zur Darstellung von Inhalten auf Open-Street-Map Karten.

msg

Das Message-Modul ermöglicht chatten unter den Benutzern. In einer Unterhaltung können beliebig viele Benutzer beteiligt sein. Das bisher einzige Modul welches über websockets, Nachrichten direkt über den Browser empfängt.

profile

Jeder Benutzer erhält ein Profil, mit den wichtigsten Informationen zur Person und Tätigkeiten im foodsharing-Netzwerk, worüber auch andere Benutzer ihn/sie wiederum kontaktieren können

quiz

Diese Modul ermöglicht die Verwaltung von multiple choice Fragebögen, die in einem Quiz mit einstellbarem Zeit-Limit als vorraussetzung für bestimmte Nutzerrollen abgefragt werden.

search

Eine Suchfunktion für mit dem Benutzer verknüpfte Inhalte. Ergebnisse werden relevanzbezogen ausgegeben.

settings

Modul zur Verwaltung von Benutzerbezogenen Einstellungen.

statistic

Seitenstatistiken werden in diesem Modul per cronjob täglich zusammengefasst und in der Datenbank angelegt.

wallpost

Ein abstrahiertes Modul welches eine Nachrichten Pinnwand zu jedem beliebigen Modul hinzufügen kann. Wird z.Z. benutzt von Benutzer-Profilseiten, Betriebs-Seiten, Fair-Teiler-Seiten und Essenskorb-Seiten.

Als zukunftsfähiges Framework kommt nun Laravel zum Einsatz. Es wurde ein Wrapper Modul geschrieben, sowie ein Repository-Layer für Daten.

Somit können innerhalb des „alten“ Quellcodes auf Laravel Model-Klassen zugegriffen werden um Datenbankmodifikationen zu managen, kommt somit auch das Laravel eigene ORM Eloquent zum Einsatz.

Für die komplette Datenbankstruktur sind nun Datenbank-Migrationen erstellt, die ein sehr einfachen Installationsprozess ermöglichen. Außerdem werden zur Installation Fake-Datensätze mitgeliefert um ein sofortiges Testen-Der Software ermöglichen.

Für alle Module werden Text-Strings in Flat-Files extrahiert und über die Laravel eigenen Sprach funktionen abgelegt. Es gibt nun ein Administratives Backend, worüber Texte übersetzt werden können.

Mit der überarbeitung von 981 SQL abfragen wurde SQL Injektion ausgeschlossen. Durch eine vereinheitlichen von an dem Browser gesendeten Javascript Komponenten wurde das einschleusen von fremd Javascript verhindert.

Die extrahierung und Überarbeitung aller Benutzerausgaben sind in 238 Mögliche Schwachstellen behoben worden.

Als nun abgeleitetes Laravel-Projekt wurde somit auch die MIT-Lizensierung übernommen.

Ich habe viele neue Erkenntnisse darüber gewonnen, welchen großen Nutzen es hat Software-Prjekte von Anfang an Framework gestützt zu realisieren. Da dies bei foodsharing nicht der Fall war, musste ich besonders „tief“ in das Laravel Framework einsteigen um meine Vorhaben auch sauber umzusetzen.

Auch war mir vorher nicht bewusst wie wichtig es bei großen Projekten sein kann Daten Logik von Ablauf-Logik zu trennen um mit den gleichen Datenabfragen jede neue Software-Version umzusetzen. Mit dem Projekt ist nun der Grundstein gesetzt und es werden nach und nach alle Daten-Abfragen in das Repository Layer übernommen, womit dann auch schnell die API-Schnittstellen ausgebaut werden können.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Schon vor Beginn der Projektphase erreichten uns viele Anfragen aus anderen Ländern. Dank der Sprach-Funktionen sind die Übersetzungen der Sprach-Dateien bereits in vollem Gange und weitere Länder dürfen sich bald darüber freuen auch foodsharing einsetzen zu können.

Dank der Veröffentlichung und der nun klaren Struktur in der Software ist es leichter neue Entwickler zu finden die mich in der Weiterentwicklung

unterstützen. Außerdem ist jetzt der Grundstein gesetzt um neue Features schneller und einfacher umzusetzen.

Mit dem auslagern der Datenlogik, ist die Weiterentwicklung von API-Schnittstellen einfach Möglich. Die erste Demo einer Vue.js App wird bereits getestet (https://beta.light.foodsharing.de) Somit steigt auch die Motivation bald noch einmal die Entwicklung einer eigenständigen Android und iOS App voranzutreiben.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Da ich zu Anfang bereits mit dem Synfony-Framework sehr vertraut war fing ich auch an die Module in Symfony Komponenten zu extrahieren.

Ich bin aber immer wieder auf Schwierigkeiten gestoßen, die sich zwar Lösen ließen, aber schon mit erheblichem Aufwand und die Ergebnisse waren nicht in klarer vom Framework vorgegebener Struktur.

Außerdem musste ich für viele gewünschte Funktionalitäten auf externe AppBundles zurückgreifen.

Als ich mich dann Entschieden habe mir andere Frameworks anzuschauen und sehr schnell bei Laravel gelandet bin, war ich überrascht wie gut durchdacht genau die Komponenten für eine Moderne Web-Software in PHP bereits integriert war und erlebte in der Umsetzung meiner Meilensteine in Laravel einen Motivationsschub, da ohne ich ohne große Besonderheiten meine Vorhaben umsetzen konnte.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Homepage
https://foodsharing.de

Repo
https://github.com/raph-ael/foodsharing

foodsharing dev Gruppe
https://foodsharing.de/?page=groups

Slack Channel #foodsharing-dev
https://yunity.slack.com/

Blog
https://foodsharing.de/news

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ein deutlich größeren Fokus wie erwartet hat die Bearbeitung von Sicherheitsrelevanten Problemen verursacht. Somit hat sich auch die Veröffentlichung von altem Code sehr verzögert.

Auch wenn ich zum Schluss hin kurzzeitig Sorge hatte nicht mehr alle Meilensteine zu erreichen, bin ich Schlussendlich sehr zufrieden mit wie viel Zeitlicher Freiheit ich in diesem halben Jahr an dem Projekt arbeiten konnte und bin auch nach dieser Intensiven Zeit noch Motivierter ehrenamtlich die Entwicklung weiter zu betreuen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Eine ungeahnt große Aufgabe war es stets die Wünsche und Vorstellungen anderer Personen aus der Community in meine Entwicklung mit einzubeziehen.

Es war mir wichtig stetige Rücksprache (wöchentliche Telefon-Konferenzen) mit anderen potentiellen freiwilligen Entwicklern zu halten um Entscheidungen gut zu treffen und nicht am Ende für Enttäuschungen zu sorgen. Dennoch stagnierte zu manchen Zeiten durch Unsicherheiten bei der Entscheidung (z.B. Framework Ja/Nein, Welches Framework, Welche Entwicklungsumgebung etc.) die Entwicklung, die sonst vielleicht schneller gewesen wäre, aber so nun eine deutlich bessere Qualität aufweist.