Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Prof. Dr. Sascha Alda
Robert Hartmann

7. Juli 2020

Praktische Hausarbeit zur Vorlesung Software Engineering II Sommersemester 2020

(Bearbeitung und Abgabe bis 21.9.2020, 12 Uhr; Abgabe ausschließlich über LEA-Assignment auf LEA. Eine verspätete Abgabe ist unter *keinen* Umständen möglich!).

Motivation und Ziel

Das Unternehmen CarLook Ltd. möchte eine Suchplattform haben, mit der registrierte Endkunden nach Autos aus dem Fuhrpark des Unternehmens suchen können. Die prototypische Implementierung *muss* dabei für eine *optimale, sehr gute* Lösung auf Basis des Web-Frameworks Vaadin sowie auf der Grundlage der Methoden aus den Vorlesungen Software Engineering 1 und 2 erfolgen.

Nach Abstimmung mit dem Kunden kann für eine *erste befriedigende* Lösung anstelle einer prototypischen Implementierung auch ein Mock-Up eines Prototyps entwickelt werden, der eine Implementierung ersetzt.

Stakeholder

Ein IT-Berater von der Firma AllesLogo GmbH hat im Vorfeld eine Stakeholder-Analyse durchgeführt und hat folgende direkte Stakeholder identifiziert:

- Dr.-Ing. Urs Maier, Geschäftsführer der CarLook (O-Ton aus einem Interview: „*Ich stehe voll und ganz hinter dem Projekt und möchte es finanziell fördern! Ich kenne die Bedürfnisse der Kunden und kann ihre Anforderungen klar umfassen*“)

- Dipl. Kauffrau Anja Schmitt, Vertrieblerin der CarLook (O-Ton: „*Ich arbeite seit 10 Jahren als Vertrieblerin und habe eine klare Vorstellung über die funktionalen Anforderungen aus Sicht des Vertriebs*“)

- Herman Müller, leitender Software-Architekt der CarLook („*Bin noch unentschlossen, da ich bisher keine Informationen über das neue Projekt vorliegen habe. Technische Anforderungen kann ich ihnen aber gerne nennen.*“)

- Ingo von Stehlbach, Student und Praktikant mit einem Vertrag bis Ende August 2020 bei CarLook („*Das Projekt ist totaler Unsinn, brauchen wir nicht, es gibt doch schon die Plattform „mobile.de“. Eine innovative Anforderung aus meiner Sicht? User sollten sich einloggen können! Ein Interview gebe ich nicht!*“)

Funktionale Anforderungen

Der Inhaber des Unternehmens Dr.-ing. Maier gibt in einem weiteren Interview folgende funktionalen Anforderungen für die **Sicht des Endkunden** an (Notizen der IT-Beratung):

- „Natürlich sollte sich ein Endkunde einloggen können, also über ein Password sowie einer E-Mail-Adresse. Fehlerhandling? Naja, wenn der Endkunde nicht vorhanden ist, sollte eine Meldung schon kommen."

- „Der Endkunde sollte sich selber aktiv in der Plattform registrieren. Die Registrierungsdaten sollten *zukünftig* in unserem CRM-System OpenCRX abgelegt werden. Für den Prototyp in diesem Projekt reicht es aus, die Daten in einer Datenbank abzuspeichern. Fehlerhandling bei der Registrierung? Für den ersten Prototyp reicht mir die Angabe, ob eine E-Mail vorhanden ist oder nicht. Dazu sollte der Endkunde neben der E-Mail lediglich seinen Namen und sein Password spezifizieren, das reicht für die Registrierung für den ersten Prototyp aus, alles Weitere können wir in einer nächsten Version nachziehen."

- „Wenn eingeloggt, sollte der Endkunde in einem Suchfeld nach Autos *suchen* können, die dann in einer Tabelle oder in einem Dialog übersichtlich aufgelistet werden. Für eine erste Anzeige von Suchtreffern sollten zumindest die Angaben „Marke" (z.B. BMW), „Baujahr" (z.B. 2002), „Beschreibung" (kleiner Text wie „Gepflegtes Auto, unfallfrei mit vielen Extras") ausgegeben werden. Eine Suche nach dem Prinzip „On-The-Fly" wäre sehr schön, es reicht aber auch für einen ersten Prototyp eine konventionelle Suche mit einem Such-Button."

- Ein Endkunde kann ein Auto seiner Wahl reservieren. Ein Endkunde kann eine beliebige Anzahl von Autos reservieren. Ein Auto kann von einer beliebigen Anzahl von Kunden reserviert werden. Eine Reservierung dient für ein späteres Release der Software als Grundlage für eine Besichtigung. Die Organisation der Besichtigung muss *nicht* Bestandteil der Software sein.

- Ein Endkunde sollte seine Reservierungen z.B. in Form einer Tabelle stets einsehen können.

Die Vertrieblerin des Unternehmens hat in einem weiteren Interview die funktionalen Anforderungen für die **Sicht eines Vertrieblers** beschrieben:

- Login ist natürlich notwendig, gerne auch eine E-Mail sowie über ein Password. Beim Fehlerhandling können sie gerne an die weiteren Vorgaben von Herrn Dr.-Ing Maier orientieren.

- Nach Absprache mit unserem Software-Architekten sollten sich aus technischen Gründen Vertriebler wie ein Endkunde zunächst registrieren. Die Registrierungsdaten sollten *zukünftig* in unserem HR-System OrangeHRM verwaltet werden. Für den ersten Prototyp reicht eine Speicherung der Daten in einer Datenbank aus. Zum Fehlerhandling: Es muss gewährleistet werden, dass die E-Mail-Adresse eine gültige Adresse des Unternehmens ist (Beispiel: name@carlook.de). Der Vertriebler muss neben einer gültigen E-Mail lediglich seinen Namen und sein Password angeben. Alle weiteren Daten können wir in einer nächsten Version integrieren."

- Als eingeloggter Vertriebler möchte ich über die Plattform neue Autos eintragen können, die später von einem Endkunden bei einer Suche gefunden werden können. Für spätere Auswertungszwecke sollte ein neu eingetragenes Auto mit einem Vertriebler assoziiert werden. Ein Vertriebler kann eine beliebige Anzahl von Autos einstellen, ein Auto ist aber stets genau einem Vertriebler zugeordnet.

- Ein eingeloggter Vertriebler kann sich zu jeder Zeit seine eingestellten Autos in Form einer Tabelle (oder ähnliches) anzeigen lassen.

Technische Anforderungen

Herman Müller, leitender Software-Architekt der CarLook, gibt ihnen folgende technische Anforderungen mit auf dem Weg:

- Die prototypische Implementierung der Plattform *muss* auf Grundlage des Web-Frameworks Vaadin, *zumindest* in der Version 8 (oder höher), erfolgen.

- Eine Einbindung einer Datenbank z.B. für die Registrierung *sollte* erfolgen, wenn möglich auf dem Datenbanksystem PostgreSQL der Hochschule („Dumbo-Server"). Dazu können sie die Tabellen in einem eigenen Schema innerhalb der Datenbank ihres Semesterprojekts ablegen. Empfohlen ist aber die Verwendung einer *eigenen* Datenbank auf Dumbo (siehe dazu das Dokument „[SS20] Integration PostgreSQL"). Falls sie Probleme sehen mit der Einbindung, so können sie Beispieldaten auch statisch in ihrer Anwendung hinterlegen (z.B. in einer zugehörigen DAO). Für eine *sehr gute* Lösung sollte jedoch eine Datenbank vorhanden sein. Die Verwendung einer lokalen Datenbank auf ihrem Rechner ist *nicht* erlaubt. Falls sie zu keinem Semesterprojekt zugeordnet waren bzw. keinen Zugriff auf eine Datenbank haben, so schreiben sie eine E-Mail an Michael Welle (michael.welle@h-brs.de) er wird ihnen eine Datenbank anlegen.

- Die in den funktionalen Anforderungen genannten System-Akteure OpenCRX und OrangeHRM müssen in der Entwicklung nicht beachtet werden. Notwendige Registrierungsdaten aus den Systemen sollten in einer Datenbank verwaltet werden.

- Die Software-Architektur sollte angemessen dokumentiert werden. Folgende Artefakte sollten vorhanden sein: semantisches (konzeptuelles) ER-Modell, eine Kontextsicht der Software-Architektur, eine Bausteinsicht der Software-Architektur, ein objektorientiertes Analyse-Modell (OOA-Modell).

- Ihr Source Code der Anwendung sollte in dem Versionskontrollsystem Git bereitgestellt werden. Sie können dazu das Repository ihres GitLab-Projekts aus dem Semesterprojekt wiederverwenden. Dazu sollte sich jeder Student einen eigenen Branch innerhalb des Repository angelegen (z.B. developer_*name*). Die Empfehlung ist aber die Anlage eines eigenes Repository z.B. in GitHub oder in GitLab. Falls sie Probleme mit Git haben, so können sie ihren Source Code auch als ZIP-Datei auf LEA hochladen. Für eine *sehr gute* Lösung sollte jedoch ein Git-Repository vorhanden sein, mit einer sichtbaren Commit-Historie.

- Die Funktionalität Login soll durch einen automatisierten Akzeptanztest mittels Selenium Web Driver getestet werden. Falls sie Probleme mit Selenium haben, so reichen sie alternativ einen schriftlichen Akzeptanztest anhand der Vorlage aus der Vorlesung ein. Für eine *sehr gute* Lösung sollte aber ein automatisierter Akzeptanztest vorhanden sein.

- Ein Deployment in der SEPP-Umgebung (Jenkins, SonarQube, Test-Server) ist *nicht notwendig*. Auch ein Tracking der Entwicklung in JIRA ist *nicht* notwendig.

Qualitätsanforderungen

Die Vertrieblerin und der Geschäftsführer einigen sich auf eine gemeinsame Qualitätsanforderung zur Steigerung der Usability: „In der Plattform sollte mindestens eine bekannte Heuristik zur Gestaltung von Benutzeroberflächen integriert werden“.

Verzichten sie aber auf eine visuelle Gestaltung der Anwendung mit CSS / SCSS; der Fokus sollte auf die *rein funktionelle* Implementierung der Anforderungen liegen.

Weitere Anforderungen zur Dokumentation

Folgende Artefakte sollten in der Dokumentation enthalten sein:

- Ein UML-basiertes Use Case Diagramm zur Modellierung der funktionalen Anforderungen aus den Sichten des Vertrieblers sowie des Endkunden. Hier auch die externen System-Akteure berücksichtigen, die *zukünftig* verwendet werden sollen.

- Ein Utility Tree zur Konkretisierung der Qualitätsanforderung Usability. Dieser Tree sollte zwei Refinements mit jeweils einem Szenario erhalten. Für ein Szenario sollten sie eine zugehörige Architekturentscheidung gemäß der ATAM-Methode dokumentieren.

- Auf Grundlage der Interviews sollten sie ein Stakeholder-Portfolio entwickeln, wie in der Vorlesung eingeführt. Daneben sollten sie eine *neue* Stakeholder-Darstellung entwickeln, in der neben den bekannten Faktoren auch eine Einschätzung über die Innovationsbereitschaft eines Stakeholders angegeben wird. Leiten sie aus den bekannten Faktoren sowie aus dem neuen Faktor zudem eine Metrik ab, mit der sie einen skalaren Wert zur quantifizierbaren Beurteilung eines Stakeholders berechnen können. Führen sie eine exemplarische Berechnung für den Geschäftsführer sowie für den Praktikanten durch.

- Berechnen sie für eine View-Klasse ihrer Wahl folgende Metriken: WMC, DIT, NOC. Die Klasse sollte mindestens zwei Methoden und mindestens eine Kontrollstruktur (z.B. eine IF-Anweisung) besitzen. Die Metrik WMC sollten sie exakt herleiten und somit den Wert zu WMC nachvollziehbar begründen.

- Als Abgabe wird ausschließlich ein (in einer Zahl ausgedruckt: 1) PDF-Dokument akzeptiert, welches die Modelle, Texte usw. aus den Anforderungen und Hinweisen aus diesem Dokument umfasst. Ihr PDF-Dokument muss als erste Seite ein Deckblatt sowie eine zweite Seite gemäß der Vorlage (siehe LEA):

  „Vorlage_Deckblatt_Praktische_Hausarbeit.doc“

  beinhalten. Bitte diese einbinden oder exakt nachbilden! Die Abgabe erfolgt ausschließlich auf LEA (Assignment: „Upload für Praktische Hausarbeit“). Eine Abgabe als ausgedrucktes Dokument ist nicht erwünscht. Es gibt weder eine minimale noch eine maximale Anzahl von Seiten, die gefordert ist.

- Strukturieren sie das PDF-Dokument ansprechend; ein Inhaltsverzeichnis sollte vorhanden sein. Für eine Gliederung können sie sich an dem Vorschlag aus der Vorlage (vgl. S.3) gerne orientieren. Pro Gliederungspunkt sollten sie in wenigen Sätzen z.B. das Artefakt kurz beschreiben oder einleiten. Dennoch gilt es zu beachten, dass ihre Hausarbeit *nicht zu textlastig* wird.

Hinweise zur Implementierung

Falls sie das Software-Projekt alleine durchführen, dann reicht eine softwaretechnische Umsetzung *einer Sicht* (Endkunde *oder* Vertriebler) für eine sehr gute Lösung aus! Falls sie sich für die Sicht des Endkunden entscheiden, so können sie für die Suche Testdaten von Autos manuell z.B. in einer Datenbank anlegen.

Falls sie das Software-Projekt zu zweit durchführen, dann sollte eine softwaretechnische Umsetzung für beide Sichten (Endkunde *und* Vertriebler) erfolgen, falls sie eine *sehr gute* Lösung anstreben. Aus der Commit-Historie aus Git sollte ersichtlich sein, dass beide Studenten/-innen *ausgewogen* an dem Source Code gearbeitet haben. Stellen sie zu-

dem schriftlich in der Dokumentation klar hervor, wer welche Anteile der Software und der Dokumentation entwickelt hat.

Hinweise für ein Semesterprojekt *ohne* Implementierung

Erscheint ihnen persönlich die Implementierung einzelner Anwendungsfälle innerhalb einer Sicht (z.B. die Registrierung) als zu komplex oder zu zeitaufwändig, so können sie diese auch im Prototyp auslassen und als Alternative dafür Mock-Ups mit dem Tool Balsamiq (oder einem anderen Tool wie z.B. „draw.io“) abliefern. Für eine *sehr gute* Lösung sollte jedoch eine vollständige prototypische Implementierung einer Sicht bzw. beider Sichten auf Basis von Vaadin entwickelt werden.

Eine Bearbeitung der Hausarbeit *ohne* eine prototypische Implementierung auf Basis von Vaadin bzw. Java kann somit erfolgen! Bei dieser Variante würde ein vollständiger Mock-Up sowie ein Page Flow eine Implementierung ersetzen. Diese Variante ergibt aber, vorausgesetzt die anderen Bereiche (z.B. Modellierung Use Case Modell; Darstellung der Software-Architektur; Stakeholder-Analyse; Utility-Tree; Mock-Ups; schriftliche Akzeptanztests etc.) werden ansprechend bearbeitet, eine Beurteilung maximal im Bereich „*befriedigend*“.

Beratung

Für *dringende* Rückfragen (z.B. bei organisatorischen Fragen oder bei *gravierenden* Verständnisproblemen der Aufgabenstellungen) stehe ich ihnen persönlich während meiner Sprechstunde donnerstags von 12-14 Uhr im Juli und im September via Zoom bereit. Rückfragen können *vereinzelt* auch per E-Mail an mich (sascha.alda@h-brs.de) gestellt werden. Eine Beratung im August wird allgemein nicht möglich sein. Eine Vorab-Beurteilung der Leistung oder aber eine Vorstellung oder Diskussion einer pre-finalen Lösung wird ebenfalls nicht möglich sein.

Benotung

Die Benotung der Hausarbeit wird rund sechs Wochen in Anspruch nehmen. Auf vorherige Anfragen zu einer Benotung bitte ich in jedem Fall abzusehen. Jede Hausarbeit erhält ein Feedback-Dokument, in der die Note begründet wird. Eine Einsichtnahme wird es für die Hausarbeit *nicht* geben. Eine Rücksprache kann im Dezember 2020 zu folgenden Terminen erfolgen:

10. Dezember 2020, 14-18 Uhr (C 218 oder WebEx; wird noch bekannt gegeben)
11. Dezember 2020, 10-12 Uhr (C 218 oder WebEx; wird noch bekannt gegeben)

Eine vorherige oder spätere Rücksprache wird nicht möglich sein.