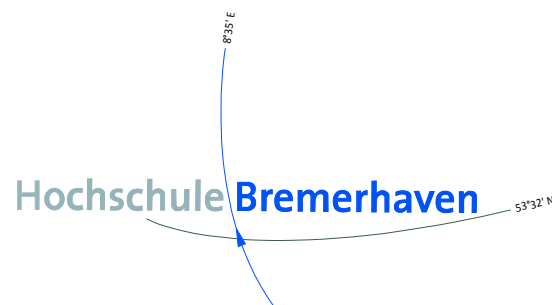


Testgetriebene Entwicklung und kontinuierliche Integration mit der SAP Mobile Platform

MAXIMILIAN AZIMI, JAN-HENRICH MATTFELD



BACHELORARBEIT

eingereicht im
Fachhochschul-Bachelorstudiengang

Wirtschaftsinformatik

in Bremerhaven

im März 2015

Erklärung

Wir erklären eidesstattlich, dass wir die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Bremerhaven, am 23. März 2015

Maximilian Azimi, Jan-Henrich Mattfeld

Inhaltsverzeichnis

Erklärung	i
Kurzfassung	iii
1 Einleitung	1
1.1 Einleitung	1
1.2 Problemstellung	1
1.3 Ziel	1
1.4 SAPs Open-Source-Initiative als Chance	2
1.5 Erkenntnisinteresse	3
2 Grundlagen	5
3 Anforderungsanalyse	6
4 Implementierung	7
5 Schlussfolgerungen	8
Quellenverzeichnis	9
Literatur	9
Online-Quellen	9

Kurzfassung

Mit aktuellen Tools und Frameworks wie SAPUI5, NW Gateway und SMP bietet die SAP neue Möglichkeiten zur Entwicklung von geräteübergreifenden mobilen Anwendungen. Diese wollen wir nutzen, um Individualsoftware der abat AG mobil nutzbar zu machen. Gleichzeitig sollen der Entwicklungs- und Auslieferungsprozess automatisiert und entsprechende Tools erprobt werden.

Kapitel 1

Einleitung

1.1 Einleitung

Die Ergebnisse entstehen im Rahmen unserer Bachelorarbeit im Fach Wirtschaftsinformatik an der Hochschule Bremerhaven, die voraussichtlich im Februar 2015 abgeschlossen sein wird. Vorausgegangen sind dieser u. a. die Zertifizierung als ISTQB Certified Tester Foundation Level und mehrere Projekte im SAP-Mobilbereich.

1.2 Problemstellung

Viele Projekte der abat AG arbeiten agil z. B. per Scrum. Hierzu ist ein umfangreiches Projektmanagement-Tool als ABAP-Eigenentwicklung vorhanden.

Dieses enthält allerdings weder ein Scrum-Board noch eine mobile Ansicht – schneller Zugriff auf wichtige Funktionen ist unterwegs unmöglich. Die Aufgaben können nur am PC mit Intranet-Zugang bearbeitet werden.

Aktueller Workflow: Ausdrucken der einzelnen Aufgaben, anpinnen, manuell verschieben und parallel per Scrum-Transaktion in das SAP-System übertragen. Dies gilt es mit aktuellen Technologien zu vereinfachen.

1.3 Ziel

Im Rahmen der Bachelorarbeit wird eine geräteübergreifende App entwickelt, die das Scrum-Board visualisiert und den Zugriff auf Projektdaten schneller und einfacher gestaltet. Während der Entwicklung sollen aktuelle Technologien und Tools zum Einsatz kommen. Kombiniert mit einem modernen Vorgehen werden die Themen Sicherheit und Zuverlässigkeit besonders betrachtet.

In Zukunft sollen die Projektaufgaben mit Zusatzinformationen auf einem Smartphone oder Tablet angezeigt und bearbeitet werden. Ein typischer

Vorgang in dieser App ist beispielsweise die Statusänderung von Aufgaben – Diese kann per Drag and Drop deutlich schneller erledigt werden.

Der bedeutendste Vorteil ergibt sich aus der ständigen Verfügbarkeit des Projektstatus: Das Scrum-Board muss nicht mehr physisch vorhanden sein, ein Blick in die App genügt. Der umständliche Zugriff über die alte, sehr umfangreiche SAP-Transaktion ist nur noch selten notwendig.

1.4 SAPs Open-Source-Initiative als Chance

Historische Entwicklungen auf Basis der SAP-eigenen Programmiersprache ABAP (Advanced Business Application Programming) lassen sich nur schwer in einem CI-Prozess automatisieren: Die entsprechenden Werkzeuge z. B. zum ABAP-Unit-Test [4] oder zur Testfallerstellung (eCATT) liegen vor, lassen sich aber nur schwer einbinden. Eine weitere Rolle spielt die grundsätzliche ABAP-Entwicklung im System selbst – Eine CI-Interaktion von außen gestaltet sich schwierig.

Deutlich mehr Möglichkeiten ergeben sich bei SAP-Entwicklungen auf Java-Basis: Hier steht die SAP NetWeaver Development Infrastructure (NWDI) zur Verfügung [3]. Alle CI-relevanten Tasks sind vorhanden und lassen sich automatisieren. NWDI ist allerdings proprietär und eignet sich nicht zur Entwicklung mit anderen Plattformen als Java EE.

Eine komplette Kehrtwende ergibt sich nun nach der Einführung von SAPUI5 als neue SAP-Mobilplattform: Sie ist auch ohne SAP-Backend lauffähig und basiert auf dem weit verbreiteten JavaScript-Framework jQuery [1].

Zusätzlich sind große Teile des Quellcodes unter dem Namen OpenUI5 auf github veröffentlicht worden – inklusive Buildkonfiguration und ausführlicher Dokumentation [5]. Sie geben einen interessanten Einblick in den SAP-internen Workflow.

In der Bachelorarbeit wird untersucht, welche Teile für eigene Projekte übernommen werden können. Wo sind größere Anpassungen und Erweiterungen notwendig? Testbarkeit spielte bei der Entwicklung des Frameworks offenbar eine größere Rolle als früher: Frei verfügbar sind bereits der QUnit-Aufsatz OPA5 (One-Page Acceptance tests for UI5) und ein MockServer zur Emulation von OData-Services [2].

Durch folgende Aspekte ergeben sich gute Voraussetzungen für eine SAPUI5-CI-Toolchain mit Jenkins (siehe Angepasste CI-Toolchain.):

1. Bewährte Basistechnologien
2. Open-Source-Vorstoß
3. Testorientierung
4. Wachsende Community

1.5 Erkenntnisinteresse

Besonders hervorzuheben ist die Kombination der verschiedenen Aspekte und Vorgehen:

1. Entwicklung einer aktuellen SAPUI5-App für ein bereits vorhandenes Altsystem auf ABAP-Basis.
2. Die Integration des neuen NetWeaver Gateways und der entsprechenden OData-Services.
3. Nutzung des Frameworks für Logon- und Offline-Funktionen
4. Erstellung von Testfällen anhand der Spezifikation.
5. Zuverlässigkeit vorhandener Features nach Updates durch automatische Regressionstests.
6. Aufbau der Open-Source-CI-Toolchain für eine SAP-UI5-Entwicklung
7. Kontinuierliche Bereitstellung neuer App-Versionen für verschiedene Gerätetypen.

Für alle folgenden Projekte werden diese Aspekte essentiell sein: Es gilt eine entsprechende Toolchain zu erproben und zu etablieren, um Softwarequalität und Erfüllung der Spezifikation nachhaltig zu gewährleisten.

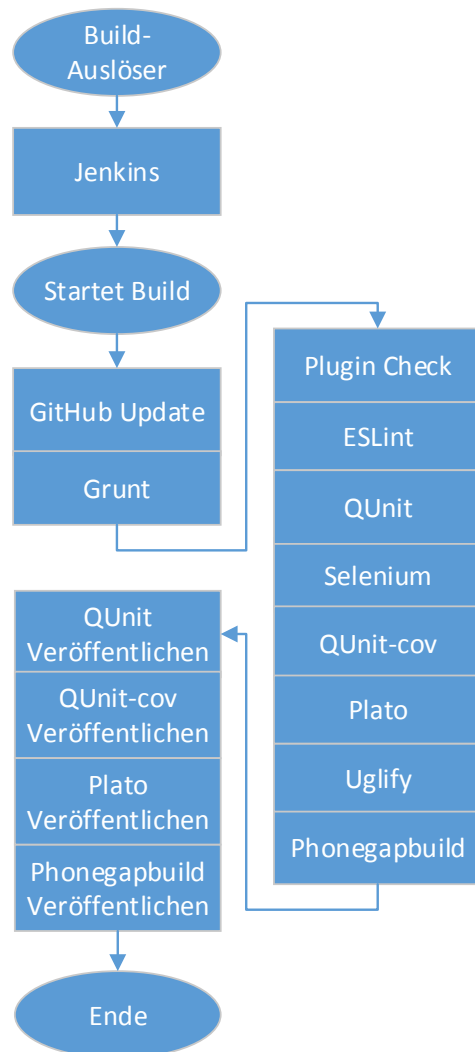


Abbildung 1.1: Angepasste CI-Toolchain.

Kapitel 2

Grundlagen

Kapitel 3

Anforderungsanalyse

Kapitel 4

Implementierung

Kapitel 5

Schlussfolgerungen

Quellenverzeichnis

Literatur

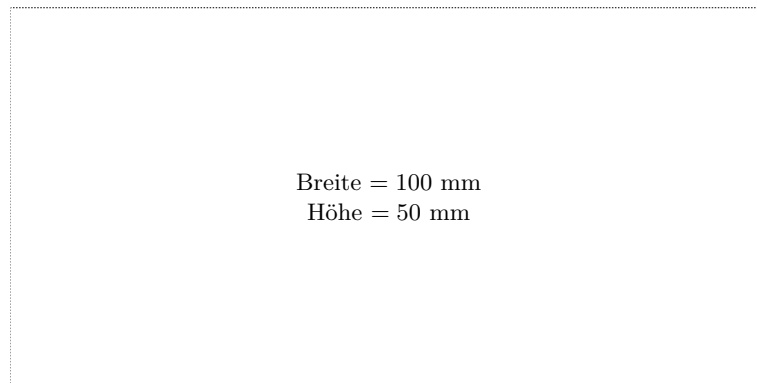
- [1] Miroslav Antolovic. *Einführung in SAPUI5*. 1. Auflage. SAP PRESS, März 2014 (siehe S. 2).
- [2] Carsten Bönner u. a. *OData und SAP Gateway*. 1. Auflage. SAP PRESS, Juni 2014 (siehe S. 2).
- [3] Marc Chan. *Installing and Configuring SAP NetWeaver Gateway 2.0*. 2. Auflage. SAP AG, Sep. 2011 (siehe S. 2).
- [4] Damir Majer. *Unit-Tests mit ABAP Unit*. 1. Auflage. dpunkt.verlag GmbH, 2009 (siehe S. 2).

Online-Quellen

- [5] SAP SE. *Development Conventions and Guidelines*. Version 68f22b0. Nov. 2014. URL: <https://github.com/SAP/openui5/blob/master/docs/guidelines.md> (besucht am 23.11.2014) (siehe S. 2).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —