

**Abschlussbericht der Projektgruppe
Cloud-basierte analytische Informationssysteme**

PG. CAIS

vorgelegt von

Jannick Asmuß
Dörthe Euhäusen
Phillip Kramer
Marcel Kuchler
Nils Lutz
Carsten Manske
Dennis Vorreiter

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	3
1.2	Lösungsansatz	3
1.3	Aufbau der Dokumentation	4
2	Organisation	6
2.1	Projektgruppe	6
2.2	Internetauftritt	6
2.3	Ressourcen	7
2.4	Rahmenbedingung und Einstieg	8
2.5	Grundlagen über Scrum	9
2.5.1	Einführung	9
2.5.2	Rollen	10
2.5.3	Artefakte	10
2.5.4	Ereignisse	11
2.6	Einstieg in Scrum	12
2.6.1	Rollenverteilung	12
2.6.2	Konzeption der Ereignisse	12
2.6.3	Definition der Artefakte	13
2.7	Anpassung von Scrum	14
3	Cloud Computing	16
3.1	Grundlagen	16
3.1.1	Public Cloud	18
3.1.2	Private Cloud	18
3.1.3	Hybride Cloud	19
3.2	X-as-a-Service	20
3.2.1	Infrastructure-as-a-Service	21
3.2.2	Platform-as-a-Service	21
3.2.3	Software-as-a-Service	22

4	Analytische Informationssysteme	23
4.1	Definition	23
4.2	In-Memory Datenbanken	24
4.3	Konzepte und Algorithmen	24
4.3.1	Extract, Transform, Load	24
4.3.2	Data Mining	24
5	Analyse der Anforderungen	26
5.1	Geschäftsprozess	27
5.2	Anwendungsfälle	29
5.3	Funktionale Anforderungen	30
5.4	Nicht-funktionale Anforderungen	32
6	Konzeption	33
6.1	Fachliches Konzept	33
6.1.1	Skalierungsalgorithmus	33
6.1.2	Billing-Modell	33
6.2	Technisches Konzept	34
6.2.1	Systemarchitektur	34
6.2.2	Datenmodell	37
6.2.3	SAPUI5 / OpenUI5 Toolkit	37
6.2.4	UI-Mockups	37
7	Implementierung des Prototypen	40
7.1	Entwicklungsumgebung	40
7.2	Infrastruktur	40
7.2.1	Cloud	40
7.2.2	Microservices	41
7.3	Backend-Implementierung	43
7.3.1	Architektur	43
7.3.2	Technologien	44
7.3.3	Prototyp	47
7.4	Frontend-Implementierung	50
7.4.1	Model-View-Controller-Architekturmuster	50
7.4.2	Coding-Beispiel	51
7.5	Testing	54
7.5.1	Allgemeines Test-Vorgehen	54
7.5.2	Unit-Tests	54
7.5.3	Systemtest	57
7.5.4	Backend-Testing	59

8 Fazit und Ausblick	64
8.1 Lessons Learned	64
8.2 Mögliche Ausbaustufen	67
Literaturverzeichnis	vii
Tabellenverzeichnis	viii
Abbildungsverzeichnis	x
Anhang	xi
Seminararbeiten	xi
P. Kramer - SCRUM	xi
J. Asmuß - Architekturmanagement	xxiii
D. Euhausen - Modellierung	xxxiii
D. Vorreiter - Business Intelligence	xliv
M. Kuchler - Software-as-a-Service	liv
C. Manske - Platform-as-a-Service	lxiv
N. Lutz - Infrastructure-as-a-Service	lxxiv
Lastenheft	lxxxv
Paper Skalierungsalgorithmen	cxiii
Protokolle	cxviii
Benutzerhandbuch	clxviii
Testdokumentation	clxxii

1. Einleitung

Die aus der vergangenen Wirtschafts- und Finanzkrisen vorhandenen hervorgegangenen Charakteristika, dazu zählen Unsicherheiten, Risiken und unvollständige Informationen, prägen zunehmend die heutige industrialisierte und globalisierte Geschäftswelt und gelten als Normalzustand. Diese Geschäftswelt fordert von Unternehmen mehr Anstrengungen zur Sicherstellung der Wettbewerbsfähigkeit. Zwangsläufig müssen Unternehmen früher oder später zunehmend agiler und schneller auf Trends und Veränderungen reagieren. Zudem muss die heutige Generation von Managern operativer arbeiten beziehungsweise agieren und zunehmend schnellere Entscheidungen treffen. [NJS15]

Der Einsatz von flexiblen, proaktiven Systemen, die zur Entscheidungsunterstützung herangezogen werden, sollen hier Abhilfe schaffen. Merkmale solcher Systeme sind eine hohe Abfrage- und Analyseleistung, ständige Erreichbarkeit auch bei einem hohen Anfrageaufkommen sowie der Umgang mit der immer größer und komplexer werdenden Datengrundlage. [NA14]

Aufgrund dieser Entwicklungen kann die Aussage getroffen werden, dass die Anforderungen an analytische Informationssysteme und speziell an Reporting-Werkzeuge steigen, denn die traditionellen und häufig monolithischen Business Intelligence-Lösungen erfüllen die erhöhten Anforderungen insbesondere im Hinblick auf Flexibilität und Skalierbarkeit nur unzureichend. Das Cloud Computing kann hier helfen, diese Anforderungen zu erfüllen, indem Business Intelligence und Cloud Computing kombiniert werden. BI in der Cloud ist keine grundlegend neue Technologie. Es gibt Vorläufer und Wurzeln in beiden Domänen. BI in der Cloud bietet neue Technologie-Kombinationen, um individuelle und differenzierte konfigurierbare, skalierbare und flexible analytische Dienstleistungen zur Verfügung zu stellen. [NS16]

Werkzeughersteller haben bereits reagiert und bieten Produkte in diesem Bereich an. Somit sind erste Produkte verschiedener Hersteller, unter anderem SAP, Microsoft, IBM und Oracle, bereits auf dem Markt. Allerdings sind diese doch sehr unterschiedlich, lassen sich nicht beliebig kombinieren oder in vorhandene Systeme integrieren. [NJS15]

Im Rahmen des Projekts Cloud-basierte analytische Informationssysteme (CAIS) soll ein solches System, also eine Kombination aus Business Intelligence und Cloud Computing, realisiert werden, das einen konkreten Anwendungsfall aus der Energiewirtschaft bedient.

Somit umfasst das zugrunde liegende Projekt dieser Abschlussdokumentation eine Business Intelligence-Anwendung, die in ihrer Gesamtheit mittels einer cloud-basierten Plattform realisiert werden soll.

Diese Anwendung sorgt für die Verarbeitung von Kennzahlen zur Gewinnanalyse von Energieverträgen, mit dem Ziel, den bestehenden Vorgang der Vertragsverhandlungen zwischen Energiekonzern und Kunde zu vereinfachen. Das Projekt vereint dafür die analytischen Vorteile einer Business Intelligence-Architektur mit der skalierbaren und dynamischen Nutzung von Cloud-Technologien. Dies umfasst auf Seiten des Business Intelligence-Bereichs sowohl eine intelligente Verarbeitung von fremden Quelldaten, ein intelligentes Analyseverfahren sowie eine angemessene und anpassbare Darstellung der Analysen. Auf Seiten der Cloud-Technologien bietet das System ein Skalierungsverfahren, um Ressourcen je nach Anfragelast verteilen zu können, ein Accounting-Management, um eine verbrauchsorientierte Abrechnung bieten zu können, sowie eine hohe Verfügbarkeit auf allen gängigen Endgeräten. Als Domänenreferenz und Ansprechpartner stellt sich Marco Haas von der Unternehmensberatung Ceyoniq zur Verfügung.

Das Projekt *Cloud-basierte analytische Informationssysteme (CAIS)* baut auf das Vorjahres-Projekt *Business Intelligence in the Cloud for Energy (BICE)* auf. Die Abgrenzung besteht darin, dass die Projektgruppe CAIS nur thematisch auf die Vorjahres-PG BICE aufbaut. Es liegen zwar ähnliche Anforderungen und Anwendungsfälle vor, jedoch ist die Erarbeitung einer von der Vorjahres-PG unabhängigen beziehungsweise nicht beeinflussten Lösung vorgesehen, weil die Verwendung anderer Technologien und somit auch das Aufsetzen auf einer eigenen Architektur im Vordergrund stehen sollen.

Die Umsetzung des Projektes soll helfen, eine genaue Vorstellung über die Vor- und Nachteile bei der Implementierung eines solchen Systems in ein Unternehmen zu erhalten. Im Rahmen des Projekts wird eine Machbarkeitsstudie zur Evaluierung einer Best-Practice-Lösung für das BI-Cloud-Umfeld durchgeführt. Betrachtungsgegenstand dieser Machbarkeitsstudie werden der Skalierungsalgorithmus und das Abrechnungsmodell sein. Hierbei erfolgt eine allgemeine Identifizierung von Algorithmen und weiterführend gezielt auf die Algorithmen, die dem Projekt zugrunde liegenden gestellten Anforderungen abdecken. Weiterhin erfolgt eine Prüfung anhand des im Rahmen der Projektgruppe entwickelten Prototypen, welche Abrechnungsmodelle für ein Business Intelligence as a Service (BlaaS) sinnvoll sind. So soll eine wissenschaftliche Grundlage für weitere Diskussionen geschaffen werden.

1.1 Problemstellung

Die Problemstellung des CAIS-Projekts resultiert aus Gesprächen mit dem Stakeholder (Kunde) und zugleich Projektansprechpartner Marco Haas von der Unternehmensberatung Ceyoniq sowie dem Projektgruppenverantwortlichen und zugleich Product Owner Oliver Norkus.

Das aktuell vorliegende Problem besteht darin, dass im Kontext der Vertragsverhandlungen zwischen Energiekonzern und Kunden, die Vertriebsmitarbeiter aktuell beim Kunden vor Ort auf keine BaaS-Lösung zurückgreifen können, die bei der Vertragsverhandlung der Unterstützung dienen würde. Der IST-Prozess ist so gestaltet, dass die Berechnung im eigenen Unternehmen durchgeführt wird, anschließend fährt der Vertriebsmitarbeiter mit einem Preisvorschlag zum Kunden. Hierbei ist zeitnahe Preisfindung nicht möglich, da beim Kunden vor Ort keine erneute Berechnung durchgeführt werden kann.

Die daraus resultierenden Nachteile sind zum einen, dass der Prozess nicht genutztes Potential hinsichtlich Abwicklung, Zeitersparnis und eine einhergehende Kostenersparnis birgt. Zum anderen ist diese Prozessgestaltung mit möglichen Wettbewerbsnachteilen verbunden, da die Konkurrenz gegebenenfalls besser aufgestellt ist.

1.2 Lösungsansatz

Mit den im Rahmen des Projekts erhobenen Anforderungen und vorliegenden Anwendungsfällen soll nachfolgend eine Kurzfassung des Lösungsansatzes präsentiert werden, der das zuvor beschriebene vorliegende Problem lösen und gleichzeitig als Zielsetzung fungieren soll:

- Entwicklung einer BaaS-Lösung
- Realisierung mittels einer Cloud-basierten Plattform
- Fokus liegt auf Verarbeitung von Kennzahlen zur Gewinnanalyse von Energieverträgen
- Vereinfachung des bestehenden Vorgangs, der Vertragsverhandlungen zwischen Energiekonzern und Kunden

Durch die Umsetzung des zuvor genannten Lösungsansatzes sollen für die Stakeholder folgende Mehrwerte generiert werden:

- Nutzer bekommt anschauliche Berichte über Kennzahlen geliefert
- Visualisierte Darstellung der Ergebnisse / Kennzahlen mittels Diagrammen

- Vertriebsmitarbeiter können während eines Kundenmeetings mit ihren mobilen Endgeräten zugreifen
- Optimierung des Geschäftsprozesses der Vertragsverhandlung
 - Echtzeitabrechnung von Verbrauchswerten
 - Gewinn- und Verlustrechnung mit Prüfung des Gewinns
- Nahezu Echtzeitberechnung durch Einsatz von In-Memory-Technologien

1.3 Aufbau der Dokumentation

Das vorliegende Abschlussdokument stellt die finale Projektdokumentation des CAIS-Projekts dar und ist auf oberster Ebene gegliedert in die folgenden Kapitel:

- Kapitel 2 behandelt projektbezogene, organisatorische Themen der PG CAIS. Zunächst wird die Projektgruppe vorgestellt und die organisatorische Einteilung der Mitglieder erklärt. Im Anschluss wird der Internetauftritt der PG CAIS behandelt, die im PG-Umfeld verwendeten Projektmanagement- und Entwicklertools beziehungsweise eingesetzten Ressourcen vorgestellt und zum Schluss die Organisation des Projekts, in der das geführte Projektmanagement rückwirkend vom Start bis zum Abschluss des Projekts näher betrachtet wird.
- Das Kapitel 3 widmet sich der Thematik des Cloud Computing. Es werden die theoretischen Grundlagen des Cloud Computings erörtert und das *Everything-as-a-Service-Paradigma* wird Betrachtungsgegenstand dieses Kapitels sein.
- In Kapitel 4 wird die Thematik der analytischen Informationssysteme, die neben dem Cloud Computing für das weitere Verständnis der Arbeit wichtig sind, betrachtet. Zum Einstieg wird eine Definition erfolgen, anschließend die InMemory-Datenbanken näher betrachtet und abschließend einen Überblick über die im Rahmen des Projekts verwendeten Algorithmen gegeben.
- Das Kapitel 5 behandelt sämtliche im Rahmen des Projekts vorhandenen Geschäftsprozesse und Anwendungsfälle sowie aufgenommenen Anforderungen. Zunächst wird ein Überblick über die Vorgehensweise der Anforderungserhebung gegeben und der Geschäftsprozess beschrieben. Anschließend werden die Anwendungsfälle Betrachtungsgegenstand dieses Kapitels sein. Abschließend werden die aufgenommenen Anforderungen, unterteilt in funktionale und nicht-funktionale, erläutert.
- Kapitel 6, das sich den bisherigen Informationen aus Kapitel 3, 4 und 5 bedient, beschäftigt sich mit der Konzeption des Prototypen, bei der sowohl eine technische als auch fachliche Unterteilung vorgenommen wird.

- In Kapitel 7 wird die Implementierung des Prototypen behandelt und richtet sich speziell an zukünftige Entwickler, die die Erweiterung des CAIS-Prototypen im Blick haben. Zunächst wird die Entwicklungsumgebung und die Infrastruktur erläutert. Anschließend wird ein Überblick über das Backend und Frontend gegeben. Zum Schluss wird dargelegt, wie der entwickelte Prototyp getestet wurde.
- Zum Abschluss erfolgt in Kapitel 8 eine Zusammenfassung in Form eines Fazits, in der die Rückbetrachtung der PG und die Lessons Learned behandelt werden, sowie ein Ausblick, der mögliche Ausbaustufen und Forschungsfragen über den Forschungsschwerpunkt der Arbeit hinaus vorschlägt.

2. Organisation

In diesem Kapitel wird die Organisation der Projektgruppe beschrieben. Dazu wird zuerst die Projektgruppe und danach der Internetauftritt vorgestellt. Abschließend werden die Ressourcen der Projektgruppe zusammengefasst.

2.1 Projektgruppe

Die Projektgruppe (PG) umfasst alle studentischen sowie alle übrigen Teilnehmer. Der Kern der Projektgruppe besteht aus sieben Studenten, die geleitet werden von Oliver Norkus und unterstützt werden durch Marco Haas, der die Firma Ceyoniq-Consulting GmbH¹ vertritt und praxisnahes Branchenwissen zur Verfügung stellt. Die Tabelle 2.1 listet alle Teilnehmer übersichtlich auf.

Beziehung zur Projektgruppe	Name
Leiter	Oliver Norkus
Teilnehmer	Jannick Asmuß
Teilnehmer	Nils Lutz
Teilnehmer	Carsten Manske
Teilnehmer	Marcel Kuchler
Teilnehmer	Dörthe Euhausen
Teilnehmer	Dennis Vorreiter
Teilnehmer	Phillip Kramer
Externer	Marco Haas

Tabelle 2.1: Auflistung aller Mitglieder der Projektgruppe

2.2 Internetauftritt

In diesem Unterkapitel wird der Internetauftritt der Projektgruppe dargestellt und der Nutzen erläutert. Der Internetauftritt dient repräsentativen Zwecken, informiert Interessenten über die Projektziele und zeigt den Progress der Projektgruppe auf. Auch die Möglichkeit der Kontaktaufnahme zur Projektgruppe ist gegeben. Unter der folgenden Adresse kann die

¹weitere Informationen unter <http://www.ceyoniq-consulting.com/>

Website aufgerufen werden: <https://cais.uni-oldenburg.de/>. Die Abbildung 2.1 zeigt die Startseite.



Abbildung 2.1: Die Startseite

Die Seite ist so strukturiert, dass über die verschiedenen Reiter die jeweiligen Kategorien aufgerufen werden können. Unter „CAIS“ wird die Projektgruppe selber beschrieben. „Projekt“ umfasst die Anforderungen, den Anwendungsfall, die Meilensteine, die Zwischenpräsentation und die Abschlusspräsentation. „Publikationen“ dient dem Zugriff auf Publikationen der Projektgruppe. Unter „Team“ wird das Team vorgestellt und unter „Impressum“ sind die obligatorischen Kontaktmöglichkeiten zu finden.

2.3 Ressourcen

Die Projektgruppe verwendete verschiedene Ressourcen während des Projekts. Da das Projekt kein finanzielles Budget hatte, wurden Ressourcen verwendet, die von der Universität Oldenburg zur Verfügung gestellt wurden sowie kostenlos beschafft werden konnten. Verwendet wurde Software zur Unterstützung der Arbeit. Über Confluence² wurden Inhalte erstellt, dokumentiert, diskutiert und archiviert. Mithilfe von Jira³ wurden die Arbeitsaufträge verwaltet. Zur Synchronisierung und Versionierung der Ergebnisse wurde Gitlab⁴ verwendet und als zentrale Datenspeicherungs- und Austausch-Möglichkeit wurde Dropbox⁵ genutzt. Zur Kommunikation wurde anfangs aufgrund der Einfachheit Whatsapp⁶ verwendet, das

²weitere Informationen unter <https://de.atlassian.com/software/confluence>

³weitere Informationen unter <https://de.atlassian.com/software/jira>

⁴weitere Informationen unter <https://about.gitlab.com/>

⁵weitere Informationen unter <https://www.dropbox.com/>

⁶weitere Informationen unter <https://www.whatsapp.com/?l=de>

aber durch Slack⁷ abgelöst wurde. Über einen Forschungsantrag an das Hasso-Plattner-Institut⁸ konnte eine SAP HANA-Datenbank in der Cloud bezogen werden. Außerdem stellte die Firma Ceyoniq-Consulting GmbH, in Person von Marco Haas, Fach- beziehungsweise Branchenwissen bereit, ebenso Testdaten als CSV-Datei.

2.4 Rahmenbedingung und Einstieg

Die Aufgabe des Projekt-Managements ist, dass das Projektziel möglichst effektiv auch mit begrenzten Ressourcen erreicht werden kann. Die Rahmenbedingungen resultieren aus den Anforderungen der Studenten der Universität Oldenburg. Die Hauptanforderung der Studenten ist es, zu berücksichtigen, dass alle Teilnehmer der Projektgruppe unterschiedliche technologische Kenntnisse aufgrund unterschiedlicher Studienschwerpunkte im Bachelor haben. Außerdem gibt es die Einschränkung, dass kaum gemeinsame Arbeitszeit möglich ist, weswegen eine Arbeitsweise nötig ist, die wenig gemeinsame Kontaktzeit erfordert. Durch die Universität wird eine Arbeitszeit, mit einem Richtwert von 14 Stunden pro Woche, sowie die Durchführung einer Alpha-, Beta- und einer Abschlusspräsentation gefordert.

Zum Unterstützen des Projekt-Managements wird obligatorisch ein Vorgehensmodell verwendet. Als Vorgehensmodell wurde zu Beginn Scrum⁹ festgelegt.

An dieser Stelle sollen die allgemeinen Vor- und Nachteile aufgezählt werden, um die grundsätzlichen Herausforderungen für die Projektgruppe im Umgang mit Scrum zu betonen. Zunächst werden die Nachteile beim Einsatz von Scrum aufgelistet: [DSC13]

- Das Framework richtet sich an erfahrene Entwickler in Vollzeit.
- Scrum erfordert Selbstorganisation.
- Scrum ist ausgerichtet, um schnell Probleme aufzudecken. Die Konsequenz ist, dass oft über Probleme geredet werden muss, was wiederum zu Problemen in der Kommunikation der Gruppe führen kann.
- Die Projektgruppe hatte keine Erfahrung im Umgang mit Scrum.

Demgegenüber stehen die Vorteile, die den Einsatz von Scrum für die Projektgruppe begründen: [DSC13]

- Der Projektgruppe wird die Möglichkeit geboten, Erfahrungen mit Scrum zu sammeln.
- Die Entscheidungsgewalt und die daraus resultierende Verantwortung wandert vom Projektleiter zum Team, da es bei Scrum keinen klassischen Leiter gibt.

⁷weitere Informationen unter <https://slack.com/>

⁸weitere Informationen unter <https://www.hpi.de>

⁹Eine empfehlenswerte Einführung in die Thematik ist zu finden unter <https://www.youtube.com/watch?v=00dAwUXmL3E&feature=youtu.be>

- Das unabhängige und selbstständige Arbeiten steht im Vordergrund.
- Scrum ermöglicht agiles Arbeiten, das bedeutet, dass neue Erkenntnisse jederzeit in das Projekt einfließen können. Das führt zu einer Konzentration auf die wesentlichen Aufgaben.

Bevor der tatsächliche Einsatz von Scrum beschrieben wird, folgt nun eine theoretische Auseinandersetzung mit dem Framework.

2.5 Grundlagen über Scrum

Nun folgen die theoretischen Grundlagen zu Scrum, die notwendig sind, um die Entwicklung von Scrum im praktischen Einsatz verstehen zu können.

2.5.1 Einführung

Scrum ist ein Rahmenwerk für das Projekt-Management, das darauf basiert, Wissen aus Erfahrungen zu gewinnen und dies in künftige Entscheidungen einfließen zu lassen. Dazu wird das Projekt in Iterationen eingeteilt, an dessen Ende jeweils ein Inkrement entsteht. Um die Erfahrung zu gewinnen, definiert Scrum drei Säulen. Die erste Säule ist die Transparenz. Transparenz bedeutet, dass jeder im Team die gleiche Einsicht auf die Aktivitäten im Scrum und ein gemeinsames Verständnis dafür hat, was erledigt werden soll. Die zweite Säule ist die Überprüfung und bedeutet, dass jeder den Fortschritt mit dem Ziel der Iteration abgleicht. Die Anpassung, dies ist die dritte Säule, führt dann dazu, dass die Soll-Ist-Abweichung reduziert wird.

Scrum beinhaltet vier Ereignisse: Sprint Planning, Daily Scrum, Sprint Review und Sprint Retrospektive [DSC13]. Zu sehen ist dies in Abbildung 2.2.

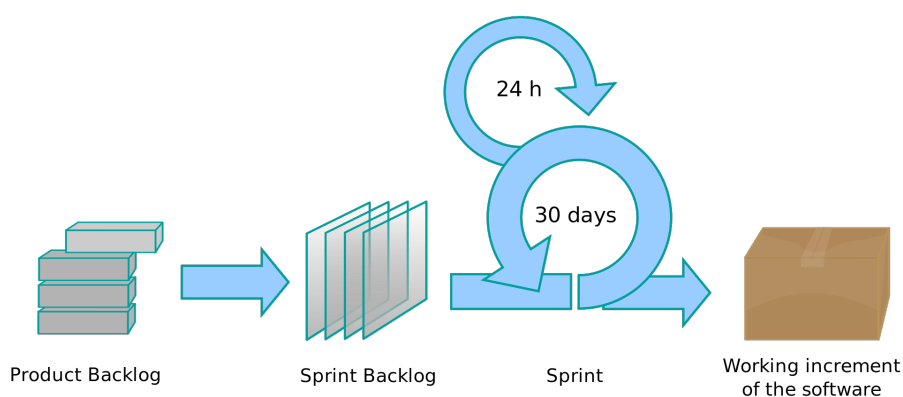


Abbildung 2.2: Der Scrum-Prozess [SCW16]

2.5.2 Rollen

Bevor nun die Ereignisse und Artefakte definiert werden, werden die Rollen im Scrum erklärt. Das Scrum-Team besteht aus dem Product Owner, dem Entwicklungsteam und dem Scrum Master. Der Product Owner ist für das entstehende Produkt verantwortlich und für die Arbeit des Entwicklungsteams. Das Entwicklungsteam arbeitet am Produkt und entwickelt dieses von Iteration zu Iteration weiter. Am Ende einer Iteration entsteht ein Inkrement, das potentiell auslieferbar ist. Die Entwickler arbeiten selbstorganisiert und werden in keine Kategorien, Rollen oder Ähnliches unterteilt. Auch wenn Mitglieder des Entwicklungsteams Spezialgebiete haben, sind sie Entwickler wie alle anderen Mitglieder im Team. Die Größe des Entwicklungsteams sollte mindestens drei und maximal neun Mitglieder umfassen.

Neben dem Product Owner und dem Entwickler existiert eine dritte Rolle: Der Scrum Master. Der Scrum Master hat die wesentliche Funktion, das Scrum-Framework zu etablieren und zu optimieren. Er unterstützt die Entwickler und den Product Owner beim Durchführen von Scrum-Ereignissen. Außerdem betreut er die Entwickler und versucht, Probleme zu beseitigen und Veränderungen herbeizuführen, die zu Produktivitätssteigerungen führen. [DSC13] Scrum kennt nur diese drei Rollen. Allerdings versteht sich das Rahmenwerk auch als solches und besonders, weil Scrum auf Adaption setzt, sind Abweichungen möglich. Im regulären Scrum werden Personen außerhalb des Teams als Stakeholder zusammengefasst. Stakeholder können an Ereignissen teilnehmen, um so externe Meinungen einfließen zu lassen. Sie sind jedoch nicht als Rolle im Sinne von Scrum zu verstehen. Eine Möglichkeit der Anpassung ist beispielsweise das Einführen von neuen Rollen. So kann zum Beispiel ein Manager definiert werden. Der Manager ist für finanzielle Aspekte verantwortlich, beschafft Aufträge und stellt Arbeitsplatzinfrastruktur bereit. Prinzipiell können Regeländerungen jederzeit eingeführt und jederzeit abgeschafft werden. [Glo13]

2.5.3 Artefakte

Artefakte repräsentieren unfertige und fertiggestellte Aufgaben. Artefakte sind so konzipiert, dass sie Transparenz und damit Möglichkeiten zur Überprüfung und Anpassung bieten. Es gibt insgesamt drei Artefakte.

Das Product Backlog beinhaltet alle Anforderungen an das Produkt. Es ist eine Liste, die nach Prioritäten geordnet ist. Der Product Owner ist für das Product Backlog verantwortlich. Es ist niemals vollständig und wird regelmäßig weiterentwickelt. Das Product Backlog wird regelmäßig überarbeitet, das bedeutet, dass die dort enthaltenen Anforderungen verfeinert, entfernt oder neue Anforderungen erfasst werden. Die wichtigsten Anforderungen sollten am weitesten ausformuliert und in der Liste höher angeordnet werden. Das Product Backlog existiert so lange, wie das Produkt entwickelt wird. [DSC13]

Im Sprint Backlog werden die Inhalte des Product Backlogs weiterverarbeitet und in die nächste Iteration als Aufgabe übernommen. Dann obliegt es dem Entwicklerteam, den Aufwand zu schätzen, und wie viele Anforderungen in der nächsten Iteration bearbeitet werden können. Das Entwicklerteam ist für das Sprint Backlog verantwortlich, das während der Iteration angepasst wird. Das Backlog stellt die Ziele der Iteration dar. [DSC13]

Das dritte Artefakt ist das Inkrement. Das Inkrement ist das Ergebnis aus allen fertiggestellten Anforderungen aus dem Product Backlog von allen Iterationen. Am Ende einer Iteration muss das Inkrement verwendbar und potentiell auslieferbar sein. [DSC13]

2.5.4 Ereignisse

Neben den Rollen und Artefakten beinhaltet Scrum außerdem Ereignisse. Die Ereignisse haben eine definierte Dauer und fördern die Transparenz, um so die Arbeitsweise überprüfen und anpassen zu können. Unterschieden werden insgesamt fünf Ereignisse. Der Sprint ist dabei als Container zu verstehen, beinhaltet die weiteren vier Ereignisse und hat einen festen Start- und Endtermin. Innerhalb dieser Zeitspanne werden die anderen Ereignisse praktiziert. Endet ein Sprint, startet danach der nächste. So wird das iterative Arbeiten realisiert. Innerhalb des Sprints entsteht ein potentiell auslieferbares Inkrement. Neben den vier anderen Ereignissen findet die eigentliche Entwicklungsarbeit statt. Ein Sprint kann vom Product Owner abgebrochen werden, wenn dieser es als notwendig erachtet. [DSC13]

Der Sprint wird mit dem Ereignis Sprint Planning eröffnet. Beim Sprint Planning wird der Sprint geplant, indem das Scrum-Team die Ziele für den Sprint festlegt. Dies wird dadurch erreicht, dass die Inhalte vom Product Backlog besprochen und die Entwickler entscheiden, was davon innerhalb des Sprints umgesetzt werden kann. Dadurch entsteht das Sprint Backlog. Am Ende des Sprint Plannings sollte das Entwicklerteam wissen, wie das Ziel erreicht und wie das Inkrement in den Zielzustand gebracht werden kann. [DSC13]

Während des Sprints findet täglich das Daily Scrum statt. Dies ist ein kurzes Ereignis, bei dem jeder Entwickler seine aktuelle Arbeit und die Herausforderungen beschreibt. Ebenso wird der Fortschritt bis zum nächsten Daily Scrum prognostiziert. Damit die Kommunikation effektiv ist, werden dazu drei Fragen beantwortet: [DSC13]

- Was habe ich gestern erreicht, das dem Entwicklungsteam hilft, das Sprint-Ziel zu erreichen?
- Was werde ich heute erledigen, um dem Entwicklungsteam bei der Erreichung des Sprint-Ziels zu helfen?
- Sehe ich irgendwelche Hindernisse, die mich oder das Entwicklungsteam vom Erreichen des Ziels abhalten?

An dem Ereignis nehmen nur das Entwicklungsteam und der Scrum Master teil. Dies führt dazu, dass die Kommunikation untereinander gefördert und Probleme identifiziert werden. Ebenso wird dadurch Wissen über Fortschritt und das Inkrement ausgetauscht. [DSC13]

Zum Anschluss eines Sprints finden die verbleibenden zwei Ereignisse statt. Zunächst wird im Sprint Review das Inkrement geprüft. Das ganze Scrum-Team sowie Stakeholder nehmen teil. Ziel ist es, Informationen über das Inkrement auszutauschen und so ein Feedback zu erhalten. Das Feedback wird genutzt, um das Product Backlog anzupassen. Es ist kein Ereignis zum Berichten des Projektprogresses. [DSC13]

In der Sprint Retrospektive prüft sich das Team selbst und den Sprintverlauf und identifiziert Verbesserungsmöglichkeiten sowie deren Umsetzung. Da das Scrum-Team sich selber prüft, nimmt auch das ganze Team an dem Ereignis teil. [DSC13]

2.6 Einstieg in Scrum

Das Scrum-Framework liefert das Regelwerk zum Durchführen von Scrum. Das Framework bietet auch Möglichkeiten der Anpassung und so soll nun die Scrum-Version der Projektgruppe beschrieben werden.

2.6.1 Rollenverteilung

Zunächst folgt die Rollen-Verteilung innerhalb der Projektgruppe. Oliver Norkus stellt den Product Owner dar, Marco Haas ist ein Stakeholder und der Rest der Projektgruppe bildet das Entwicklerteam. Einer der Entwickler nimmt außerdem die Rolle des Scrum Masters ein. Die Rolle des Scrum Masters ist dabei so ausgelegt, dass der Scrum Master ebenfalls am Inkrement entwickelt.

2.6.2 Konzeption der Ereignisse

Wie bereits zuvor erwähnt, beinhaltet Scrum fünf Ereignisse. Den Rahmenbedingungen der Projektgruppe entsprechend gilt es, die Ereignisse anzupassen. Der Sprint ist soweit unverändert und bildet den Container für die anderen Ereignisse. Die Sprintdauer ist flexibel angesetzt. Der Tabelle 2.2 ist die genaue Definition zu entnehmen.

Teilnehmer	Input	Verarbeitung	Output
Entwickler	Product Inkrement	Sprintziel festlegen	Sprint Backlog
	Product Backlog	Was kann erreicht werden?	
	Kapazitäten der Entwickler	Wie kann es umgesetzt werden?	

Tabelle 2.2: Definition Sprint Planning

Das Daily Scrum laut Framework umzusetzen, ist für die Projektgruppe nicht möglich, da alle Projektmitglieder viel von zuhause entwickelt haben und so nur wenig Kontaktzeit

stattgefunden hat. Die drei W-Fragen sind in einer modifizierten Version im Confluence zu berichten. Die W-folgenden Fragen stellen in der modifizierten Version Bezug zum Sprintziel her:¹⁰

- Was habe ich im Hinblick auf das Sprintziel erreicht?
- Was will ich im Hinblick auf das Sprintziel als Nächstes erreichen?
- Was behindert mich im Hinblick auf das Sprintziel?

Die Tabelle 2.3 zeigt die Beschreibung der Umsetzung des Sprint Reviews.

Teilnehmer	Input	Verarbeitung	Output
Scrum Team	Sprintziel	Übersicht über Fortschritt	Ergebnisse in User Stories
	Sprint Backlog	Inkrement demonstrieren	überarbeitetes Product Backlog
	Inkrement	Diskutieren und Inspektion	

Tabelle 2.3: Definition Sprint Review

Die Tabelle 2.4 zeigt die Beschreibung der Umsetzung der Sprint Retrospektive.

Teilnehmer	Input	Verarbeitung	Output
Entwickler Scrum Master	Feedback	Diskussion über Menschen	Ergebnisse in User Stories
		Beziehungen Tools und Prozesse Probleme identifizieren	überarbeitetes Product Backlog

Tabelle 2.4: Definition Sprint Retroperspektive

2.6.3 Definition der Artefakte

Es folgt nun die Beschreibung, wie die Projektgruppe die Artefakte für sich definiert hat. Hierbei gibt es Unterschiede zum Scrum-Regelwerk. Ebenso ist es erforderlich, um die Transparenz zu gewährleisten, genau zu definieren, wie die Artefakte im Scrum-Prozess involviert sind.

Product Backlog Als Product Backlog dient einerseits das klassische Whiteboard bei gemeinsamen Treffen. Außerhalb der Treffen wird JIRA verwendet. Die Verantwortung dafür liegt bei dem Entwicklungsteam und nicht, wie im Regelwerk vorgegeben, beim Product Owner. Die Anforderungen werden hier in Form von User Stories erfasst und durch den Scrum-Prozess transportiert.

¹⁰weitere Informationen unter <http://boeffi.net/scrum/agile-smells/smells-im-daily-scrum/die-beruhmten-drei-fragen-werden-nicht-richtig-gelebt/>

Sprint Backlog Auch das Sprint Backlog wurde einerseits über ein Whiteboard sowie über das Software-Tool JIRA realisiert. Die Verantwortung dafür liegt ebenfalls beim Entwicklungsteam.

Inkrement Das Inkrement ist das Ergebnis aller bisher erledigten User Stories und befindet sich im auslieferungsfähigen Zustand.

Non-Scrum-Elemente Aufgrund der Einbindung in universitäre Abläufe werden einige Ereignisse praktiziert, die nicht dem Scrum-Regelwerk entsprechen. Dies ist der Jour Fixe, der regelmäßige wöchentliche Termin mit dem Projektleiter, sowie die Alpha-, die Zwischenstands- und die finale Abschlusspräsentation. Dabei wird auf Rollen, Artefakte etc. verzichtet. Es sind Elemente aus dem klassischen Projekt-Management.

2.7 Anpassung von Scrum

Das Scrum-Framework setzt auf Adaption und damit Anpassungen zur Optimierung. Die Notwendigkeit der Anpassung resultiert daraus, dass Qualitätssicherung und Testen im regulären Framework kaum berücksichtigt werden. Abbildung 2.3 zeigt den üblichen Scrum-Prozess.

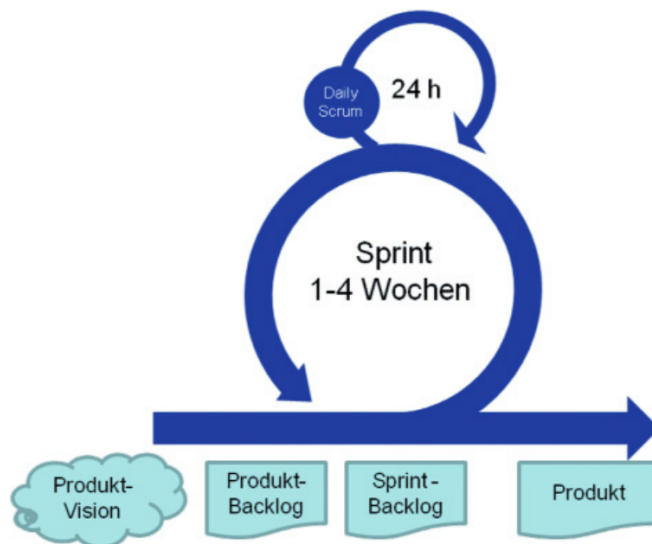


Abbildung 2.3: Der Scrum-Prozess ohne Testen [SC112]

Um Testing und Qualitätssicherung Scrum-konform praktizieren zu können, wird das Regelwerk erweitert. Abbildung 2.4 zeigt den erweiterten Prozess. Hierbei gibt es neben dem Entwicklungssprint einen Sprint zum Erstellen der Anforderungen für das Product Backlog und einen Sprint zum Testen des Produkts. Unterstützend dazu sind drei weitere Artefakte notwendig.

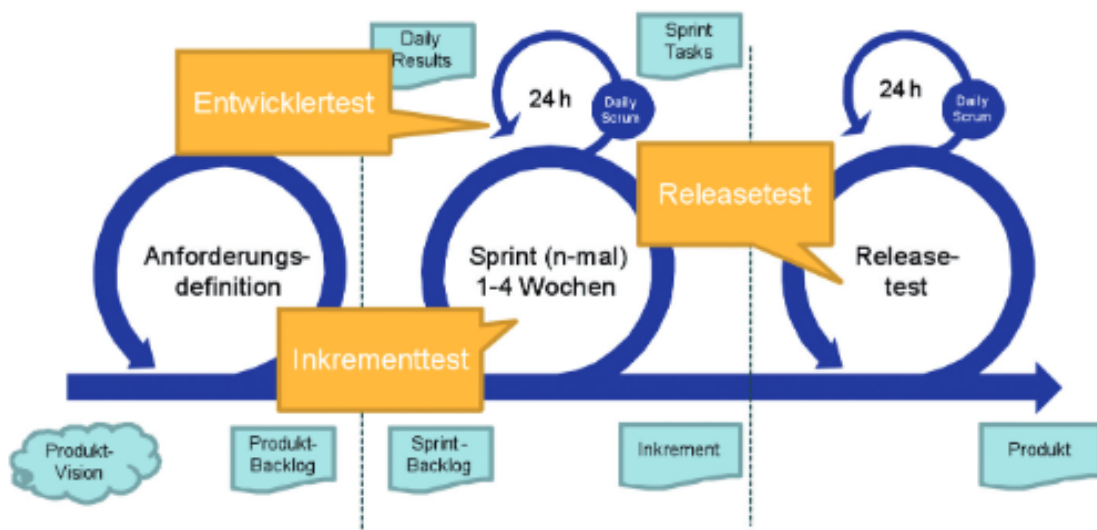


Abbildung 2.4: Der Scrum-Prozess inklusive Testen [SC112]

Definition of Ready Die sogenannte Definition of Ready beschreibt, was erforderlich ist, sodass der Product Owner die User Story als erledigt betrachtet. Ohne diese Akzeptanzkriterien kann eine User Story nicht an den Sprint Backlog übergeben werden und somit nicht im Entwicklungsprozess umgesetzt werden.¹¹

Definition of Done Die Definition of Done ist eine Liste mit Kriterien, die beschreiben, welche Voraussetzungen erfüllt sein müssen, damit die User Story von den Entwicklern abgeschlossen werden und im Sprint Review analysiert werden kann. Die Definition of Done beinhaltet, was und wie getestet wird, um so die Fertigstellung der User Story feststellen zu können.¹²

Definition of Test Die Definition of Test beschreibt die Teststrategie. Die Teststrategie wird iterativ überarbeitet, um so die Erfahrungen aus den Sprints einfließen lassen zu können. Sie wird in kurzen Aussagen formuliert und über die Sprints erweitert und angepasst.¹³

¹¹weitere Informationen unter <https://www.scrum-events.de/was-ist-die-definition-of-ready-dor.html>

¹²weitere Informationen unter <https://www.scrumalliance.org/community/articles/2008/september/definition-of-done-a-reference>

¹³weitere Informationen unter <http://www.agiletestingmanifesto.de>

3. Cloud Computing

In diesem Kapitel wird die Thematik des Cloud Computing erörtert. Es wird ein Überblick über die wichtigsten Grundlagen zum Cloud Computing gegeben. Betrachtungsgegenstand dieses Kapitels werden dabei die Definition des Cloud Computing, dessen Ziele und Eigenschaften sowie die Cloud-Architektur mit seinen Service-Bausteinen sein. Basistechnologien wie Virtualisierung, Service-orientierte Architekturen und Web Services zeichnen das Cloud Computing aus und finden in dieser Thematik Anwendung. Neben der Virtualisierung sind Service-orientierte Architekturen (SOA) und Web Services als fundamentale Voraussetzungen für das Cloud Computing zu verstehen, die im Rahmen dieses Kapitels nicht weiter aufgegriffen werden, da das gewählte Hauptaugenmerk auf die verschiedenen Klassen von Diensten bzw. Services und den unterschiedlichen Betriebs- bzw. Liefermodelle des Cloud Computing liegen wird.

3.1 Grundlagen

Cloud Computing ist in der Informationstechnologie (IT) ein allgegenwärtiger Begriff, zu dem es viele Interpretationen gibt, aber dem keine standardisierte oder gar einheitliche Definition zu Grunde liegt. Cloud Computing ermöglicht die Bereitstellung von Plattformen, von IT-Infrastrukturen und von Anwendungen aller Art als im Web elektronisch verfügbare Dienste. Der Begriff *Cloud* zielt darauf ab, dass die Erbringung der Dienste von einem Anbieter im Internet (beziehungsweise im Intranet) erfolgt, und weiterführend können die Nutzer solcher Cloud-Dienste wiederum selbst ihre eigenen Angebote als Dienste anbieten. [BKNT11]

Unabhängig davon, ob eine einheitliche und standardisierte Definition des Begriffs Cloud Computing vorliegt, sind die grundlegenden Konzepte als auch die generellen Ziele des Cloud Computing unbestritten:

- Cloud-Ressourcen sind in der Regel virtualisiert
- Cloud-Dienste sind darüber hinaus dynamisch skalierbar
- Cloud Computing folgt den Ideen des Utility Computing [BKNT11]

Hieraus lässt sich ableiten, dass Cloud Computing das moderne Web und Virtualisierung nutzt, um Ressourcen verschiedenster Art als elektronisch verfügbare Dienste dynamisch bereitzustellen. Dabei sollen die Dienste von mehreren Konsumenten skalierbar und verlässlich, einschließlich der Verfügbarkeit sowohl auf Abruf als auch nach Bedarf, nutzbar sein. [BKNT11]

Das Nationale Institut für Standards und Technologie (NIST) in den USA hat eine Definition von Cloud Computing erstellt, die in diesem Themengebiet sehr häufig zitiert, bis heute auf weitgehende Akzeptanz stößt und verschiedene Definitionsansätze bündelt: [MG11]

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models."

Die Definition nach NIST ist dynamisch, das heißt es erfolgt regelmäßig eine Anpassung der Definition an die neuen Entwicklungen. [MRV11]

Das NIST bestimmt fünf wesentliche Eigenschaften des Cloud Computings, drei verschiedene Klassen von Diensten beziehungsweise Services und vier unterschiedliche Betriebs- beziehungsweise Liefermodelle. Als wesentliche Eigenschaften werden benannt: [MG11],[BKNT11]

- **Diensterbringung auf Anforderung:** Dienste sind auf Anforderung und selbständig von Konsumenten ohne erforderliche menschliche Interaktion mit dem Anbieter nutzbar.
- **Netzwerkbasierter Zugang:** Dienste können netzwerkbasierend in Echtheit durch Verwendung von Standardtechnologien abgerufen werden.
- **Ressourcen Pooling:** Ressourcen sind in Pools konsolidiert und erlauben eine parallele Diensterbringung für mehrere Nutzer (Mandanten), die dem tatsächlichen Bedarf eines jeden Nutzers angepasst ist.
- **Elastizität:** Ressourcen werden schnell und in verschiedenen, auch fein granularen Quantitäten, zur Verfügung gestellt und erlauben so die Skalierung von Systemen. Dem Nutzer gegenüber entsteht die Illusion unendlich verfügbarer Ressourcen.
- **Messbare Dienstqualität:** Dienste sind quantitativ und qualitativ messbar, so dass eine nutzungsabhängige Abrechnung und Validierung der Dienstqualität gegeben ist.

Bei der Betrachtung von Cloud-Architekturen erfolgt eine Unterscheidung zwischen zwei verschiedenen Perspektiven, nämlich der organisatorischen und der technischen Sicht. Die organisatorische Sicht unterscheidet nach einer Trennung der organisatorischen Einheiten

von Anbietern und Benutzern, während sich die technische Sicht an funktionalen Eigenschaften orientiert. Damit entspricht die technische Sicht den Klassen der Dienste oder Services beziehungsweise Cloud-Servicemodellen und die organisatorische Sicht dem Betriebsbeziehungsweise Liefermodell aus der Definition des NIST [BKNT11]. Die einzelnen Clouds werden auch als Cloud-Liefermodelle verstanden, denn alle können die Nutzungsvarianten IaaS, PaaS, SaaS und XaaS, erläutert in Kapitel 3.2, dem Endanwender „ausliefern“. [MRV11]

Nachfolgend werden die Betriebsbeziehungsweise Liefermodelle (Kapitel 3.1.1; 3.1.2; 3.1.3) und weiterführend die Klassen der Dienste oder Services beziehungsweise Cloud-Servicemodellen (Kapitel 3.2.1; 3.2.2; 3.2.3) erläutert.

3.1.1 Public Cloud

Eine Public Cloud ist eine Cloud, die öffentlich zur Verfügung steht. Als Public Cloud bezeichnet man sämtliche Cloud-Angebote, bei denen die Anbieter und die potenziellen Benutzer nicht derselben organisatorischen Einheit angehören. Sie ist im Gegensatz zu der Private Cloud nicht mehr auf interne Anwendungen einer einzelnen Institution, eines Departments oder eines Unternehmens beschränkt. Bei der Public Cloud machen die Anbieter ihre Cloud öffentlich zugänglich und bieten den Benutzern meist ein Web-Portal an, in dem diese ihre gewünschten Leistungsumfänge spezifizieren können. Statt eines übergeordneten Rahmenvertrags, wird in der Regel bevorzugt die vertragliche Bindung im Rahmen der Leistungsspezifikation eingegangen. Die tatsächlich über die Zeit benutzten Ressourcen bilden die Basis für die Abrechnung der Dienste. [BKNT11] Es können weitere Unterformen gebildet werden. Bei der Exclusive Cloud kennen sich sowohl Anbieter als auch Nutzer. Das Gegenstück hierzu ist die Open Cloud, bei denen sich im Gegensatz zu den Exclusive Clouds die Anbieter und Nutzer nicht kennen. [MRV11]

3.1.2 Private Cloud

Im Kontrast zu der Public Cloud steht die so genannte Private Cloud, bei der die Anbieter- und die Benutzerseite derselben organisatorischen Einheit angehören. Häufig ist Anbieter gleich Betreiber gleich den Anwendern im Falle von Private Clouds. Ein Alleinstellungsmerkmal der Private Cloud ist, dass sie ohne Internetzugang auskommen kann, sofern sie nur beim Kunden inhouse betrieben wird. [MRV11] Ein wichtiges Entscheidungskriterium für den Einsatz einer Private Cloud statt einer Public Cloud beruht auf Sicherheitsgründen, da in der Private Cloud die Kontrolle über die Daten beim Benutzer beziehungsweise innerhalb dessen Organisation bleibt. Hervorzuheben ist, dass die Dienste einer Private Cloud zwar auf organisationseigenen Ressourcen betrieben werden, jedoch teilweise versucht wird, die gleichen technischen Schnittstellen wie in der Public Cloud zu realisieren. Dieses Vorgehen ermöglicht, die in der Public Cloud vorhandenen Werkzeuge auch in der Private Cloud zu

benutzen und hält die Option offen, zunächst für die Private Cloud entwickelte Anwendungen später in die Public Cloud skalieren zu können. [BKNT11]

Es wird weiterführend zwischen den verschiedenen Evolutionsstufen unterschieden. Bei der Exploratory Cloud steht das Ausprobieren und Testen von Cloud-Funktionalitäten - eine Art Spielwiese für Versuche mit Cloud Computing - innerhalb eines Unternehmens oder einer Gruppe im Vordergrund. Die Departmental Cloud ist eine Cloud, die sich innerhalb eines Unternehmens auch lediglich innerhalb einer Abteilung befindet. Diese Cloud-Art dient mehr als nur zu Testzwecken. Die Enterprise Cloud unterscheidet sich wiederum gegenüber der Departmental Cloud darin, dass hier die Anbieter und Nutzer aus unterschiedlichen und nicht ein und derselben Unternehmensabteilungen stammen. [MRV11]

3.1.3 Hybride Cloud

Bei einer Hybrid Cloud handelt es sich um Szenarien, in denen Dienste aus der Public und Private Cloud zusammen benutzt werden. In der Regel erfolgt der Betrieb über die privaten Ressourcen und bestimmte Funktionalitäten oder Lastspitzen werden in die Public Cloud ausgelagert. Dabei erfolgt eine Abwägung der zuvor genannten Sicherheitsüberlegungen, mit dem Resultat, dass nur unkritische Funktionen und Daten ausgelagert werden dürfen. [BKNT11] Abbildung 3.1 zeigt, wie sich die drei zuvor beschriebenen Typen Public Cloud, Private Cloud und Hybrid Cloud zueinander positionieren:

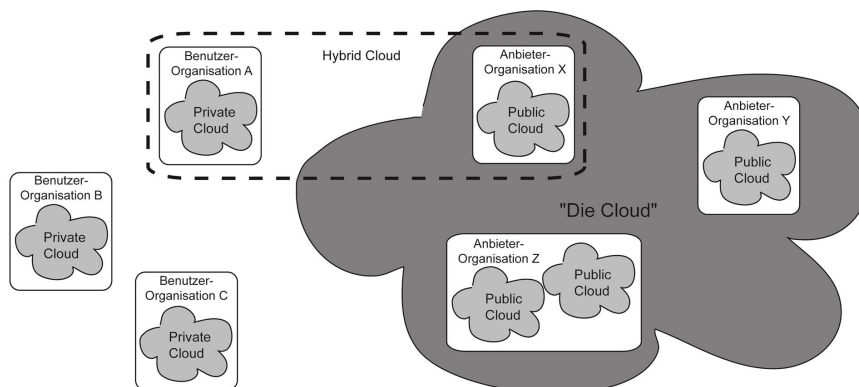


Abbildung 3.1: Public Cloud, Private Cloud und Hybrid Cloud (eigene Darstellung, angelehnt an [MRV11])

3.2 X-as-a-Service

Die von der Cloud erbrachten Leistungen umfassen verschiedene Architekturebenen beziehungsweise -Schichten, von Softwarediensten über Plattformen für Entwicklung und Betrieb bis hin zur Bereitstellung grundlegender IT-Infrastrukturdienste. Die Cloud-Architektur beruht auf dem Grundmuster der Schichtung, bei dem die einzelnen Schichten wie gewöhnlich nach ihrem Abstraktionsgrad angeordnet sind. Hervorzuheben ist, dass nach diesem Muster die höheren abstrakteren Schichten die Dienste der tieferen konkreteren Schichten zu ihrer eigenen Dienstrealisierung benutzen können. Das besondere Merkmal hierbei ist, dass die Schichtung nicht streng ist, das heißt, eine höhere Schicht kann die Dienste aller unterliegenden Schichten nutzen und nicht nur die der nächst tieferen Schicht. [MRV11]

Um die genannten Informationen verwerten zu können, folgt in der Abbildung 3.2 eine Übersicht über das durch NIST definierte Cloud-Architektur-Modell, dessen Bestandteile im weiteren Verlauf dieses Kapitels erläutert werden.

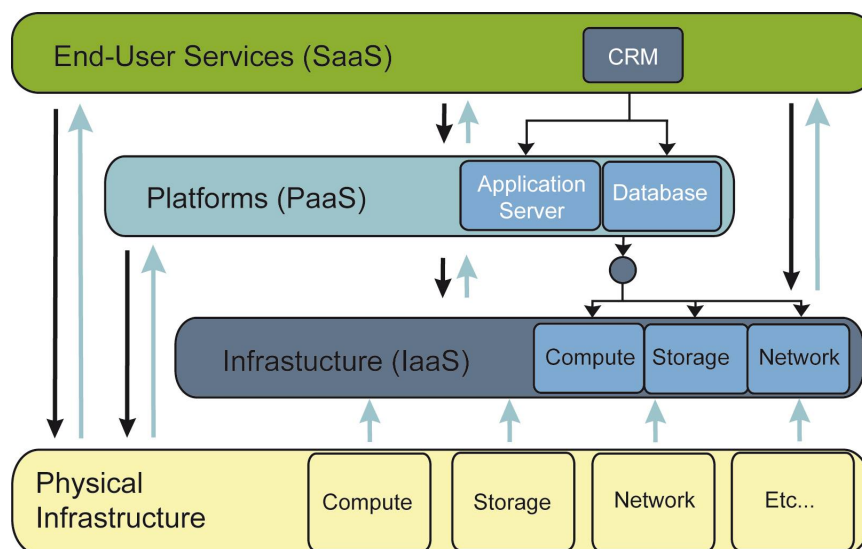


Abbildung 3.2: Cloud Computing Architektur (eigene Darstellung, angelehnt an [AR10])

Die drei Hauptschichten, nach denen das resultierende Schichtenmodell in Abbildung 3.2 unterteilt wird, folgen dem Everything as a Service-Paradigma (XaaS) mit den vier Hauptvertretern Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) und Humans as a Service (HaaS). Neben diesen vier Vertretern existieren noch der Mashup-as-a-Service (MaaS) und der Business-Process-as-a-Service (BPaaS). Der HaaS wird zusammen mit den beiden weiteren genannten Vertretern MaaS und BPaaS nicht Betrachtungsgegenstand dieses Kapitels sein. Es werden lediglich die drei Cloud-Servicemodelle beziehungsweise Dienste (IaaS, PaaS, SaaS) in den folgenden Unterkapiteln genauer erläutert. [MRV11]

C. Metzger et. al. haben eine Erläuterung zu den Klassen der Dienste oder Services beziehungsweise Cloud-Servicemodellen, die hier als Einleitung dienen soll: [MRV11]

”Cloud Computing (deutsch etwa „Rechenleistung aus der Wolke“) verfolgt den Ansatz, abstrahierte IT-Infrastrukturen – z.B. Rechenkapazität, Datenspeicher (IaaS = Infrastructure-as-a-Service), fertige Software (SaaS = Software-as-a-Service) und Programmierumgebungen (PaaS = Platform-as-a-Service) – dynamisch an den Bedarf angepasst über ein Netzwerk zur Verfügung zu stellen. Die dynamische Anpassung erfolgt in erster Linie über die Bündelung von Infrastrukturdienstleistungen.”

Aufbauend auf die Erläuterung von C. Metzger et. al. lassen sich die drei Cloud-Dienste beziehungsweise -Servicemodelle mit wenigen Worten wie folgt erklären:

- IaaS Cloud basiert auf virtuellem Zugang
- PaaS bietet grundlegende Software
- SaaS bietet Verantwortung für den Betreiber

Für die Vertiefung der von C. Metzger et. al. zuvor genannten Erläuterung werden nachfolgend die Cloud-Servicemodelle im einzelnen erläutert.

3.2.1 Infrastructure-as-a-Service

Infrastructure-as-a-Service ist der Teil einer Cloud-Dienstleistung, der dem Kunden IT-Infrastruktur über das Internet zur Verfügung stellt. Typische Beispiele sind virtuelle Telefonanlagen, Festplattenkapazität oder Backup-Services über das Internet. [MRV11] In der IaaS-Schicht wird den Benutzern eine abstrahierte Sicht auf die Hardware angeboten, das heißt auf unter anderem Rechner, Massenspeicher und Netzwerke. [BKNT11]

3.2.2 Platform-as-a-Service

C. Metzger et. al. definieren Platform-as-a-Service wie folgt: [MRV11]

”Mit Platform-as-a-Service ist die Dienstleistung gemeint, die einem Entwicklerteam die Deployment-Plattform für Softwareentwicklung zur Verfügung stellt. Im Unterschied zu einer typischen und bekannten Entwicklungsplattform wie Eclipse, die noch keinerlei Aussage darüber macht, wo beispielsweise der entwickelte Java-Code ausgeführt wird, ist beim PaaS die Zielplattform und deren Betrieb ganz klar das Serviceangebot. Typischerweise liegen viele der Komponenten eines Entwicklungsprojektes als fertige Services bereits vor. Die Entwickler nutzen die Plattform, die als Cloud-Service erreichbar ist, und „deployen“ ihren Code dort für den Betrieb.”

3.2.3 Software-as-a-Service

C.Metzger et. al. definieren Software-as-a-Service wie folgt: [MRV11]

”Grundsätzlich wird hier das Konzept verfolgt, dass Software nicht mehr auf jedem Rechner, der die Software nutzen will, installiert zu sein braucht. Anstelle dessen wird davon ausgegangen, dass bestimmte Software einfach über einen Browser aufruf- und nutzbar gemacht wird – unabhängig davon, wo die Software genau ausgeführt wird. Dem Nutzer ist es im Idealfall sogar egal, ob die Ausführung in mehreren Teilen an verschiedenen Orten im Hintergrund passiert, solange er ein einheitliches Graphical User Interface (GUI) nutzen kann. Dies ermöglicht auch für gewöhnlich die Nutzung derselben Software von vielen Standorten aus und von vielen verschiedenen Usern gleichzeitig.”

Die in Kapitel 3 bereitgestellten Informationen sind die Voraussetzung dafür, die getroffenen Entscheidungen und die erarbeitete Lösung, siehe Kapitel 6 und 7, einfacher nachvollziehen zu können. Für weitere tiefgehendere Informationen zum Themengebiet Cloud Computing empfehlen sich die Werke von [BKNT11], [MRV11] und [AR10] sowie die im Anhang enthaltenen Seminararbeiten, die im Rahmen dieser Projektgruppe verfasst wurden. Neben dem Themengebiet des Cloud Computing nimmt das Themengebiet der analytischen Informationssysteme ebenfalls einen hohen Stellenwert ein, das nachfolgend Betrachtungsgegenstand sein wird.

4. Analytische Informationssysteme

Für ein betriebswirtschaftliches Unternehmen ist es heutzutage nötig, zu jeder Zeit einen genauen Überblick über die eigenen Daten, wie Entwicklung, Wachstum, Trends und Gewinn, zu besitzen. Um dies bei der Masse an Daten zu gewährleisten, sind analytische Informationssysteme (AIS) unumgänglich. In dem folgenden Kapitel soll deshalb ein kurzer Überblick auf analytische Informationssysteme gegeben werden. Dabei wird der Begriff zunächst definiert. Danach sollen ein Überblick über die InMemory-Datenbank-Technologie gegeben und einige Algorithmen vorgestellt werden.

4.1 Definition

Analytische (oder dispositive) Informationssysteme stellen einen Sammelbegriff zu analytischen Anwendungen und Technologien innerhalb der Datenverarbeitung von großen, zumeist heterogenen Datenbeständen dar. Es gibt keine eindeutige Definition von AIS, kann aber auch mit dem Begriff *Business Intelligence* ersetzt werden. [GC15] Der Begriff umfasst Konzepte wie *Data Warehouse*, *Online Analytical Processing (OLAP)*, *Data Mining* und *Reporting*. [GC15]

Ein *Data Warehouse* bezeichnet dabei eine Sammelstelle für firmeninterne, homogene, historische Datensätze sowie externe heterogene Datensätze, die in simpler oder aggregierter Form bereitgestellt werden. Die multidimensionale Sicht auf diese Datensätze und eine Umformung in analysierbare Daten bezeichnet das Konzept des *Online Analytical Processings*. *Data Mining* und *Reporting* schließlich beinhalten Konzepte zur automatischen Mustererkennung und gesteuerten Erkennung von Wissen sowie der darauffolgenden geeigneten Präsentation dieser Daten. [Lau11] Im Gegensatz zu den operativen Systemen, die mit entscheidbaren Problemen, besonders administrativen und Bewertungsaspekten, klare und eindeutige Lösungen bieten, stellen AIS mit diffuseren Daten eine größere instabilere Herausforderungen dar. [GC15]

4.2 In-Memory Datenbanken

Um eine zeitnahe Informationsversorgung trotz hoher Datendichte und anspruchsvoller Verarbeitung bieten zu können, muss von einer Datenbanklösung mit simplen Festplattenspeicher abgesehen werden. InMemory-Datenbanken verfolgen somit den Ansatz der Hauptspeicherbasierten Datenbanken, also Datenbanken, die einen Hauptspeicher zur Datenhaltung verwenden. Möglich macht dies die immer steigende Kapazität der Hauptspeicher sowie die konstant fallenden Preise für ebendiese. InMemory-Datenbanken dienen mit ihrer schnellen Verarbeitungszeit Grundlage für ein modernes Data Warehouse dar. [GC15]

4.3 Konzepte und Algorithmen

Unter den analytischen Informationssystemen stehen eine Vielzahl von Algorithmen bereit, die in diesem Kapitel kurz angedeutet werden sollen (ausführlicher im Anhang, Seminararbeit Dennis Vorreiter).

4.3.1 Extract, Transform, Load

Extract, Transform, Load (ETL) bezeichnet eine Schnittstelle für AIS. Die angeforderten Daten entstammen dabei mehreren Quellsystemen und durchlaufen einen bestimmten Verarbeitungsdurchlauf. Die Daten werden zunächst extrahiert. Danach werden sie bereinigt und an die Bedürfnisse des Zieldatenbanksystems angepasst (*Transform*). Die so transformierten Daten werden dann in die Zieldatenbank geladen.

4.3.2 Data Mining

Das *Data Mining* umfasst die Konzepte zur Muster- und Erkenntnisgewinnung aus vorliegenden Datensätzen, zumeist unter Zuhilfenahme von Trainingsdaten. Hierbei stehen mehrere Konzepte bereit, die diese Aufgaben bewältigen.

Clustering-Algorithmus Dient zur Erkennung von größeren Gruppen innerhalb der Datensätze. Häufig verwendet wird der K-means-Algorithmus, bei dem der Abstand zwischen den Punkten soweit minimiert wird, dass eindeutige Gruppierungen entstehen.

Klassifikations-Algorithmus Dient zur Einteilung der Daten in bestimmte Klassen. Hierbei kann beispielsweise über Entscheidungsbäume festgelegt werden, welche Eigenschaften Daten erfüllen müssen, um einer bestimmten Klasse anzugehören.

Assoziationsanalyse Dient der Analyse von Mustern. Sie zeigt Wenn-Dann-Beziehungen auf. Häufig genutzte Algorithmen sind dafür die Frequent-Pattern-Analyse, bzw. der Apriori-Algorithmus.

Weiterführend beinhaltet das Data-Mining auch die Analyse von Ausreißern und die Regressionsanalyse. Die Ergebnisse werden dann mittels Zusammenfassungen in ihrer Komplexität möglichst vereinfacht. [Lau11]

5. Analyse der Anforderungen

Zu Beginn der Projektarbeit haben zwei Interviews stattgefunden, in denen die Anforderungen an die zu entwickelnde Softwarelösung ermittelt wurden. Während des ersten Interviews wurden Fragen bezüglich des Funktionsumfangs und der technischen Voraussetzungen geklärt. Das zweite Interview baute auf den Ergebnissen des ersten Interviews auf und diente dazu, die bereits aufgenommenen Anforderungen weiter zu spezifizieren.

Aus den Protokollen der Interviews wurde im Anschluss ein Lastenheft erstellt (s. Anhang) und die erhobenen Anforderungen wurden mit dem Kunden diskutiert und nach der MoSCoW-Methode¹⁴ priorisiert.

Dieses Kapitel soll einen Überblick über die Ergebnisse der Anforderungsanalyse geben. Begonnen wird dabei mit dem vom Kunden beschriebenen Geschäftsprozess und der Einordnung in die Wertschöpfungskette der Energie-Branche. Darauf folgend wird auf die Anwendungsfälle und die Unterstützung durch die zu entwickelnde Software-Lösung eingegangen. Die Unterkapitel 5.3 und 5.4 zeigen eine tabellarische Übersicht aller erhobenen Anforderungen und der zugehörigen Priorisierung durch den Kunden.

¹⁴Das Akronym MoSCoW setzt sich aus den vier, im Folgenden absteigend aufgeführten, Prioritätsstufen MUST (muss), SHOULD (soll), COULD (kann) und WON'T HAVE (nicht zu erfüllen) zusammen. Weitere Informationen sind beispielsweise unter <https://www.projektmagazin.de/glossarterm/moscow> zu finden.

5.1 Geschäftsprozess

Der vorgegebene Geschäftsprozess kann dem Bereich „Vertrieb“ zugeordnet werden, der sich, wie der Wertschöpfungskette in Abbildung 5.1 zu entnehmen ist, an die Stufen „Rohstoffförderung und Beschaffung“, „Energieerzeugung“ und „Energietransport und Speicherung“ anschließt.



Abbildung 5.1: Wertschöpfungsdiagramm zur Einordnung des Geschäftsprozesses

Die ereignisgesteuerte Prozesskette in Abbildung 5.2 zeigt den Geschäftsprozess wie er mit der Einführung des CAIS-Systems ablaufen soll. Bereits zur Vorbereitung einer Vertragsverhandlung nutzt ein Vertriebsmitarbeiter das System, beispielsweise um einen neuen Kunden in das System einzupflegen oder um sich über einen Bestandskunden zu informieren. Die Optimierung des Prozesses durch die Einführung des CAIS-Systems zeigt sich jedoch hauptsächlich während der eigentlichen Vertragsverhandlung. Ist der Kunde mit dem Angebot der Vertriebsmitarbeiters nicht einverstanden, muss dieser nun nicht mehr zurück in sein Büro, um dort ein neues Angebot vorzubereiten. Stattdessen kann er beim Kunden vor Ort mit einem mobilen Endgerät auf das CAIS-System zugreifen und sofort berechnen lassen, ob der Gegenvorschlag des Kunden noch in der Gewinnzone liegt.

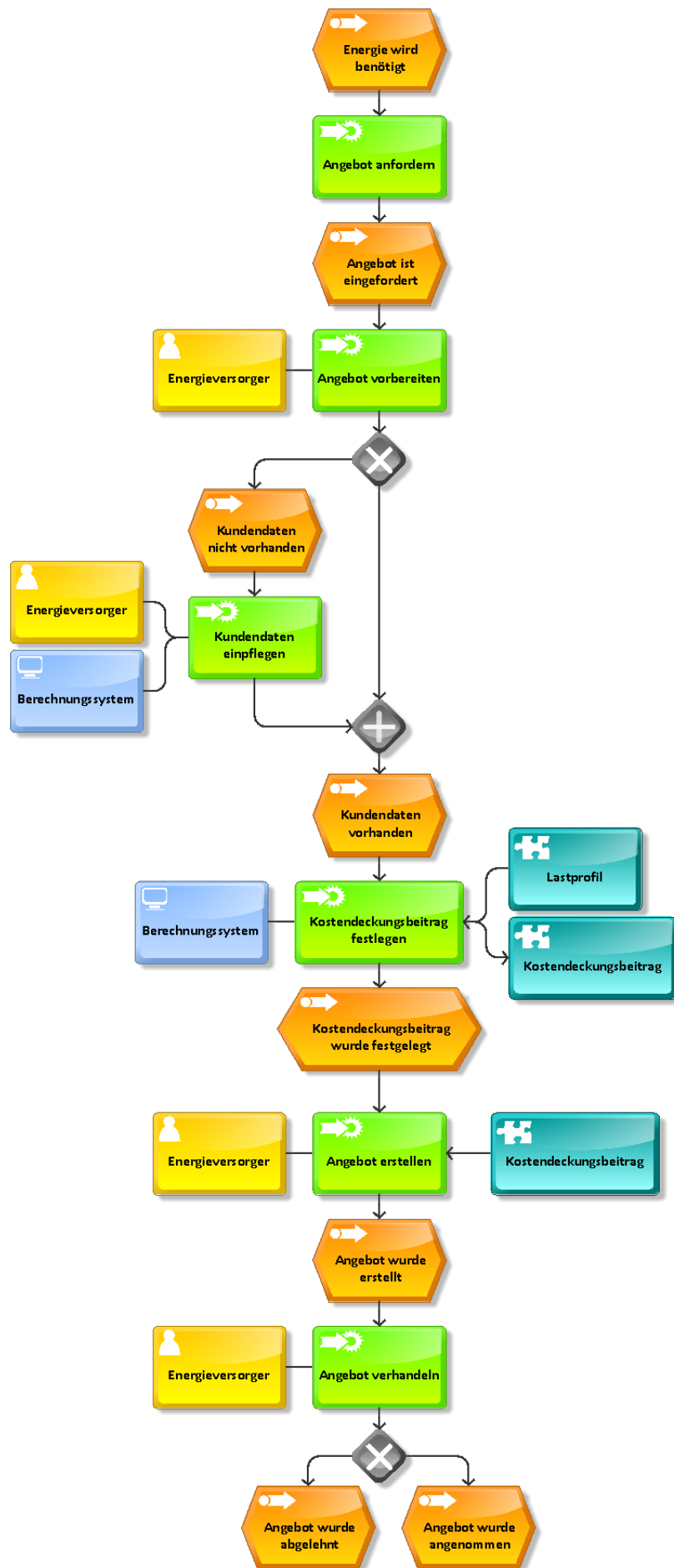


Abbildung 5.2: Ereignisgesteuerte Prozesskette

5.2 Anwendungsfälle

Im Zuge der Anforderungsanalyse wurden die drei verschiedenen Rollen Admin, Power User und Business User identifiziert, die an den in Abbildung 5.3 aufgeführten Use Cases beteiligt sind. Anwender mit der Rolle des Admin führen am CAIS-System administrative Aufgaben, wie beispielsweise das Anlegen und Freischalten von Benutzern, durch. Der Business User besteht aus einer rein konsumierenden Rolle. Anwender, die dieser Gruppe angehören, können sich Kennzahlen anzeigen und Diagramme ausgeben lassen. Business User können darüber hinaus die Granularität der Dimensionen anpassen. Anwender mit der Rolle des Power Users sind ebenso berechtigt, alle Anwendungsfälle des Business Users durchzuführen, besitzen darüber hinaus aber Änderungsrechte in Bezug auf Kennzahlen und Ausgabefunktionen. Sie sind somit in der Lage neue Funktionen für Ausgaben zu erstellen.

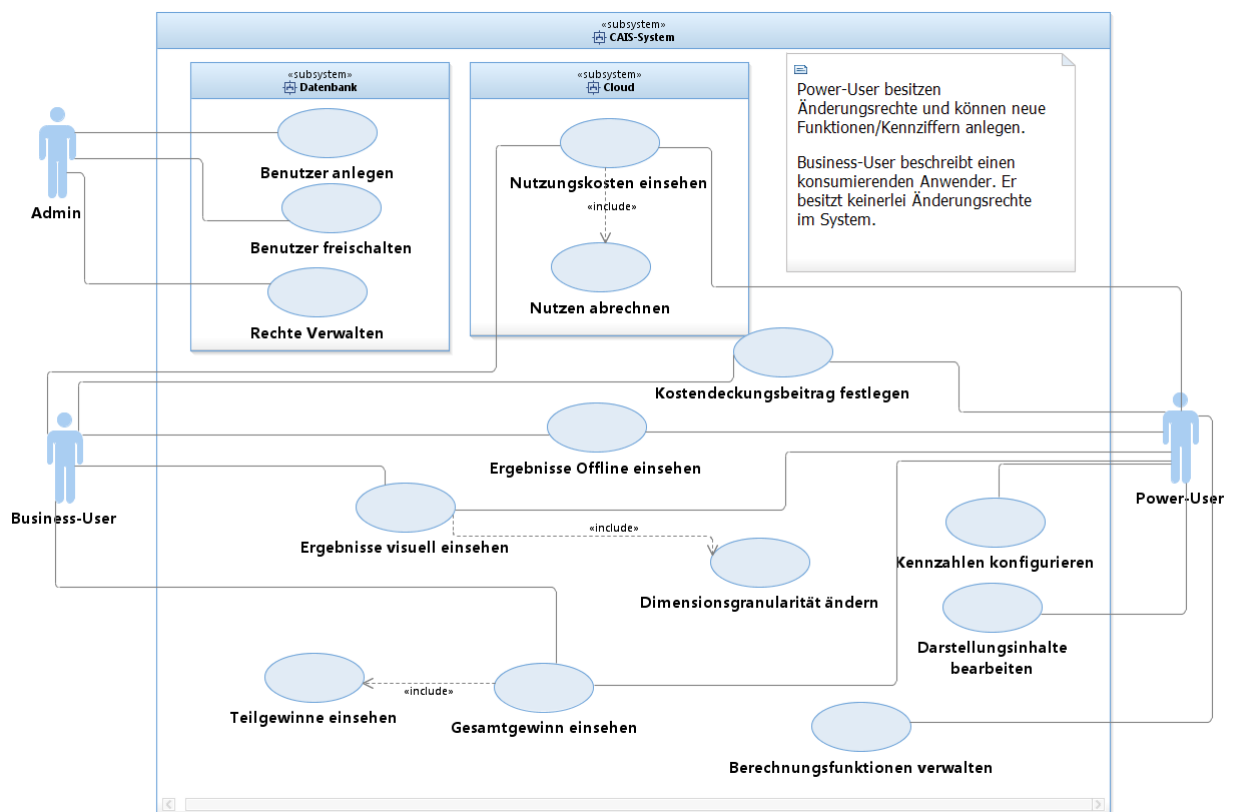


Abbildung 5.3: Use Case-Diagramm zum Projekt

Eine weitere Ausführung der Use Cases erfolgte nach IEC 62559-Template innerhalb des auszuliefernden Lastenheftes (s. Anhang).

5.3 Funktionale Anforderungen

ID	Typ	Beschreibung	Priorität
CAIS_REQ101	Funktion	Das System muss Kennzahlen darstellen können.	MUST
CAIS_REQ102	Funktion	Das System muss die Kennzahlen in Abhängigkeit zu unterschiedlichen Dimensionen (Kunde, Quartal, ...) auswerten können.	MUST
CAIS_REQ103	Funktion	Das System muss einen Gesamtgewinn errechnen können.	MUST
CAIS_REQ104	Funktion	Das System muss die Profite pro Kraftwerk und pro Kunde auswerten können.	MUST
CAIS_REQ105	Funktion	Die Kennzahlen sollen über verschiedene Zeiträume angezeigt werden können. (stundengenau, tagesgenau, pro Quartal, ...)	MUST
CAIS_REQ106	Funktion	Das System muss Ergebnisse visuell darstellen können.	MUST
CAIS_REQ106-1	Funktion	Der User soll die Darstellung der Ergebnisse frei wählen können. (bspw. Kreis-/Balkendiagramm)	MUST
CAIS_REQ106-2	Funktion	Das System soll besonders gute Werte (Gewinnzone) grün und besonders schlechte Werte (Zerlustzone) rot markieren.	MUST
CAIS_REQ301	Schnittstelle	Das System muss Testdaten per Bulk Load einlesen können.	MUST
CAIS_REQ401	Funktion	Das System muss eine Accountverwaltung besitzen.	MUST
CAIS_REQ401-1	Funktion	Der User soll seinen eigenen Account nach dem Self Service-Prinzip verwalten können.	MUST
CAIS_REQ401-2	Funktion	Der User muss seinen Accountnamen ändern können.	MAY
CAIS_REQ401-3	Funktion	Der User muss seine Adresse angeben können.	MUST
CAIS_REQ401-4	Funktion	Der User muss seine Adresse ändern können.	MUST
CAIS_REQ401-5	Funktion	Der User muss sein Passwort ändern können.	MUST
CAIS_REQ401-6	Funktion	Das System muss falsche Zugangsdaten erkennen.	MUST

CAIS_REQ402	Funktion	Das System muss eine Rechteverwaltung besitzen.	MUST
CAIS_REQ403	Funktion	Das System muss eine Session-Verwaltung besitzen.	MUST
CAIS_REQ404	Funktion	Die Rechteverwaltung muss eine Admin-Rolle (User anlegen/freischalten) besitzen.	MUST
CAIS_REQ405	Funktion	Die Rechteverwaltung muss eine Power User-Rolle besitzen.	MUST
CAIS_REQ406	Funktion	Die Rechteverwaltung muss eine Business User-Rolle besitzen.	MUST

Tabelle 5.1: Übersicht über die funktionalen Anforderungen

5.4 Nicht-funktionale Anforderungen

ID	Typ	Beschreibung	Priorität
CAIS_REQ201	Technologie	Das System muss skalierbar sein. Dazu muss der mit dem Kunden vereinbarte Skalierungsalgorithmus implementiert werden.	MUST
CAIS_REQ202	Technologie	Das System soll eine Verfügbarkeit von 99,99% aufweisen.	SHOULD
CAIS_REQ203	Technologie	Die Daten eines zu bestimmenden Kunden sollen abgespeichert werden können.	MUST
CAIS_REQ204	Technologie	Das System muss über ein nutzungsorientiertes Abrechnungsverfahren verfügen.	MUST
CAIS_REQ205	Technologie	Das System muss einer Multi-Tenancy-Architektur zugrunde liegen. (Mandantenfähigkeit)	SHOULD
CAIS_REQ206	Technologie	Wartbarkeit	SHOULD
CAIS_REQ207	Technologie	Performance	MUST
CAIS_REQ208	Technologie	Erweiterbarkeit	MUST
CAIS_REQ209	Technologie	Daten eines bestimmten Kunden sollen offline zur Verfügung stehen.	MAY
CAIS_REQ501	Benutzeroberfläche	Die Anwendung muss über alle gängigen Browser erreichbar sein.	MUST
CAIS_REQ502	Benutzeroberfläche	Die grafische Oberfläche der Anwendung muss ein responsives Design aufweisen.	SHOULD

Tabelle 5.2: Übersicht über die nicht-funktionalen Anforderungen

6. Konzeption

Das Kapitel beschreibt die Konzeptionierung der zu entwickelnden Software-Lösung und ist in zwei Teile untergliedert.

Der erste Teil, nämlich das fachliche Konzept, beschreibt die Auswahl eines passenden Skalierungsalgorithmus und die Konzeptionierung des Billing-Modells zur verbrauchsabhängigen Abrechnung. Das technische Konzept befasst sich mit der Planung der Systemarchitektur, dem zugrunde liegenden Datenmodell sowie dem SAP UI5 Framework und dazugehörigen Mockups der Benutzungsoberflächen.

6.1 Fachliches Konzept

Das fachliche Konzept befasst sich mit der Auswahl eines passenden Skalierungsalgorithmus und einer kurzen Erläuterung des Billing-Modells.

6.1.1 Skalierungsalgorithmus

Eine der Anforderungen, die mit der höchsten Priorität MUST eingestuft wurde, ist CAIS_REQ201. Gefordert wird, dass das System skalierbar ist und der mit dem Kunden vereinbarte Skalierungsalgorithmus implementiert wird.

In einem Paper (s. Anhang) wurden daher die grundlegenden Algorithmen Random, Round Robin, Fastest und Least Connections präsentiert und gegenübergestellt. Implementiert wurde nach Absprache mit dem Kunden der Round Robin-Skalierungsalgorithmus.

6.1.2 Billing-Modell

Ebenfalls mit der höchsten Priorität bewertet wurde die Anforderung CAIS_REQ204, die ein nutzungsorientiertes Abrechnungsverfahren für das System verlangt. Statt einer beispielsweise monatlichen oder jährlichen Gebühr für die Nutzung des Systems, sollen die Kosten von der tatsächlichen Nutzung durch den Anwender abhängig sein. Weitere Informationen zur Implementierung sind im Kapitel 7, Abschnitt 7.3.3 Abrechnung zu finden.

6.2 Technisches Konzept

Das technische Konzept beinhaltet sämtliche technische Entscheidungen der Projektgruppe. Begonnen wird auf Seiten des Backends mit der Vorstellung der gewählten Systemarchitektur, gefolgt vom Datenmodell und der Lastverteilung. Anschließend wird das Frontend-Konzept vorgestellt, das das SAP UI5 Framework und UI-Mockups umfasst.

6.2.1 Systemarchitektur

Aus der Anforderungserhebung war es der Gruppe möglich genauestens zu ermitteln, welchen Komponenten das System benötigt und wie diese zusammenhängen. Daraus ergab sich für das Pflichtenheft folgende zu realisierende Architektur in Abbildung 6.1.

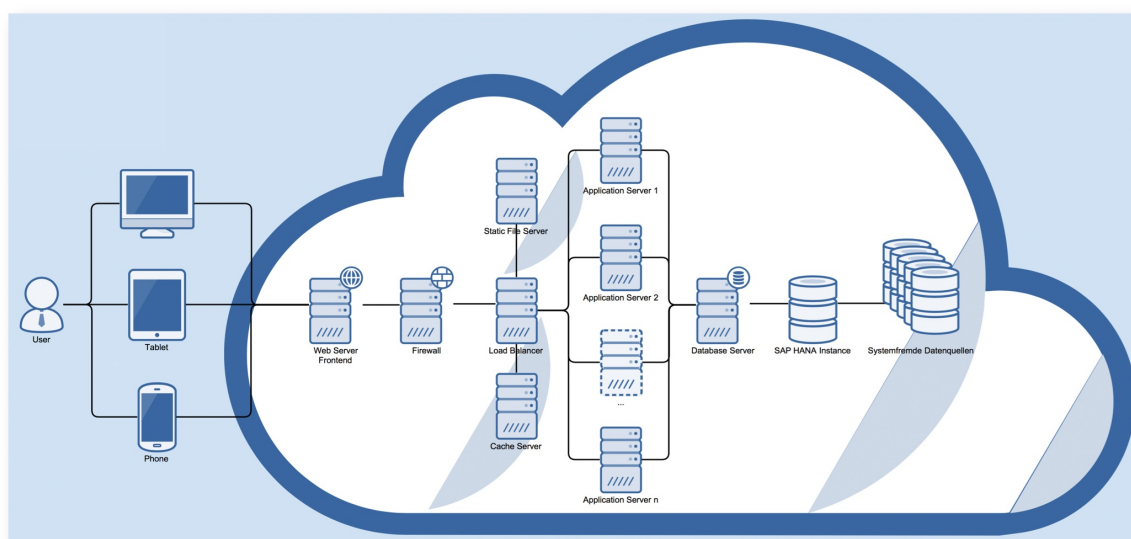


Abbildung 6.1: Architektur des Systems

Für die Umsetzung der Architektur wurden die wichtigsten Komponenten definiert und die Umsetzungsentscheidungen festgelegt.

Komponenten	Funktionalitäten	Umsetzung
Devices	Internet-fähiges Eingabegerät wie z.B. Smartphone, Tablet oder ein klassischer Desktop-Rechner	Android, iOS, Desktop (IE11, Chrome, Firefox, Safari)
Frontend Server	Stellt die Web-App über eine öffentliche Domain im Internet bereit und übersetzt Anwendereingaben in Datenabfrage für den App-Server.	Nginx Stack + UI5
Firewall	Schützt die dahinter liegende System Architektur vor unbefugtem Zugriff.	Linux, m0n0wall

Load Balancer	Sorgt für die Lastverteilung der Anfragen auf die App-Server. Vordefinierte Trigger starten oder beenden neue Instanzen der App-Server.	Nginx Stack
Cache Server	Häufig gestellten Anfragen an die Datenbasis sollen vorgehalten werden.	Varnish
App-Server	Stellt eine API bereit um Anfragen vom Frontend Server zu bedienen.	NodeJS Stack
Database Server	Stellt eine Datenbank für die User Verwaltung bereit.	MySQL Stack

Tabelle 6.1: Komponenten und Umsetzungsentscheidungen

Daraus ergab sich das folgende Komponentendiagramm in Abbildung 6.2.

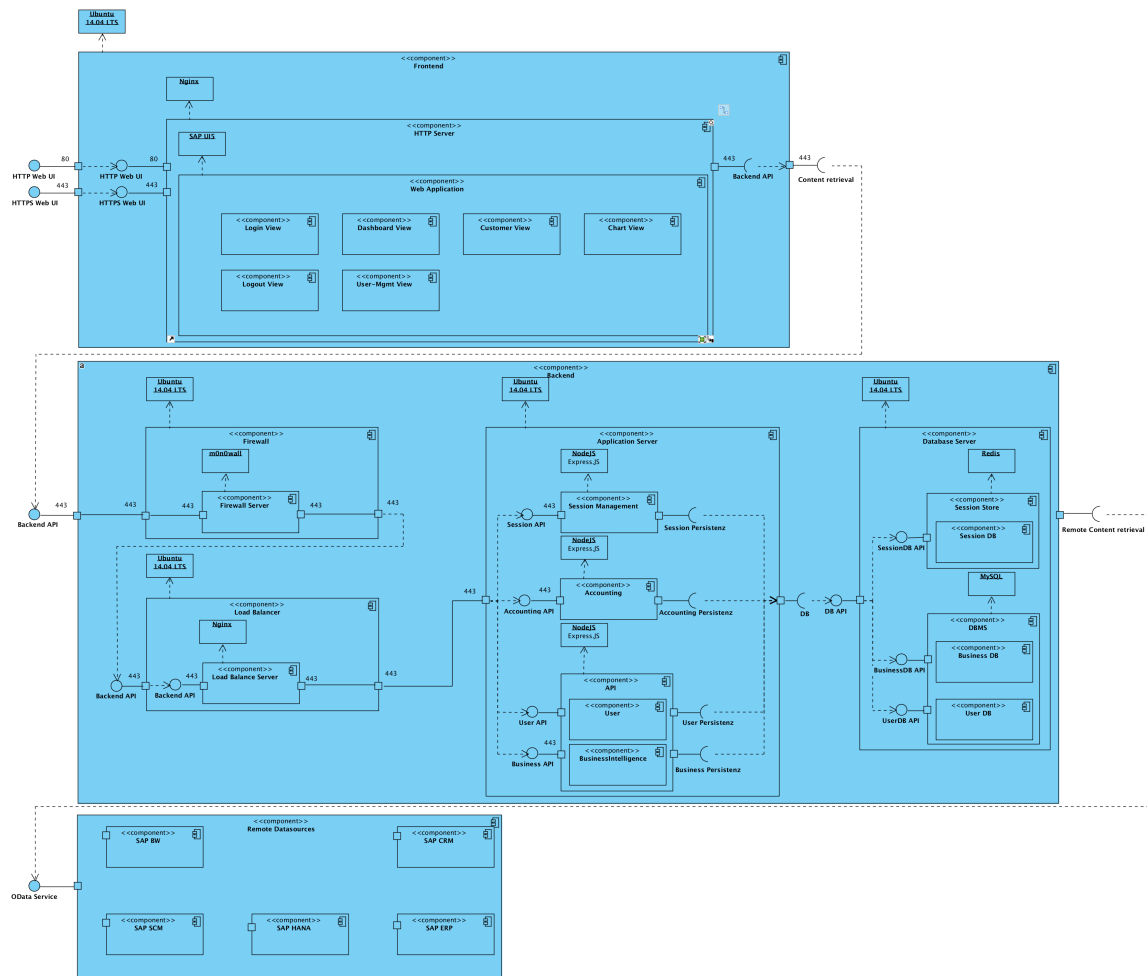


Abbildung 6.2: Komponentendiagramm des Systems

6.2.2 Datenmodell

Die Abbildung 6.3 zeigt das Datenmodell in Form eines Entity-Relationship-Diagramms. Die Entität User beinhaltet alle Attribute des Anwenders, die er selbst zum größten Teil selbst über einen Self Service-Dialog ändern kann. Zusätzlich werden in dieser Tabelle Metainformationen (Erstelldatum, Ersteller, Änderungsdatum, Änderer) gespeichert. Die drei Entitäten Grundpreis_je_kwh, Arbeitspreis_je_kwh und Netznutzungspreis_je_kwh werden zur Berechnung des Deckungsbeitrags benötigt.

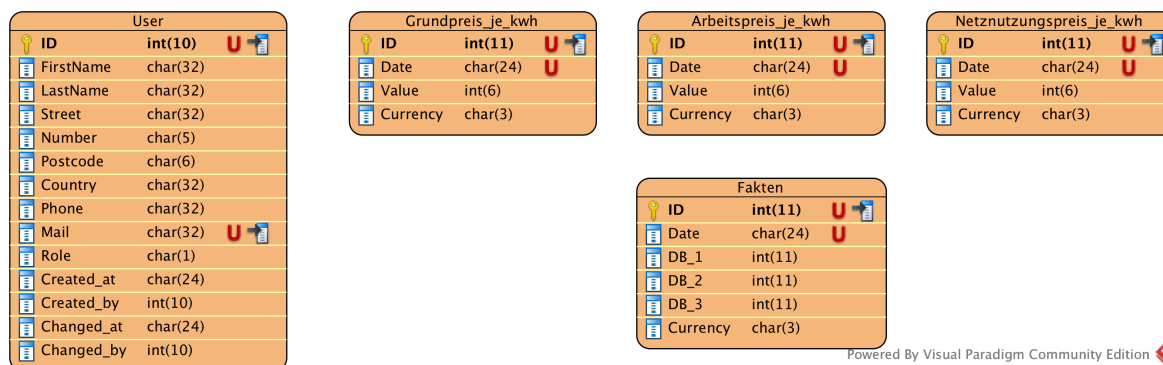


Abbildung 6.3: Entity-Relationship-Diagramm

6.2.3 SAPUI5 / OpenUI5 Toolkit

Bei SAPUI5 handelt es sich um ein Toolkit, das auf HTML5 und JavaScript basiert und von SAP entwickelt wird. Der Kern dieses Toolkits wurde im Jahr 2013 als Open Source-Variante unter dem Namen OpenUI5 veröffentlicht. Für die Entwicklung des CAIS-Frontends wurde diese Variante verwendet. Bei UI5 handelt es sich um ein sogenanntes One-Stop-Shop-Toolkit, das heißt, dass keine weiteren Bibliotheken benötigt werden. Sollte für einen sehr speziellen Anwendungsfall doch etwas fehlen, besteht jedoch die Möglichkeit, viele weitere Bibliotheken einzubinden. [BRW16]

SAPUI5 ist Teil von SAPs neuer User Experience-Strategie Fiori und bildet die technische Grundlage. Sämtliche Oberflächen der Applikationen des SAP ERP-Nachfolgers S/4 HANA werden mit ebendieser Technologie gestaltet. [BRW16] SAPUI5- und OpenUI5-Anwendungen sind auf der SAP HANA Cloud Platform, dem SAP NetWeaver Application Server ABAP oder dem SAP NetWeaver Application Server Java und auch auf Open Source-Plattformen lauffähig. [Ant16]

6.2.4 UI-Mockups

Nach der Aufnahme sämtlicher Anforderungen des Kunden wurden zunächst Mockups der Benutzungsoberflächen erstellt. Die Mockups dienen dazu, dem Kunden einen ersten,

groben Eindruck der späteren Applikation vermitteln zu können und Änderungswünsche zu diskutieren bevor mit der eigentlichen Entwicklung der Oberflächen begonnen wird.

Die Abbildung 6.4 zeigt exemplarisch, wie ein SAP Fiori Login Screen beziehungsweise Launchpad aussehen würde. Anhand der unterschiedlichen Endgeräte wird außerdem aufgezeigt, dass die Softwarelösung der Projektgruppe ein responsives Design aufweist und sich daher automatisch an unterschiedliche Displaygrößen und auch an die Bedienung mit einem Finger statt einer Maus anpasst.



Abbildung 6.4: UI-Mockups des Login-Screens und Launchpads

In Abbildung 6.5 ist zu sehen, dass sowohl die tabellarische Darstellung der benötigten Informationen als auch eine grafische möglich ist. Auch hier passt sich die Softwarelösung

wieder an die unterschiedlichen Displaygrößen an. Es empfiehlt sich jedoch besonders für die Diagrammansicht ein Tablet zu verwenden, um die angezeigten Daten korrekt lesen und interpretieren zu können.



Abbildung 6.5: UI-Mockups der Anwendung

7. Implementierung des Prototypen

Dieses Kapitel beschreibt das Handwerkszeug, in Form von Technologien und Software-Tools, das bei der Implementierung des Prototypen verwendet wurde.

7.1 Entwicklungsumgebung

Da mit den unterschiedlichsten Technologien gearbeitet wurde, sind auch mehrere Entwicklungsumgebung zum Einsatz gekommen. Im Frontend-Bereich wurde die WebIDE von SAP verwendet. SAP WebIDE ist eine gehostete IDE in der Public Cloud von SAP. Die WebIDE unterstützt den Entwicklungsprozess unter anderem mit einem Layout-Editor zum Gestalten der Oberflächen. Als Einschränkung können mit der WebIDE nur UI5-Applikationen entwickelt werden.

Im Backend Bereich wurde mit WebStorm von JetBrains entwickelt. WebStorm ist eine IDE für den Web-Bereich und kann mit nützlichen Integrationen dienen.

7.2 Infrastruktur

Die Infrastruktur für das System ist aus verschiedenen Komponenten aufgebaut. Das Kapitel soll die gewählten Bestandteile aufzeigen und im Detail erläutern.

7.2.1 Cloud

Die gestellten Anforderungen an das Projekt haben dazu beigetragen, dass sämtliche Infrastruktur-Komponenten in die Cloud ausgelagert wurden. Dabei sind zwei unterschiedlich dimensionierte Cloud-Ressourcen zum Einsatz gekommen. Zum einen standen Ressourcen vom Hasso-Platner-Institut zur Verfügung und zum anderen konnte auf Ressourcen der Universität zurückgegriffen werden.

HPI Ressourcen

Das Hasso-Platner-Institut (HPI) hat vier Maschinen mit wählbarer Ausstattung und eine SAP HANA Datenbank bereitgestellt. Zugang zu den Maschinen wurde über ein VPN gewährt. Die vier Maschinen besitzen dabei folgende Eigenschaften:

- 8 Kerne
- 64 GB Arbeitsspeicher
- 256 GB SSD Speicher

Die SAP HANA Datenbank-Instanz hat folgende Ausstattung:

- SUSE Linux Enterprise Server 11.4
- HP Enterprise Hardware
- AFL Plugin

Die HANA-Instanz konnte nicht dimensioniert werden. Um Zugang zu den Ressourcen des HPI zu bekommen, ist eine Verbindung zum VPN nötig.

Uni VM

Die Universität hat eine virtuelle Maschine bereitgestellt. Die Maschine ist wie folgt ausgestattet:

- 8 Kerne
- 4 GB Arbeitsspeicher
- 300 GB HDD Speicher

7.2.2 Microservices

Um die zur Umsetzung nötige Infrastruktur möglichst von der darunterliegenden Hardware zu trennen, wurde auf Microservices zurückgegriffen. Microservices sind Kleinstkomponenten von ausführbarer Software. So werden in einer monolithischen Architektur meist alle funktionalen Komponenten in einem System beziehungsweise Prozess zusammengelegt. Microservices brechen diesen Ansatz auf und lagern jede Funktionalität in ein eigenes System aus. Diese Systeme werden Container genannt. Container werden aus Images erzeugt, wobei das Image selbst jedoch nicht verändert wird während ein Container ausgeführt wird. Der Vorteil von Microservices besteht darin, dass sie auf jedem Container-Host ausgeführt werden können ohne darauf achten zu müssen, ob die Hardware oder das Betriebssystem die nötigen Abhängigkeiten erfüllt. Der Container beinhaltet sämtliche Abhängigkeiten, die nötig sind, um seine spezifische Aufgabe ausführen zu können. Im Folgenden sollen kurz die verwendeten Images und die daraus generierten Container und ihr Einsatzzweck im System erläutert werden.

Images

Service	Basis Image	verlinkt mit
Gitlab	Ubuntu 14.04	PostgreSQL, RedisDB
PostgreSQL	Ubuntu 14.04	Gitlab
RedisDB	Ubuntu 14.04	Gitlab
HAProxy	Ubuntu 14.04	Nginx
Nginx	Ubuntu 14.04	HAProxy
MongoDB	Ubuntu 14.04	Nginx

Tabelle 7.1: Docker Images

Container

Aus den aufgeführten Images wurde ein System für das Versionsmanagement. Zusätzlich eine dokumentenorientierte NoSQL Datenbank und die eigentliche skalierbare Web-Applikation mit einem Load Balancer. Dabei besteht das Versionsmanagement aus dem Gitlab-Service, der zum Arbeiten eine relationale Datenbank benötigt - in diesem Fall PostgreSQL. Die Weboberfläche von Gitlab benötigt außerdem einen Key-Value-Store, der mit dem RedisDB Image abgebildet ist. Das Nginx Image hat das API bereitgestellt. Für die Persistenz ist die MongoDB zuständig. Das HAProxy Image stellt den Load Balancer dar und wird mit den vorhandenen Nginx Instanzen verlinkt. Eintreffender Traffic wird dann vom HAProxy per Algorithmus an alle verlinkten Container verteilt. Abbildung 7.1 zeigt grafisch die Verlinkung von Docker Container am Beispiel des Gitlab Services.

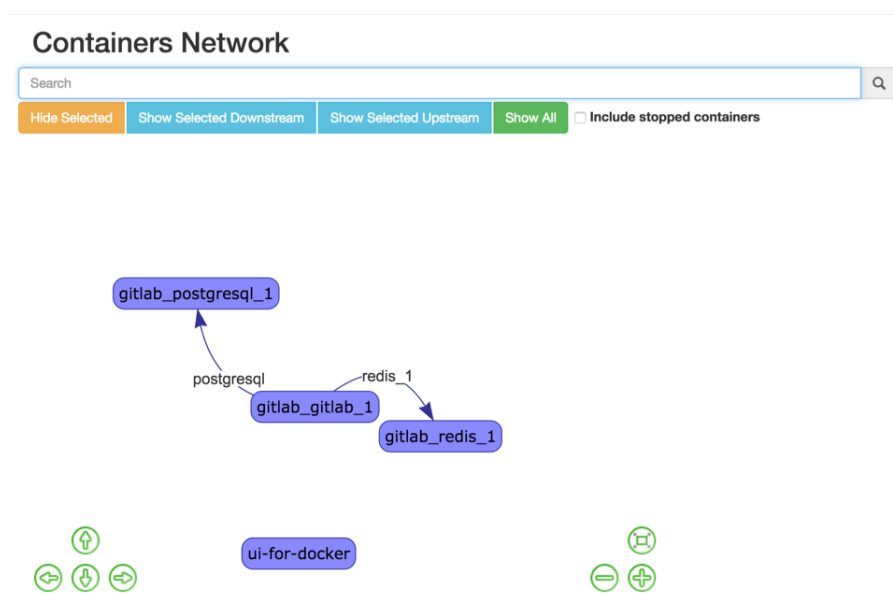


Abbildung 7.1: Docker Container Netzwerk

7.3 Backend-Implementierung

Im Folgenden soll ein genauer Überblick über den Backend-Prototypen gegeben werden. Dabei werden zunächst die Architektur und genutzten Technologien beschrieben. Danach wird ein Überblick über die wichtigsten Funktionen und Algorithmen des Prototypen gegeben.

7.3.1 Architektur

Das Backend des Projektes bildet eine API, die dem Frontend alle benötigten Funktionen zur Verfügung stellt. Da es sich bei dem Projekt um eine Webanwendung handelt, folgt die API dabei dem REST-Paradigma. Die dadurch erzeugte strenge Trennung von Client (Frontend) und Server (Backend) ermöglicht es somit einen modularen Aufbau zu gewährleisten. Es ist also möglich, die API auch ohne das gelieferte Frontend oder mit einer anderen Version des Frontends zu bedienen. Dies garantiert somit auch eine erhöhte Wartbarkeit und Skalierbarkeit beider Module, sowie eine einfache Wieder- und Weiterverwendung innerhalb anderer Projekte.

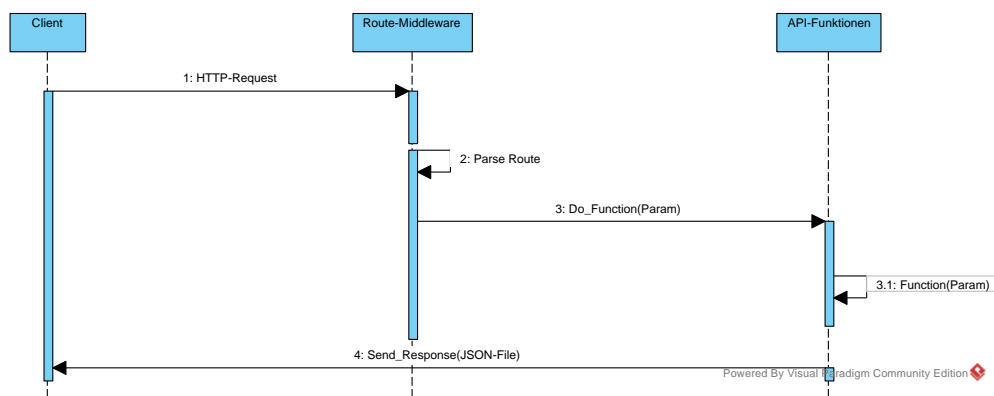


Abbildung 7.2: API-Ablauf

Die API bietet somit einheitliche Schnittstellen für jede beliebige Frontend-Anwendung. Um die Funktionen der API nutzen zu können, müssen die gegebenen Routen der API mittels einer HTTP-Anfrage angesprochen werden. Die Abbildung 7.2 skizziert den Ablauf der Verarbeitung einer solchen Anfrage. Diese wird über eine Zwischenschicht an die jeweilige Funktion weitergeleitet und bearbeitet. Als Antwort werden die Daten in einem JSON-Format zurückgegeben, die dann frei verwendet werden können.

Die Abbildung 7.3 gibt eine genaue Übersicht über den Umfang der im Projekt untersuchten API-Routen.

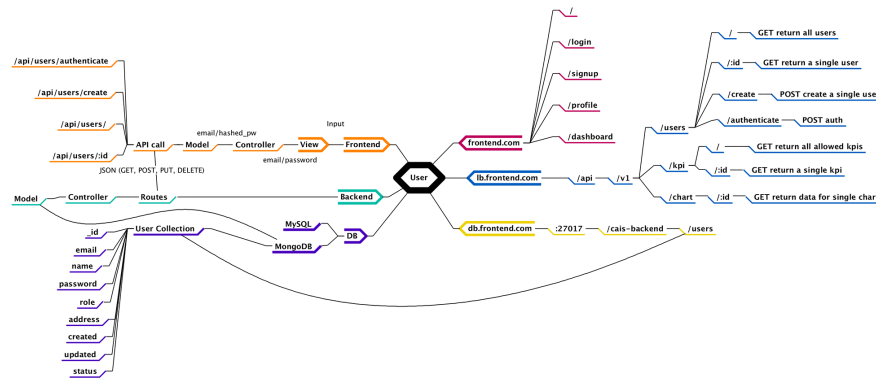


Abbildung 7.3: API-Routen

7.3.2 Technologien

Im folgenden Abschnitt sollen allen genutzten Technologien aufgelistet und dargestellt werden, die innerhalb des Projektes benutzt werden. Außerdem soll verdeutlicht werden, warum die Entscheidung auf eben diese Technologien fiel.

Programmiersprachen

Als Programmiersprache wurde innerhalb der Backend-Entwicklung einheitlich nur JavaScript verwendet. Durch das Verwenden des Frameworks Node.js ist es möglich, neben der Client-Anwendung JavaScript ebenfalls auf dem Server zu nutzen.

Der Kommunikationsdatentyp ist das JSON-Format, das sich durch die einfache Lesbarkeit und Einbindung innerhalb von JavaScript als geeignet herausstellt.

Webserver

Ziel des Projektes ist es, eine performante skalierbare Cloud-Anwendung zu entwickeln. In Hinblick auf diese Ziele ist die Entscheidung des zu nutzenden Frameworks auf Node.js gefallen. Im Gegensatz zu PHP bietet Node.js eine Grundlage für performante Anfrageverarbeitung und unterstützt insbesondere dynamische Webseiten. Dies entspringt der Tatsache, dass Node.js JavaScript sowohl client- als auch serverseitig benutzt. Damit kann die Event-orientierte Eigenschaft von JavaScript in Gänze ausgenutzt werden und so eine performante Lösung entwickelt werden. Des Weiteren ist es bei einer häufigen Kommunikation mit einer Datenbank, wie es in diesem Projekt benötigt wird, von Vorteil. Die Module-Struktur des Frameworks ermöglicht, aus einer Großzahl von vorhandenen Funktionen beliebige mit dem Projekt zu verknüpfen und so die Programmierung auf das Wesentliche zu fixieren. Es hat sich insbesondere herausgestellt, dass Node.js sich gerade für skalierbare Anwendung eignet. Durch seine Architektur wird die Kommunikation über kontinuierliche Kanäle offen gehalten, sodass es auch bei systemkritischen Anfragen nicht zu einem Blockverhalten kommt. Das

Framework bietet somit auch zusätzliche Sicherheit für eine größere Version des Projektes. [HCWD12]

Um eine moderne Web-Anwendung entwickeln zu können, muss neben responsivem Design und performanter Verarbeitung auch auf ein intuitives Design geachtet werden. Für diese Anforderung ist die Entscheidung auf das Express-Framework gefallen. Es ist speziell auf die Web- und Mobil-Entwicklung ausgerichtet. Express synergisiert vorteilhaft mit Node.js. Da Node.js sowohl Webserver als auch Client-Anwendung auf dem Server hält, können so beide Seiten einen geteilten Code nutzen. Mit Hilfe von Express werden so alle wichtigen und ressourcenlastigen Bearbeitungen auf dem Server verarbeitet. Der Webbrowser erhält schließlich eine fertige Version der Webseite, ohne großen Rechenaufwand. Express bietet insbesondere eine Middleware für Routen-Verarbeitung, die für die Umsetzung der REST-API dieses Projekts ausgezeichnet geeignet ist. Es unterstützt ebenfalls die Verarbeitung von HTTP-Anfragen und Verwaltung von Sessions. [Mar14]

Authentifizierung

Grundlage des Projektes bildet die Entwicklung einer kostenanalytischen Webanwendung. Hierbei handelt es sich um das Abbilden und Sammeln firmeninterner sicherheitskritischer Daten. Damit verbunden geht eine Anforderung an hoher Sicherheit einher. Um diese Anforderung zu bedienen, wurde sich innerhalb des Projektes auf das Sicherheitsprotokoll OAuth 2.0 festgelegt. Mittels des Moduls Passport wird so innerhalb der entwickelten Node.js API eine Authentifizierungs-Middleware integriert. Dies bietet zwei grundlegende Vertrauensbasen. Zum einen wird das Passwort des Nutzers nach seiner Erstellung nie im Klartext übermittelt, wie es bei einfacher JavaScript-Kommunikation durchaus vorkommen kann. Zum anderen wird es innerhalb des "OAuth 2.0"-Protokolls vermieden, das Passwort überhaupt zu verschicken. Die Verschlüsselung des Passwortes läuft innerhalb der Anwendung über einen SHA-256-Hash-Algorithmus.

Wie in Abbildung 7.4 zu sehen, ist es dem Nutzer nach erfolgreicher Authentifizierung möglich, ein Authentifizierungs-Token von einer sicheren Instanz, hier der API, zu erhalten. Mit diesem Token ist es möglich, jeden Dienst der API nutzen zu können, ohne eine weitere Authentifizierung notwendig zu machen oder das Nutzerpasswort angeben zu müssen. So können auch bei späteren Erweiterungen der API ohne großen Aufwand diese Sicherheitsaspekte eingepflegt werden. Es ist so theoretisch ebenfalls möglich, weitere externe Dienste anzubinden, die den Token akzeptieren. So wird ebenfalls vermieden, Nutzerdaten an Dritte weiterzugeben. Dieses Vorgehen bietet außerdem noch einen weiteren Vorteil hinsichtlich des Skalierungsfaktors: Dank des Token-Prinzips kann der Nutzer auf jede beliebige Instanz der API zugreifen. Es ist also keine Sessionverwaltung notwendig. Jede Anfrage des Clients kann also immer an eine andere Instanz der API weitergeleitet werden. Der Nutzer hat dennoch das Gefühl, mit nur einer Instanz zu kommunizieren.

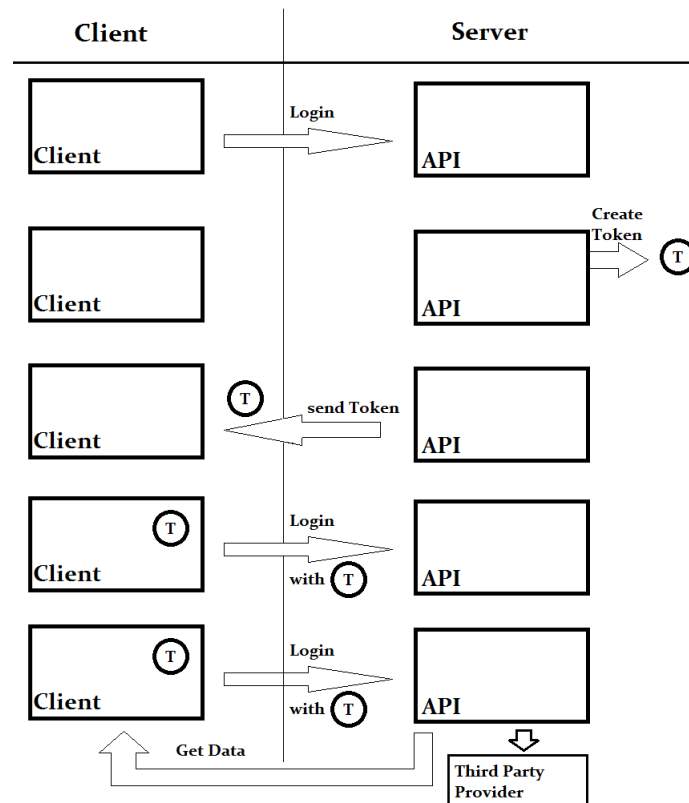


Abbildung 7.4: Token

Datenbankanbindung

Die Datenbankkommunikation der API zur HANA-Datenbank wird mittels einer speziellen Kommunikations-Middleware (*SAP HANA Database Client for Node*) geregelt. Die Kommunikation mit der Datenbank läuft dabei kaskadiert mittels Callbacks ab. Dies ist nötig, da gerade bei größeren Datenmengen die angefragten Daten nicht sofort bereit stehen, da sie erst komplett gelesen oder entsprechend berechnet werden müssen. Jede Anfrage an die Datenbank läuft dabei wie in Abbildung 7.5 ab.

```
// connect.js

//Erzeuge einen Asynchronen Ablauf
async.waterfall([connect, execute, fetchRows], done);

// 1. Stelle Connect zur Datenbank her
function connect(cb) {
  client.connect(cb);
}

// 2. Statement
function execute(cb) {
  client.execute(sqlStmt, cb);
}

// 3. Lese Werte aus
function fetchRows(rs, cb) {
  rs.fetch(cb);
}

// 4. Fehler- oder Abschlussbehandlung
function done(err, rows) {
  if(err)
    doSomething();
  else {
    doSomethingElse();
    client.end() //Disconnect
  }
}

}
```

Abbildung 7.5: Code-Beispiel: Datenbankkommunikation

7.3.3 Prototyp

Im folgenden Kapitel sollen Funktionen und detaillierte Erklärungen über die Funktionsweise des Backend-Prototypen gegeben werden. Hierbei wird zunächst ein grober Überblick über alle Module gegeben. Es folgt eine Beschreibung des Self-Services. Danach folgt eine Erläuterung der Datenverarbeitung und der Abrechnungsfunktion.

Module-Überblick

Zunächst sei mit Abbildung 7.6 ein Überblick über die wichtigsten Komponenten der API gegeben. Hierbei werden die einzelnen Modelle und Controller in Verbindung gebracht. Die wichtigsten Modelle der API bilden die *User* und die *Costumer*. Sie dienen mit ihrer ID als Ankerpunkte für die weiteren Daten. Bedient werden User von dem *Change*-Controller, der alle wichtigen Funktionen zum Ändern des jeweiligen Nutzers bereitstellt (siehe auch 7.3.3). Der *CostumerMethods*-Controller bietet dagegen alle Funktionen, die wichtig sind,

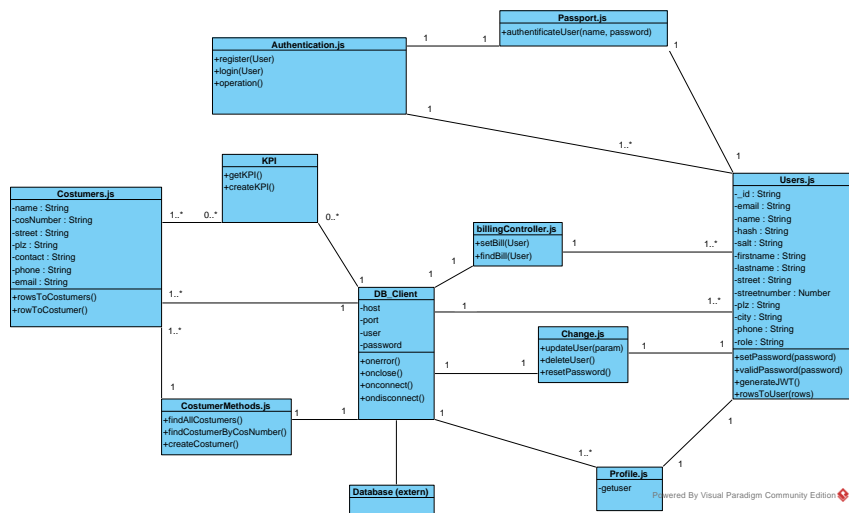


Abbildung 7.6: API-Module

um Kundeninformationen zu verarbeiten. Anhand der Kunden sind jeweils die Kennzahlen geknüpft.

Selfservice

Damit nur autorisierte Personen die Funktionen der API nutzen können, ist eine Userverwaltung wichtig. Der Prototyp bietet dafür zunächst die Möglichkeit, einen Account zu erstellen. Dabei stellt der Anwender einen POST-Request (siehe Abbildung 7.7) an die API. Über das Frontend werden gewünschter Nutzernamen und Email innerhalb des Requests verschickt. Besonderheit ist hierbei, dass das eingegebene Passwort nicht direkt in die Datenbank

```
router.post('/users/register', ctrlAuth.register);
```

Abbildung 7.7: Code-Beispiel: POST-Request Registrierung

geschrieben wird, sondern zunächst mittels SHA256 gehasht wird. Hierbei wird ein Hash und ein Schlüssel (*salt*) erzeugt. Nur mittels dieser beiden Informationen ist es möglich, beim Login-Vorgang das Passwort zu verifizieren.

Möchte sich der Nutzer einloggen, so muss er beim erstmaligen Versuch seine Nutzerdaten korrekt eingeben. Für alle weiteren Besuche und Anfragen an die API besitzt er nun einen Token, mit dem er sich authentifizieren kann. Das Passwort wird dabei mittels des Hashs überprüft.

Damit der Nutzer weitere persönliche Daten für sein Profil angeben kann, steht die Möglichkeit zum Ändern dieser Daten bereit. Hierbei werden mittels eines PUT-Requests die zu ändernden Daten verschickt. Die API gleicht neuen Daten mit den alten ab. Bei Änderungen werden die neuen Daten eingefügt. Sollte sich ein Teil des Namens (zum Beispiel

der Nachname) geändert haben, wird der volle Name automatisch aktualisiert.

Zum Löschen des eigenen Accounts kann der Nutzer einen DELETE-Request an die Users-Route schicken. Dabei wird die persönliche Objekt-ID des Nutzers als Schlüssel verwendet. Ist die Löschung bestätigt, kann so der Nutzereintrag komplett aus der Datenbank entfernt werden.

Kundeninformationen

Die Darstellung der Kunden ist für das Projekt eine essentielle Anforderung gewesen. Den Nutzern der API ist es somit möglich, neue Kundeninformationen anzulegen oder abzurufen. Hierbei kann der Nutzer sowohl alle Kunden im Überblick als auch einzelne Kundeninformationen über die jeweilige Kundennummer abrufen.

Datenverarbeitung

Um bei großen Datenmengen ein möglichst effizientes analytisches Informationssystem zu implementieren, steht die API mit einer SAP-HANA-Datenbank in Verbindung. Für die Kommunikation müssen dafür die tabellarischen Datenreihen in die für API und Client verständlichen JSON-Formate umgewandelt werden. Die Verarbeitung durchläuft dabei folgende Berechnung:

```
Data: HANA-Datenreihen  
Result: JSON-File  
initialization;  
Erstelle neues JSON-File;  
for jede Reihe i in Datenreihen do  
|   Erstelle neues JSON-Object[i];  
|   for jeden Wert j in Reihe do  
|   |   JSON-Object[i].Wert[j] := Reihe[i].Wert[j];  
|   end  
end
```

Algorithm 1: Datenverarbeitung aus der HANA

Abrechnung

Das Abrechnungssystem ist Teilanforderung des im Projekt angestrebten Cloud-Szenarios. Der Nutzer soll anhand des Gebrauchs des Systems eine bestimmte Rechnung erhalten. Das Abrechnungssystem innerhalb des Prototypen wird über Abrufe geregelt. Die Abrechnung

läuft somit nicht über ein gewisses zeitliches Intervall, sondern wird pro Aufruf von Daten berechnet. Hierbei wird vereinfacht folgender gewichteter Term genutzt:

$$\text{Rechnung} = (\text{Gewicht} * \text{Kosten}) + \text{Grundgebuehr} \quad (7.1)$$

Das Gewicht richtet sich dabei an die jeweilige Art des Abrufs. Handelt es sich um voraussichtlich kleine Datenmenge, wie die eigenen Nutzerinformationen, ist das Gewicht gering. Handelt es sich um die Abfrage aller Kundeninformationen oder gar um die Kennzahlen für bestimmte Kunden, so ist das Gewicht höher. Das Gewicht wird anschließend mit einer gewählten Konstante (Kosten) verrechnet. Ebenso ist das Nutzen einer Grundgebühr möglich.

7.4 Frontend-Implementierung

Im Folgenden soll ein genauer Überblick über die Frontend-Implementierung des CAIS-Projektes gegeben werden. Dabei wird zunächst auf die Projektstruktur innerhalb des SAP UI5-Projektes und anschließend auf ein Beispiel aus dem Coding der Frontend-Implementierung eingegangen.

7.4.1 Model-View-Controller-Architekturmuster

Das sogenannte Model-View-Controller-Architekturmuster strukturiert die SAP UI5-Entwicklung in die drei Einheiten Datenmodell (Model), Präsentation (View) und Steuerung (Controller). Durch diese Trennung können die einzelnen Komponenten leichter erweitert, ausgetauscht oder wiederverwendet werden.

Model Das Model in der SAP UI5-Entwicklung repräsentiert die Schicht des Datenmodells und hält die Applikationsdaten bereit. Von dem Model werden unter anderem auch die Datenzugriffe koordiniert. Ebenso befindet sich in dem Model die dahinterliegende Geschäftslogik, die für die Durchführung eines Prozesses genutzt wird. Vom Model gehen alle relevanten Daten aus.

View Die Präsentationsschicht ist für die Darstellung auf dem jeweiligen Endgerät (Smartphone, Tablet, Desktop) zuständig. In der View werden die Benutzerinteraktionen entgegen genommen, jedoch nicht verarbeitet. In der View werden die Benutzerinteraktionen an den jeweiligen Controller weitergeleitet.

Controller Der Controller verwaltet eine oder mehrere Views, nimmt die Benutzerinteraktionen entgegen und wertet diese aus. Zu jeder View existiert ein Controller. Sollte es bei

einer Benutzerinteraktion zu einer Datenänderung kommen, so übernimmt der Controller die Kommunikation zum Model.

7.4.2 Coding-Beispiel

Anhand dieser oben aufgeführten Grundlagen zum Aufbau einer SAP UI5-Applikation wird im Folgenden ein kleiner Teil als Beispiel aus dem Frontend-Coding der Projektgruppe dargestellt und näher erläutert.

Als Beispiel fungiert die Darstellung der einzelnen Daten im Diagramm, wie sie in Abbildung 7.8 zu sehen ist.

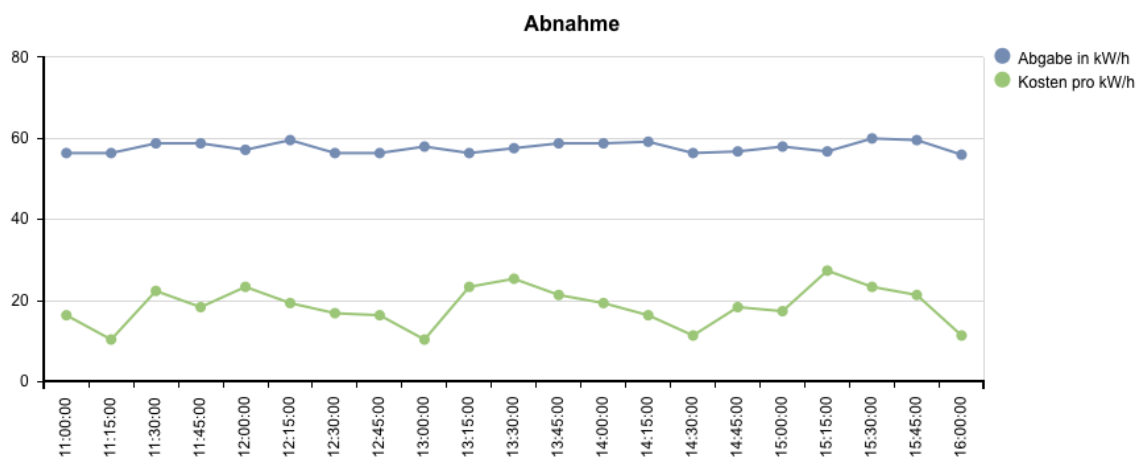


Abbildung 7.8: Beispieldiagramm aus dem UI

Auch in dieser Anwendung wird das zuvor beschriebene Model-View-Controller-Architekturmuster der SAP UI5-Entwicklung angewendet.

Model Folgende Daten (s. Abbildung 7.9) werden für die Darstellung der ForecastCharts in der CAIS-Applikation benötigt. Hier dargestellt ist lediglich ein einzelner Punkt auf dem Chart. Dabei ist der Profilwert die Grundlage der blauen ChartLine, Profilwert 2 die grün dargestellte ChartLine.

```
{
  "Ab-Datum": "01.01.2005",
  "Ab-Zeit": "11:00:00",
  "Profilwert": "56.4",
  "Profilwert2": "16.4"
},
```

Abbildung 7.9: Code-Beispiel: JSON-Model der anzuzeigenden Daten

View In der `Detail.view.xml` (Ausschnitt s. Abbildung 7.10) wird das `Forecast.Chart` in die Oberfläche eingebunden.

```
<core:FragmentDefinition xmlns:core="sap.ui.core"
  xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:viz="sap.viz.ui5" xmlns:types="sap.viz.ui5.types">
  <viz:Line id="lineChart" width="100%" height="400px"
    selectData="handleForecastDataSelect"></viz:Line>
</core:FragmentDefinition>
```

Abbildung 7.10: Code-Beispiel: Einbindung des Diagramms in die View

Controller In der `Detail.controller.js` (Ausschnitt s. Abbildung 7.11) werden alle relevanten Daten für die View des `ForecastChart` ausgewertet:

```
configQuantityChart : function() {
  var oDatasetForecast = new sap.viz.ui5.data.FlattenedDataset({
    dimensions : [ {
      axis : 1,
      name : 'Zeit',
      value : "{Ab-Zeit}"

    } ],
    measures : [ {
      name : 'Abgabe in kW/h',
      value : '{Profilwert}'
    },
    {
      name : 'Kosten pro kW/h',
      value : '{Profilwert2}'
    } ],
    data : {
      path : "QuantityData"
    }
  });

  var oLineChart = this.getView().byId("lineChart");
  oLineChart.setTitle(new sap.viz.ui5.types.Title({
    visible : true,
    text : "{i18n>ForecastChartTitle}"
  }));
  oLineChart.setDataset(oDatasetForecast);
}
```

Abbildung 7.11: Code-Beispiel: Controller zum Diagramm

7.5 Testing

Dieses Unterkapitel befasst sich mit allen evaluierungsbezogenen Themen, die mit der Entwicklung der Software einhergehen. Zunächst wird das allgemeine Vorgehen zu den Testing-Verfahren beschrieben, anschließend werden die durchgeführten Tests im Einzelnen erläutert. Dazu zählen Unit-Tests für Frontend und Backend sowie Systemtests in Form eines Testdrehbuchs.

7.5.1 Allgemeines Test-Vorgehen

Mit dem Testen einer Software soll unter anderem überprüft werden, ob diese der vorgegeben Softwarespezifikation entspricht und die zuvor im Pflichtenheft beschriebenen Funktionen auch ordnungsgemäß sowie frei von Fehlern ablaufen [Noe07], [Aki06]. Fehler werden dabei als „jede Abweichung der tatsächlichen Ausprägung eines Qualitätsmerkmals von der vorgesehenen Soll-Ausprägung, jede Inkonsistenz zwischen der Spezifikation und der Implementierung und jedes strukturelle Merkmal des Programmtextes, das ein fehlerhaftes Verhalten des Programms verursacht“ [Bal99] verstanden. Aus diesem Grund sollten durchgehend Tests parallel zur Konzeption, Spezifikation und Entwicklung der Software durchgeführt werden. [And03] Im Folgenden werden unsere durchgeführten Testarten erläutert sowie Auszüge unserer durchgeführten Tests gegeben.

7.5.2 Unit-Tests

Unit-Tests sind allgemein als ein wichtiges Mittel für die Verbesserung von Software anerkannt, wobei die einzelnen Methoden eines Moduls parallel zur Entwicklung getestet werden. [TDHX10], [And03] Dabei stellt ein Modul die kleinstmögliche zu testende Einheit eines Programms dar, bei dem ein erfolgreicher Test grün und ein nicht gelungener Test rot markiert wird. [JQu] Es wurden Unit-Tests für das Backend und Frontend entworfen und durchgeführt. Für das Frontend-Testing wurde QUnit für die Erstellung von Unit-Tests herangezogen, das nachfolgend behandelt wird.

Frontend-Testing

QUnit, das sich im Zuge von jQuery entwickelt hat, ist ein Open-Source-Framework. Die Installation ist mit sehr wenig Aufwand verbunden, da es sich bei QUnit um ein clientseitiges Testframework handelt. Bei der Verwendung von QUnit ist der Benutzer nicht an ein bestimmtes System gebunden. Das Vorhandensein eines JavaScript-fähigen Browsers ist die einzige Voraussetzung. QUnit ist eine eigenständige Bibliothek, die nur eine JavaScript-Datei (`qunit.js`) und eine CSS-Datei (`qunit.css`) für den Betrieb benötigt. Die JavaScript-Datei enthält den Quellcode von QUnit, der zusammen mit dem Stylesheet, das für die

Formatierung der Ergebnisse verantwortlich ist, über QUnit-Website¹⁵ bezogen werden kann. Zusätzlich zu diesen beiden Dateien ist noch eine Datei selbst zu erstellen. Diese Datei stellt den Test-Runner dar, der sowohl den Quellcode als auch das Stylesheet von QUnit einbindet und die Infrastruktur zur Ausgabe der Tests zur Verfügung stellt. Außerdem werden in dieser Datei der eigene JavaScript-Quellcode und die eigenen Tests eingebunden [Spr15].

Im Folgenden werden die von uns festgelegten Bedingungen und die Vorgehensweise für die Erstellung der durchgeführten Unit-Tests mit QUnit beschrieben:

Bedingungen:

- **Primär:** Durchführung von positiven Tests, das heißt, durch die Tests wird versucht die gestellten Anforderungen zu verifizieren und das korrekt erwartete Verhalten als Ergebnis zu erzielen.
- **Sekundär:** Durchführung von negativen Tests, das heißt, durch die Tests wird versucht absichtlich herbeigeführte Fehler aufzudecken und zu überprüfen, ob die Anwendung eine entsprechende Gegenmaßnahme in Form der Ausgabe einer Fehlermeldung ausführt. Diese Art von Tests, die neben den positiven Tests der Gruppe der funktionsorientierten Tests angehören, dienen der Prüfung der Robustheit unserer Anwendung.

Vorgehensweise:

1. Für jede zu testende Klasse wird eine Testklasse erstellt.
2. Für jede zu testende Methode innerhalb der zu überprüfenden Klasse wird eine Testmethode erstellt. Mit der Testmethode wird eine korrekte Funktionsweise der zu testenden Methode überprüft.
3. Test-Verifikation: QUnit Test Grün → erfolgreich oder QUnit Test Rot → nicht erfolgreich.

Zur Veranschaulichung werden in der Abbildung 7.12 ein paar Unit-Tests, die mit QUnit erstellt und durchgeführt wurden, dargestellt. Es wurden absichtlich fehlerhafte Tests erstellt, um die Darstellung bei erfolgreichen und nicht erfolgreichen Tests zu zeigen.

¹⁵<http://jquery.com/>

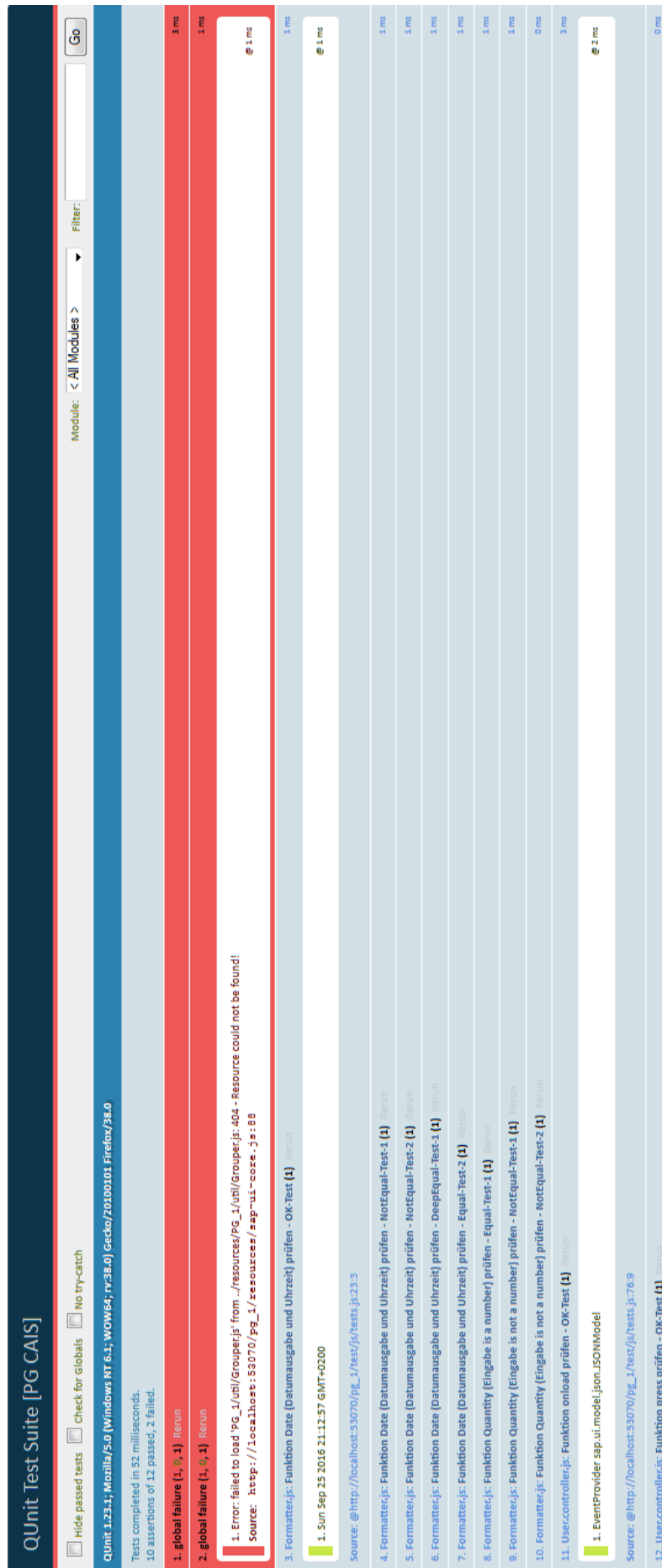


Abbildung 7.12: Auszug der QUnit Test Suite der PG CAIS

7.5.3 Systemtest

Der Systemtest, ist ein Test des komplett integrierten Systems als fertiges Produkt. Dies geschieht üblicherweise unmittelbar vor Auslieferung der Software an den Kunden. Zur Veranschaulichung wird in der Abbildung 7.13 das Testdrehbuch gezeigt, womit die Durchführung des Systemtests erfolgte.

Testfall-ID	Testfallname	Testfallkurzbeschreibung	Testgrundlage	Getestete Prototyp-Version	Priorität	Vorbedingung	Status Testdefinition	Tester	Testdatum	Sollverhalten	Testergebnis	Jira Ticket	Bemerkungen	Ausprägung
CAIS_TF_01	Kennzahlen darstellen	Das System muss sicherstellen, dass Kennzahlen dargestellt werden.	CAIS_REQ_101	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_02	Kennzahlen auswerten	Das System muss sicherstellen, dass die Kennzahlen in Abhängigkeit zu unterschiedlichen Dimensionen (Kunde, Quartal,...) ausgewertet werden können.	CAIS_REQ_102	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_03	Gesamtgewinn	Das System muss sicherstellen, dass der Gesamtgewinn errechnet wird.	CAIS_REQ_103	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_04	Profitauswertung	Das System muss sicherstellen, dass die Profite pro Kratwerk und pro Kunde ausgewertet werden.	CAIS_REQ_104	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_05	Kennzahlen Zeiträume	Das System muss sicherstellen, dass die Kennzahlen über verschiedene Zeiträume angezeigt werden (stundengenau, tagessgenau, pro Quartal).	CAIS_REQ_105	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_06	Visuelle Darstellung der Ergebnisse	Das System muss sicherstellen, dass durch den User im Rahmen der visuellen Darstellung der Ergebnisse folgende Schritte durchgeführt werden können: - Markierung der Ergebnisse frei wählbar (bspw. Kreis-/Balkendiagramm) - Markierung von besonders guten Werten (Gewinnzone) grün und besonders schlechte Werte (Verlustzone) rot	CAIS_REQ_106 CAIS_REQ_106-1 CAIS_REQ_106-2	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Fehlerhaft	-	Darstellung der Ergebnisse sind nicht frei wählbar (nur Liniendiagramm). Verlustzone (z.B. Gewinn- und Verlustzone) ist nicht angelegt (für die Abgabe in KUH und die Kosten pro kWh)	Core
CAIS_TF_07	Testdaten per Bulk Load	Das System muss sicherstellen, dass Testdaten per Bulk Load eingelesen werden können.	CAIS_REQ_301	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Fehlerhaft	-	Fehlverhalten ist an den fehlenden Rechten innerhalb der HANA zurückzuführen	Core
CAIS_TF_08	Accountverwaltung	Das System muss sicherstellen, dass durch den User im Rahmen der Accountverwaltung folgende Schritte durchgeführt werden können: - Änderung des Accountnamens - Angabe der Adresse - Änderung der Adresse - Änderung des Passworts	CAIS_REQ_401 CAIS_REQ_401-1 CAIS_REQ_401-2 CAIS_REQ_401-3 CAIS_REQ_401-4 CAIS_REQ_401-5	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_09	Falsche Zugangsdaten	Das System muss sicherstellen, dass im Rahmen der Accountverwaltung falsche Zugangsdaten eingegeben werden.	CAIS_REQ_401-6	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_10	Rechteverwaltung	Das System muss sicherstellen, dass eine Rechteverwaltung vorhanden ist.	CAIS_REQ_402	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_11	Session-Verwaltung	Das System muss sicherstellen, dass eine Session-Verwaltung vorhanden ist.	CAIS_REQ_403	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_12	Rollen-Prüfung	Das System muss sicherstellen, dass im Rahmen der Rechteverwaltung folgende Rollen vorhanden sind: - Admin-Rolle (User anlegen/freischalten) - Power-User-Rolle - Business-User-Rolle	CAIS_REQ_404 CAIS_REQ_405 CAIS_REQ_406	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_13	Speicherung Kundendaten	Das System muss sicherstellen, dass Daten eines bestimmten Kunden abgespeichert werden können.	CAIS_REQ_203	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_14	Kundendaten offline verfügbar	Das System muss sicherstellen, dass Daten eines bestimmten Kunden offline zur Verfügung stehen können.	CAIS_REQ_114	1.0	Mittel	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Nicht ausfüllbar	-	Nicht implementiert und kann als zukünftige Ausbaustufe betrachtet werden	Core
CAIS_TF_15	Browsierbarkeit	Das System muss sicherstellen, dass die Anwendung über alle gängigen Browser erreichbar ist.	CAIS_REQ_115 CAIS_REQ_XXX	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	Das System lief in Safari, Chrome, IE, Firefox und auf dem iPhone 7 Plus, iPhone 7, iPhone 6 Plus, iPhone 5s, iPhone 4s, iPad Air, iPad Pro und iPad mini	Core

Abbildung 7.13: Testdrehbuch zur Durchführung des Systemtests der PG CAIS

7.5.4 Backend-Testing

Im Folgenden sollen einige Testfälle und ihre Ergebnisse vorgestellt werden, die innerhalb der Test-Phase durchgeführt worden sind. Bei den Tests handelt es sich um Blackbox-Tests, die der simplen Überprüfung des Ein-/Ausgabeverhaltens der API dienen. Die Tests wurden in Form von Unit-Tests angelegt und durchgeführt.

Selfservice

Die Tabellen 7.14 und 7.15 zeigen einige Testergebnisse für die Selfservice-Funktionalität des Backends.

#	Klasse/ Grenze	Eingabe	Umgebung	Ausgabe	Erwartetes Verhalten	Ergebnis
1	gültig	POST /register JSON{ email: max@mustermann.de,password: test}	Leere User- Datenbank	JSON { email: max@mustermann.de, password: test,name:, firstname: lastname: street: sNumber: plz: city: phone:role:}	JSON { email: max@mustermann.de, password: test,name:, firstname: lastname: street: sNumber: plz: city: phone:role:}	Korrekt
2	ungültig	POST /register JSON{ email: max@mustermann.de,password: test}	Account bereits vorhanden	DB-FAILURE	"Account already exists"	Falsch
3	ungültig	POST /register JSON{ email: max@mustermann.de,password: } }	Leere User- Datenbank	"All fields required"	"All fields required"	Korrekt
4	ungültig	POST /register JSON{ email: .password: test }	Leere User- Datenbank	"All fields required"	"All fields required"	Korrekt
5	gültig	GET /users JSON{ email: max@mustermann.de,password: test}	Account vorhanden	JSON { email: max@mustermann.de, password: test,name:, firstname: lastname: street: sNumber: plz: city: phone:role:}	JSON { email: max@mustermann.de, password: test,name:, firstname: lastname: street: sNumber: plz: city: phone:role:}	Korrekt
6	ungültig	GET /users JSON{ email: max@mustermann.de,password: test}	Account nicht vorhanden	'User not found'	'User not found'	Korrekt
7	ungültig	GET /users JSON{ email: max@mustermann.de,password: wrongPassword}	Account erstellt	'Password is wrong'	'Password is wrong'	Korrekt
8	gültig	PUT /users JSON{ email: max@mustermann.de,password: wrongPassword, firstname: Max, lastname: Mustermann}	Account erstellt	"Successfully updated your Account!" JSON { email: max@mustermann.de,password: wrongPassword, firstname: Max, lastname: Mustermann, name: Mustermann_Max}	"Successfully updated your Account!" JSON { email: max@mustermann.de,password: wrongPassword, firstname: Max, lastname: Mustermann, name: Mustermann_Max}	Korrekt

Abbildung 7.14: Tabelle: Testen für Selfservice 1

#	Klasse/ Grenze	Eingabe	Umgebung	Ausgabe	Erwartetes Verhalten	Ergebnis
9	gültig	DELETE /users JSON{ email: max@mustermann.de,password: wrongPassword, token}	Account erstellt, eingeloggt, Token erhalten	"Successfully delete your Account!"	"Successfully delete your Account!"	Korrekt
10	ungültig	DELETE /users JSON{ email: max@mustermann.de,password: wrongPassword, Oldtoken}	Account erstellt, (eingeloggt), (falsches/altes Token)	"Account not found!"	"UnauthorizedError: not logged in"	Inkorrekte Ausgabe

Abbildung 7.15: Tabelle: Testen für Selfservice 2

Kundeninformationen

Die Tabelle in 7.16 zeigt ebenfalls einen Ausschnitt aus dem Testverfahren.

#	Klasse/ Grenze	Eingabe	Umgebung	Ausgabe	Erwartetes Verhalten	Ergebnis
1	gültig	GET /costumers JSON{Token}	Eingeloggt, keine Kundeninformationen vorhanden	JSON {}	JSON {}	Korrekt
2	ungültig	GET /costumers JSON{}	keine Kundeninformationen vorhanden	"UnauthorizedError: not logged in"	"UnauthorizedError: not logged in"	Korrekt
3	gültig	GET /costumers:CosNumber JSON{Token}	Kunde mit CosNumber in Datenbank	Kunde(CosNumber)	Kunde(CosNumber)	Korrekt
4	ungültig	GET /costumers:CosNumber JSON{Token}	Kunde mit CosNumber in Datenbank	DB_FAILURE	"Couldn't find costumer"	Falsch
5	ungültig	POST /costumers JSON{Token}	Datenbank leer	"Costumername and Costumernumber is required!"	"Costumername and Costumernumber is required!"	Korrekt
6	gültig	POST /costumers, JSON{cosName: Ikea, CosNumber: 12345 Token}	Datenbank leer	"Success! Costumer created!" JSON{cosName: Ikea, CosNumber: 12345 Token}	"Success! Costumer created!" JSON{cosName: Ikea, CosNumber: 12345 Token}	Korrekt

Abbildung 7.16: Tabelle: Testen für Kundeninformationen

8. Fazit und Ausblick

In diesem Kapitel soll ein kurzes Resümee der Projektgruppen-Arbeit gezogen werden. Es wird dargelegt, was zukünftige Projektgruppen in der täglichen Arbeit besser machen können bzw. vermeiden sollten. Zum Schluss wird noch ein kurzer Blick in eine mögliche Zukunft der in der PG entstandenen BaaS-Lösung gewagt.

8.1 Lessons Learned

Die meisten PG-Mitglieder haben bereits zuvor erste umfangreichere Erfahrungen mit Projektarbeit gesammelt. Dennoch war es für alle sieben PG-Mitglieder eine große Herausforderung, einen auf die Mitglieder zugeschnittenen optimalen Arbeits- sowie Organisationsprozess zu finden und zu leben. Das Arbeitsklima innerhalb der Gruppe war harmonisch und bei Fragen und Schwierigkeiten wurde stets untereinander ausgeholfen.

Das kollaborative Arbeiten, die Aufgabenverteilung und das Arbeiten mit zuvor unbekanntem Projektmanagement- und Entwicklerprogrammen und -methoden war für viele eine neue und zugleich wertvolle Erfahrung, die mit hohem Wert mit Blick auf den weiteren Berufsweg einzuschätzen ist. Die Aufgabenverteilung ist größtenteils gelungen, hätte aber durch eine präzisere Definition und Abgrenzung der Aufgaben noch verbessert werden können. Die Projektgruppe hatte sich bereits zu Beginn des Projekts einstimmig darauf geeinigt, keine Zeiterfassung vorzunehmen, sondern vertraute darauf, dass jedes Mitglied selbstständig die geforderten Arbeitsstunden in das Projekt investieren würde. Für Projektgruppen empfiehlt sich eine Zeiterfassung und -vorausplanung dennoch, um beispielsweise für Klausurenphasen eine niedrigere wöchentliche Stundenzahl zu planen, die in den davor- und dahinterliegenden Wochen ausgeglichen werden kann.

Der Einsatz von Projektmanagement- und Softwareentwicklungswerkzeugen wie JIRA, Confluence und Gitlab haben den Aufwand für die Koordination und Durchführung des Entwickler-Prozesses reduziert. Ein Teil der Mitglieder konnte bereits Erfahrungen mit einigen der Tools vorweisen und teilten dieses Wissen mit der Gruppe. Dennoch wurde die Dokumentation der individuellen Tätigkeiten in JIRA ab und an nicht konsequent durchgeführt beziehungsweise erfolgte in größeren Abarbeitungsblöcken. Die Informationssammlung und

-dokumentation mithilfe von Confluence stellte sich als die richtige Entscheidung heraus. Confluence ermöglichte es, über den gesamten Projektverlauf Informationen abzuspeichern und mit anderen Mitgliedern zu teilen und zu diskutieren. Als Kommunikationsmedium stellte sich der webbasierte Instant-Messaging-Dienst Slack als hervorragende Wahl heraus, da hiermit beispielsweise Fragen und Abstimmungen zwischen zwei Treffen virtuell gestellt beziehungsweise durchgeführt werden konnten.

Gut war auch der Erfahrungsgewinn über die Arbeit mit Scrum. Durch die vielen Varianten, die wir ausprobiert haben und der anschließenden Umstrukturierung gab es eine sehr gute Lernkurve. Negativ bezüglich der Scrum-Rituale war, dass die meisten Scrum-spezifischen Anwendungen eher schleppend und erzwungen liefen. So wurde beispielsweise Weekly Scrum teilweise nicht korrekt eingehalten. Im Bezug auf den Umfang des Projektes und dessen Umstände waren viele Scrum-interne Anwendungen zu zeitaufwändig. Scrum wurde genutzt und angepasst, konnte sich aber aufgrund unserer Arbeitsorganisation nie richtig etablieren. Ebenfalls kostete die Durchführung der Scrum-Aktivitäten über die Projektlaufzeit gesehen zu viel Zeit, die dringend für die zeitaufwändige Entwicklung benötigt wurde. Sämtliche an die Projektgruppe gestellten Anforderungen waren von vorne rein fix und haben sich unerwarteter Weise nicht fortlaufend auf Änderungen reagieren können. Für diese Voraussetzungen ist Scrum eher suboptimal, da durch die hohe Agilität jederzeit auf Änderungen reagiert wird. Scrum selbst fordert auch Erfahrung, die zum Start der Projektgruppe nicht vorhanden war. Scrum hätte daher in der Projektgruppe etabliert werden müssen bevor, die eigentliche Entwicklung begonnen wurde. Aufgrund dieser Gegebenheiten musste Scrum zeitgleich mit dem Prototyp entwickelt werden. Das Vorgehensmodell ist sehr interessant und hat seine Stärken, konnte diese aber innerhalb der Projektgruppe nicht ausspielen. Zukünftige Projektgruppen sollten zu Projektanfang ausreichend Zeit in Teambuilding und die Projekt- sowie Arbeitsorganisation investieren. Weiterhin sollte das richtige Verhältnis zwischen gemeinsamer Arbeitszeit und Arbeit im Home Office gewählt werden.

Wie bereits zuvor beschrieben, wurde als Technologie zur Entwicklung des Frontends Open UI5 verwendet. Unserer Meinung nach hat sich diese Entscheidung aus mehreren Gründen als die richtige herausgestellt. SAPUI5 beziehungsweise die in der Projektgruppe verwendete Open Source-Variante Open UI5 ist Teil der aktuellen SAP UX-Strategie und wird bereits für die neuen SAP-Benutzungsoberflächen verwendet. Würde unsere Lösung also bei einem Kunden produktiv eingesetzt werden, der bereits über ein SAP NetWeaver-System verfügt, kann sie ebenfalls auf einem SAP Fiori Launchpad deployt werden und Anwender müssten nicht zwischen ihrem Fiori Launchpad und einer weiteren Website wechseln und sich auch nicht an eine andere Optik und Bedienung gewöhnen. Auch ist die Integration zu anderen Systemen möglich. Sinnvoll wäre beispielsweise die Anbindung an ein CRM-System, um die benötigten Kundendaten nicht redundant pflegen zu müssen und damit mögliche Inkonsis-

tenzen zu vermeiden.

Im Hinblick auf moderne Technologien und Skalierung war die Wahl zum Aufbau einer REST-API mit Node.js eine gute Wahl. Dies folgt nicht nur aus den vielen Möglichkeiten, die die bereits vorhandenen Module bieten können, sondern auch die Effizienz, die dieses Framework für Webserver bietet. Auch die Auswahl des OAuth2.0 Protokolls zur Authentifizierung und Nutzung von Express als Framework haben sich als gute Möglichkeit zum Aufbau eines sicheren Webservers erwiesen. Die Möglichkeit, den Webserver auf mehreren Instanzen zugänglich zu machen, war dank der Umsetzung des Sicherheitsprotokolls sehr einfach. Die Umleitung und Verteilung der Anfragen auf die einzelnen Funktionalitäten der API wurden ebenfalls ausreichend vom Framework unterstützt. Die Verwendung des JSON-Datentyps zum Datenaustausch zwischen Client und Server ergab ebenfalls keine Nachteile. Die leichte Lesbarkeit dieses Datentyps und die Unterstützung durch Javascript haben nur Vorteile bei der Umsetzung geliefert. Einzig die Umstellung der Datenbank-Anbindung zwischen den Prototyp-Versionen auf die SAP-HANA-Plattform zeigte einen Mehraufwand, der allerdings nicht zu vermeiden gewesen wäre. Die Implementierung erfüllt also alle notwendigen Funktionalitäten und ist möglichst modular und übersichtlich gehalten. Die Implementierung hat allerdings ebenfalls genügend Ansatzpunkte für Erweiterungen. Die Architektur ist auf solche Erweiterung ausgelegt. Insbesondere der Aspekt der Skalierung wurde innerhalb der Implementierung besonders fokussiert und durchgesetzt. Da die genutzten Technologien für die Umsetzung zum ersten Mal verwendet wurden, konnten so auch viele neue Kompetenzen und Erfahrungen gesammelt werden, die insbesondere für eine moderne Webserver-Entwicklung sehr hilfreich sind.

Das Backend wurde sehr strukturiert erstellt und modular aufgebaut. Die Absprachen waren gut und die Aufgabenverteilung recht klar. Weniger gut waren die knappen Arbeitsressourcen, denn ein oder zwei zusätzliche Mitglieder wären durchaus hilfreich gewesen. Der recht späte Zugriff auf HANA-Instanz hat den Zeitplan stark durcheinander geworfen. Die Zusammenarbeit mit den Entwicklern des Frontends lief gut, jedoch mit viel Potenzial nach oben. Es herrschte eine recht gute Kommunikation sowie eine offene und angenehme Arbeitsweise. Es fanden regelmäßig mehrere interne Besprechungen und Absprachen mit dem Frontend-Team statt. Allerdings hätte ein größeres Einbinden der Teams in die Arbeit des jeweils anderen wahrscheinlich einen weiteren Gewinn, beispielsweise durch Team-Coding oder Mitgliedertausch, erzielt. Hier gibt es noch Verbesserungsbedarf. Das Zusammenführen zwischen Backend und Frontend gelang nur teilweise, da durch die späte Einführung der HANA-Instanz viele Funktionen, Umstrukturierung der Datentypen von NoSQL zu HANA noch einmal überarbeitet werden mussten. Der Fokus stand auf die fehlerfreie Ausführung beider Teilbereiche.

Direkt zu Projektanfang wurde die Entscheidung für die agile Entwicklung und Projektmanagement nach Scrum getroffen. Jedoch fand kein agiles Testen statt, da kein PG-Mitglied mit dieser Thematik stark vertraut war und ein unterschiedliches Verständnis von Testing vorlag. Eine testgetriebene Entwicklung kam ebenfalls nicht zum Einsatz, da dies sehr viel Erfahrung und Know-How benötigt, das zum damaligen Zeitpunkt nicht vorhanden war. Zu Anfang wurden Muster-Testdrehbücher für die Abdeckung von Systemtests/Funktionstests, ein Muster-Testbericht, eine Testfallverteilung und ein Testkonzept entworfen. Diese Idee beziehungsweise der Entwurf wurde nach interner Abstimmung zunächst zurückgestellt, da agil getestet werden sollte. Der Fokus lag somit auf den Unit-Tests, die zufriedenstellend mit gut gewählten Frameworks erstellt und durchgeführt wurden. Von dem ursprünglichen Entwurf wurde die Nutzung der Testdrehbücher für die Durchführung von Systemtests beschlossen, die sich unserer Meinung nach bewährt haben.

Eine große Herausforderung beim Testen stellten Problematiken mit der Hardware dar. Verbindungsfehler oder Unregelmäßigkeiten in der Hardware konnten Fehler bei den Tests erzeugen, die nicht auf Software-Ebene begründet lagen. Das Testen hat innerhalb der Projektgruppe viele neue Kompetenzen entstehen lassen und geholfen, eine eingehendes Befassen mit dem Quellcode durchzuführen. Aufgrund der Gruppengröße war es allerdings nicht möglich, das Testing in adäquatem und ausreichendem Umfang durchführen zu können. Zukünftigen Projektgruppen ist die sofortige Berücksichtigung von Testaspekten, die im Rahmen eines Projekts zu überlegen und anzugehen sind, zu empfehlen. Neben der Implementierung der gestellten Anforderungen ist das Testen der implementierten Funktionen mit seiner korrekten Funktionsweise von großer Bedeutung.

Im Nachhinein war die Projektgruppe eine sehr gute Erfahrung. Neben der Verbesserung der sozialen Kompetenzen wurden auch die fachlichen erweitert. In seiner Gesamtheit ist das Projekt gut umgesetzt worden, in Anbetracht der Teilnehmerzahl, da viele Dinge ausprobiert wurden. Ideen konnten frei diskutiert und Ansätze ausprobiert werden. Abschließend bleibt festzuhalten, dass die Durchführung einer solchen Projektgruppe sehr sinnvoll ist, da die Studierenden vielfältige Erfahrungen sammeln und die Möglichkeit erhalten, ihr vorhandenen Know-How in ein unbekanntes Thema einzubringen und die Einarbeitung in neue Projektmanagementmethoden- sowie Werkzeuge und Technologien ermöglicht bekommen. Außerdem werden die Studierenden durch die intensive Beziehung zu den Stakeholdern und Betreuern sehr gut auf den Berufseinstieg nach dem Studium vorbereitet.

8.2 Mögliche Ausbaustufen

Anhand der Anforderungen und der Architektur wurden User Stories definiert, die daraufhin, wie in Tabelle 8.1 nachzuvollziehen ist, zu Ausbaustufen zusammengefasst wurden. Die

Ausbaustufen richten sich hierbei nach den Zwischenstandspräsentationen und sind betitelt als Alpha, Beta und Final. Sie dienen als die drei Haupt-Meilensteine des Projekts.

Name	Nummer	Erweiterung	Sprint
Alpha	0.1	Webanwendung mit Dashboard	3
Alpha	0.1	User-Management-System	3
Alpha	0.1	Webanwendung mit Login-System	3
Beta	0.5	Hana Integration	4,5,6,7
Beta	0.5	Load Balancer	4,5,6,7
Beta	0.5	Deckungsbeitrag-Rechnung ausführbar	4,5,6,7
Final	1.0	Accounting	8,9,10,11
Final	1.0	OLAP-Funktionen	8,9,10,11

Tabelle 8.1: Komponenten und Umsetzungsentscheidungen

Mit dem Final-Prototypen als Ausgangspunkt sind folgende Ausbaustufen zukünftig sinnvoll und könnten von nachfolgenden Projektgruppen angegangen werden:

- Erweitertes Modul zur Berechnung von Kennzahlen
- Ausbau der administrativen Verwaltung von Nutzern
- umfangreicheres Rechtesystem
- Ausbau der Skalierungsalgorithmen der VMs
- Offline-Nutzung

Unabhängig von den bisherigen sowie zukünftigen Ausbaustufen sind generell drei Ebenen, dazu zählen die technische Umsetzung, Kundensicht und Vertriebsicht, gedanklich zu berücksichtigen. Für das Projekt gibt es zwei Sichten. Für die wissenschaftliche Sicht sollte das Projekt im Bezug auf die Skalierung erweitert werden. Insbesondere der Load-Balancer sollte erweitert und bearbeitet werden. Für die betriebswirtschaftliche Sicht sollte der Fokus auf den Ausbau der Funktionalität des Projektes gesetzt werden. Insbesondere auf eine effiziente Darstellung und Verarbeitung der Kennzahlen sollte großen Wert gelegt werden. Sinnvoll wäre auch die Überlegung, wie Daten von Neukunden erfasst werden können. Oder, dass statt dem bisherigen Szenario ein Lastgang pro Kunde, zukünftig eine Szenario-basierte Evaluation erfolgt.

Folgende Forschungsfragen im Rahmen des Projektthemas sollte man zukünftig angehen bzw. im Blick haben:

- Das Verhalten der Lastverteilung bei großen Anfragemassen, insbesondere im Bezug auf die Flaschenhals-Wirkung des Load-Balancers. Muss vielleicht selbst der Load-Balancer skaliert werden, d.h. ist mehr als ein Load-Balancer nötig? Wie sieht das davor geschaltete Verteilungssystem aus?
- Ist es lohnenswert Lasttests zu entwerfen und durchzuführen, d. h. wie verhalten sich die Antwortzeiten bei der Erweiterung von Tages- und Monatswerte um Jahres- und 10-Jahreswerte? Wie verhält sich das System beziehungsweise der Prototyp, wenn die Testdaten erweitert werden? Entstehen Beeinträchtigungen hinsichtlich der Performance des Systems beziehungsweise Prototypen?
- Zukünftig ebenfalls interessant sind heuristische Verfahren zum Skalieren abseits der üblichen Parameter. Kann eine intelligente Skalierung entwickelt werden?

Literaturverzeichnis

- [Aki06] AKINGBEHIN, K. (Hrsg.): *Towards Destructive Software Testing*. In 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, 7 2006 . – S. 374-377, Print-ISBN: 0-7695-2613-6
- [And03] ANDRESEN, Andreas: *Komponentenbasierte Softwareentwicklung mit MDA, UML 2 und XML* -. 1. Aufl. München : Hanser, 2003. – ISBN 978-3446222823
- [Ant16] ANTOLOVIC, Miroslav: *Einführung in SAPUI5*. Rheinwerk Verlag, 2016. – ISBN 978-3-8362-3908-0
- [AR10] ARCHIE REED, Stephen G. B.: *Silver Clouds, Dark Linings - A Concise Guide to Cloud Computing*. Amsterdam : Pearson Education, 2010. – ISBN 978-0-132-59964-1
- [Bal99] BALZERT, Helmut: *Lehrbuch Grundlagen der Informatik - Konzepte und Notationen in UML, Java und C++, Algorithmik und Software-Technik, Anwendungen*. Heidelberg [u.a.] : Spektrum Akademischer Verlag, 1999. – ISBN 978-3827403582
- [BKNT11] BAUN, Christian ; KUNZE, Marcel ; NIMIS, Jens ; TAI, Stefan: *Cloud Computing - Web-basierte dynamische IT-Services*. 2. Aufl. Berlin Heidelberg New York : Springer-Verlag, 2011. – ISBN 978-3-642-18436-9
- [BRW16] BEIER, Jens ; RÖCKELEIN, Wolfgang ; WASSILTSCHENKO, Alexander: *Mobile Geschäftsprozesse mit SAP*. Rheinwerk Verlag, 2016. – ISBN 978-3-8362-4028-4
- [DSC13] *Der Scrum Guide*. 2013. – <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf>, Stand: 21.08.2016
- [GC15] GLUCHOWSKI, P. ; CHAMONI, P.: *Analytische Informationssysteme: Business Intelligence-Technologien und -Anwendungen*. Springer Berlin Heidelberg, 2015. – ISBN 978-3-6624-7763-2

- [Glo13] GLOGER, Boris: *Scrum - Produkte zuverlässig und schnell entwickeln*. Carl Hanser Verlag, 2013
- [HCWD12] HUGHES-CROUCHER, T. ; WILSON, M. ; DEMMIG, T.: *Einfuehrung in Node.js*. O'Reilly, 2012. – ISBN 978-3-8689-9797-2
- [JQu] JQUERYFOUNDATION16: *QUnit.com: Introduction to Unit-Testing*. <http://qunitjs.com/intro/>, . – zuletzt aufgerufen am 29.05.2016
- [Lau11] LAUX, F.: *Analytische Informationssysteme*. http://dblab.reutlingen-university.de/tl_files/TeachingResources/CPMwithBI/AIS.pdf, 2011. – zuletzt aufgerufen am 11.09.2016
- [Mar14] MARDAN, A.: *Pro Express.js: Master Express.js: The Node.js Framework For Your Web Development*. Apress, 2014 (Expert's voice in Web development). – ISBN 978-1-4842-0037-7
- [MG11] MELL, Peter ; GRANCE, Timothy: *The NIST Definition of Cloud Computing / National Institute of Standards and Technology (NIST)*. Version: September 2011. <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. Gaithersburg, MD, September 2011 (800-145). – Forschungsbericht
- [MRV11] METZGER, Christian ; REITZ, Thorsten ; VILLAR, Juan: *Cloud computing - Chancen und Risiken aus technischer und unternehmerischer Sicht*. München : Hanser, 2011. – ISBN 978-3-446-42454-8
- [NA14] NORKUS, O. (Hrsg.) ; APPELRATH, H.-J. (Hrsg.): *Towards a Business Intelligence Cloud*. Lodz University of Technology; Lodz/Poland : Third International Conference on Informatics Engineering and Information Science (ICIEIS2014), September 2014 . – S. 55-66, ISBN: 978-8-132-22526-3
- [NJS15] NORKUS, O. ; JOSEFIOK, M. ; SAUER, J.: Einordnung der verschiedenen Dienstarten: Business Intelligence aus der Cloud. In: *BI-SPEKTRUM* (2015), Mai, S. 34-37. – Herausgeber: SIGS DATACOM GmbH
- [Noe07] NOE, Manfred: *Projektbegleitendes Qualitätsmanagement - Der Weg zu besserem Projekterfolg*. New York : John Wiley & Sons, 2007. – ISBN 978-3-895-78614-3
- [NS16] NORKUS, O. ; SAUER, J.: *Towards an Architecture of BI in the Cloud*, In: *Altmann, J., Silaghi, G.C., Rana, O. F. (Hrsg.): Economics of Grids, Clouds,*

- Systems, and Services - 12th International Conference, GECON 2015, Cluj-Napoca, Romania, September 15-17, 2015, Revised Selected Papers.* Berlin, Heidelberg : Springer, 2016. – ISBN 978–3–319–43177–2. – S. ?
- [SC112] *Agiles Testen in Scrum.* 2012. – http://www.sigs-datacom.de/uploads/tx_mwjournals/pdf/geisen_gueldali_0S_Agility_2012_8e65.pdf,
Stand: 21.08.2016
- [SCW16] *Scrum.* 2016. – www.upload.wikimedia.org/wikipedia/commons/thumb/a/ad/Scrum_process-de.svg/2000px-Scrum_process-de.svg.png,
Stand: 21.08.2016
- [Spr15] SPRINGER, Sebastian: *Testgetriebene Entwicklung mit JavaScript - Das Handbuch für den professionellen Programmierer.* 1. Auflage. Heidelberg : dpunkt.verlag, 2015. – ISBN 978–3–864–91648–9
- [TDHX10] TILLMANN, N. (Hrsg.) ; DE HALLEUX, J. (Hrsg.) ; XIE, T. (Hrsg.): *Parameterized Unit Testing: Theory and Practice.* In 32nd International Conference on Software Engineering, 2010 . – S. 483–484, Print-ISBN: 978-1-60558-719-6

Tabellenverzeichnis

2.1	Auflistung aller Mitglieder der Projektgruppe	6
2.2	Definition Sprint Planning	12
2.3	Definition Sprint Review	13
2.4	Definition Sprint Retrospektive	13
5.1	Übersicht über die funktionalen Anforderungen	31
5.2	Übersicht über die nicht-funktionalen Anforderungen	32
6.1	Komponenten und Umsetzungsentscheidungen	35
7.1	Docker Images	42
8.1	Komponenten und Umsetzungsentscheidungen	68

Abbildungsverzeichnis

2.1	Die Startseite	7
2.2	Der Scrum-Prozess	9
2.3	Der Scrum-Prozess ohne Testen	14
2.4	Der Scrum-Prozess inklusive Testen	15
3.1	Public Cloud, Private Cloud und Hybrid Cloud	19
3.2	Cloud Computing Architektur	20
5.1	Wertschöpfungskettendiagramm	27
5.2	Ereignisgesteuerte Prozesskette	28
5.3	Use Case-Diagramm zum Projekt	29
6.1	Architektur des Systems	34
6.2	Komponentendiagramm des Systems	36
6.3	Entity-Relationship-Diagramm	37
6.4	UI-Mockups des Login-Screens und Launchpads	38
6.5	UI-Mockups der Anwendung	39
7.1	Docker Container Netzwerk	42
7.2	API-Ablauf	43
7.3	API-Routen	44
7.4	Token	46
7.5	Code-Beispiel: Datenbankkommunikation	47
7.6	API-Module	48
7.7	Code-Beispiel: POST-Request Registrierung	48
7.8	Beispieldiagramm aus dem UI	51
7.9	Code-Beispiel: JSON-Model der anzuzeigenden Daten	51
7.10	Code-Beispiel: Einbindung des Diagramms in die View	52
7.11	Code-Beispiel: Controller zum Diagramm	53
7.12	Auszug der QUnit Test Suite der PG CAIS	56
7.13	Testdrehbuch zur Durchführung des Systemtests der PG CAIS	58
7.14	Tabelle: Testen für Selfservice 1	60

7.15	Tabelle: Testen für Selfservice 2	61
7.16	Tabelle: Testen für Kundeninformationen	63
1	Kundeninformationen anzeigen	clxviii
2	Deckungsbeitrag anzeigen	clxix
3	Self Service	clxx
4	Verwendung der Applikation auf einem Tablet	clxx
5	Verwendung der Applikation auf einem Smartphone	clxxi
6	Testdokumentation Deckblatt	clxxii
7	Testdokumentation Allgemeines	clxxiii
8	Testdokumentation Übersicht	clxxiv
9	Testdokumentation Auswertung	clxxv
10	Testdokumentation Listen	clxxv

Anhang

Seminararbeiten

Die Projektgruppe begann mit der Seminarphase in der jedes Projektgruppenmitglied eine Seminararbeit zu einem für die spätere Entwicklung relevanten Thema verfasst hat. Diese Seminararbeiten sind auf den folgenden Seiten zu finden.

Agile Vorgehensmethodik anhand von Scrum

Phillip Kramer¹

Abstract: Bei der Betrachtung bzgl. der Entwicklung des Prozessmodells in den vergangenen 30 Jahren ist auffallend, dass ein deutlicher Wandel einhergegangen ist. Agile Modelle wie Scrum rücken verstärkt in den Vordergrund. Scrum macht sich wie die weiteren agilen Modelle zur Aufgabe, das agile Manifest und deren Leitsätze umzusetzen. In der heutigen Zeit wird es immer wichtiger auf sich ändernde Anforderungen zu reagieren, wofür der Einsatz von Scrum besonders eignet ist. Durch den richtigen Einsatz von Scrum steigt die Mitarbeiterzufriedenheit, die Kundenzufriedenheit und es wird ein wirtschaftlicher Nutzen erzielt. Diese Seminararbeit zeigt auf, wie solch ein „richtiger Einsatz“ aussehen könnte.

Keywords: Agile Softwareentwicklung, Agile Manifest, Prozessmodell, Scrum, Scrum-Flow, Scrum-Projektrollen, Scrum-Artefakte

¹ Carl von Ossietzky Universität Oldenburg, Department für Informatik, Escherweg 2, 26121 Oldenburg,
phillip.kramer@uni-oldenburg.de

1 Einleitung

Im Rahmen der Seminararbeit in der Projektgruppe "CAIS - Cloudbasierte Analytische Informationssysteme", soll das Thema - Agile Vorgehensmethodik anhand von Scrum – vorgestellt werden.

Beginnend mit dem Phasenmodell in den 1970er Jahren über das Spiralmodell in den 1980er Jahren bis hin zum agilen Modell aus der heutigen Zeit, ist ein deutlicher Wandel bezüglich der Entwicklung des Prozessmodells einhergegangen. Die Entwicklung der agilen Modelle ist unter anderem der Tatsache geschuldet, dass Anforderungen an das Software-Produkt nicht mehr nur am Anfang des Projekts definiert, sondern diese fortlaufend bis hin zum Prozessabschluss geändert werden. Grundsätzlich wird zwischen schwergewichtigen Prozessmodellen (unter anderem V-Modell) und leichtgewichtigen Prozessmodellen (agile Modelle) unterschieden. Diese Seminararbeit handelt ausschließlich um die agile Vorgehensmethodik „Scrum“.²

Ziel dieser Seminararbeit ist es, das Agile Manifest aufzuzeigen und einen Überblick über den Scrum-Prozess zu geben. Anschließend ist es jedem frei überlassen, ob Scrum für zukünftige eigene Projekte eingesetzt werden soll.

Für die Anfertigung meiner Seminararbeit wurde Datenmaterial benötigt. Die meiste Verwendung hatte das Werk von Pichler, Roman: "Scrum - Agiles Projektmanagement erfolgreich einsetzen" und das Werk von Hanser, Eckhart: "Agile Prozesse: Von XP über Scrum bis MAP". Zu den bereits genannten Werken verwendete ich noch weitere Literatur in Form von Internetseiten.

Nach folgendem Schema ist die Seminararbeit aufgebaut. Nach der jetzigen Einleitung, die unter anderem die Geschichte des Prozessmodells beinhaltet, soll ein Einblick in agile Prozessmodelle und dem Agilen Manifest gegeben werden. Darauf folgend wird der Scrum-Prozess inkl. der Scrum-Projektrollen/-Artefakte näher gebracht. Anschließend zeige ich Handlungsempfehlungen im weiteren Sinne für die Nutzung von Scrum auf und im engeren Sinne nehme ich Bezug auf unsere aktuell bestehende Projektgruppe. Abschließend folgt das Fazit, das meine eigene Meinung beinhalten soll.

² Vgl. Hanser, Eckhart: Agile Prozesse, S.3 ff., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

2 Agile Prozessmodelle und das Agile Manifest

Im Gegensatz zu den beiden Prozessmodellen - V-Modell und dem Unified Software Development Process -, die als *schwergewichtig* einzustufen sind, gelten die agilen Modelle als *leichtgewichtig*. Die Besonderheit ist, dass nicht die Dokumentation, sondern funktionierender Code im Vordergrund steht. Agile Projekte sind keinesfalls anarchisch, denn weder ist die Vorgehensweise im Projekt chaotisch, noch ist es verboten, zu dokumentieren. Prozesse die dem *agilen* Manifest entsprechen, dürfen grundsätzlich als *agil* bezeichnet werden. Das *Agile Manifest* ist eine Reaktion der Entwickler auf die sog. schwergewichtigen Prozess- bzw. Vorgehensmodelle. Obwohl beide Ansätze sehr unterschiedlich sind, gelten beide als sehr dokumentenlastig, was zumindest bei Anfängern erheblichen zusätzlichen Aufwand bedeutet. Software-Entwickler haben die Aufgabe, Software-Lösungen zu designen und zu implementieren, doch eine umfangreiche Dokumentation ist ihnen eher lästig. Den Entwicklern die Notwendigkeit einer aufwändigen Dokumentation aufzuzeigen, verlangt große Überzeugungsarbeit. Das Agile Manifest zeigt einen Ausweg aus dieser Problematik.³

Es definiert vier Leitsätze für eine effiziente und erfolgreiche Software-Entwicklung. Die Leitsätze liest man "von rechts nach links". Das heißt, obwohl der Wert der Elemente auf der rechten Seite von Wichtigkeit ist, wird der Wert der Elemente auf der linken Seite mehr geschätzt.⁴

- "Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan"⁵

Die Interpretation des zweiten Leitsatzes ist besonders interessant, da dies eben nicht heißt, dass in agilen Projekten nicht dokumentiert werden muss. Es wird lediglich eine Aussage über die Priorität der Dokumentation gesagt. Das Agile Manifest stellt eine Art Randbedingungen der leichtgewichtigen Softwareentwicklung dar. Die diversen Prozessmodelle haben die Aufgabe, die agilen Leitsätze umzusetzen. Weiterführend wird das agile Prozessmodell „Scrum“ erörtert um zu verdeutlichen, wie genau agile Softwareentwicklung aussieht.⁶

³ Vgl. Hanser, Eckhart: Agile Prozesse, S.7 ff., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

⁴ Vgl. Beck K et al. (2001) www.agilemanifesto.org, Zugriff Oktober 2015

⁵ Beck K et al. (2001) www.agilemanifesto.org, Zugriff Oktober 2015

⁶ Vgl. Hanser, Eckhart: Agile Prozesse, S.10, 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

3 Scrum

3.1 Grundlegendes zu Scrum

Im Gegensatz zu den agilen Prozessmodellen wie das V-Modell und dem Unified Software Development Process, ist Scrum eine agile Projektmanagementmethode. Fragen der Zusammenarbeit und der Teamstruktur stehen nicht im Vordergrund, sondern der Ablauf des Projekts. Scrum, die agile Methode mit zunehmender Verbreitung, folgt der Idee des Lean Managements, des „schlanken“ Managements. In Scrum, das in den frühen 1990er Jahren entwickelt wurde, sind die festgelegten Projektrollen (s. Kapitel 3.2.1) und damit insbesondere das Fehlen einer „klassischen“ Rolle, der des Projektmanagers auffallend. Neben den Aufgaben der Projektrollen regelt Scrum den Projektablauf in Form von Sprints (s. Kapitel 3.2.2).⁷

Scrum ist ein agiles Managementframework, das aus wenigen klaren Regeln besteht und zur Entwicklung von Software dient. Diese beinhalten die Anwendung der Rollen (s. Kapitel 3.2.1), die Verwendung eines priorisierten Product Backlog (s. Kapitel 3.2.2) sowie das Erstellen von Produktinkrementen innerhalb kurzer Arbeitszyklen, die Sprints (s. Kapitel 3.2.2) genannt werden. Scrum lässt sich auf alle Arten der Softwareentwicklung anwenden. Als agiles Framework verkörpert Scrum die Werte des Agilen Manifest (s. Kapitel 2). Dieses stellt den Menschen in den Mittelpunkt der Softwareentwicklung, da schließlich die Software nur durch die Interaktion und Kollaboration von Menschen entsteht. Scrum fordert und fördert die enge Zusammenarbeit der Beteiligten und ist keinesfalls technologie- oder toolorientiert.⁸

Scrum ist ein empirischer Prozess, in der die Arbeitsweise und das Produkt regelmäßig begutachtet und angepasst wird. Scrum ist keine herkömmliche Methode und keine Komplettlösung. Es wird nicht detailliert vorgeschrieben, was wann zu tun ist, sondern die Kreativität der Mitarbeiter gefördert. Daher beinhaltet Scrum auch keine Verfahrensanweisungen oder Templates. Soweit es als hilfreich erachtet wird, müssen diese für das jeweilige Projekt und der zugrundeliegende Organisation selbst erarbeitet werden. Scrum ist kein Wundermittel, das, einmal in eine Organisation eingeführt, von selbst alles besser werden lässt. Das Gegenteil ist der Fall, da oft die ersten *Sprints* für Projektmitarbeiter und Management schwierig sind. Alle Beteiligten müssen nicht nur die neuen Spielregeln lernen, sondern auch alte Angewohnheiten ablegen. Das erfolgreiche Anwenden von Scrum ist also ein Lernprozess. Daher ist darauf zu achten, dass nicht Scrum, sondern die eigenen Arbeitspraktiken geändert werden.⁹

⁷ Vgl. Hanser, Eckhart: Agile Prozesse, S.61, 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

⁸ Vgl. Pichler, R.: Scrum, S.1, 1. Auflage, dpunkt.verlag, Heidelberg, 2008

⁹ Vgl. Ebenda., S.2 ff.

3.2 Zusammenspiel der Scrum-Rollen anhand des Scrum-Kreislaufs

Um einen Überblick über den Scrum-Prozess und dessen Ablauf zu erhalten, ist eine grafische Darstellung besonders gut geeignet. Diese Abbildung ist in ähnlicher bzw. abgewandelter Form in vielen verschiedenen Werken veröffentlicht. Die Gemeinsamkeit besteht darin, dass sämtliche Abbildungen den gleichen Informationsgehalt beinhalten, d.h. die Scrum-Rollen (s. Kapitel 3.2.1) sowie die Scrum-Artefakte (s. Kapitel 3.2.2) sind immer inbegriffen. Die nachfolgende Abbildung 1 ist dem Scrum-Flow aus dem Werk [HA10] nachempfunden, die nachfolgend in gekürzter Form wiedergegeben wird:

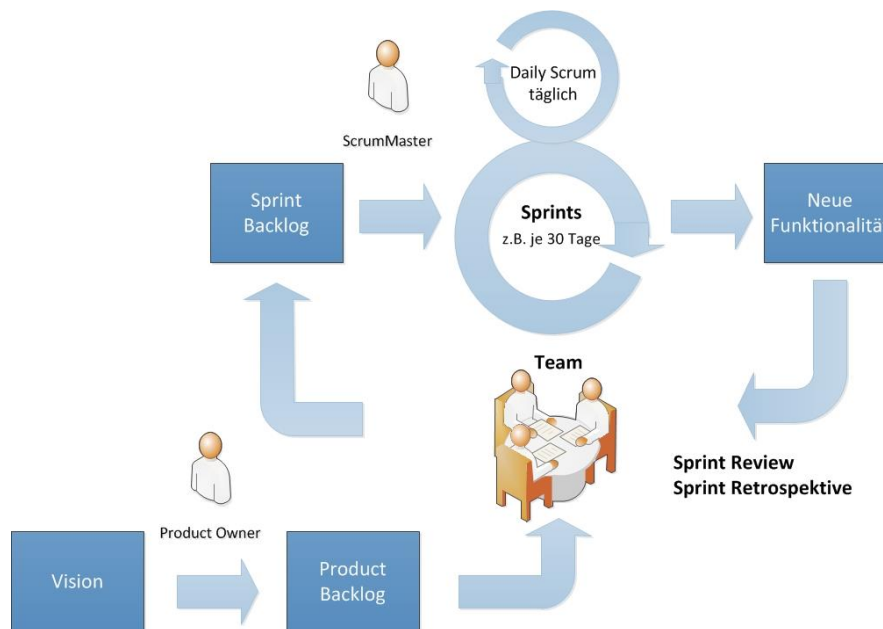


Abb. 1: Der Scrum-Kreislauf [HA10]¹⁰

Am Anfang steht die *Vision* des Product Owners, die eine kurze Beschreibung des Produkts sowie eine Abschätzung des Aufwands und möglicher Meilensteine, beinhaltet. Sobald die Entscheidung gefallen ist, ob ein Projekt angegangen wird, tritt man anschließend in den eigentlichen Scrum-Prozess ein. Das *Product Backlog*, in dem alle Anforderungen an die Software festgehalten und priorisiert sind, wird vom Product Owner gefüllt. Das Team erstellt die Software in Iterationszyklen, den sog. *Sprints*. Sämtliche *Sprints* haben die gleiche festgelegte zeitliche Dauer. Für einen *Sprint* ist eine maximale zeitliche Dauer von 30 Tagen vorgesehen. Dabei ist für jeden *Sprint* ein

¹⁰ Vgl. Hanser, Eckhart: Agile Prozesse, S.69., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

Endtermin (*timeboxing*) zu wählen, der zwischen Product Owner und Team vereinbart wird. Zu Beginn des *Sprints* findet eine *Sprint-Planungssitzung* statt, in der das Team Anforderungen auswählt und das *Sprint Backlog* erstellt. Hier wird festgelegt welche Anforderungen für das nächste Inkrement der Software abgearbeitet werden sollen und welche Aktivitäten dazu notwendig sind. Nach den Vorgaben im *Product Backlog* geschieht die Planung des *Sprints*. Darauf folgt der Beginn des *Sprints*, also die eigentliche Implementierung des nächsten Inkrements der Software. Das Team entwickelt hier das neue Software-Inkrement mit den im *Sprint Backlog* festgelegten Features. Hier werden die Anforderungen des *Sprint Backlogs* in Personentage geschätzt und der Entwicklungsfortschritt täglich in so genannten *Feature-Burndown-Charts* protokolliert. Die Teambesprechung, auch *Daily Scrum* genannt, findet jeden Tag am selben Ort zur selben Zeit statt. Diese Sitzung, in der das Team und der ScrumMaster anwesend sein müssen, ist dabei so kurz wie möglich zu halten. Das *Daily Scrum* dient dazu dem Team die Gelegenheit zu geben, die Arbeit zu koordinieren und Probleme anzusprechen und entweder gleich zu lösen oder an den Product Owner / Scrum Master weiterzugeben. Das Ziel des *Sprints* ist immer ein am System erkennbarer Fortschritt. Dazu kann am Ende eines *Sprints* ein Prototyp, eine Testversion oder eine auslieferbare Version stehen. Am Ende des *Sprints* steht das *Sprint Review*, in dem das entstandene Software-Inkrement vom Product Owner geprüft und abgenommen wird. Akzeptanz finden lediglich nur vollständige und fehlerfreie Arbeitsergebnisse. Partiiell fertiggestellte oder defekte Arbeitsergebnisse gelten dabei als nicht erledigt. Der *Sprint* schließt mit der *Sprint Retrospektive*, die dem Team die Möglichkeit gibt, über den Ablauf des *Sprints* nachzudenken, insbesondere über Verbesserungsmaßnahmen und die Zusammenarbeit im Team. Auf Basis des *Sprint Reviews* und des *Product Backlogs* wird der nächste *Sprint* geplant. Details zum *Sprint - Sprint-Planungssitzung, Daily Scrum, Ablauf, Sprint Review, Sprint Retrospektive* - werden nicht Teil der Seminararbeit sein, da es den Rahmen der Seminararbeit sprengen würde.^{11, 12, 13}

Einen Überblick über den Scrum-Prozess wurde soeben gegeben, nachfolgend soll auf die Projektrollen und Artefakte in Scrum, die in den Kapiteln 3.2.1 und 3.2.2 zu finden sind, näher drauf eingegangen werden.

¹¹ Vgl. Hanser, Eckhart: Agile Prozesse, S.68 ff., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

¹² Vgl. Lippert, M. et al.:eXtreme Programming, S.134 ff., 2. Auflage, dpunkt.verlag, Heidelberg, 2005

¹³ Vgl. Pichler, R.: Scrum, S.7, 1. Auflage, dpunkt.verlag, Heidelberg, 2008

3.2.1 Scrum-Projektrollen

Im Folgenden werden die in Scrum zwingend geforderten Projektrollen - *Product Owner, Team, Scrum Master* - näher beleuchtet.

- Der **Product Owner** übernimmt die Sichtweise des Endkunden und steuert die Software-Entwicklung in enger Zusammenarbeit mit dem (Entwicklungs-)team. Er ist für das Entwicklungsteam ständig verfügbar und bestimmt im Projekt alleine „wohin die Reise geht“. Die Rolle des Product Owners ist bivalent, d.h. er darf nicht nur neue Features wünschen, sondern wird auch sofort auf die Realisierungsmöglichkeit achten. Tolle Ideen werden nur dann realisiert, wenn der Product Owner an ihre Realisierungsmöglichkeit glaubt. Der Product Owner hat Aufgaben in den nachfolgenden Bereichen, die kurz genannt werden:
 - 1) Anforderungsmanagement: Der Product Owner erfasst die Anforderungen des Kunden an die Software und beschreibt sie in Form eines *Produktkonzepts* und des *Product Backlogs* (s. Kapitel 3.2.2). Der Product Owner priorisiert und schätzt die Anforderungen als Basis der des Teams.
 - 2) Release Management: Der Product Owner entscheidet alleine über Funktionalität, Kosten und Termine. Somit trägt er auch die Verantwortung für die Auslieferung der diversen Releases. Er erstellt einen *Releaseplan* und passt diesen ggf. an. Die Aufwandschätzungen aus dem Team sind die Basis für seine Planung.
 - 3) Kommunikation: Der Product Owner muss beständig mit dem Team und mit allen beteiligten Interessengruppen (Stakeholdern), insbesondere beim Endkunden, kommunizieren. Er steht dem Team idealerweise pro Tag eine entsprechende *Zeitdauer* zur Verfügung. Der Product Owner kommuniziert insbesondere mit den Endkundengruppen, um ein möglichst frühes Feedback über neue Funktionalität der Software zu bekommen.^{14 15}
- Das **Team** entwickelt („liefert“) das Produkt, also die Software. Seine Zusammensetzung ist entscheidend für das Gelingen des Projektes. Neben der fachlichen Qualifikation, ist das „Zusammenpassen“ der Teammitglieder von hoher Bedeutung. Ein Team muss selbstorganisiert und klein sein, d.h. innerhalb eines Sprints (also einer Iteration) organisiert sich das Team selbst. Das Team ist bevollmächtigt, d.h. ausschließlich die Teammitglieder entscheiden, welche Aufgaben gelöst werden müssen, um ein *Sprint-Ziel* zu erreichen, und wer für die Aufgabe verantwortlich ist. Gemäß Scrum kann auf den Teamleiter verzichtet werden, weil das Team seine tägliche Arbeit in Form von Berichten und Besprechungen durchführt und somit Leitplanken für die Projektdurchführung - unter anderem *Sprint-Backlog* sowie *Daily-Scrum* - hat (s. Kapitel 3.2.2). Bedingung

¹⁴ Vgl. Hanser, Eckhart: Agile Prozesse, S.62 ff., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

¹⁵ Vgl. Pichler, R.: Scrum, S.9 ff., 1. Auflage, dpunkt.verlag, Heidelberg 2008

für die agile Arbeitsweise nach Scrum ist die Vollzeitmitgliedschaft der Mitglieder im Team. Das Team muss osmotisch kommunizieren, darunter ist unter anderem die räumliche Nähe zu verstehen, d.h. es muss ein gemeinsames Büro zur Verfügung stehen. Die Teammitglieder müssen alle Fähigkeiten abdecken, die zum Erreichen der Projektziele erforderlich sind. Es befinden sich beispielsweise Software-Architekten, Datenbankspezialisten, Programmierer, Qualitätssicherer, Tester sowie Dokumentierer im Team. Es gilt besonders darauf zu achten, dass diese Spezialisten nicht auf ihre traditionelle Projektrolle beharren. Grundsätzlich muss jedes Teammitglied jede Aufgabe im Projekt übernehmen (können) und damit auch sein Wissen an die anderen weitergeben. Das Scrum-Team muss unabhängig sein, d.h. sobald ein *Sprint-Ziel* definiert ist, muss das Team in der Lage sein, das Ziel ohne externe Abhängigkeiten oder Hilfe von außen zu erreichen.^{16 17}

- Der **ScrumMaster** ist der Prozessverantwortliche. Er ist der Freund des Teams und coacht diese hinsichtlich der Einhaltung des Prozesses. Das Team arbeitet nicht für den ScrumMaster, sondern der ScrumMaster arbeitet beim Team. Der ideale ScrumMaster zeichnet sich durch Offenheit und bewusstes, reflektiertes Sprechen und Handeln aus. Weiterhin sollte er ein guter Zuhörer sein und über genügend Rückrad verfügen. Der ScrumMaster kann durch das Team, dem Management oder im äußersten Fall durch Selbstnominierung gewählt werden. Der ScrumMaster hat mehrere Aufgaben zu bewältigen, dazu zählen unter anderem folgende:^{18, 19}
 - Einführung von Scrum und die Etablierung des Prozesses im Team
 - Beantwortung von Fragen zum Prozess und Schützen des Teams vor äußeren Einflüssen
 - Unterstützung des Product Owners sowie Sicherstellung, dass Product Owner und Team direkt zusammenarbeiten
 - Verhinderung, dass der Product Owner sowie der ScrumMaster selbst, in die Rolle des Projektleiters verfallen
 - Beseitigung und Identifizierung von Hindernissen, bspw. verkrustete Denkweisen und „veraltete“ Arbeitsweisen, sowie keine Aufgabenverteilung im Team, da dies dem Team selbst obliegt

¹⁶ Vgl. Hanser, Eckhart: Agile Prozesse, S.63 ff., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

¹⁷ Vgl. Pichler, R.: Scrum, S.13 ff., 1. Auflage, dpunkt.verlag, Heidelberg, 2008

¹⁸ Vgl. Ebenda., S.19 ff.

¹⁹ Vgl. Hanser, Eckhart: Agile Prozesse, S.65 ff., 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

3.2.2 Scrum-Artefakte

Die wichtigsten Scrum-Artefakte (Dokumente) sind das *Sprint Backlog*, das *Product Backlog* und der *Releaseplan* mit *Burndown Chart*. Nachfolgend werden diese Artefakte kurz beschrieben:

- **Product Backlog:**
Das *Product Backlog* ist das zentrale Mittel zum Erfassen und Managen von Anforderungen in Scrum, das von dem Product Owner erstellt und verfeinert wird. Es enthält alle bekannten Anforderungen und Arbeitsergebnisse, die zur Erreichung des Projektziels erbracht werden müssen. Dazu gehören unter anderem funktionale und nicht funktionale Anforderungen. Nachdem die Funktionalitäten, dabei stellt eine Funktionalität üblicherweise eine Zeile dar, im *Product Backlog* gesammelt wurden, werden sie priorisiert, und der Realisierungsaufwand wird geschätzt. In der Praxis hat sich neben der Form von Karteikarten, auch die elektronische Darstellung (als Tabelle) des *Product Backlogs* bewährt. Die Items können dann auf einfache Art und Weise beim *Sprint* in das *Sprint Backlog* übernommen werden.
- **Sprint Backlog:**
Das *Sprint Backlog* beinhaltet alle Aktivitäten, die zur Umsetzung der Anforderungen notwendig sind, zu denen sich das Team in der *Sprint-Planungssitzung* verpflichtet hat. Bei der *Sprint-Planungssitzung* legt das Team fest, welche und wie viele der vom Product Owner vorbereiteten Anforderungen bis zum Ende des nächsten *Sprints* umgesetzt werden können. Falls Anforderungen aus dem *Product Backlog* übernommen worden sind, deren Aufwand größer ist, müssen sie in (Teil-)Anforderungen zerlegt werden. Damit geht eine Präzisierung der Anforderungen einher. Das Team weiß zu jedem Zeitpunkt über den Status jedes *Sprint Backlog Items* (also jedes Arbeitspakets) Bescheid. Es ist sinnvoll, die *Sprint Backlog Items* als *User Stories* darzustellen. Es wird täglich, idealerweise vor dem *Daily Scrum*, angepasst und zeigt jederzeit den aktuellen Status des *Sprints* auf.
- **Releaseplan & Feature Burndown Charts:**
Der *Releaseplan* in Scrum enthält im Wesentlichen die Anzahl der benötigten *Sprints* und, soweit möglich, die aus dem *Product Backlog* antizipierte Reihenfolge der Umsetzung der Anforderungen. Der *Releaseplan* dient dazu, den Rahmen für die *Sprint-Planung* vorzugeben. Der *Sprint* liefert die tatsächliche Entwicklungsgeschwindigkeit, die Verwendung zur Aktualisierung des *Releaseplans* findet. Der *Release Burndown-Bericht* beschreibt den aktuellen Projektfortschritt. D.h. die Entwickler (Team) schätzen die Anforderungen des *Sprint Backlogs* und protokollieren den Entwicklungsfortschritt in so genannten *Feature-Burndown-Charts*. Diese zeigen auf, wann das Projekt alle Anforderungen aus dem *Sprint Backlog* abgearbeitet haben wird.^{20 21 22}

²⁰ Vgl. Pichler, R.: Scrum, S.27 ff. u. 49 ff. u. 70 u. 93 u. 102, 1. Auflage, dpunkt.verlag, Heidelberg, 2008

²¹ Vgl. Hanser, Eckhart: Agile Prozesse, S.73 ff. u. 75, 1. Auflage, Springer-Verlag, Berlin Heidelberg, 2010

²² Vgl. Lippert, M. et al.:eXtreme Programming, S.135, 2. Auflage, dpunkt.verlag, Heidelberg, 2005

4 Handlungsempfehlung

In diesem Kapitel soll eine Handlungsempfehlung bzgl. des Einsatzes von Scrum im Rahmen unserer Projektgruppe abgegeben werden. D.h. falls Scrum eingesetzt werden soll, wie sollte das Vorgehen gestaltet sein. Die Handlungsempfehlung ist nachfolgend stichpunktartig festgehalten:

- Der Product Owner, der die (priorisierten) Anforderungen an die Software bereitstellt, steht bereits fest und ist somit erledigt
- Die Rolle des ScrumMaster kann nur von einer Person besetzt werden, die idealerweise Erfahrung mit Scrum vorweist. Es muss entschieden werden, ob der ScrumMaster durch das Team, das Management oder durch Selbstnominierung gewählt wird
- Die Strukturierung des Teams kann so erfolgen, dass Mitglieder verschiedene Fachbereiche zugeordnet werden. Mögliche Bereiche wären z.B.: Anforderungsmanagement, Datenbanken, Programmierung, Qualitätssicherung & Test sowie Dokumentation. Das Vorhandensein eines Teamleiters ist nicht zwingend erforderlich
- Festlegung der maximalen zeitlichen Dauer sowie Bestimmung von Endterminen der jeweiligen *Sprints*
- Organisation bzgl. der Teambesprechung bzw. *Daily Scrum* (Tag, Ort, Uhrzeit)

5 Schlusswort

Softwareentwicklung ist schwierig und herausfordernd, auch der Einsatz von Scrum ändert nichts an dieser Tatsache. Das Wesen der Softwareentwicklung ist Innovation und Kreativität, das vielen Organisationen nicht leicht fällt. Sind die Ziele in Gefahr, so sollen häufig die Projektmitarbeiter länger arbeiten und neue Mitarbeiter werden dem Projekt hinzugefügt. Die Softwarequalität und die Moral der Mitarbeiter sinkt, bedingt durch Hektik, Stress, längere Arbeitszeiten und schlecht eingearbeitete Projektmitglieder. Dadurch werden vom Management mehr Kontrollen eingeführt, die die Entwicklung weiter verlangsamen. Das zentrale Problem traditioneller Vorgehensweisen besteht darin, dass erst spät im Projekt, eine Rückmeldung über den tatsächlichen Fortschritt, in der Regel bei der Integrierung und dem Testen der Software, zurückgegeben wird. In Scrum hingegen wird bereits nach wenigen Wochen Rückmeldung über den Fortschritt und etwaige Probleme und Hindernisse gegeben. Scrum ermöglicht das frühe Auffinden von Problemen, das wiederum einen größeren Handlungsspielraum und Flexibilität eröffnet. Warum sollte man Scrum letztendlich einsetzen? Durch den richtigen Einsatz von Scrum steigt die Mitarbeiterzufriedenheit, die Kundenzufriedenheit und es wird ein wirtschaftlicher Nutzen erzielt.²³

²³ Vgl. Pichler, R.: Scrum, S.3 ff., 1. Auflage, dpunkt.verlag, Heidelberg, 2008

Literaturverzeichnis

- [BE15] Beck K et al. (2001), www.agilemanifesto.org, Stand: 31.10.2015.
- [HA10] Hanser, E.: Agile Prozesse - Von XP über Scrum bis MAP. 1. Auflage, Springer-Verlag, Berlin-Heidelberg, 2010.
- [LRW05] Lippert, M.; Roock, S.; Wolf, H.: eXtreme Programming: Eine Einführung mit Empfehlungen und Erfahrungen aus der Praxis, 2. Auflage, dpunkt.verlag, Heidelberg 2005.
- [PR08] Pichler, R.: Scrum - Agiles Projektmanagement erfolgreich einsetzen, 1. Auflage, dpunkt.verlag, Heidelberg 2008.

Architektur-Management, Deployment, EDI

Author: Jannick Asmuß, B.Sc.³

Abstract: Dieser Seminarbeitrag ist über Architektur-Management, Deployment und Electronic Data Interchange. Der Beitrag erklärt die Bedeutung von Architektur-Management und die zugehörigen Aufgaben. Es wird ein Zusammenhang zwischen einer Unternehmensarchitektur und einer IT-Architektur beschrieben. Frameworks bieten dabei Unterstützung. Es gibt verschiedene Rollen die an einer Architektur beteiligt sind, sowie Methoden und Diagramme die beim Aufbau oder Ändern verwendet werden. Innerhalb der Architektur werden Daten ausgetauscht. Dazu werden Standards und mögliche Umsetzungen gezeigt. Der Seminarbeitrag kann als Einstiegspunkt verwendet werden, um eine Organisationsstruktur aufzubauen.

1 Einleitung

Die Einleitung umfasst die Motivation und den Aufbau der Arbeit. In der *Motivation* wird der Zweck des Seminarbeitrags beschrieben und die Relevanz des Themas begründet. Die Struktur der Arbeit wird unter *Aufbau der Arbeit* erläutert. Dort wird beschrieben wie die Kapitel und Unterkapitel strukturiert sind.

1.1 Motivation

Ziel ist es, durch diesen Seminarbeitrag, alle Mitglieder der Projektgruppe auf einen gemeinsamen Wissensstand zu bringen. Für die Mitglieder wird eine Übersicht über die Thematik verschafft, so dass gängige Begriffe verstanden werden können und wichtige Standards, Modelle, Software und Werkzeuge bekannt sind. Ebenso soll das Thema von anderen Themen abgegrenzt werden um thematische Überschneidungen mit den anderen Seminarbeiträgen zu vermeiden.

1.2 Aufbau der Arbeit

Nach der Einleitung folgt das erste Hauptkapitel *Architektur-Management*. Das Kapitel besteht aus drei Unterkapiteln, in denen zusammengefasst wird, was Architektur-Management bedeutet, wie es umgesetzt werden kann und welche Methoden dazu existieren. Das zweite Hauptkapitel *Deployment* bietet eine Übersicht darüber, wie eine Architektur aufgebaut werden kann und welche Diagramme dafür verwendet werden können. Das dritte Hauptkapitel beinhaltet den Datenaustausch. In den drei Unterkapiteln wird der Begriff definiert, mögliche Umsetzungen gezeigt und Standards beschrieben.

³ Universität Oldenburg, Department Informatik, Ammerländer Heerstraße 114-118, 26129 Oldenburg, jannick.asmuß@uni-oldenburg.de

2 Architektur-Management

In diesem Kapitel wird veranschaulicht, was Architektur-Management ist und wofür es bedeutsam ist. Dazu wird gezeigt, was die Aufgaben beim Management sind. Im nächsten Schritt wird der Zusammenhang zwischen einer IT-Architektur und Geschäftsprozessen erläutert. Zum Aufbauen oder Ändern einer Architektur können Frameworks verwendet werden. Diese werden im Unterkapitel *Enterprise Architecture Frameworks* eingeführt.

2.1 Definition

Wenn Prozesse einer Organisation durch die IT unterstützt werden, ist eine IT-Architektur notwendig. Eine IT-Architektur beschreibt die IT innerhalb einer Organisation. Sie bestimmt die Grundstrukturen und das Zusammenspiel der Komponente. Die Infrastruktur besteht aus Hardware, Standorte, Netzwerke, Software und Daten. Außerdem gehören Schnittstellen zur Architektur. Das Management beschäftigt sich mit Konfigurations- und Kapazitätsplanung, Lastverteilung, Datensicherung, Verfügbarkeit, Ausfallsicherheit, Notfall-Planung, ect. (vgl.[IT15]).

Eine IT-Architektur kann durch eine IT-Strategie beeinflusst werden. Eine Enterprise-Strategie ist die ganzheitliche Betrachtung von Geschäft, Infrastruktur und Anwendungen zur Bildung einer Enterprise Architecture. Enterprise Architecture Management verfügt über bereichsübergreifenden Aufgaben. Dazu gehören Projekt- und Produktplanung (Zuordnung von Anwendungen, Infrastruktur, Geschäftsprozesse), Anbieten von Business-Services und das Verwalten der System-Landschaft(Landscape Management) (vgl.[EA15]).

Zum Erstellen und Bearbeiten einer Architektur kann Software als Hilfsmittel verwendet werden. Eine Software dafür ist ARIS. ARIS unterstützt durch viele Funktionen die beim Enterprise Architecture Management von Bedeutung sind. ARIS unterstützt u.A. beim entwickeln, analysieren, steuern und verwalten des Geschäftsmodells, Modellieren von Geschäftsprozessen und Dokumentieren von Unternehmensarchitekturen (vgl.[Ar15]).

2.2 Unternehmensarchitektur

Eine Unternehmensarchitektur verknüpft die IT mit den Geschäftszielen. Die Architektur kann von Unternehmen zu Unternehmen verschieden sein. Zunächst wird die Unternehmensarchitektur definiert:

„Eine Enterprise Architecture ist ein konzeptioneller Entwurf, welcher die Struktur und Arbeitsweise einer Organisation beschreibt. Ziel einer Enterprise Architecture ist es, zu ermitteln, wie die betrachtete Organisation möglichst effektiv aktuelle und zukünftige Ziele erreichen kann (S.13[St09]).“

Die Unternehmensarchitektur besteht aus verschiedenen Bereichen. Die Bereiche können selbst definiert werden. Die Bereiche entsprechen unterschiedliche Sichten. Die Bereiche

können unterschiedlich abgegrenzt werden. Im Wesentlichen enthält die Architektur immer die selben Bereiche. Beispielhaft folgt nun eine Einteilung der Architektur in vier Bereiche.

Geschäfts-Architektur „Die Geschäfts-Architektur beschreibt eine abstrakte Sicht auf die fachlichen, betriebswirtschaftlichen Aktivitäten und Beziehungen innerhalb einer Organisation. Es handelt sich dabei um eine überblicksartige Beschreibung der betriebswirtschaftlichen Sicht auf das Unternehmen (S.13[St09]).“

Daten-Architektur „Die Daten-Architektur beschreibt die im Rahmen der Geschäftstätigkeit des Unternehmens anfallenden bzw. beteiligten Geschäftsobjekte, Informationen und Daten. In der gemeinsamen Betrachtung von Geschäfts- und Anwendungs-Architektur ist sie die Schnittstelle zwischen fachlichen Inhalten und der Informationstechnologie (S.13[St09]).“

Anwendungssystem-Architektur „Die Anwendungs-Architektur zeigt auf, welche informationstechnologische Unterstützung benötigt wird, um das betriebswirtschaftliche Ziel des Unternehmens zu erfüllen. Sie beschreibt auf hoher Ebene die in der betrachteten Organisation vorhandenen Softwarelösungen und deren Beziehungen untereinander (S.13[St09]).“

Infrastruktur-Architektur „Die Infrastruktur-Architektur beschreibt die erforderliche IT-Infrastruktur zum Betrieb der Anwendungssystem-Architektur und damit das IT-technische Fundament einer Organisation (S.14[St09]).“

Die Architektur-Pyramide (Siehe Abbildung:1) ist ein Modell zum Strukturieren der Architektur. Sie beginnt oben mit der Strategie, welche die Architektur beeinflusst. Die Geschäftsarchitektur beinhaltet das Geschäftsmodell und das Geschäftsprozess-Management. Danach folgen drei IT-Architekturen. Die Informationsarchitektur mit der Kern-Anwendungslandschaft, die Anwendungsarchitektur mit den Support-Anwendungen und die Basisstruktur, mit der System- und Infrastruktur-Architektur. Im Projektportfolio werden alle Projekte gebündelt, die erforderlich sind um die Architektur in einen bestimmten Soll-Zustand zu transformieren (vgl.[Wa15]).

2.3 Enterprise Architecture Framework

Zum Bilden einer Unternehmens-Architektur können Entwurfsmuster verwendet werden. Diese können auch als Enterprise Architecture Frameworks bezeichnet werden. Es „sind Rahmenwerke, die eine systematische Sammlung von Strukturen und manchmal auch Methoden und Werkzeugen bereitstellen. Die meisten EA Frameworks kann man nach einer primär statischen oder dynamischen Ausrichtung unterscheiden [St09].“

Ein statisches Framework ist eine Vorlage für ein Modell zur Beschreibung einer Organisation. Ein bekanntes statisches Framework ist das Zachman Framework. Es ist eine Matrix

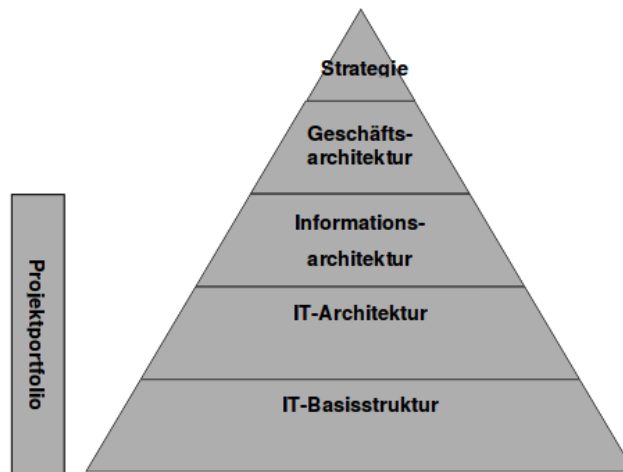


Abb. 1: Architektur-Pyramide[Ge09]

und betrachtet eine Organisation aus 6 Perspektiven und 6 Rollen (Siehe Abbildung:2). Es gibt keine offizielle Methodik zum Anwenden des Frameworks. Es ist eine Vorlage die als Ausgangspunkt beim Bilden einer Struktur unterstützt (vgl.S.12[St09]).

Dynamische Frameworks liefern Phasenmodelle und Arbeitsanweisungen zu den Vorlagen (vgl.S.12[St09]). Ein dynamisches Framework ist das TOGAF. TOGAF unterscheidet zwischen 4 Architekturen(Geschäfts-, Anwendungs-, Daten- und Technische-Architektur). Außerdem gehört ein Modellierungsprozess dazu. Die TOGAF Architecture Development Method. Dieser definiert bestimmte Schritte, die dazu führen, dass die Zielarchitektur erreicht werden kann (vgl.[2.15]).

3 Deployment

Deployment ist die Verteilung und kann unterschiedliche Bedeutungen haben. So kann Deployment den Prozess der Verteilung von Software beschreiben. Also die Installation und Konfiguration von Betriebssystem und Anwendungssoftware (vgl.[Th15]). Im Rahmen des Themas wird Deployment allerdings aufgefasst, als die Verteilung von Komponenten auf Rechnerknoten. Als erstes wird das Vorgehen betrachtet, durch dass die Architektur bestimmt werden kann und welche Rollen involviert sind. Danach wird das Deployment Diagram vorgestellt. Dies ist ein Diagramm der UML und dient zum Darstellen der Verteilung (vgl.[Ve15a]).

3.1 Enterprise Architecture Modelling

Als erstes werden die beteiligten Rollen vorgestellt und deren Aufgaben beim Entwickeln einer Architektur. An dem Entwickeln einer Architektur sind verschiedene Rollen betei-

	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>
Objective/Scope (contextual) <i>Role: Planner</i>	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) <i>Role: Owner</i>	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) <i>Role: Designer</i>	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) <i>Role: Builder</i>	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representation (out of context) <i>Role: Programmer</i>	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise <i>Role: User</i>	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

Abb. 2: Zachman Framework[Ab15]

ligt. Das Management kommuniziert die IT-Strategie, bewertet und priorisiert IT-Projekte, überwacht Projekte und führt eine abschließende Bewertung durch. Business Analysten ermitteln fachliche Anforderungen, erstellen ein Fachkonzept und Testen die Lösung auf fachlicher Korrektheit.

Die technische IT überführt das fachliche Konzept in ein technisches Konzept und überprüft die Umsetzbarkeit und die Risiken (vgl.S8f.[St09]). Das Entwickeln einer Architektur kann durch ein Framework unterstützt werden. Das TOGAF-Framework liefert dafür die Architecture Development Method. Die Methode (Siehe Abbildung:3) wird iterativ angewendet. Es beginnt mit einer Phase zum Vorbereiten der Organisation für das IT-Projekt. Als nächstes wird in der Architektur-Vision das Ziel und die Beteiligten festgelegt. Danach folgen die Phasen, wo der aktuelle Architekturzustand und der geplante Zustand der Geschäfts-, Informationssystem- und der Technologie-Architektur definiert werden. Danach werden Maßnahmen festgelegt um eine Änderung herbei zu führen. Die anderen Phasen beschreiben die Projektüberwachung. Es werden die Maßnahmen beschrieben sowie die Durchführung überwacht und ausgewertet (vgl.[TO15]).

3.2 Deployment Diagram

Eine Möglichkeit den Sachverhalt der Verteilung zum Modellieren und Visualisieren ist das Verwenden eines Verteilungsdiagramm. Ein Deployment Diagram oder Verteilungsdiagramm ist ein Diagramm aus der UML. Abbildung :4 zeigt ein Beispiel des Diagramms. Es ist ein Strukturdiagramm, dass die Komponenten auf Rechnerknoten verteilt. Es dient dem Modellieren von Zusammenhängen. Es veranschaulicht wie Artefakte des Systems

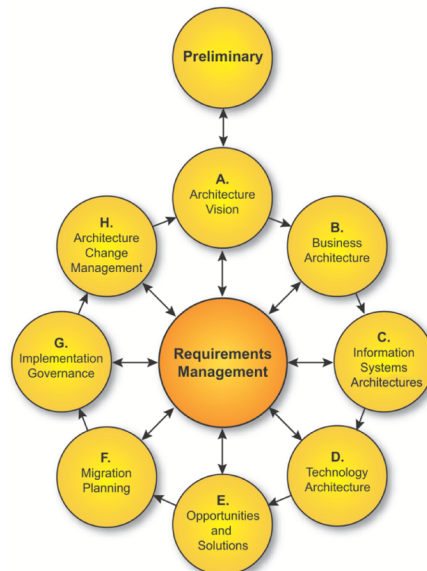


Abb. 3: TOGAF[TO15]

auf Rechnerknoten verteilt werden. Zusätzlich werden Kommunikationsverbindungen und Abhängigkeiten gezeigt. Das Diagramm besteht aus zwei Symbolkategorien. Kanten und Knoten. Die wichtigsten Knotentypen sind (vgl.[E115]):

Knoten „Ein Knoten stellt eine Ressource dar, auf die Artefakte verteilt werden können ([E115]).“

Gerät „Der spezielle Knoten Gerät stellt einen bestimmten Hardware-Typ dar ([E115]).“

Ausführungsumgebung „Der spezielle Knoten Ausführungsumgebung stellt einen bestimmten Software-Typ dar ([E115]).“

Artefakt „Artefakte stellen eine physische Informationseinheit dar, z.B. Modelle, Quellcode, Skripte oder Dokumente, die im Entwicklungsprozess oder zur Laufzeit eines Systems erzeugt oder benötigt werden ([E115]).“

Einsatzspezifikation „Das spezielle Artefakt Einsatzspezifikation regelt die Verteilung der mit ihm verbundenen Artefakte auf die Knoten ([E115]).“

Instanzbeschreibung „Die Instanzbeschreibung beschreibt eine zur Laufzeit vorhandene Einheit des modellierten Systems anhand eines Beispiels ([E115]).“

Randbedingung „Die Randbedingung ist ein einschränkender, logischer Ausdruck, der die Semantik eines oder mehrerer Modellelemente präzisiert ([E115]).“

Kommunikationspfad „Der Kommunikationspfad ist eine spezielle Assoziation, die es den verbundenen Knoten ermöglicht, Nachrichten auszutauschen ([E115]).“

Link „Ein Link ist die Instanz einer Assoziation, die zur Herstellung einer Verbindung zu einer anderen Instanzbeschreibung dient. Die zu instanzierende Assoziation wird beim Anlegen abgefragt ([E115]).“

Generalisierung „Eine Generalisierung ist eine gerichtete Beziehung von einem speziellen Classifier zu einem generelleren Classifier. Instanzen des speziellen Classifiers sind damit auch Instanzen des generelleren Classifiers ([E115]).“

Verteilungsbeziehung „Die Verteilungsbeziehung ist eine gerichtete Beziehung, die angibt, dass ein Artefakt auf einen Knoten verteilt wird ([E115]).“

Manifestation „Eine Manifestation ist eine Abhängigkeitsbeziehung, die besagt, dass ein Artefakt das verbundene Modellelement physisch realisiert ([E115]).“

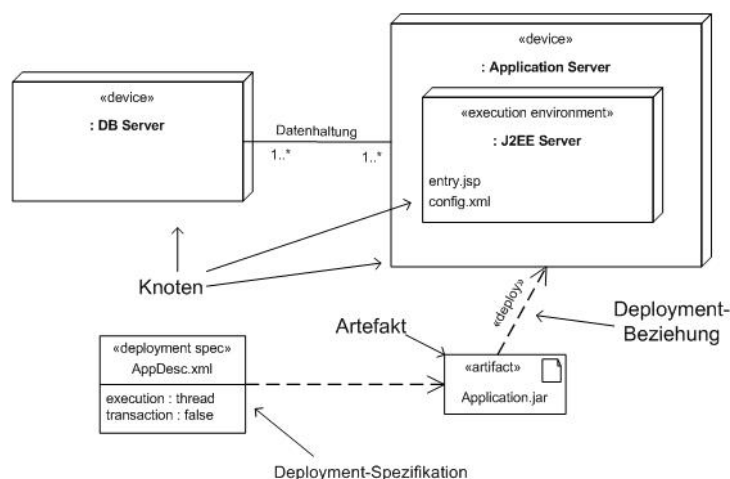


Abb. 4: Deployment Diagram [Ve15b]

4 EDI

EDI ist die Abkürzung für den elektronischen Datenaustausch. In diesem Kapitel wird der Begriff definiert und die Arten der Umsetzung gezeigt. Abschließend werden beispielhafte Standards zum elektronischen Datenaustausch gezeigt.

4.1 Definition

„Unter Electronic Data Interchange (EDI) versteht man ganz allgemein die elektronische Übertragung von Geschäftsdaten. Ziel von EDI ist es, eine Kommunikation zu ermöglichen, mit der über Unternehmensgrenzen hinweg Geschäftsprozesse gesteuert werden. EDI soll die zahlreichen Papierdokumente wie Bestellungen, Bestätigungen, Aufträge, Rechnungen, Lieferscheine usw. ersetzen, die bei einem Geschäftsprozess anfallen. EDI steht für einen elektronischen Dokumentenaustausch mit formatierten Strukturen, der in Rechen-systemen weiterverarbeitet werden kann [ED15b].“

4.2 Umsetzung

Durch EDI werden Inhalte von Daten so formatiert, dass die Daten von anderen Rechnern weiterverarbeitet werden können. Geschäftstätigkeiten können automatisiert werden und organisationsübergreifende Prozesse beschleunigt werden. EDI ist vorteilhafter, wenn es eine hinreichende Integrationstiefe besitzt. EDI kann in drei Grade unterteilt werden (vgl.[ED15b]).

„Bei Stand-Alone-EDI ist der elektronische Datenaustausch in keinerlei interne Prozesse des Unternehmens oder in das Hauptrechnersystem integriert. Beim Integrated EDI sind die Applikationssysteme der beiden Kommunikationspartner miteinander verbunden. Informationen, die von einem der beiden Geschäftspartner generiert werden, gelangen in das Applikationssystem des anderen Geschäftspartners. Von Full-EDI spricht man dann, wenn über die Integration der unternehmensinternen Applikationen mit EDI hinaus, eine optimale Ausgestaltung des Unternehmens vorgenommen und die unternehmensinternen Prozesse den EDI-Erfordernissen angepasst werden.

Der elektronische Datenaustausch (EDI) wird zunehmend von Extensible Markup Language (XML) und Webservices abgelöst [ED15b].“

4.3 Standards

„United Nations/Electronic Data Interchange for Administration, Commerce and Transport ist der internationale Standard, der von den Vereinten Nationen entwickelt wurde. Die Wartung und weitere Entwicklung des Standards erfolgt über das United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) unter der UN- Wirtschaftskommission für Europa. Der EDIFACT-Standard liefert mehrere Syntaxregeln für die

Struktur, ein interaktives Austauschprotokoll und mehrere Standardnachrichten, anhand welcher multinationale und branchenübergreifende Unternehmen elektronische Geschäfts-Dokumente austauschen können. EDIFACT wird weit verbreitet in Europa eingesetzt, was hauptsächlich auf die Tatsache zurückzuführen ist, dass ihn viele Unternehmen schon frühzeitig übernommen haben. EDIFACT wurde auch im ASPAC-Bereich übernommen, obwohl dort derzeit mehr XML-Standards in diesem spezifischen Bereich heute eingesetzt werden [ED15a].“

Ein Format zum Austauschen von Informationen ist XML. XML wird über das World Wide Web Consortium (W3C) definiert. Das W3C hat die Aufgabe Web-Standards und Richtlinien zu entwickeln um ein langfristiges Wachstum des Webs zu sichern (vgl.[Ub15]).

XML wird in vielen IT-Systemen verwendet. Es kann verwendet werden, egal welche Hard- oder Software verwendet wird. XML speichert nur Informationen. Es wird zusätzliche Software benötigt, die XML senden, empfangen, speichern oder anzeigen kann. XML ist HTML sehr ähnlich. Vorkenntnisse über HTML und auch Javascript helfen beim Umgang mit XML. XML verwendet ebenso Tags wie HTML allerdings gibt es bei XML keine vordefinierten Tags. Diese können selbst bestimmt werden (vgl.[In15]).

Eine Alternative zu XML ist JSON. JSON wird im Standard ECMA-404 definiert (vgl.[Th15]).

5 Fazit

Das erarbeitete Wissen kann für die Projektgruppe verwendet werden. Dies wird unter *Ausblick* aufgezeigt. Zuvor wird im *Rückblick* das Thema rückblickend zusammengefasst.

5.1 Rückblick

Das Thema bietet sehr viel in der Breite als auch in der Tiefe. Da der Seminarbeitrag an eine bestimmte Seitenzahl gebunden ist, musste dementsprechend an einigen Stellen auf inhaltliche Tiefe oder Breite verzichtet werden. So wurde *Service-orientierte Architektur* und *Geschäftsprozessmodellierung* ausgegrenzt, obwohl es inhaltliche Überschneidungen gibt. Ebenso *ITIL*. Außerdem gibt es einige Berührungspunkte mit anderen Beiträgen. Deswegen wurde die Software *Aris* nicht tiefer behandelt. So wurde auf das Definieren der ISO über Architektur-Beschreibungen, sowie auf einige Software-Beispiele verzichtet. Ebenso wurde *Data Mapping* entfernt, dafür *JSON* zugefügt.

5.2 Ausblick

Das Thema kann in vielen Bereichen vertieft werden. Allerdings kann das erarbeitete Wissen bereits genutzt werden, bei den Überlegungen, wie die Projektgruppe organisiert und strukturiert werden kann. Dazu gibt es Hilfestellungen wie Frameworks und das Verteilungsdiagramm. Zu beachten ist auch der Austausch von Daten innerhalb der Architektur.

Dafür wird EDI und der EDIFACT-Standard erklärt. Mit XML und JSON werden zwei Datentypen genannt, die dafür genutzt werden können.

Literaturverzeichnis

- [2.15] 2. Core Concepts, <http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>.
- [Ab15] About the Zachman Framework, <http://www.zachman.com/about-the-zachman-framework>.
- [Ar15] Aris, <http://www.softwareag.com/de/products/az/aris/>.
- [EA15] EAM (enterprise architecture management), <http://www.itwissen.info/definition/lexikon/enterprise-architecture-management-EAM.html>.
- [ED15a] EDI-Dokumentenstandards, <http://www.edileitfaden.de/edi-resources/document-standards/>.
- [ED15b] EDI (electronic data interchange), <http://www.itwissen.info/definition/lexikon/electronic-data-interchange-EDI-Elektronischer-Datenaustausch.html>.
- [El15] Element-UML-Verteilungsdiagramm, http://help.innovator.de/12.2/de_de/Innovator/Content/Ref.MetaM.UML/CLDeploymentDia.htm.
- [Ge09] GernotDern: Management von IT-Architekturen: Leitlinien für die Ausrichtung, Planung und Gestaltung von Informationssystemen. Vieweg + Teubner, Wiesbaden, 2009.
- [In15] Introduction to XML, http://www.w3schools.com/xml/xml_what_is.asp.
- [IT15] IT-Architektur, <https://de.wikipedia.org/wiki/IT-Architektur>.
- [St09] Stähler, D.; I.Meier; R. Scheuch, C.Schmülling; Somssich, D.: Enterprise Architecture, BPM und SOA für Business-Analysten. Carl Hanser Verlag, München, 2009.
- [Th15] The JSON Data Interchange Format, <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
- [TO15] TOGAF Standard Courseware V9.1 Edition, <http://www.togaf.info/togafSlides91/TOGAF-V91-M3-Intro-ADM.pdf>.
- [Ub15] Über das W3C, <http://www.w3c.de/about>.
- [Ve15a] Verteilungsdiagramm, <https://de.wikipedia.org/wiki/Verteilungsdiagramm>.
- [Ve15b] Verteilungsdiagramm, cs.uni-magdeburg.de/tutorial/UML2/Verteilungsdiagramm-Dateien/image001.jpg.
- [Wa15] Was ist EAM + IT-Strategien, <https://www.hpi.uni-potsdam.de/hirschfeld/teaching/past/itua12/media/Itua12.02.ITStrategie.pdf>.

Modellierung

Dörthe Euhäusen¹

Abstract: In ihrem Bericht *An Approach for a Cloud-based Contribution Margin Dashboard in the Field of Electricity Trading* berichten Studierende der Carl von Ossietzky Universität Oldenburg von der Anforderungsanalyse sowie deren Ergebnisse und präsentieren einen ersten Prototypen ihres Business Intelligence in the Cloud-Service für den Bereich des Stromhandels. Die Architektur ihres Cloud Service-Systems veranschaulichen sie mit einem Komponenten- und einem Sequenzdiagramm in der Standardnotation UML und einem Entity Relationship-Diagramm. Diese Seminararbeit beschäftigt sich zunächst mit den Grundlagen der Modellierung und gibt anschließend eine Empfehlung, welche Diagrammtypen für die weitere Arbeit in der Projektgruppe Cloud-basierte analytische Informationssysteme (CAIS) entwickelt werden sollten.

Keywords: Modellierung, UML (Unified Modeling Language), Entity Relationship-Modellierung, Business Intelligence in the Cloud

1 Einleitung

Im Wintersemester 2014/2015 begann die Projektgruppe Business Intelligence as a Service mit der Anforderungsanalyse und prototypischen Entwicklung eines cloud-basierten Business Intelligence-Systems für die Verwendung im Bereich des Stromhandels namens BICE [No15] (Business Intelligence in the Cloud for Energy).

Ziel der im Wintersemester 2015/2016 beginnenden Projektgruppe Cloud-basierte analytische Informationssysteme (CAIS), in deren Rahmen diese Seminararbeit verfasst wurde, ist es, für einen noch zu wählenden Anwendungsfall ein analytisches Informationssystem in der Cloud zu realisieren. In [No15] ist eine mögliche Architektur eines solchen Informationssystems bereits beschrieben und anhand von Modellen dargestellt. Im Rahmen der geplanten Realisierung eines analytischen Informationssystems müssen detaillierte Modelle entworfen werden, die das Informationssystem beschreiben.

Diese Seminararbeit befasst sich deshalb zunächst mit den Grundlagen der Softwaremodellierung. Es folgt die Vorstellung der Standardnotation UML und einiger ausgewählter Diagrammtypen, die entweder von der Projektgruppe Business Intelligence as a Service verwendet wurden oder aber eine detailliertere Darstellung des geplanten analytischen Informationssystems erlauben. Das abschließende Fazit fasst nochmals alle wichtigen Aspekte der Softwaremodellierung zusammen und gibt darüber hinaus eine Empfehlung zur Modellierung im Rahmen der beginnenden Projektgruppe.

¹ Carl von Ossietzky Universität Oldenburg, doerthe.euhausen@uni-oldenburg.de

2 Grundlagen

Dieses Kapitel beschäftigt sich mit den Grundlagen der Modellierung, die für das Verständnis der folgenden Kapitel notwendig sind. Das erste der beiden Unterkapitel befasst sich zunächst mit Definitionen der verwendeten Begriffe Modell, Modellierung und Software-Architektur und erläutert anschließend, weshalb die Modellierung ein nicht zu vernachlässigender Schritt einer jeden Projektdurchführung ist und weshalb zur Modellierung eine Notation verwendet werden sollte. Das zweite der beiden Unterkapitel stellt die Grundsätze ordnungsgemäßer Modellierung vor, die eine Handlungsempfehlung zur qualitativ hochwertigen Modellierung darstellen.

2.1 Modellierung

Als Ursprung des Begriffs Modell wird in der Literatur zum einen das lateinische *modulus*, was Maß oder Maßstab bedeutet [KKB08], und zum anderen das italienische *modello*, was Muster oder Entwurf bedeutet [Le08], genannt. Die Definitionen des Begriffs für den Bereich der Informatik ähneln sich jedoch, so wird das Modell in [KKB08] als „im Allgemeinen abstrakte Abbilder oder Vorbilder zu konkreten oder abstrakten Originalen“ und in [Le08] als „ein konstruiertes, vereinfachtes Abbild“ definiert. In [Le08] wird zusätzlich darauf hingewiesen, dass die Begriffe Modell und Diagramm zwar sehr häufig synonym verwendet werden, aber ein Modell genau genommen aus mehreren Diagrammen und auch aus zusätzlichen textuellen Beschreibungen bestehen kann.

Das Modellieren geschieht in den Anfangsphasen eines Projektes. Es dient dazu, darzustellen, wie Anforderungen verstanden wurden und wie die Umsetzung aussehen soll [KKB08]. Für die Erstellung eines Modells wird eine Modellierungssprache, auch Notation genannt, verwendet. Informale Sprachen, wie Skizzen, die nicht den Regeln einer Notation folgen oder natürliche Sprachen sind zur Modellierung weniger geeignet, denn es besteht stets die Gefahr der Unvollständigkeit, Mehrdeutigkeit und Inkonsistenz [Le08]. Formale Sprachen, wie Petrinetze, besitzen eine mathematisch präzise Syntax und Semantik und verzichten soweit möglich auf Elemente natürlicher Sprache. Mit formalen Sprachen ist die Simulation von Abläufen und damit auch eine Aufdeckung von Fehlern möglich. Der große Nachteil der formalen Sprachen ist, dass sie erlernt werden müssen und zumeist nicht intuitiv verständlich sind. Als Diskussionsgrundlage sind sie damit ungeeignet. Einen Kompromiss bieten semiformale Sprachen. Sie verfügen über eine vorgegebene Struktur und eine präzise definierte Syntax, verbinden dies aber mit einigen grafischen Elementen und Elementen aus der natürlichen Sprache [Le08]. Ein Beispiel für ein Modell in semiformaler Sprache ist das Anwendungsfalldiagramm aus der UML, das neben einigen anderen Diagrammen im folgenden Kapitel vorgestellt wird.

Im Rahmen der bevorstehenden Projektarbeit wird eine Software-Architektur für ein cloud-basiertes Informationssystem weiterentwickelt. In [RH06] wird Software-Architektur in Anlehnung an die englischsprachige IEEE-Definition als „die grundlegende Organisation eines Systems, dargestellt durch dessen Komponenten, deren Beziehungen zueinander

und zur Umgebung, sowie die Prinzipien, die den Entwurf und die Evolution bestimmen“ definiert. In [RH06] werden außerdem einige Vorteile aufgeführt, die die Modellierung einer guten Software-Architektur mit sich bringt. So kann durch eine gute Architektur beispielsweise der Entwicklungsprozess effizienter gestaltet werden, denn die Entkopplung der einzelnen Teilaufgaben ermöglicht die Arbeit in mehreren Entwicklerteams. Auch das frühzeitige Erkennen und Behandeln von Risiken wird durch die Entwicklung einer guten Software-Architektur ermöglicht. Wie bereits zuvor erwähnt, dient ein Modell der geplanten Software-Architektur auch als Diskussionsgrundlage und unterstützt damit die Kommunikation zwischen allen Projektbeteiligten.

2.2 Grundsätze ordnungsgemäßer Modellierung

Die insgesamt sechs Grundsätze ordnungsgemäßer Modellierung (GoM) wurden im Jahr 1995 von Becker, Rosemann und Schütte veröffentlicht. Sie stellen eine Modellierungsempfehlung dar, die eine hohe Modellqualität garantieren soll, die über die rein syntaktische Korrektheit [En12] hinausgeht.

- *Grundsatz der Richtigkeit:* Der in der Veröffentlichung [BRS95] aus dem Jahr 1995 erstgenannte der sechs Grundsätze bezieht sich sowohl auf die semantische als auch syntaktische Korrektheit eines Modells. Die semantische Richtigkeit, also ob ein System oder ein Prozess der Realität entsprechend modelliert wurde, kann grundsätzlich nicht bewiesen werden. Hierzu ist die Überprüfung durch einen Fach- und Methodenexperten notwendig [En12]. Die syntaktische Richtigkeit beschreibt die Einhaltung der Regeln der verwendeten Notation.
- *Grundsatz der Relevanz:* Dieser Grundsatz besagt, dass nur so viel modelliert werden soll, wie relevant ist [En12]. Dazu muss zunächst das Ziel der Modellierung definiert werden. Ist das geschehen, kann sowohl das notwendige Abstraktionsniveau bestimmt als auch eine passende Modellierungstechnik gewählt werden [En12].
- *Grundsatz der Wirtschaftlichkeit:* Der Grundsatz der Wirtschaftlichkeit besagt, dass bei der Modellierung auf ein sinnvolles Kosten-Nutzen-Verhältnis geachtet werden muss, was bedeutet, dass die Kosten der Modellierung den entstehenden Nutzen nicht übersteigen dürfen [En12].
- *Grundsatz des systematischen Aufbaus:* Ein Sachverhalt kann aus verschiedenen Sichten gezeigt werden. Diese Sichten können wie beispielsweise bei der UML in Struktur- und Verhaltenssicht oder wie in ARIS in Funktions-, Daten- und Organisationssicht unterteilt werden [BRS95]. Mit dem Grundsatz des systematischen Aufbaus wird gefordert, zusätzlich zu den Modellen, die die verschiedenen Sichten zeigen, ein sichtenübergreifendes Metamodell zu entwerfen, das den Zusammenhang der anderen Modelle erklärt [En12].
- *Grundsatz der Klarheit:* Ein Modell ist für den Modelladressaten nur dann von Nutzen, wenn es verständlich ist. Deshalb ist zunächst darauf zu achten, „die Modellierungstechnik [...] nutzeradäquat“ [En12] auszuwählen. Weiterhin ist besonders bei

komplexen Prozessen auf ein sinnvolles Maß an Detailreichtum zu achten. Um die Klarheit eines Modells zu gewährleisten, wird deshalb empfohlen, das Modell vertikal oder horizontal zu dekomponieren [BRS95]. Durch den Einsatz der vertikalen und horizontalen Dekomposition wird ermöglicht, die Semantik eines Prozesses präzise abzubilden, während das detailärmerere Modell weiterhin verständlich bleibt.

- *Grundsatz der Vergleichbarkeit:* Der Grundsatz der Vergleichbarkeit ist mit dem der Klarheit verbunden [BRS95]. Modelle müssen semantisch vergleichbar sein, auch wenn unterschiedliche Modellierungstechniken angewendet wurden. Von besonderer Wichtigkeit ist das im betriebswirtschaftlichen Bereich, um Ist- und Soll-Modelle vergleichen zu können [En12].

3 UML (Unified Modeling Language)

Bei UML, kurz für Unified Modeling Language, handelt es sich um eine visuelle Sprache beziehungsweise Notation, die für das Spezifizieren, Konstruieren und Dokumentieren von Systemen [OM11] verwendet wird.

Die UML wurde in den 1980er und -90er Jahren von Grady Booch, James Rumbaugh und Ivar Jacobson entwickelt. Booch, Rumbaugh und Jacobson entwickelten zunächst unabhängig voneinander verschiedene Darstellungsmethoden, entschieden sich aber Mitte der 1990er Jahre dazu, gemeinsam an einer Modellierungssprache weiterzuarbeiten. Kurz darauf wurde ein UML-Konsortium gegründet, an dem sich beispielsweise auch Hewlett Packard, Microsoft, Oracle und Texas Instruments beteiligten. Im Jahr 1997 wurde die erste Version der UML vorgestellt. Das Konsortium arbeitete bis zur Version 1.1 an der UML und übergab diese dann an die OMG (Object Management Group), die die Pflege und Weiterentwicklung der Modellersprache übernahm [Sc03]. Im März 2015 wurde die UML Version 2.5 [OM15] veröffentlicht.

In [Ke05] werden sechs Vorteile der UML benannt, die im Folgenden kurz vorgestellt werden.

- *Eindeutigkeit:* Alle Notationselemente der UML sind semantisch präzise und eindeutig, sodass Missverständnisse vermieden werden.
- *Verständlichkeit:* Die UML beinhaltet viele einfach gestaltete Notationselemente, mit denen verschiedene Sichten auf ein Softwaresystem modelliert werden. Die Kommunikation aller Projektbeteiligten wird durch die Verwendung von UML erleichtert.
- *Ausdrucksstärke:* Die UML beinhaltet viele verschiedene Notationselemente, womit ermöglicht wird, alle Details eines Softwaresystems mit Diagrammtypen der UML darzustellen.
- *Standardisierung und Akzeptanz:* Der OMG, die sich, wie bereits zuvor erwähnt, mit der Pflege und Weiterentwicklung der UML beschäftigt, gehören mehr als 800 Un-

ternehmen an. Die UML ist daher weit verbreitet und damit sowohl bei den meisten Softwareentwicklern bekannt als auch akzeptiert.

- *Plattform- und Sprachunabhängigkeit:* Die UML ist nicht von einer bestimmten Plattform oder Programmiersprache abhängig. Die Wahl der Sprache und Plattform wird daher durch die Verwendung von UML nicht eingeschränkt.
- *Unabhängigkeit und Vorgehensmodelle:* Bei der UML handelt es sich um Werkzeuge zur Spezifikation, Konstruktion und Dokumentation von Softwaresystemen. Die UML ist nicht abhängig von einem bestimmten Vorgehensmodell zur eigentlichen Entwicklung, was bedeutet, dass jedes beliebige Vorgehensmodell in Kombination mit UML verwendet werden kann.

Da die Verwendung von UML, wie eben gezeigt, einige Vorteile, allen voran die Standardisierung und hohe Akzeptanz, mit sich bringt, empfiehlt es sich, die UML auch weiterhin für die Projektarbeit zu nutzen.

Die verschiedenen Diagrammtypen der UML werden in die Gruppen der Strukturdiagramme und der Verhaltensdiagramme unterteilt. Zu den Strukturdiagrammen gehören Diagrammtypen, die „statische, zeitunabhängige Elemente des Systems“ [Ke05] zeigen. Diagrammtypen aus der Gruppe der Verhaltensdiagramme sind dagegen dynamisch und zeigen das System- und Komponentenverhalten [Ke05].

Die folgenden Unterkapitel stellen jeweils einen Diagrammtypen vor, der entweder von der Projektgruppe Business Intelligence as a Service für die Modellierung ihres BICE-Systems verwendet wurde oder zur weiteren Detaillierung dient.

3.1 Komponentendiagramm

Das Komponentendiagramm gehört zu der Gruppe der Strukturdiagramme. Es wird zur Spezifikation von Softwarearchitekturen eingesetzt und zumeist parallel zu Verteilungsdiagrammen modelliert [Ke05]. Die Abbildung 1 zeigt das Komponentendiagramm des BICE-Systems der Projektgruppe Business Intelligence as a Service. Der Abbildung sind sämtliche Softwarekomponenten des Systems und benötigte Schnittstellen zu entnehmen. Dadurch eignen sich Komponentendiagramme nicht nur dazu, einen Überblick über die Architektur der Software zu geben, sondern in großen Projekten auch, um die eigentliche Entwicklung auf mehrere Teams aufzuteilen [Ke05].

3.2 Kompositionsstrukturdiagramm

Auch das Kompositionsstrukturdiagramm gehört zur Gruppe der Strukturdiagramme. Es eignet sich zur weiteren Detaillierung eines Komponentendiagramms, da es die interne Komponentenstruktur und die Interaktionspunkte zu den weiteren Komponenten eines Softwaresystems aufzeigt [Ke05]. Da sich die Projektgruppe CAIS mit der Weiterentwicklung und Verbesserung der Software-Architektur der vorangegangenen Projektgruppe

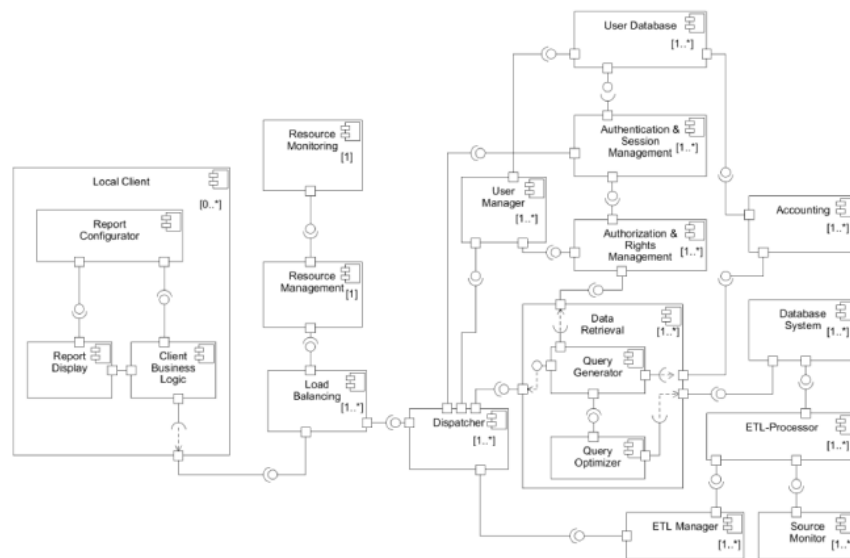


Abb. 1: Komponentendiagramm des BICE-Systems [No15]

beschäftigen wird, empfiehlt es sich, ein solches Kompositionsstrukturdiagramm zu entwickeln und damit einen detaillierten Überblick über die Software-Architektur zu bieten.

3.3 Verteilungsdiagramm

Wie bereits zuvor angedeutet, gehört auch das Verteilungsdiagramm zur Gruppe der Strukturdiagramme und wird häufig parallel zu Komponentendiagrammen eingesetzt. Ein Verteilungsdiagramm stellt die Spezifikation der Hardware- und Softwareumgebung dar und wird meist in der Entwurf und Design-Phase modelliert, um auf seiner Basis notwendige Hard- und Softwarekomponenten zu beschaffen.

3.4 Sequenzdiagramm

Das Sequenzdiagramm gehört zu einer Untergruppe der Verhaltensdiagramme, die Interaktionsdiagramme genannt wird. Die Diagramme dieser Untergruppe zeigen die Interaktion und den Nachrichtenaustausch zwischen Objekten [Ke05]. Sequenzdiagramme können sowohl in der Phase der Analyse und Definition als auch in der Entwurf- und Design-Phase eingesetzt werden. In der Phase der Analyse und Definition dient das Sequenzdiagramm dazu, den zu analysierenden Geschäftsprozess mit allen beteiligten Personen und Systemen besser verstehen zu können. Während der Phase des Entwurfs und Designs werden

Sequenzdiagramme eingesetzt, um die Interaktionen innerhalb des Systems in ihrem zeitlichen Rahmen abzubilden. Sequenzdiagramme zeigen also die Reihenfolge einer Interaktion, aber auch welche Befehle vom Benutzer gegeben werden müssen, um eine Interaktion zu starten und welche Nachricht an den Benutzer zurückgegeben wird [Ke05].

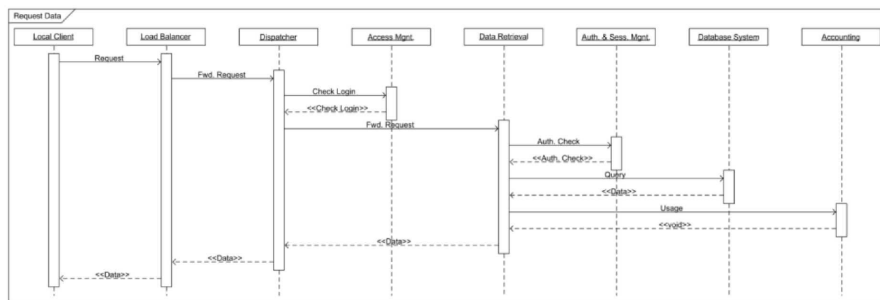


Abb. 2: Sequenzdiagramm des BICE-Systems [No15]

Wie in der Abbildung 2 zu sehen ist, hat die Projektgruppe Business Intelligence as a Service ein Sequenzdiagramm modelliert, um den Nachrichtenaustausch im BICE-Systeme zu zeigen, wenn ein Report vom Benutzer angefordert wird.

3.5 Anwendungsfalldiagramm

Auch das Anwendungsfalldiagramm gehört zur Gruppe der Verhaltensdiagramme. Im Gegensatz zu anderen hier vorgestellten Modellen zeigt ein Anwendungsfalldiagramm auf einer hohen Abstraktionsebene alle Anwendungsfälle, die für einen externen Nutzer wahrnehmbar und von Nutzen sind [Ke05].

Sollte noch kein solches Diagramm für das BICE-System modelliert worden sein, empfiehlt es sich, dies nachzuholen. Damit würde ermöglicht werden, sich nicht nur an fachkundige Beteiligte und Interessierte zu wenden, sondern auch späteren Anwendern des Systems mit einem einfach verständlichen Diagramm die Möglichkeiten aufzuzeigen, die das System ihnen bietet.

4 Weitere Diagrammtypen

Dieses Kapitel beschäftigt sich mit zwei weiteren Diagrammtypen, die nicht zur UML-Notation gehören. Zunächst wird das Entity Relationship-Modell vorgestellt, das zur Datenmodellierung eingesetzt wird und anschließend das Dekompositionsdiagramm, das ähnlich wie ein Organigramm eingesetzt wird, um Daten- und Kontrollstrukturen aufzuzeigen.

4.1 Entity-Relationship-Modell

Die zuvor vorgestellten Diagrammtypen eignen sich zur Modellierung von Software-Systemen oder zeigen, wie das Anwendungsfalldiagramm, alle Funktionen, die ein System bietet. Für den konzeptuellen Entwurf von Datenbanken wird meist das Entity-Relationship-Modell verwendet, das hier im Folgenden kurz vorgestellt werden soll.

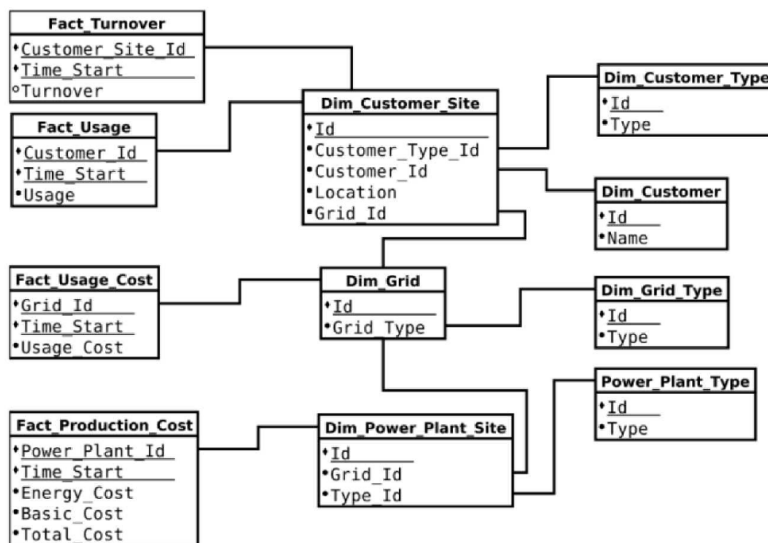


Abb. 3: Entity Relationship-Modell des BICE-Systems [No15]

Ein Entity-Relationship-Modell zeigt einen Ausschnitt der realen Welt durch sogenannte Entity-Mengen und ihren Beziehungen untereinander [Fa07]. Eindeutig identifizierbare Objekte aus der realen Welt, wie beispielsweise eine Person oder ein Gegenstand, werden als Entity bezeichnet. Die Eigenschaften der Entities werden durch Attribute und Attributwerte beschrieben. Die minimale Menge von Attributen, die zur Identifizierung einer Entity benötigt wird, heißt Schlüssel. Der Zusammenhang dieser Entities wird durch Beziehungen dargestellt. Ein Entity-Relationship-Modell des BICE-Systems ist in Abbildung 3 zu sehen.

4.2 Dekompositionsdiagramm

Bei dem Dekompositionsdiagramm, das auch als Jackson-Diagramm bezeichnet wird [SN92], handelt es sich um einen Diagrammtyp, der häufig als Organigramm eingesetzt wird [IB15].

Ab 1975 stellte Michael A. Jackson diese Entwurfsmethodik vor, die sich besonders für Entwicklungsprojekte, bei denen mit großen Datenmengen gearbeitet wird, eignet [IT15]. Das Grundprinzip der Entwurfsmethodik von Jackson besteht darin, zunächst die Datenstruktur zu bestimmen und zu modellieren und anhand dieses Modells die Software zu

entwickeln [Kö15]. Zur Modellierung können die drei Strukturierungselemente Sequenz (Aneinanderreihung), Iteration (Wiederholung) und Selektion (Auswahl) verwendet werden [IT15], die je nach dem, ob sie in der Darstellung von Datenstrukturen oder in der Darstellung von Kontrollstrukturen verwendet werden, eine unterschiedliche Semantik besitzen [Kö15].

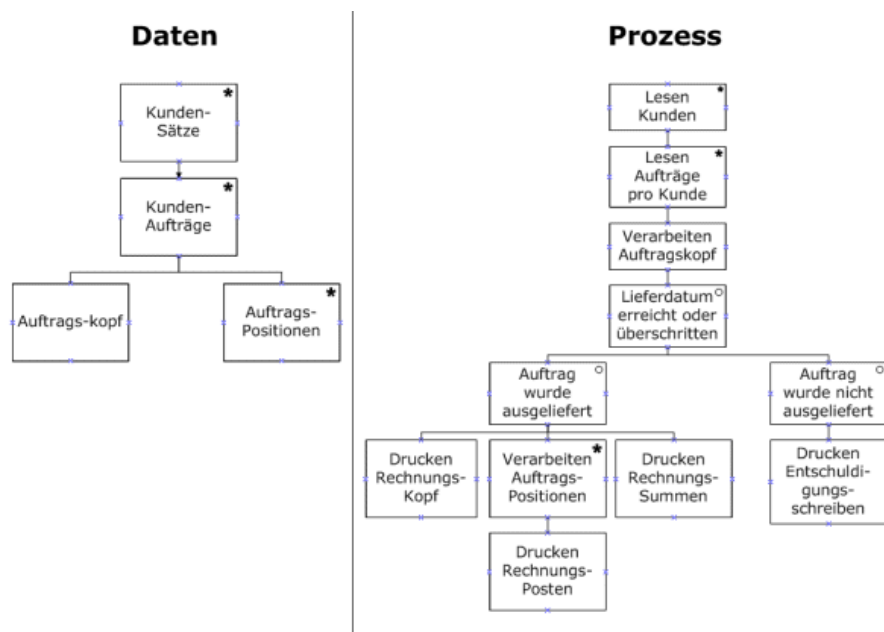


Abb. 4: Dekompositionsdiagramm [Wi15]

Die Abbildung 4 zeigt auf der linken Seite ein Beispiel für die Datenmodellierung mit der Jackson-Methode und auf der rechten Seite ein Beispiel für die Prozessmodellierung. Der zuvor bereits erwähnte hierarchische Aufbau ist hier gut zu erkennen.

5 Fazit und Empfehlung

In der vorliegenden Seminararbeit wurde zunächst die Frage geklärt, was unter einem Modell verstanden wird und weshalb die Modellierung einen wichtigen Schritt im Prozess der Softwareentwicklung darstellt. Darauf folgend wurden zunächst die Grundsätze ordnungsgemäßer Modellierung von Becker, Rosemann und Schütte vorgestellt, die bei Einhaltung ein hohes Maß an Modellqualität gewährleisten. Das dritte Kapitel befasste sich mit der Notation UML und einigen ausgewählten Diagrammtypen, die von der vorangegangenen Projektgruppe verwendet wurden oder deren Modellierung bei der Weiterentwicklung der Systemarchitektur empfehlenswert ist. Des Weiteren wurden das Dekompositionsdiagramm und das Entity-Relationship-Modell vorgestellt.

Die Arbeit hat noch einmal den hohen Stellenwert der Modellierung für den Erfolg eines Projektes gezeigt. Zu Beginn der Projektarbeit sollte, den Grundsätzen ordnungsgemäßer Modellierung folgend, überprüft werden, für welchen Zweck weitere Modellierung stattfinden muss. Eine Empfehlung für einen bestimmten Diagrammtyp kann an dieser Stelle nicht gegeben werden. Es bietet sich jedoch an, zu überdenken, ob die Erstellung eines Verteilungsdiagramms zum bereits vorhandenen Komponentendiagramm oder die Erstellung eines Kompositionsstrukturdiagramms zur Planung der Weiterentwicklung am System sinnvoll ist. Unbedingt vermieden werden sollten jedoch Modellierungsvorhaben, die für das Projekt nicht relevant sind oder deren Kosten (in der Projektgruppe wahrscheinlich Kosten in Form von Zeit) den mit sich bringenden Nutzen übersteigen.

Literaturverzeichnis

- [BRS95] Becker, Jörg; Rosemann, Michael; Schütte, Reinhard: Grundsätze ordnungsgemäßer Modellierung. WIRTSCHAFTSINFORMATIK 37, S. 435 – 445, 1995. http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_wirtschaftswissenschaften/wi/sysent/studium/lehre_ss07/modprakt/downloads/Becker1995.pdf, Stand: 28.10.2015.
- [En12] Enzyklopädie der Wirtschaftsinformatik - Grundsätze ordnungsgemäßer Modellierung (Prof. Dr. Jörg Becker, zuletzt bearbeitet: 31.10.2012), <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaeten-der-Systementwicklung/Problemanalyse-/Grundsätze-ordnungsgemäse-Modellierung>, Stand: 28.10.2015.
- [Fa07] Faeskorn-Woyke, Heide; Bertelsmeier, Birgit; Riemer, Petra; Bauer, Elena: Theorie und Praxis mit SQL2003, Oracle und MySQL. Pearson Verlag, 2007.
- [IB15] IBM Knowledge Center - Dekompositionsdiagramme, http://www-01.ibm.com/support/knowledgecenter/SS6RBX_11.4.2/com.ibm.sa.process.doc/topics/c_Decomposition_Diag.html?lang=de, Stand 03.11.2015.
- [IT15] IT Wissen Online Lexikon: Jackson-Diagramm, <http://www.itwissen.info/definition/lexikon/Jackson-Diagramm-Jackson-chart.html>, Stand 03.11.2015.
- [Kö15] Köning, Rainer: Jackson-Diagram, http://kroening-online.info/Method/Jackson/m_jackson.php Stand: 03.11.2015.
- [Ke05] Kecher, Joseph: UML 2.0 - Das umfassende Handbuch. Galileo Press GmbH, Bonn, 2005.
- [KKB08] Kastens, Uwe; Kleine Büning, Hans: Modellierung - Grundlagen und Methoden. Hanser Verlag, München, 2008.
- [Le08] Lehmann, Frank R.: Integrierte Prozessmodellierung mit ARIS. dpunkt.verlag, Heidelberg, 2008.
- [No15] Norkus, Oliver et al.: An Approach for a Cloud-based Contribution Margin Dashboard in the Field of Electricity Trading. INFORMATIK 2015, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn, 2015.

-
- [OM11] OMG Unified Modeling Language Version 2.4.1 (Dokumentation), <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>, Stand 20.10.2015.
- [OM15] OMG Unified Modeling Language Version 2.5 (Dokumentation), <http://www.omg.org/spec/UML/2.5/PDF/>, Stand 19.10.2015.
- [RH06] Reussner, Ralf; Hasselbring, Wilhelm, Hrsg. Handbuch der Software-Architektur. dpunkt.Verlag, Heidelberg, 2006.
- [Sc03] Schmuller, Joseph: UML - Der einfache Einstieg in die visuelle Objektmodellierung. Markt + Technik Verlag, München, 2003.
- [SN92] Schönthaler, Frank; Németh, Tibor: Software-Entwicklungswerkzeuge: Methodische Grundlagen. B. G. Teubner Verlag, Stuttgart, 1992.
- [Wi15] Wikipedia: Jackson-Diagramm, <https://de.wikipedia.org/wiki/Jackson-Diagramm>, Stand: 01.10.2015.

Business Intelligence

Dennis Vorreiter¹

Abstract: *Business Intelligence* wird ein immer prägnanterer Ausdruck für eine besondere Art von analytischen Informationssystemen, die heutzutage vermehrt in großen und kleinen Unternehmen zum Einsatz kommen. Diese Arbeit befasst sich mit der Erläuterungen von *Business Intelligence* und erklärt welche Komponenten und Phasen essentiell für eine Handhabung mit den heutigen Datenvolumina und ihrer Heterogenität sind, um eine zeit- und kosteneffiziente Planungs-, Entscheidungs- und Controllings-Unterstützung bieten zu können.

Keywords: Business Intelligence, Analytische Informationssysteme, Data Warehouse, ETL, OLAP

1 Einführung

Das Ziel eines jeden Unternehmens ist es in der Regel das eigene Wachstum und eine damit einhergehende Gewinnmaximierung. Hierfür ist es insbesondere wichtig einen genauen Überblick über den aktuellen Status des Unternehmens und die derzeitige Marktentwicklung zu haben, sowie Prognosen und Kalkulationen über einen kommenden Zeitraum treffen zu können und frühzeitig Trends abzusehen. Dies erfordert Daten, aus denen diese Informationen gezogen werden können. Herausforderung hierbei stellt in der heutigen Zeit allerdings eben genau die Größe und die Heterogenität derartiger Daten dar. Denn neben firmeninterner Daten sollen möglichst auch externe Daten hinzugezogen werden können. Zusammen mit immer ausgereifteren Sensordaten(RFID)([ML13], S.1) und dem Datenfluss des Web2.0 entsteht somit ein riesiges Datenvolumen, das es zu verarbeiten gilt. Bei einem Großteil davon handelt es sich nicht einmal um operative, geschäftsbedingt anfallende Daten (Vgl. [ML13], S.1).

Allein 2010 schätzte man die weltweit gesamte Datenmenge auf mehrere Zetabyte und sagte einen Zuwachs von jährlich 100 % voraus (Vgl. [ML13], S.1).

Des Weiteren geht der Trend immer mehr in Richtung orts- und zeitunabhängiger Kommunikation. Daten sollen möglichst überall zu jeder Zeit und auf jedem Endgerät zur Verfügung stehen, sodass innerhalb des Datenflusses eine hohe Dynamik entsteht. Zeitgleich ändern sich auch durch Faktoren wie virtuelle Arbeitsplätze die Arbeitsprozesse. Gerade um zeitnahe Planungs- und Entscheidungsaufgaben erledigen zu können, sind Anforderungen an verfügbare Daten und Analysen unabdingbar (Vgl. [ML13], S.1).

Um diese Anforderungen an Informationssystemen leisten zu können, wurde der Begriff *Business Intelligence (BI)* geprägt. Dieser soll den Ansatz verfolgen eine umfassende Integration von Strategien, Prozessen und Technologien zum Verarbeiten interner und ex-

¹ Carl von Ossietzky Universität Oldenburg, Fakultät 2: Informatik, Wirtschafts- und Rechtswissenschaften, dennis.vorreiter@uni-oldenburg.de

terner Daten unter Hilfe analytischer Informationssysteme zu bieten (Vgl. [In15]). Dabei soll eine optimale Datenbereitstellung trotz verteilter und heterogener Datenbestände gewährleistet sein. Sodass BI einen Fokus darauf legt, Daten entsprechend so zu strukturieren, dass eine modellgestützte Auswertung kostengünstig möglich ist. Somit können externe und interne Daten in einem Data Warehouse verbunden werden und mittels OLAP-Abfragen oder Data-Mining zu analytischem Wissen ausgewertet werden (Vgl. [ML13], S.2).

Im Folgenden sei die Architektur eines solchen BI-System beschrieben und erläutert. Hierfür werden die Datenintegration mittels des ETL-Prinzips beschrieben. Darauf folgt die Beschreibung der Datenspeicherung innerhalb eines Data Warehouse. Danach werden die Möglichkeiten zur Datenanalyse diskutiert. Abschluss stellt ein kurzes Fazit über die zukünftige Entwicklung der Business Intelligence dar.

2 Business Intelligence Konzept

Um ein Verständnis vom Konzept BI zu bekommen, sollte man sich zunächst einen Überblick über die Komponenten und ihre Funktionen verschaffen, die innerhalb eines solchen Systems, auch Data Warehouse Architektur genannt, agieren.

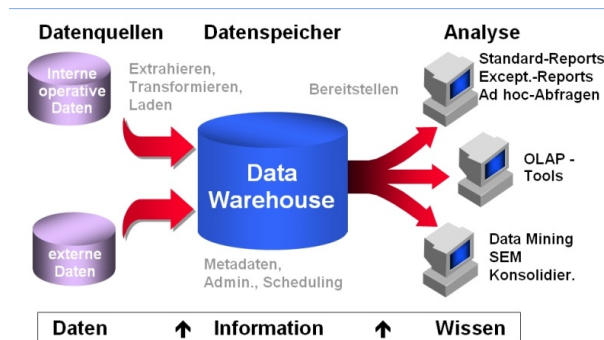


Abb. 1: Überblick einer BI-Architektur [Bu13].

Wie in Abbildung 1 gezeigt, lässt sich eine Data Warehouse Architektur in drei Bereiche aufteilen: Daten, Information und Wissen.

Der Daten-Bereich beschreibt den Integrationsvorgang der Datenquellen. Hierfür werden operative und externe Daten mittels des ETL-Prinzips (siehe 2.1) verarbeitet. Im Informations-Sektor steht als physische Einheit das Data Warehouse, das als Datenbanksystem die integrierten Daten speichert. Dafür werden spezielle Prozesse genutzt, die sich von einem üblichen Datenbanksystem unterscheiden (siehe 2.2). Aus diesen gespeicherten Daten wird letztlich das relevante Wissen genommen und für den Endanwender sichtbar gemacht. Um Wissen zu generieren werden hierfür *Online-Analytical-Processing* (OLAP)-, Ad-Hoc-, oder Data-Mining-Prozesse genutzt (siehe 2.3). Bei einer BI-Architektur handelt es sich nicht nur um ein Data Warehouse. Das Data Warehouse ist vielmehr als grundlegendes System zu betrachten. (Vgl. [BH05]).

BI-Systeme grenzen sich klar von gewöhnlichen dispositiven Systemen ab. Die sogenannten *Online Transactional Processing* (OLTP) sind darauf ausgelegt hauptsächlich wenig Daten innerhalb ihrer Transaktionen zu verarbeiten. Diese Transaktionen bestehen aus Lese-, Änderungs- und Löschoptionen. Der Datenbestand besteht hauptsächlich aus detaillierten aktuellen Werten. Für die entscheidungsunterstützenden BI-Systeme sind diese Eigenschaften ungeeignet. Für Analyseverfahren werden hauptsächlich Leseoperationen benötigt. Ebenso müssen viele historische Daten bereitliegen, was Löschoptionen selten macht. Aus diesem Grund werden BI-Architekturen als OLAP-Systeme angelegt. Hier liegen die Daten verdichtet, aufbereitet und mehrdimensional vor. Um den historischen Bezug zu berücksichtigen müssen dafür große Datenmengen in den Transaktionen verarbeitet werden (Vgl. [ML13], S.14)(Vgl. [BH05], S.5).

Essentielle Eigenschaft des BI ist die Datenintegration, an Durchführung dieser letztendlich die Qualität der Analysen gemessen werden kann. Die Integration koordiniert den riesigen Fluss an Daten, um sie in ein geeignetes Informationszentrum zusammenzufassen, das neben wertvollem Analysematerial auch eine Abstimmung des Wissensstands und der Geschäftsprozesse liefert (Vgl. [BI07]).

2.1 Datenintegration durch ETL

Um Wissen aus gespeicherten Daten ziehen zu können, müssen relevante Daten zunächst in geeigneter Form zur Verfügung stehen. Für diese Aufgabe wurde der ETL-Prozess (Extract, Transform, Load) entwickelt. Er wird als wichtigster Bestandteil innerhalb der BI-Funktionalität angesehen, da auf ihm letztlich die Vollständigkeit, Fehlerfreiheit, Widerspruchsfreiheit und Aktualität des gesamten System aufbaut (Vgl. [ML13], S.31, Z.11ff.). Zugleich ist er der teuerst Prozess und soll als dieser möglichst im Hintergrund effizient agieren. Es gilt somit, dass der ETL-Prozess möglichst kurz durchgeführt werden sollte (Vgl. [Gr08]).

2.1.1 Extraktion

Der ETL-Prozess beginnt mit der Extraktion der internen (operativen) und externen Daten. In diesem Teilprozess ist auch die Bereinigung der Daten mit beinhaltet. Untersucht werden diese Daten bezüglich vier V's: "Volume (Datenvolumen), Velocity (Veränderungsgeschwindigkeit), Variety (Datenvielfalt) und Veracity (Wahrheitswert)"([ML13], S27,Z.8-9). Die größte Herausforderung in diesem Schritt ist neben des hohen Datenvolumens und der Komplexität der Daten, den Wahrheitswert dieser Daten zu ermitteln, um die Widerspruchsfreiheit zu garantieren (Vgl. [ML13], S27,Z.8ff.).

Ebenso muss das System mit fehlerhaften oder inkompatiblen Datenformaten umgehen können, indem es geeignete Filterungen benutzt (Vgl. [BI07]).

Um die Aktualität zu gewährleisten muss dieser Schritt regelmäßig ausgeführt werden. Hier gibt es sowohl Ansätze, die diesen Schritt wöchentlich, täglich oder sogar minütig, in Echtzeit ("Real-Time-Data-Warehousing"[ML13], S.31,Z.17), durchführen. Um möglichst

kurze Sperrzeiten zu erzeugen, werden in der Regel nur Daten selektiert, die für das System neu sind (Vgl. [Gr08]).

2.1.2 Transformation

Darauf folgt der Transformation-Schritt, der die Daten in das vom Data Warehouse genutzte Zielschema und Format aggregiert. Dafür muss die Gesamtheit der Daten zunächst auf einen homogenen Stand gebracht werden. Dies wird innerhalb der sogenannten *data migration* bzw. des *data mapping* erledigt (Vgl. [ML13], S.32, Z. 19.). Diese besteht im Groben aus Filterung, Harmonisierung, Aggregation und Anreicherung der Daten.

Bei der Filterung werden verschiedene Klassen von Mängeln in den Datensätzen erkannt und möglichst behoben. Unterschieden wird hierbei auf syntaktischer und semantischer Ebene. Die erste Klasse an Mängeln umfasst einfache Fehler in bestimmten Zeichen (syntaktisch) und das fehlen bestimmter Datenwerte (semantisch), die algorithmisch behoben werden können. Die zweite Klasse beinhaltet inkompatible Formatierungen (syntaktisch) und unglaubliche Werte bzw. Ausreißer (semantisch). Die letzte Klasse umfasst alle übrigen semantischen Fehler, die nicht algorithmisch erkannt werden können. (Vgl. [Gr08], S.4).

Die Harmonisierung führt eine Angleichung der Datenfaktoren durch, die semantisch zwar gleich, syntaktisch jedoch unterschiedliche Nennungen besitzen. Dabei treten Konflikte auf, die sich in Namens- und Datenkonflikte unterteilen lassen. Als Konfliktlösung werden hierbei Datentypen auf einen gemeinsamen Datentyp angepasst, die Kodierung einzelner Datenbestände angeglichen, Zeichenketten (bspw. Datumsangaben) vereinheitlicht, Synonyme zusammengeführt, Homonyme getrennt, Skalierungen von Einheiten angeglichen und entsprechende Attributswerte kombiniert oder separiert. Zusammengefasst bezieht sich die Handhabung der Daten stark auf das *Natural Language Processing*, was durch die Heterogenität der Daten hervorgerufen wird. Außerdem wird eine Schlüsselbehandlung durchgeführt. Diese dient dem Finden oder künstlichen Erzeugen primärer (und damit für ein Datenbanksystem essentiell wichtiger) Primärschlüssel (Vgl. [ML13], S.32, Z. 20ff.). Am Ende dieses Teilschrittes liegen die Daten in einer Form vor, die generell für eine Analyse nutzbar wäre (Vgl. [Gr08], S.6).

Im nächsten Schritt, der Aggregation, wird dafür gesorgt, dass die Daten eine entsprechende Granularität aufweisen. Hier werden Hierarchietabellen angelegt, um Vorberechnungen entwickeln zu können, die bereits in das Data Warehouse gespeichert werden sollen. Als Wurzel der Hierarchie dienen in der Regel Ausdrücke wie "Gesamtöder Älle", wobei die Folgeknoten abhängig sind von der Zielgruppe bzw. dem Endanwender. Der Datenbestand wird somit personalisiert. Dieser Vorgang dient insbesondere der Performance-Steigerung (Vgl. [Gr08], S.7).

Im letzten Schritt, der Anreicherung der Daten, wird der Datenbestand um Kennziffern und Hintergrundwissen erweitert, das für mehrere Zielgruppen relevant sein könnte. Die ermöglicht durch die Vorberechnungen eine effizientere Abfrage und Analyse, gewährleistet

eine Konsistenz der Daten und gibt eine gemeinsame Wissensbasis für das Unternehmen (Vgl. [Gr08], S.8).

2.1.3 Laden

Alle bis dahin erzeugte Daten innerhalb der Extraktion und Transformationen liegen in einem Teilbereich des Data Warehouses, dem Zwischenspeicher. Somit werden zuletzt die bis dahin selektierten, replizierten und transformierten Daten im Laden-Prozess fest in den Kern des Data Warehouse geladen, damit sie für die weitere Verwendung in den Data Marts bereitstehen. Dies geschieht allerdings nicht immer vollständig, da es sich hierbei um nicht unbedeutende Datenmengen handelt, sondern geschieht auf Prinzip der Aktualisierung. Nur neue oder veränderte Daten werden geladen. In diesem Schritt entsteht auch die eigentliche Blockierung der Datenbank. Da während des Ladens ein inkonsistenter Zustand entsteht, muss die Datenbank für Anfragen gesperrt sein. Dies darf allerdings aus wirtschaftlicher Sicht nur in einem bestimmten Zeitfenster geschehen (meist Nachts)(Vgl. [Gr08], S.9).

2.1.4 Datenqualität

Anhand qualitativ-hochwertiger Daten können geeignete Analysen erstellt werden. Deshalb wird die Qualität der Daten auch dadurch beschrieben, wie sehr sie sich für die Analyse eignen, sprich wie viel Wissen aus ihnen entnommen werden kann (Vgl. [ML13], S.38, Z. 8ff.).

Charakterisiert wird die Qualität anhand vier Faktoren: Glaubwürdigkeit, Nützlichkeit, Interpretierbarkeit und der Schlüsselintegrität. Unter Glaubwürdigkeit fällt ein Großteil der im ETL-Prozess liegenden Anforderungen widerspruchsfreie, korrekte und vertrauenswürdige Daten zu erhalten. Die Nützlichkeit beschreibt die Lückenlosigkeit der Daten. Besitzen Datenbestände beispielsweise fehlende Einträge, sind nur Default-Werte eingetragen, gibt es unzulässige Werte oder besitzen sie Duplikate oder Tippfehler, so sind sie als nutzlos anzusehen. Die Interpretierbarkeit baut dagegen auf die Harmonisierung der Daten auf und beschreibt Eindeutigkeit und Verständlichkeit der Daten. Die Schlüsselintegrität schließlich umfasst die korrekte Verwendung von Schlüsselattributen innerhalb des Datenbestandes (Vgl. [ML13], S.39).

2.2 Datenspeicherung im Data Warehouse

Das Data Warehouse beschreibt ein Datenbankkonzept, das für analytische Daten ausgelegt wurde. Es ähnelt einem gängigen Informationssystem, sammelt jedoch hauptsächlich geschäftsbezogene und geschäftsrelevante Daten, und ist nicht eine umfassende Datensammlung. Grundsätzlich lässt sich das Data Warehouse als ein Datenbestand charakterisieren, der subjektorientiert, integriert, zeitraumbezogen und persistent ist (Vgl. [ML13], S.14, Z. 8ff.).

Das Data Warehouse unterteilt sich in mehrere Komponenten:

Die Daten, die innerhalb des ETL-Prozesses selektiert und generiert werden, befinden sich in der sogenannten *Staging Area*, dem Zwischenspeicher. Erst wenn der Ladeprozess abgeschlossen ist, werden die Daten fest in das *Core Data Warehouse* gespeichert und sind somit persistent. Hier werden sogenannte *Data Marts* angelegt, entweder zusätzlich oder als Hauptspeicher, die in Form von Datenwürfeln (Data Cubes) personalisierte Datenbestände repräsentieren. (Vgl. [ML13], S.18).

Diese Data Cubes weisen eine besondere Art der Speicherungsform auf, die sich für die analytischen Zwecke der Daten als besonders effizient erwiesen hat. Da bei der Analyse der Daten auf mehrere Faktoren eingegangen wird (Menge pro Zeit u.Ä.), verfolgt man hier einen mehrdimensionalen Ansatz. Zugrunde liegt hier das 3D-Prinzip, das drei Orthogonale, also unabhängige Kenngrößen, als Dimensionen postuliert. Die Grunddimensionen sind die Raum-, Zeit- und Sachdimension, auf die sich alle Dimensionen zurückführen lassen. Von dort aus kann im Folgenden die Sachdimension in weitere Dimensionen, wie "Verkäufer", "Kunde" oder "Produkt", aufgespalten werden. Die darin gespeicherten Daten werden Fakten genannt und können über verschiedene Funktionen (siehe 2.3) analytisch dargestellt werden. (Vgl. [ML13], S.15).

Um die analytischen Vorgänge weiterhin effizienter zu gestalten, ist man dazu übergegangen, die Analysen für den Speicherumfang gängiger Hauptspeicher zu optimieren. So wird der große Verlangsamungsfaktor der Festplatte umgangen (Vgl. [ML13], S.25-26).

Zur Organisation der Data Marts gibt es verschiedene Möglichkeiten. Diese Organisation sind über die Dimensionen der mehrdimensionalen Datenwürfel definiert. So kann die gesamte Architektur darauf ausgelegt sein, unabhängige Data Marts, quasi für jede Abteilung kleine Data Warehouses zu erzeugen. Das kann eine günstige Möglichkeit darstellen, da ein Unternehmen so Schritt für Schritt eine BI Architektur implementieren können, birgt dennoch die Gefahr eines uneinheitlichen Wissensstands. Deshalb gibt es die Alternative die Data Marts mittels gemeinsamer Dimensionen zu verknüpfen. Das macht es überdies möglich gefahrlos zwischen den Datenbeständen zu wechseln. Eine ähnliche Organisation stellen abhängige Data Marts dar. Hier wird bei Einführung des System bereits unternehmensweit ein einheitliches Schema angelegt (Vgl. [ML13], S.21).

Bei großen oder sich häufig ändernden Quelldaten hat es sich außerdem als effizient herausgestellt, die Daten direkt aus den Quellsystem zu erfragen und so keine Kopien innerhalb des eigenen System zu nutzen. Hierbei spricht man auch von föderierter Architektur. Bei großen Datenmengen wird heute zu einer parallelen Verarbeitung des Datenstroms geraten, der mittels *Map-Reduce*-Prozesse in drei Phasen verarbeitet und entsprechend komprimiert wird (Vgl. [ML13], S.22).

Wie jedes Datenbanksystem besitzt auch das Data Warehouse außerdem ein Log-Verzeichnis, um konsistente Transaktionen gewährleisten zu können, einen Manager, zur Ablaufsteuerung, und eine Optimierungseinheit, um Anfragen effektiv verarbeiten zu können (Vgl. [ML13], S.20).

2.3 Datenanalyse

Ist der Datenintegrationsprozess und die Speicherung abgeschlossen, folgt die Auswertung der Daten über Analysen. Die zwei gängigsten Analysevorgänge seien im Folgenden erklärt.

2.3.1 OLAP-Analyse

Die Speicherung der Datenbestände in mehrdimensionale Datenwürfel bietet es an, die (OLAP)-Operationen auf die Daten anzuwenden. Bei dieser Analyse wird auf mehrere Dimensionen Bezug genommen, was sie besonders flexibel und effizient machen soll. Hierbei werden nicht, wie in vielen relationalen Datenbanken, 2-dimensionale Datenbestände auf Punktabfragen untersucht, sondern mehrdimensionale Daten auf bestimmte Bereiche (bspw. Zeitraum, per Standort, usw.), also Bereichsabfragen, untersucht. Für den Endanwender müssen somit aggregierte Daten, Kreuzprodukte von Dimensionen, als Datenwürfel, vorliegen, deren Verarbeitung innerhalb gewöhnlicher Datenbanksysteme hohen Aufwand erzeugen (Vgl. [ML13], S.51).

Die Operationen auf einen Data Cube, einem OLAP-Datenwürfel, kann man sich als eine Art Navigation über den Datenwürfel vorstellen. So ist es, wie in Abbildung 2 dargestellt, möglich bestimmte Tupel über eine angegebene Bedingung zu selektieren (*Slicing*). Ebenso können kleinere Würfel aus dem Ursprungwürfel projiziert werden (*Dicing*).

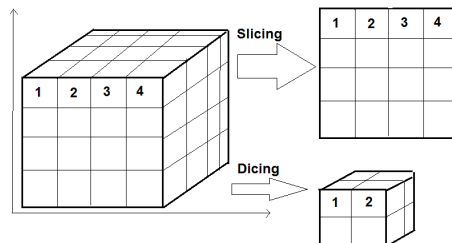


Abb. 2: Skizzenhafte Darstellung von OLAP-Operationen: Slicing und Dicing.

Unter Beachtung sogenannter Hierarchiemengen, die eine Art Ordnungsrelation der Attribute darstellen, lassen sich, wie in Abbildung 3, Generalisierungen mittels Aggregationen (*Roll-Up*) und Spezifizierungen (*Drill-Down*) durchführen.

Bietet es die zugrundeliegende Architektur an, kann außerdem zwischen den Datenwürfel gewechselt werden, wenn beispielsweise eine gemeinsame Dimension vorliegt (*Drill-across*) oder darunterliegende Daten betrachtet werden sollen (*Drill-through*).

Schließlich kann der Würfel rotiert werden, sodass Dimensionen nach Wunsch ausgetauscht werden können, wie Abbildung 4 zeigt (*Pivotierung*) (Vgl. [ML13], S.54-55).

Die Anforderungen an eine solche Verarbeitung wurden innerhalb des Kürzels *FASMI* definiert. Das System darf hiernach nicht länger als eine Sekunde, bei komplexeren Auf-

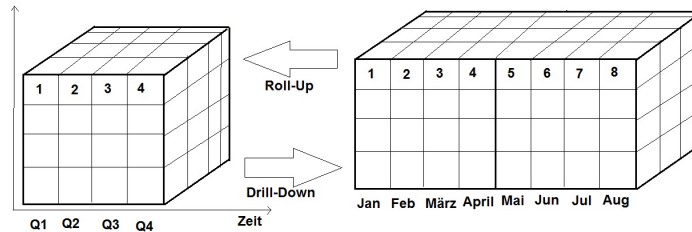


Abb. 3: Skizzenhafte Darstellung von OLAP-Operationen: Roll-Up und Drill Down anhand der Zeit.

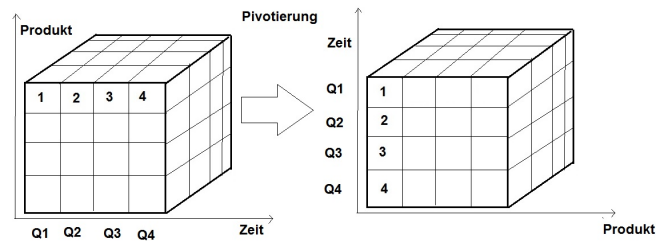


Abb. 4: Skizzenhafte Darstellung von OLAP-Operationen: Pivotierung.

gaben maximal 20 Sekunden brauchen, um zu antworten (*Fast*) und muss dem Anwender intuitive Werkzeuge zur Analyse mit wenig Programmieraufwand bereitstellen (*Analysis*). Rücksicht muss dabei auf den Mehrbenutzer-Betrieb genommen werden (*Shared*), indem die Zugriffe auf die mehrdimensionalen Daten (Multidimensional) entsprechend verwaltet werden. Die daraus resultierende Analyse darf nicht durch das System eingeschränkt werden und muss durchgehend transparent sein (*Information*) (Vgl. [ML13], S.51).

2.3.2 Data-Mining

Neben der reinen Analyse von Daten, um einen momentanen Status Quo einsehen zu können, interessieren sich Unternehmen außerdem für die Möglichkeit, Prognosen zu erstellen und Vorräusplanungen treffen zu können. Hierfür bietet das Prinzip des Data-Minings eine Grundlage. Es ermöglicht innerhalb großer, mehrdimensionaler Datenbestände Muster zu erkennen, die bei Vorhersagen (zum Beispiel durch *wenn-dann*-Beziehungen) helfen sollen.

Aus dem Prozess des Data-Minings soll Wissen extrahiert werden, das bisher noch nicht aus den vorhandenen Daten zu lesen war. Data-Mining ist somit ein Teilschritt des Knowledge Discovery in Databases (KDD). Das Data-Mining bedient sich verschiedener Verfahren, die im folgenden kurz erläutert werden sollen. In der Regel basieren sie aber alle auf dem erkennen von Mustern und dem Lernen durch Lerndaten (Vgl. [ML13], S.76).

Innerhalb des *Clustering* wird anhand vorhandener Daten versucht, gemeinsame Gruppen zu erstellen. Daten, die in gewissen Abstand zueinander liegen, sollen zusammengefasst werden und so einer Gruppe zugeordnet werden können. Das *Clustering*-Verfahren iteriert hierfür Gruppierungen über einen Datenbestand und sortiert sie dann nach Auswahlverfahren entsprechend ein (Vgl. [ML13], S.83).

Die Assoziationsanalyse befasst sich mit der Analyse von *wenn-dann(-wahrscheinlich)-auch*-Beziehung. Hierbei wird untersucht, ob, wenn ein oder mehrere Aspekte zutreffen, auf das Zutreffen eines weiteren Falls geschlossen werden kann. Diese Art der Analyse wird oft in Verbindung mit der Warenkorb-Analyse bei Online-Händlern gebracht, da sie ermöglicht einzusehen, was ein Kunde eventuell noch kaufen möchte, wenn er bereits schon einige Artikel im Warenkorb hinterlegt hat (Vgl. [ML13], S.89).

Das Verfahren der Assoziationsanalyse beruht dabei auf der Suche nach starken Regeln. Hierfür wird heutzutage in der sogenannte *Frequent-Pattern-Algorithmus* verwendet, der über Iterationen versucht, die Häufigkeit auftretender Muster zu finden, um so mögliche gültige neue Regeln ableiten zu können (Vgl. [As08]).

Des Weiteren setzt man beim Data-Mining auch auf die Klassifizierung von Daten. Hierbei wird versucht die Daten in bestimmte Klassen aufzuteilen, um für diese Klasse eine Aussage treffen zu können. Die Klassifikation verläuft in der Regel über Entscheidungsbäume, deren Pfad die Eigenschaften einer bestimmten Klasse definieren und dessen Blatt schließlich die Klasse definiert (Vgl. [Ü08]).

Ein Beispiel bildet dabei Abbildung 5. Hier wird vereinfacht untersucht, ob eine Person einen Kredit erhalten würde oder nicht. Dafür werden bestimmte Attribute festgelegt, die für die Entscheidung relevant sind. In diesem Beispiel gehe man davon aus, dass es für den Banker wichtig sei, Alter, Beruf und Einkommen zu wissen.

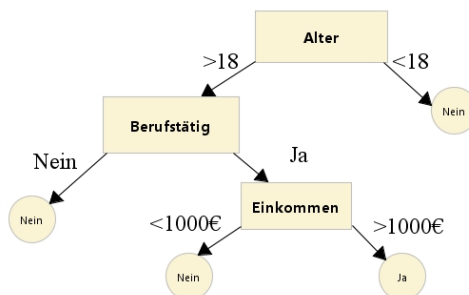


Abb. 5: Entscheidungsbaum zur Kreditwürdigkeit

Anhand der Abfrage von Daten, kann nun die Kreditwürdigkeit über diesen Entscheidungsbaum entschieden werden. Hierbei kann es noch zu Fehlern kommen. So kann eine Person zwar als Kreditwürdig eingestuft werden, den Kredit dennoch nicht zurückzahlen (falsch positive Klassifizierung). Ebenso könnte ein Kunde anderweitig in der Lage sein, ohne Beruf den Kredit zurückzubezahlen (falsch negative Klassifizierung).

2.4 Fazit

Abschließend wird deutlich, dass das Business Intelligence Konzept durch seine effiziente Verknüpfung von Datenintegration, Data Warehouse, OLAP- und Data-Mining-Analysen eine wesentlich größere Unterstützungsmöglichkeit bietet, als statische, dispositive OLTP-Ansätze. Durch zukünftige Entwicklung insbesondere im Bereich der Hardwaremöglichkeiten und neuen Erkenntnissen des maschinellen Lernens, birgt Business Intelligence noch ein sehr großes Potential insbesondere im Bereich der sogenannten Industrie 4.0. Gewinn wird dabei insbesondere durch den Einfluss des immer weiter wachsenden Social-Webs gemacht, in dessen Datenmengen wertvolle Kundeninformationen gewonnen werden können, sodass auch Vorhersagen zu zukünftigen Trends und Wünschen immer präziser werden. Die bisherigen Entwicklungen zeigen einen immer essentiellen Einfluss der Business Intelligence, deren Zukunft durch Begriffe wie *Business Intelligence 2.0* und *Predictive Analytics* geprägt werden (Vgl.[Bu15]).

Literaturverzeichnis

- [As08] Association Rule Mining, <http://people.cs.pitt.edu/iyad/AR.pdf>.
- [BH05] Architektur von Data Warehouses und Business Intelligence Systemen, https://www.fbi.h-da.de/fileadmin/personal/b.humm/Publikationen/Humm_Wietek_-_Architektur_DW_-_Informatik-Spektrum_2005-01_.pdf.
- [BI07] BI-Systeme stehen und fallen mit der Datenintegration, <http://www.computerwoche.de/a/bi-systeme-stehen-und-fallen-mit-der-datenintegration,594464>.
- [Bu13] Business Intelligence, <http://www.io solutions.de/businessIntelligence.html>, Stand: 25.10.2015.
- [Bu15] Business Intelligence ist zurück in der Zukunft, <http://www.bigdata-insider.de/analytics/articles/498436/>.
- [Gr08] Grundlagenserie Business Intelligence, http://www.tecchannel.de/server/sql/1746250/bi_datenmanagement_teil_1_datenaufbereitung_durch_den_etl_prozess/.
- [In15] Institut für Business Intelligence, <http://www.i-bi.de/home/index.html>, Stand: 25.10.2015.
- [ML13] Müller, R.M.; Lenz, H.J.: Business Intelligence. eXamen.press. Springer Berlin Heidelberg, 2013.
- [Ü08] Überblick zu Data Mining-Verfahren, <http://dbs.uni-leipzig.de/file/dw-kap6.pdf>.

Business Intelligence in der Cloud: Software-as-a-Service

Marcel Kuchler¹

Abstract: Aufgrund der inzwischen immer weiter ausgebauten Infrastruktur und der Vernetzung von Unternehmen ist es unausweichlich, einen Zusammenhang zwischen Business Intelligence und Cloud-Lösungen zu finden und umzusetzen. In der folgenden Ausarbeitung werden die Themen „Cloud-BI“ und der Begriff Software-as-a-Service näher betrachtet und einige Beispielanwendungen vorgestellt. Dabei stellt der Bereich von Software-as-a-Service einen wesentlichen Bereich in dem Einsatz von Business Intelligence-Lösungen in kleinen bis mittleren Unternehmen (KMU) dar, da diese mit Hilfe der SaaS-Anwendungen kostengünstig und ohne großen Aufwand Business Intelligence-Lösungen im kleinen und mittleren Unternehmen integrieren können. SAP Business byDesign, Tableau oder auch Microsoft Dynamics sind unter anderem einige gängige Lösungen für KMU, welche auf dem SaaS-Prinzip basieren.

Keywords: SaaS, Software as a Service, BI in der Cloud, Business Intelligence, SAP Business byDesign, Tableau, Microsoft Dynamics

1 Einleitung

IT-Experten der heutigen Zeit kennzeichnen das erste Jahrhundert des dritten Jahrtausends einhellig mit den Schlagworten zunehmende Datenvolumina bzw. Big Data, orts- und zeitunabhängige (mobile) Kommunikation, virtuelle Arbeitsplätze und verkürzte „Planungs- und Entscheidungsdauern durch die zunehmende Globalisierung“ [CC01]. Für 2010 wurde eine globale Datenmenge in der Größe von einem Zetabyte vorausgesagt, dazu eine Wachstumsrate von 100 % pro Jahr [CC01]. Steigende Datenvolumina und erhöhte Datenraumdimensionalität, eine stärkere Verteilung von Daten in sogenannten Clouds, hohe Erwartungen von Endanwendern und zunehmende Komplexität der Datenverarbeitung erfordern stete Anpassungen der IT-Systeme, um diese Entwicklung voranzutreiben und sich zudem entsprechend anzupassen.

Auf Unternehmen übertragen bedeuten diese Entwicklungen laut den Autoren Müller und Lenz „steigende Volumina von Bewegungs- und Bestandsdaten und gestiegene Anforderungen des Managements“ und „dessen Informationsbedürfnisse umfassend, zeitnah und ortsgebunden zu decken“ [CC01].

Eng mit diesen Entwicklungen verbunden stehen die Begriffe Cloud Computing und Business Intelligence. Als „Sammelbegriff für den IT-gestützten Zugriff auf Informationen, sowie die IT-gestützte Analyse und Aufbereitung dieser Informationen“

¹ Carl von Ossietzky Universität Oldenburg, Department für Informatik, Ammerländer Heerstraße 114-118, 26129 Oldenburg, marcel.kuchler@uni-oldenburg.de

[CC02] soll Business Intelligence (BI) als Prozess „aus dem im Unternehmen vorhandenen Wissen“ neues Wissen generieren. Das neue Wissen soll relevant und handlungsorientiert sein und damit Managemententscheidungen in der Unternehmenssteuerung unterstützen [CC02].

Cloud Computing „beinhaltet Technologien und Geschäftsmodelle um IT-Ressourcen dynamisch zur Verfügung zu stellen“ [CC02]. Es stellt über das Internet oder ein Intranet als dienstleistungsbasiertes Geschäftsmodell bedarfsorientiert und flexibel IT-Ressourcen (beispielsweise Server oder Anwendungen) zur Verfügung. [CC02]

Stellt man die beiden Begriffe in den Zusammenhang der aktuellen Entwicklungen, so scheint ihre Kombination als „Cloud-BI“ unausweichlich. Die folgenden Ausführungen werden aus dem Grund neben einer Erläuterung des Business-Intelligence-Begriffes und des Cloud-Computing auch die Kombination, deren Vor- und Nachteile sowie einige Geschäftsmodelle des Cloud-BI – gerade im Hinblick auf den Usus „Software-as-a-Service“ - darstellen.

2 Grundlagen

Software as a Service (SaaS) als ein Teilbereich des Cloud Computings kann auf mehreren Ebenen für den Einsatz von Business Intelligence-Lösungen in Frage kommen. Um diese Kombination der drei Dinge besser im Zusammenhang verstehen zu können, werden im Folgenden die Strukturen von Software-as-a-Service, die Grundlagen des Cloud Computings sowie allgemeine Grundlagen von Business Intelligence aufgegriffen und anschließend in einem Zusammenhang dargestellt, sodass Software-as-a-Service als eine Business Intelligence-Lösung in der Cloud genutzt werden kann.

Auch werden die unterschiedlichen Möglichkeiten der BI-Lösungen in der Cloud – gerade in Hinblick auf die Unterschiede zwischen SaaS, Platform as a Service (PaaS) sowie Infrastructure as a Service (IaaS) kurz angerissen.

2.1 Cloud Computing

Als zentrale Begriffe für die Entwicklung der Gesellschaft hin zu einer digitalisierten Dienstlandschaft hat sich das „Cloud Computing“ bereits stark etabliert. Dabei bezeichnet der Begriff Cloud Computing ein Konzept für die Bereitstellung von IT-Leistungen als Versorgungsgut. Ähnlich wie Wasser, Strom und Telekommunikationsdienste sollten Internetdienste jederzeit erreichbar sein und eine verlässliche Qualität haben, zum Beispiel in Bezug auf Latenz, Bandbreite, Volumen und Sicherheit. Die von der Cloud erbrachten Leistungen umfassen verschiedene Architekturebenen – von Softwarediensten über Plattformen für Entwicklung und

Betrieb bis hin zur Bereitstellung grundlegender IT-Infrastrukturdienste. Der Begriff Cloud ist dabei eine Metapher für das Internet, das grafisch häufig als eine ebensolche Wolke dargestellt wird. [CC03]

Aktuelle Erhebungen und Bestandsaufnahmen zeigen, dass immer mehr Unternehmen dem Thema Cloud Computing aufgeschlossen gegenüberstehen. [CC04]. Während einerseits die Erfahrungswerte der Cloud-Nutzer auffallend gut sind – 75 Prozent beurteilen ihre Erfahrungen als durchweg positiv –, geben andererseits Entwicklungen wie der Fall des amerikanischen Whistleblowers Edward Snowden, der die umfassenden Abhörpraktiken der amerikanischen NSA aufdeckte, den Skeptikern weitere Argumente gegen die Nutzung von Cloud- oder sogar allen internetbasierten Diensten an die Hand, da die zugrunde liegende Kommunikationsinfrastruktur ebenfalls systematisch kompromittiert wurde. [CC04]

Der Großteil der Unternehmen bevorzugt unverändert private, das heißt für eine Organisation exklusiv zur Verfügung gestellte Dienste gegenüber solchen Diensten, die für die breite Öffentlichkeit verfügbare Cloud-Lösungen bieten. [CC04] Insbesondere große Unternehmen setzen verstärkt auf private Clouds, die vorwiegend in Eigenregie betrieben werden, um vor allem die interne IT neu auszurichten.

2.2 Software-as-a-Service (SaaS)

Software-as-a-Service, kurz SaaS, ist ein Teilbereich des Cloud Computings (siehe 2.1). Dabei basiert der Grundsatz der Technologie auf dem Modell, dass die Software und die IT-Infrastruktur bei einem externen IT-Dienstleister zur Verfügung gestellt und über das Internet durch den jeweiligen Anwender aufgerufen wird.

Im Jahre 2006 rückte SaaS in das Bewusstsein von IT-Entscheidern, gerade mit dem erfolgreichen Release der SaaS-CRM-Lösung von salesforce.com. SaaS ist dabei kein neues Konzept, in Zeiten des ersten Internetbooms (Web1.0) nannte man es “Application Service Providing” (ASP). Das Web1.0-ASP scheiterte jedoch, da die Anwendungen handwerklich schlecht gemacht wurden und weil die für die Nutzung notwendige Bandbreite und ständige Verfügbarkeit der Systeme und Internetzugänge bei den Unternehmen noch nicht vorhanden waren, um darüber das “Daily Business” abwickeln zu können. Einige Jahre später versuchte IBM mit einer “On-Demand“-Initiative diese Idee wieder aufzurufen, was mit dem Einsatz des Web2.0 und die Kooperation mehrerer Unternehmen auf dieser Ebene auch funktionierte.

Dabei hat das National Institute of Standards and Technology (NIST) fünf Charakteristika formuliert, welche Cloud-Dienste, wie eben der Dienst des SaaS, besitzen sollten: [SE01]

1. Resource Pooling: Die gemeinsame Nutzung physischer Ressourcen wird durch den Dienst ermöglicht. Kernaspekte hierbei sind Virtualisierung

(bezüglich der Hardware) sowie Mehrmandantenfähigkeit (bezüglich der Datenhaltung)

2. **Broad Network Access:** Ein allgemeiner Netzwerkzugriff auf den betrachteten Dienst ist möglich. Dies wird vor allem durch die Bereitstellung einer ausreichenden Bandbreite und die Einhaltung technischer Standards ermöglicht.
3. **Rapid Elasticity:** Der Dienst erlaubt eine unverzügliche Anpassung an den aktuellen Bedarf der aufrufenden Nutzer. Dies kann „von Hand“ durch den Nutzer oder automatisch durch den Dienst erfolgen. Diese Elastizitätseigenschaft darf jedoch nicht mit der Preiselastizität der Nachfrage verwechselt werden
4. **On-Demand Self-Service:** Der Dienst kann von den Anwendern nach Bedarf ohne einen fest vereinbarten Abnahmezeitpunkt selbst aufgerufen werden. Dies wird oftmals durch eine Automatisierung auf der Anbieterseite ermöglicht, sodass keine arbeitsintensive Konfiguration des Dienstes notwendig ist.
5. **Measured Service:** Es ist eine individuelle, nachvollziehbare Erfassung der Servicenutzung möglich. Dies ist vor allem mit dem Erlösmodell Pay-per-Use verbunden.

Zudem gibt es aktuell folgende Cloud-Dienste, welche aktuell zur Verfügung gestellt werden:

- **Infrastructure-as-a-Service (IaaS):** Virtuelle Hardware oder Infrastruktur, zum Beispiel, Speicherplatz, Rechenleistung oder Netzwerkbandbreite
- **Platform-as-a-Service (PaaS):** Programmierframeworks, Bibliotheken und Werkzeuge, um Anwendungen unter eigener Kontrolle auf Cloud-Infrastrukturen bereitstellen zu können
- **Software-as-a-Service (SaaS):** Vollständige Anwendungen, die auf Cloud-Infrastrukturen betrieben und beispielsweise über einen Webbrowser aufrufbar sind.
- **Mashup-as-a-Service (MaaS):** Verknüpfung einzelner Software-Komponenten zu einem aggregierten Cloud-Dienst
- **Business-Process-as-a-Service (BPaaS):** Konkrete Geschäftsanwendungen (z.B. CRM) als Verknüpfung einzelner Software-Komponenten

In diesem Beispiel wird nun intensiver auf den dritten Punkt eingegangen: Software-as-a-Service.

Für die Anwendung von Software-as-a-Service in der Cloud wird ein internetfähiger Computer sowie eine Internetverbindung an den externen IT-Dienstleister benötigt. Die komplette Hard- und Software, auf denen beispielsweise Business Intelligence-Lösungen installiert sind und zur Verfügung gestellt werden, werden extern gehostet und entsprechend von extern aufgerufen. Der Zugriff auf die Software wird in der Regel über den Webbrowser realisiert. Der Serviceanbieter übernimmt somit die gesamte Wartung und Pflege des Systems und dem Anwender bleiben somit hohe Anschaffungskosten durch den entsprechenden Hardwareeinsatz erspart. Ein weiterer Aspekt, der für das SaaS-Konzept spricht, ist die Skalierbarkeit von SaaS-Lösungen und die damit einhergehende Flexibilität.

Der Cloud-Provider sollte jederzeit in der Lage sein, flexibel auf neue Anforderungen des Kunden reagieren zu können – unabhängig davon, ob höhere Leistungen, größere Kapazitäten, neue Anwendungen oder ein 24-Stunden-Support in Anspruch genommen werden will. Ein immer wieder angesprochener und inzwischen sehr großer Aspekt ist die Sicherheit: SaaS-Lösungen sorgen entsprechend für Entlastung auf Nutzer-Seite, da die Sicherheit auf Seiten des Providers gegeben sein muss. Gleichzeitig ist der SaaS-Nutzer allerdings auch deutlich stärker von den Leistungen des Software-Anbieters und vor allem einer funktionierenden Internet-Anbindung abhängig.

Kurz zusammengefasst kann ein pro und contra für den Einsatz von SaaS-Lösungen innerhalb von Unternehmen aufgestellt werden: [SE02]

Vorteile von SaaS	Nachteile von SaaS
Kostentransparenz und –flexibilität	Abhängigkeit vom Anbieter
Keine zusätzlichen Investitionen	Daten werden nach außen verlagert
Fokus auf das eigene Kerngeschäft	Rechtliche Fragen
Zusätzliche Dienstleistungen	Untransparenter Markt

2.3 BI-Lösungen in der Cloud

In der Literatur gibt es unzählige Ansätze zur Definition des Begriffes Business Intelligence: Im Jahr 1989 wurde er ursprünglich von Howard Dresner, einem Analysten der Gartner Group, geprägt [BI01], anschließend folgten diverse weitere Definitionen.

Gluchowski und Kemper aus dem Jahr 2006 veröffentlichten die allgemeine Definition, die auch von einschlägigen Beratungshäusern verwendet wird:

„Unter Business Intelligence wird ein integriertes IT-Gesamtkonzept verstanden, das für die unterschiedlichen Ausprägungen der Anforderungen an geeignete Systeme zur Entscheidungsunterstützung tragfähige und miteinander verknüpfte Lösungen anbietet.“ [BI01]

Business Intelligence wird demnach zur Entscheidungsfindung in Unternehmen herangezogen. Müller und Lenz verstehen es als eine Verbindung interner und externer Daten in einem Data Warehouse mit Abfragemöglichkeiten für „Entscheidungs-, Planungs- und Controllingzwecke“ [CC01] in Verbindung mit explorativer Datenanalyse. Der Business Intelligence-Einsatz wird hierfür in der Praxis an notwendige Erfordernisse angepasst, da an ihn gestellte Anforderungen je nach Unternehmen variieren. [BI01]

Heutzutage ist die notwendige Infrastruktur inzwischen soweit entwickelt, dass BI-Lösungen in der Cloud ohne Probleme zur Verfügung gestellt werden können.

3 Anwendungen

In diesem Teil der Ausarbeitung werden einige Beispielanwendungen aus dem Bereich des Cloud-BI als Software-as-a-Service-Lösung vorgestellt und beschrieben.

Zunächst wird dabei näher auf den Bereich SaaS bei der SAP Deutschland AG eingegangen, die mit ihrem System „SAP Business byDesign“ SaaS realisiert hat. Aber auch Tableau als eine reine BI-Lösung wird – neben der SaaS-Lösung von Microsoft, dem Microsoft Dynamics Online - kurz vorgestellt.

3.1 SAP Business byDesign

SAP Business byDesign ist ein Angebot der SAP AG, welches sich vor allem an kleine und mittlere Unternehmen (KMU) richtet, welche zwischen 100 bis 500 Mitarbeiter besitzen. Die Anwendungsfunktionalitäten des Systems sind entsprechend durchdacht und modularisiert, sodass die vielen verschiedenen Geschäftsprozesse eines Unternehmens – gerade in der Wachstumsphase – gut durchdacht und entsprechend abgebildet werden können. [SAP01].

SAP Business byDesign wurde am 19. September 2007 in New York offiziell vorgestellt und steht Kunden in sieben Ländern (Deutschland, USA, Großbritannien, Frankreich, China, Indien und Südafrika) zur Verfügung. [SA03]

SAP beschreibt diese Lösung als „eine umfassende und anpassungsfähige On-Demand-Lösung – erschwinglich und einfach einzuführen“. [SA02]

SAP bietet viele verschiedene Lösungen, unter anderem auch verschiedenste Lösungen für KMU in der Cloud. Die Abbildung 1 zeigt die Aufteilung von Services der SAP AG, unterteilt in On-premise deployment sowie Cloud deployment, wobei das SAP Business byDesign in die Sparte des Cloud deployment eingeordnet wird:

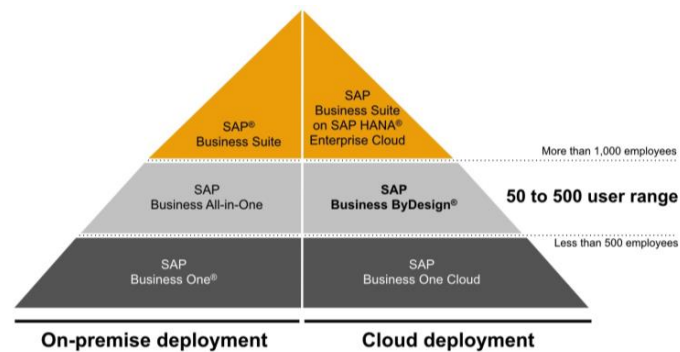


Abbildung 1 | Cloud & On-premise deployment von SAP für KMU [SA02]

SAP Business byDesign ist in einem SAP-eigenen Rechenzentrum installiert, vor allem, um in Bezug auf Datensicherheit das Risiko zu reduzieren. Das Softwareprodukt selbst wird von SAP entwickelt, die Konfiguration und Anpassung der Software – das Customizing – an die Kundenbedürfnisse, inklusive der Erstbefüllung der Datenbanken und der Projektberatung werden in der Regel von SAP-Partnern durchgeführt. Zudem kann gewählt werden, ob eine Public Edition oder eine Private Edition genutzt werden soll. Die Public Edition beinhaltet eine gemeinsam mit anderen Unternehmen genutzte Hardware, die Private Edition eine eigene Hardware, die ausschließlich vom jeweiligen Unternehmen genutzt wird.

SAP Business byDesign stellt dabei eine Lösung zur Verfügung, die die Belangen eines KMU in Gänze abbilden lässt, gerade was die Analyse, also den Bereich des Business Intelligence, angeht: Unterstützung der Geschäftsleitung, Unternehmensziele zu erreichen; Unterstützung des Finanzmanagements, das Finanzwesen zu optimieren; Unterstützung des Kundenbeziehungsmanagements, sich auf die Kundenwünsche zu konzentrieren; Unterstützung des Human Resource Managements, den optimalen Personaleinsatz zu berechnen; Unterstützung des Supply Chain Managements, das Supply Chain bestmöglich in die betrieblichen Abläufe zu integrieren; Unterstützung des Projektmanagements oder die Integration des Lieferantenmanagements, um so die Einkaufsprozesse optimal zu unterstützen.

Ein weiterer wichtiger Aspekt in Hinblick auf die Nutzung eines solchen Systems in der Cloud ist die, dass keinerlei Hardware-Anschaffungen getätigt werden müssen, worauf das System serverseitig aufgerufen werden kann. Lediglich Arbeitsplatzrechner müssen zur Verfügung gestellt werden – und die entsprechenden Lizenzen der SAP AG: Der Kunde bezahlt dabei lediglich eine Pauschalgebühr von 133,- EUR pro User und Monat.

Die Mindestabnahme liegt bei 25 Usern. Dieses Einstiegspaket kann der Kunde jederzeit um weitere Pakete von 5 oder 10 Usern erweitern. Die Wartungsgebühr beträgt monatlich 17% der Userkosten. [SA05]

3.2 Tableau Online – Die BI-Lösung in der Cloud

Neben dem SAP Business byDesign gibt es – selbstverständlich – viele weitere Anbieter auf dem globalen SaaS-Markt. Eine weitere Möglichkeit in dem Bereich der Business Intelligence-Cloud als Software-as-a-Service-Lösung stellt dabei die Software-Lösung „Tableau“ dar. Tableau ist eine SaaS-Lösung, die mit Hilfe eines WYSIWYG-Editors schnell zu einer optimierten Darstellung der erforderlichen Ergebnisse im BI-Bereich kommt. „Mit Tableau Online können die Benutzer ihre Analysen in wenigen Minuten durchführen und bereitstellen und Benutzer mit wenigen Klicks hinzufügen“, erklärt Chris Stolte, Chief Development Officer und Mitgründer von Tableau Software. Tableau ist dabei „Skalierbar, sicher und benötigt keine Infrastruktur“. [TA01]

Auch ist ein wesentlicher Aspekt der, dass „immer mehr Unternehmen in die Cloud kommen und ihre Daten mitbringen“, so Dan Jewett, Vice President of Product Management bei Tableau. [TA01]

In Tableau lassen sich wichtige Informationen mit Hilfe von Dashboards direkt visualisiert darstellen und veröffentlichen. Nutzer können Kollegen oder potentielle Kunden oder Partner einladen, diese Informationen zu betrachten. Die Daten sind damit zwar außerhalb der Unternehmensfirewall, aber dennoch legt Tableau großen Wert darauf, dass gerade diese Funktionen gesichert sind. Tableau wird online über einen Webbrowser aufgerufen und ist entsprechend auch ohne Probleme an jedem beliebigen Ort nutzbar – Voraussetzung hierfür ist natürlich ein Internetzugang.

Über den zentralen Datenspeicher, dem Data-Server, welcher von Tableau zentral betrieben wird, lassen sich Daten und Metadaten zentral speichern und verwalten. Auch eine Skalierung des Systems ist – auch im Nachhinein – ohne Probleme jederzeit möglich. Anders als beim SAP Business byDesign ist bei Tableau keine Mindestnutzeranzahl notwendig. Wächst ein Unternehmen, so kann jederzeit auf eine höhere Server-Version migriert oder entsprechend weitere User eingepflegt werden. [TA02]

Neben dem Live-Zugriff auf Cloud-Lösungen von Partnern wie Salesforce.com, Google Analytics, Google BigQuery oder Amazon RedShift bietet Tableau auch online weitere Schnittstellen, um schnell an die Auswertung von Unternehmensdaten zu gelangen. Über einen Cloud-Integrations-Service lassen sich hierbei viele Quellen, die zur Auswertung von Geschäftsfällen benötigt werden, in das BI-Tool integrieren.

3.3 Microsoft Dynamics

Microsoft Dynamics ist ein weiteres Beispiel für eine SaaS-Unternehmenslösung, welche aus dem Hause Microsoft kommt, und dafür sorgt, dass eine professionelle Ansprache und Betreuung der Kunden – gerade in kleinen bis mittleren Unternehmen (KMU) reibungslos abläuft. Von Lösungen für das Customer Engagement, gestützt durch Customer Relationship Management-Software, bis zu Enterprise Resource Planning, und von Supply Chain Management bis zu Business Intelligence und Reporting – Microsoft Dynamics unterstützt alle diese oben genannten Punkte. Mit dem MS Dynamics werden Unternehmensanwendungen, Produktivitätswerkzeuge, Daten, Dokumente und Endgeräte miteinander vereint und über die Cloud verbunden. [MS01]

Microsoft Dynamics Online ist eine On-Demand-Lösung mit vollem Funktionsumfang, die zu einem Festpreis pro Monat pro Benutzer genutzt werden kann. Die Software läuft als Service (SaaS) in Datenzentren mit Geo-Replikation (Speicherung der Daten in geografisch voneinander getrennten Rechenzentren, um einen Verlust von Daten zu verhindern), womit eine Verfügbarkeit von 99,9% im Jahr seitens Microsoft garantiert wird. [MS02]

4 Fazit

Software-as-a-Service-Anbieter behaupten, dass das Konzept „SaaS“ sich für alle Unternehmensgrößen eignen würde, doch die meisten Vorteile können sich KMU (kleine und mittlere Unternehmen) erzielen. Diese Unternehmen haben meistens keine IT-Abteilungen oder spezielle IT-Mitarbeiter und können folglich davon profitieren, dass der komplette Betrieb der Software vom SaaS-Anbieter geleistet wird. Weitere Vorteile für KMU bestehen darin, dass in den SaaS-Rechenzentren die Software nach den neuesten Sicherheitsstandards abgesichert ist. In der Regel sind die Daten damit sicherer, als wenn sie auf privaten Servern in der Firma gelagert wären, da viele kleinere Unternehmen ihre IT-Sicherheit stark vernachlässigen oder schlichtweg keine finanziellen Mittel dafür aufwenden können. Auch ein Kostenvorteil hat der SaaS-Einsatz für KMU: Das Unternehmen spart Kosten für eigene Administratoren und IT-Fachkräfte und benötigt keine besonders hohen Anfangsinvestitionen, um Softwarelizenzen und die entsprechend geeignete Hardware anzuschaffen.

Abschließend lässt sich sagen, dass das SaaS-Konzept sehr viel Potenzial mit sich bringt, gerade was die Software-Landschaft im Bereich der KMU betrifft, da sich diese zum größten Teil kostengünstiger und einfacher realisieren lassen, als große, hausinterne, auf privaten Serverlandschaften gehostete, ERP-Systeme, mit denen BI-Abfragen durchgeführt werden können. Ein weiterer Vorteil bei der Nutzung von SaaS-Lösungen ist der Bereich des „BYOD“ (Bring-your-own-Device), da durch die Cloud-Anwendungen die Daten in der Regel über einen Webbrowser aufgerufen und die dahinterstehenden Datenbanken angesprochen werden können und es somit kein Problem ist, diese auf den unterschiedlichsten Devices darzustellen.

Literaturverzeichnis

- [CC01] Müller, R.; Lenz, H.-J. (2013): Business Intelligence. Springer-Verlag Berlin Heidelberg.
- [CC02] Brich, S.; Hasenbalg, C. (2013): Kompakt-Lexikon Wirtschaftsinformatik: 1.500 Begriffe nachschlagen, verstehen, anwenden, Wiesbaden.
- [CC03] Krishnan, S.: Programming Windows Azure, O'Reilly Verlag Sebastopol 2010
- [CC04] Wallraff, B., Weber, M.: Cloud-Monitor 2013. Cloud-Computing in Deutschland – Status quo und Perspektiven, BITKOM 2013.
- [SE01] Vossen, G., Haselmann, T. Hoeren, T.: Cloud-Computing für Unternehmen: Technischer, wirtschaftliche, rechtliche und organisatorische Aspekte. Heidelberg: dpunkt.Verlag GmbH 2012
- [SE02] Marktüberblick über SaaS-Lösungen: <http://t3n.de/magazin/marktueberblick-saas-loesungen-business-applikationen-224066/2/>; Letzter Zugriff: 02.11.2015
- [BI01] Röger, M. (2010): Konzeption und Realisierung eines Data Warehouses zur Analyse chirurgischer Workflows, Hamburg.
- [SA01] Leonhardt, Despina (2008): Software as a Service – ein innovatives Geschäftsmodell, Norderstedt
- [SA02] Cloud & On-premise deployment von SAP für KMU: <http://diginomica.com/wp-content/uploads/2014/10/jpg>; Letzter Zugriff: 03.11.2015
- [SA03] SAP byDesign: Die umfassende On-Demand-Lösung: <http://www.3-as.de/wp-content/uploads/2011/08/SAPByDDieUmfassendeOnDemandSoftwareLoesung.pdf>; Letzter Zugriff: 06.11.2015
- [SA04] Erläuterung: SAP Business byDesign: https://de.wikipedia.org/wiki/SAP#Business_ByDesign; Letzter Zugriff: 02.11.2015
- [SA05] SAP Business by Design: <http://go.sap.com/germany/product/enterprise-management/business-bydesign.html>; Letzter Zugriff: 06.11.2015
- [TA01] Tableau als SaaS-Lösung: <http://www.silicon.de/41586931/tableau-server-jetzt-auch-als-saas-version/>; Letzter Zugriff: 07.11.2015
- [TA02] Analysen mit Tableau in der Cloud: <http://www.prnewswire.com/news-releases/analysen-in-der-cloud-sind-mit-tableau-online-90-schneller-als-je-zuvor-504070321.html>; Letzter Zugriff: 07.11.2015
- [MS01] Microsoft Dynamics als SaaS-Lösung: <http://www.microsoft.com/de-de/dynamics/kleine-und-mittelstaendische-unternehmen.aspx>; Letzter Zugriff: 07.11.2015
- [MS02] Erläuterung: Microsoft Dynamics als CRM: https://de.wikipedia.org/wiki/Microsoft_Dynamics_CRM; Letzter Zugriff: 07.11.2015

Business Intelligence in der Cloud: State-of-the-art PaaS

Abstract: Da die Menge produzierter Daten heute durch viele Unternehmensprozesse immer größer wird, ist auch der Bedarf an Business Intelligence-Lösungen größer geworden. Anbieter solcher Lösungen stellen diese häufig in einer Cloud bereit. Ziel dieser Seminararbeit ist es deshalb, den aktuellen Stand von Business Intelligence auf Basis einer Plattform-as-a-Service zu untersuchen. Um die Hintergründe dieser Technologie besser verstehen zu können, wird zunächst in einem Kapitel kurz beschrieben, was Cloud Computing und Business Intelligence ist. Weiter wird erklärt, was generell unter einer Plattform in der IT zu verstehen ist. Anschließend werden unterschiedliche PaaS-Anbieter vorgestellt, die unterschiedliche Lösungen anbieten. Außerdem werden Beispiele gegeben, die ein besseres Verständnis über diese Technologie geben sollen und zeigen, wie weit sie schon entwickelt ist.

Keywords: Paas, Platform-as-a-Service, BI in der Cloud, Business Intelligence, SAP HANA Cloud Platform, Google App Engine, Windows Azure

1 Einleitung

Die steigende Menge an Daten die weltweit produziert werden, bedarf komplexer Systeme, die helfen, der Daten herr zu werden. Bei diesen Systemen handelt es sich um Business Intelligence (BI) Lösungen, welche Daten aufbereiten, auswerten und die gesammelten Ergebnisse auf vorgegebenen Reports ausgeben. Mit Hilfe dieses Reports ist es dann möglich, Entscheidungen für die Zukunft zu treffen. Heutzutage können solche Systeme auch über eine Cloud bezogen werden.

Diese Seminararbeit hat zum Ziel, den aktuellen Stand dieser Technik zu untersuchen und Beispiele für ihre Anwendung zu finden. Dazu werden in Kapitel 2 Grundlagen gesammelt, die helfen sollen, sich mit dem Thema BI in der Cloud auf Basis von Plattform-as-a-Service zu befassen.

In Kapitel 3 wird die Plattform-as-a-Service Lösung einiger Anbieter beschrieben. Dabei wird untersucht, worin jeweilige Vorteile der unterschiedlichen Systeme liegen. Dies kann im Rahmen dieser Seminararbeit selbstverständlich nicht im vollen Umfang erfolgen. Weiter werden einige Beispiele gezeigt, an denen der Nutzen des Einsatzes der vorgestellten Lösungen verdeutlicht wird.

Abgeschlossen wird diese Seminararbeit von einem Fazit, in dem die gesammelten Ergebnisse noch einmal zusammengefasst werden.

2 Grundlagen

Da sich diese Seminararbeit mit dem Thema Business Intelligence in der Cloud in Bezug auf Platform-as-a-Service beschäftigt, werden in diesem Kapitel Grundbegriffe erklärt, die dem Leser beim Verstehen des Themas helfen sollen. Dabei wird nur in Ansätzen kurz auf Cloud Computing eingegangen, da sonst der Rahmen dieser Seminararbeit überschritten werden würde.

Im Anschluss wird beschrieben was unter einer Plattform allgemein verstanden wird. Weiter wird Platform-as-a-Service beschrieben, welches ein Dienst zum Bereitstellen einer Plattform in der Cloud ist. Zum Schluss wird noch untersucht, was eine BI Plattform ist und aus welchen Komponenten sie besteht.

2.1 Cloud Computing

Cloud Computing bietet Privatpersonen und Unternehmen die Möglichkeit, Rechenleistung oder Datenspeicher auszulagern. Das bedeutet, dass beispielsweise Daten nicht mehr auf eigener Hardware gespeichert werden, sondern auf der eines Cloud-Anbieters. Auch kann so Software auf externe Hardware ausgelagert werden. Dies hat den Vorteil, dass keine eigenen Ressourcen mehr gekauft werden müssen. So kann Geld für die Anschaffung und Wartung gespart werden. Auch lassen sich Cloud Lösungen skalieren, wenn beispielsweise ein Unternehmen wächst. [CN01 S.23]

Auf der anderen Seite ist aber zu beachten, dass mit dem Auslagern von Unternehmensdaten auf externe Speichermedien einige Risiken verbunden sind. Zum einen ist man von der Verfügbarkeit des Services eines anderen Unternehmens abhängig. Sollte es dort zu einem Datenverlust kommen, kann es sein, dass die Daten unwiderruflich verloren sind. Zum anderen hängt es auch von der Art der Cloud ab, wie gut geschützt die Daten sind. So werden in einer Public Cloud die Hardware Ressourcen mit anderen Unternehmern geteilt, ohne dass darauf selbst Einfluss genommen werden kann. [FH01] Um mehr Kontrolle über die Ressourcen einer Cloud haben zu können, kann auf eine Private Cloud zurückgegriffen werden. Hierbei können Vorgaben an einen Cloud-Anbieter gestellt werden. [TK01]

Die Möglichkeiten, die das Cloud Computing mitbringt, sind sehr vielseitig. Sie erlauben es Unternehmen, die nicht so viel Geld für ihre IT ausgeben können, trotzdem auf moderne IT Lösungen zurück zu greifen. Da dies aber nicht ohne Risiken möglich ist, muss von einem Unternehmen geprüft werden, ob dieses Risiko den Nutzen übersteigt.

2.2 Plattform

Eine Plattform wird in der IT als eine Basis gesehen, auf der Software betrieben werden und Komponenten angeschlossen werden können. Die Plattform bildet sich aus einer Hardware-, und Softwareplattform. Die Hardwareplattform wird durch den Aufbau der

Rechnerarchitektur gebildet. Die Softwareplattform wird durch das Betriebssystem und die Anwendungsprogramme gebildet, die auf der Plattform betrieben werden sollen. [IW01]

Eine Plattform kann aus unterschiedlichen Komponenten bestehen. Dies hängt unter anderem von der gewünschten Anwendung ab. Typischerweise wird eine Plattform durch die folgenden Bestandteile aufgebaut: [IW01]

- Die Rechnerarchitektur
- Das Betriebssystem
- Die Programmiersprache
- Schnittstellen

2.3 Platform-as-a-Service (PaaS)

Unter Platform-as-a-Service in dieser Seminararbeit abgekürzt mit PaaS versteht man Programmier- oder Laufzeitumgebungen, wie beispielsweise Entwicklungsumgebungen, die über eine Cloud bereitgestellt werden. [CC01 S.155] PaaS basiert meist auf einer Public Cloud. [CC01 S.38f]

Bei PaaS wird dem User eine Umgebungen bereitgestellt, in der beispielsweise große Webanwendungen erstellt werden können. Dabei wird von der Hardware und der Software alles bereitgestellt, was nötig ist, diese Anwendung zu erstellen und zu hosten. So muss keine teure und wartungsintensive Hardware bzw. Software angeschafft werden, um entwickeln zu können. [GO01] Weiter wird dieser Service je nach Anforderung abgerechnet, was es ermöglicht, Projekte kostengünstig durchzuführen. Dabei ist immer sichergestellt, dass Bibliotheken für die gewünschte Programmiersprache aktuell sind und die Verfügbarkeit der Entwicklungsumgebung gewährleistet ist. Nach Abschluss einer Entwicklung kann diese gleich über die Cloud getestet werden. [CC01 S.155]

Beim Einsatz von PaaS sind auch einige Nachteile zu beachten, die die Entwicklung beeinflussen können. Zum einen gibt es keinen einheitlichen Standard, der es ermöglicht, flexibel zwischen den am Markt vertretenen Anbietern zu wechseln. Jeder Anbieter stellt dem Kunden nur seine PaaS Lösung zur Verfügung, was den Wechsel zu einem späteren Zeitpunkt erschweren kann. Zum anderen werden die Rahmenbedingungen bezüglich Programmiersprache und Entwicklungsumgebung vom Anbieter vorgegeben. So ist der Kunde im Gegensatz zu einer lokalen Lösung auf die Vorgaben des Anbieters angewiesen. [CC01 S.155]

2.4 BI-Plattform in der Cloud

Business Intelligence hilft Unternehmen und Organisationen dabei, aus bestehenden Daten Trends für die Zukunft aufzudecken. Hierfür wird Hard- und Software benötigt, die in der Lage sind, in kürzester Zeit große Mengen an Daten zu analysieren. Da diese sehr teuer sein kann, profitieren meist nur große Unternehmen von den Vorteilen einer Business Intelligence Lösung. Um auch kleinen und mittelständischen Unternehmen die Möglichkeit zu geben, Business Intelligence zu nutzen, kann dieses über eine Cloud bezogen werden. So ist es möglich, Kosten für Hardware, Lizenzen und Wartung zu sparen. Den Vorteilen steht dabei meist die Angst vor einem unzureichenden Datenschutz der Unternehmensdaten gegenüber. [TT01]

Auch bei der Entwicklung von Business Intelligence Lösungen kann ein Unternehmen auf eine PaaS Cloud zurückgreifen, um Kosten zu sparen. In dieser Cloud kann dann die Entwicklung selber vorgenommen werden. Weiter können ausführliche Tests durchgeführt und so die Qualität der Business Intelligence Lösung sichergestellt werden. Für den produktiven Einsatz ist der PaaS-Ansatz aber nicht ausgelegt. Dies kommt unter anderem daher, dass in PaaS Clouds meist nur Standardschnittstellen verwendet werden. [TC01]

3 Anwendungen

In diesem Teil der Seminararbeit wird nun eine Übersicht über einige PaaS Lösungen gegeben, die sich mit Business Intelligence befassen. Dabei sollen auch Beispiele aus Wirtschaft und Forschung den Nutzen von PaaS hervorheben.

Zunächst wird die SAP HANA Cloud Plattform vorgestellt. Anschließend wird die Google APP Engine beschrieben. Zum Schluss wird ein kurzer Überblick über Microsofts Windows Azure Plattform gegeben.

3.1 SAP HANA Cloud Platform

Die SAP HANA Cloud Platform ist eine PaaS Lösung der Firma SAP. Mit dieser Plattform können Unternehmen eigene Cloud BI Lösungen entwickeln. Dabei wird die auch hier die für SAP HANA genutzte In-Memory-Technologie verwendet. Mit ihr können noch schnellere Auswertungen erstellt werden, da diese im Hauptspeicher durchgeführt werden. [SA01 S.2f]

Auf der SAP HANA Cloud Platform können Unternehmen Applikationen entwickeln, die sowohl über einen Webbrowser als auch über mobile Endgeräte wie Smartphones und Tablets genutzt werden können. Dies erlaubt es, Anwendern zu jederzeit auf wichtige Unternehmensinformationen zugreifen zu können. Die entwickelten Applikationen können dabei ganz auf die Wünsche und Vorgaben eines Unternehmens

angepasst werden. Dabei hat ein Unternehmen alle Vorteile der Cloud wie beispielsweise die Kostenreduktion. [SA01 S.3] Auch werden hier alle Schnittstellen mitgeliefert, die von einem Unternehmen benötigt werden, um nicht SAP Anwendungen nutzen zu können und so aussagekräftige Auswertungen zu erhalten. [SA02 S.4] Die untenstehende Abbildung 1 zeigt den schematischen Aufbau der SAP HANA Cloud Platform.

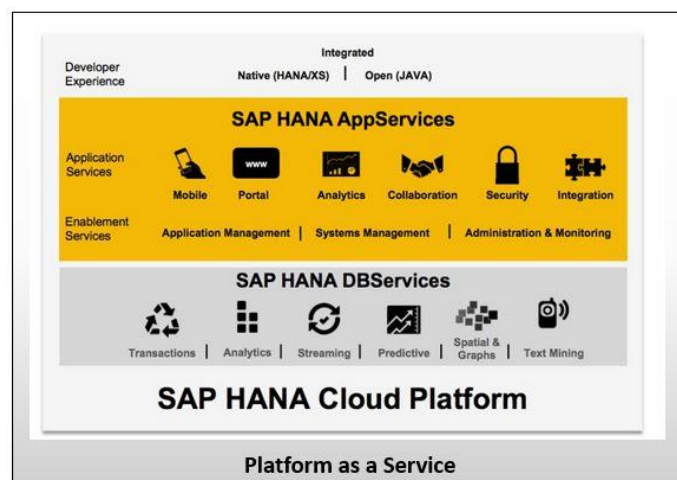


Abbildung 1 [AB01]

Viele Unternehmen und andere Organisationen haben bereits die SAP HANA Cloud Plattform im Einsatz und können so Business Intelligence aus der Cloud nutzen. Dazu gehören unter anderem Volkswagen und Shell, die zurzeit ein Projekt durchführen, welches das Autofahren in großen Städten vereinfachen soll. Dazu wurde mit der SAP HANA Cloud Plattform eine Lösung entwickelt, die bei der Parkplatzsuche hilft und über die auch die Bezahlung des Parkplatzes ermöglicht wird. Weiter kann über ein Smartphone die nächste Shell Tankstelle gefunden werden. Diese Tankstelle wird über den anstehenden Besuch benachrichtigt und übernimmt den Tankvorgang für den Kunden. Bezahlt wird im Anschluss ebenfalls über das Smartphone. In Zukunft könnte so mit dieser PaaS das Autofahren revolutioniert werden. [SA03]

Die SAP HANA Cloud Plattform bietet Unternehmen eine moderne Business Intelligence Lösung in der Cloud. Dadurch, dass sie nun auch über eine Cloud zu nutzen ist, wird sie besonders für Unternehmen interessant, die bisher durch hohe Anschaffungskosten abgeschreckt wurden. Das genannte Beispiel zeigt, dass mit dieser Technologie viele Probleme des Alltags gelöst oder vereinfacht werden können.

3.2 Google App Engine

Die Google App Engine ist eine PaaS Lösung, die von der Firma Google angeboten wird. Mit dieser Lösung können Entwickler Applikationen erstellen und über ihre Infrastruktur verbreiten. Dabei richtet sich die Google App Engine hauptsächlich an App-Entwickler, die für Google Dienste Applikationen erstellen. Apps lassen sich hierbei in vier Programmiersprachen entwickeln. [CC01 S.159f]

Der Vorteil bei der Google App-Engine liegt vor allem in dem Bezahlssystem. Dies ermöglicht eine Testphase in der Entwickler die App Engine gratis nutzen können. Erst wenn die Anforderungen an die Entwicklungsumgebung wachsen, muss für den Service bezahlt werden. [GO01]

Auch mit der Google App Engine lassen sich Business Intelligence Lösungen entwickeln. So gibt es am Markt einige Hersteller, die auf Basis der zur Verfügung stehenden Programmiersprachen Applikationen anbieten. Ein Beispiel hierfür ist das Unternehmen InetSoft, das eine Java basierte Reporting App anbietet. [IS01]

Mit der App Engine hat Google einen Rahmen geschaffen, in dem Entwickler für Googles eigene Systeme entwickeln können. Dies ermöglicht es, Apps zu programmieren ohne viel über benötigte Ressourcen nachzudenken. Sollte sich dieses Vorgehen bewähren, könnte dies in Zukunft der Weg sein, wie Software entwickelt wird. Das Einbinden von Business Intelligence Anwendungen ist dabei nicht so leicht wie bei anderen Anbietern. Mit Hilfe von Drittanbietern ist es aber möglich Reports zu erstellen. Dies zeigt, dass es auf jeder Plattform möglich sein kann, unabhängig davon, ob eine Plattform für den Einsatz von Business Intelligence Software vorgesehen ist, es auf jeder Plattform möglich sein kann. Es zeigt aber auch, dass dies nicht immer ohne weitere Software möglich ist, wodurch sich die Kosten für eine Plattform erhöhen können.

3.3 Windows Azure Platform

Windows Azure ist ein PaaS der darauf zielt, Entwicklern bei ihrer Arbeit möglichst viele Freiräume zu geben. Dazu lässt sich die Plattform je nach Wunsch konfigurieren. Durch diese Flexibilität ist Windows Azure für jede Unternehmensgröße geeignet. Wie auch bei anderen PaaS Anbietern wird die Cloud hier je nach Wunsch skaliert. Weiter bietet Microsoft mit dieser Plattform die Möglichkeit, selber zu entscheiden, ob Daten in der Cloud oder auf einem unternehmenseigenen Speicher abgelegt werden. Außerdem zeichnet sich Windows Azure durch einen hohen Datenschutz und eine hohe Verfügbarkeit, in vielen Bereichen, von mindestens 99,9 % aus. [MS01]

Für den Bereich Business Intelligence bietet Windows Azure einige Werkzeuge, die bei der Auswertung von Daten helfen. Dabei ist es für das Reporting unwichtig, ob die Daten in strukturierter oder unstrukturierter Form vorliegen. Die verwendeten Werkzeuge sind unter anderem:[MS01]

- **Machine Learning**

Machine Learning bietet Anwendern die Möglichkeit schnell und unkompliziert Datenflussdiagramme zu erstellen. Mit diesen können Vorhersagen über zukünftige Ereignisse erstellt werden. [MS02] So kann beispielsweise die Auslastung eines Fahrradverleihs unter Berücksichtigung von Faktoren wie Wetter, Jahreszeit und Wochentag erstellt werden.

- **Power BI**

Power BI führt Daten aus allen möglichen Quellen eines Unternehmens zu aussagekräftigen Reports zusammen. Dabei können Excel-Tabellen genauso genutzt werden wie eine SQL Datenbank. Die Reports können sowohl in Echtzeit genutzt werden als auch nach Bedarf neu angefragt werden. Dazu muss lediglich eine Frage eingegeben werden. Power BI sucht dann alle Informationen zusammen. [MS03] Beispielsweise könnte man fragen „In welcher Region wurde unser Produkt am meisten verkauft?“ Power BI zeigt dem Fragenden darauf eine Karte mit der gewünschten Antwort.

- **Stream Analytics**

Da heutzutage sehr viele Daten von Sensoren und anderen Geräten erzeugt werden, stellt Windows Azure mit Stream Analytics ein Werkzeug bereit, mit dem man in Echtzeit solche Ströme von Daten analysieren kann. Bei der Analyse können mehrere Streams miteinander verglichen werden. Es können aber auch Streams mit historischen Daten verglichen werden, um so Aussagen für die Zukunft zu erstellen. Aber auch Fehler im Stream können aufgedeckt und dadurch Warnungen generiert werden. [MS04] Mit diesem Werkzeug können beispielsweise Sensordaten von vernetzten Autos überwacht werden. Lässt sich aus einem Stream ein Fehler ableiten, etwa weil die Sollwerte eines Sensors nicht mehr erreicht werden, könnte der Fahrer gewarnt und an seine Werkstatt verwiesen werden.

Die untenstehende Abbildung 2 zeigt den schematischen Aufbau der Windows Azure Landschaft

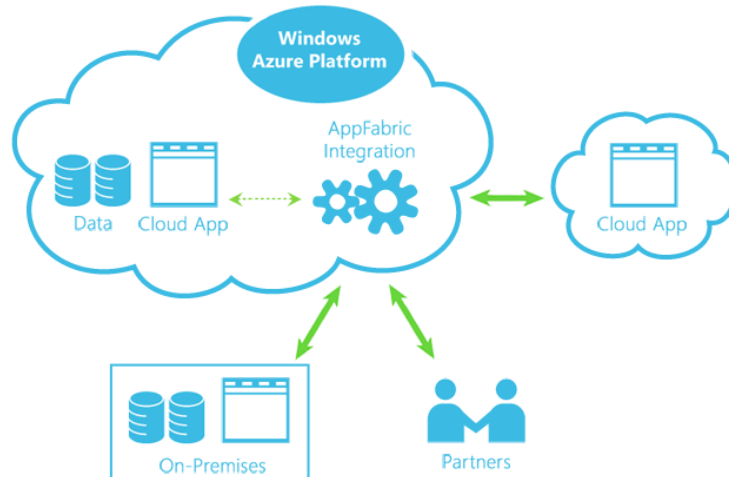


Abbildung 2 [AB02]

Mit Windows Azure hat Microsoft eine Plattform geschaffen, mit der jedes Unternehmen seine IT über eine Cloud abwickeln kann. Dabei sind besonders die Anpassungsmöglichkeiten sehr umfangreich, so dass Windows Azure sich an das Unternehmen anpasst und nicht umgekehrt. Dies ist wie anfangs beschrieben ein großer Kritikpunkt bei PaaS. Windows Azure zeigt hier möglicherweise eine Anforderung die jeder PaaS Anbieter bedienen muss, möchte er nicht von anderen Anbietern vom Markt verdrängt werden.

4 Fazit

Ziel dieser Seminararbeit war es, den aktuellen Stand der Business Intelligence Lösungen auf Grundlage von Platform-as-a-Service aufzuzeigen. Um dieses Ziel zu erreichen, wurden die Kapitel 2 und 3 erstellt. Dabei ist Kapitel 2 als Sammlung von Grundlagen zu verstehen, die einen Einstieg in das Thema BI PaaS nach dem aktuellen Stand ermöglicht.

Das dritte Kapitel zeigt mit einer Übersicht über die SAP HANA Cloud Platform, die Google App Engine und Windows Azure, was mit dem heutigen Stand der Technik möglich ist. Es wurde gezeigt, dass Cloud Computing ein Weg ist, kleine und mittelständische Unternehmen, die bisher nicht über moderne Business Intelligence Lösungen verfügen, in die Lage versetzen, die Technologie kostengünstig und ohne großen Aufwand im Unternehmen zu implementieren. Dabei wurde auch gezeigt, dass jeder Anbieter unterschiedliche Schwerpunkte mit seiner Lösung abdeckt. So setzt SAP eher darauf, ihren Kunden eine Plattform anzubieten, mit der diese auf das eigene

Unternehmen abgestimmte Reports erstellen und verbreiten können. Google hingegen gibt seinen Kunden eine Plattform zum Erstellen von Applikationen für die Google eigenen Systeme. Dabei wird dem Kunden aber alles bereitgestellt, was für die Entwicklung nötig ist. So muss keine eigene Entwicklungsumgebung geschaffen werden, in der Google Apps erstellt werden können. Microsoft versucht mit Windows Azure vor allem eine Lösung zu bieten, in der der Kunde seine Anforderungen an die Plattform so realisieren kann, wie er es möchte. Dabei soll zum Beispiel die Kontrolle über die Daten immer beim Kunden bleiben.

Abschließend lässt sich sagen, dass Business Intelligence auf Basis von Plattform-as-a-Service eine Technologie ist, die in der heutigen Zeit gebraucht wird. Dies kommt daher, dass zum einen auch kleinere Unternehmen den Nutzen von Business Intelligence entdeckt haben, zum anderen aber auch viel mehr Daten ausgewertet werden müssen, da heute sehr viele Prozesse von Sensoren in Echtzeit überwacht werden. Weiter ist die Möglichkeit, die Cloud einfach je nach Anforderung skalieren zu können und so Kosten zu sparen, ein Grund, PaaS zu nutzen. Es bleibt aber die Frage, ob der Nutzen die Risiken beispielsweise im Datenschutz rechtfertigt. Sicher ist, dass in Zukunft Konzepte erstellt werden müssen, die die Sicherheit der Unternehmensdaten gewährleisten. Sollte dies realisiert werden, wird es wohl noch mehr Nutzer geben.

Literaturverzeichnis

- [AB01] saphanatutorial: <http://saphanatutorial.com/wp-content/uploads/2015/01/SAP-HANA-Cloud-Platform-2.jpg>. Stand: 03.11.2015
- [AB02] Sys-Con: <http://res.sys-con.com/author/10294/diag-integration.gif>. Stand: 03.11.2015
- [CW00] Computerwoche.: PaaS Anbieter im Vergleich. <http://www.computerwoche.de/a/paas-anbieter-im-vergleich,3066351>, Stand: 19.10.2015
- [CC01] Tarplan, K.; Voigt, C.: Cloud Computing. Verlag mitp, 2011.
- [CN01] Köhler-Schute, C.: Cloud Computing: Neue Optionen für Unternehmen. KS-Energy-Verlag, Berlin 2011.
- [FH01] Fraunhofer: Was bedeutet Public, Private und Hybrid Cloud?. <http://www.cloud.fraunhofer.de/de/faq/publicprivatehybrid.html>. Stand: 23.10.2015
- [TK01] Telekom: Welche Cloud-Modelle gibt es? Public, Private und Hybrid. <https://www.telekom.com/medien/medienmappen/cloud-computing/135560>. Stand: 23.10.2015
- [GO01] Google: <https://support.google.com/a/answer/91077?hl=de>. Stand: 26.10.2015.
- [IW01] IT Wissen: Plattform. <http://www.itwissen.info/definition/lexikon/Plattform-plattform.html>. Stand: 26.10.2015
- [TT01] Techtag; <http://www.techtage.de/business/mittelstand-vorteile-durch-business->

- [intelligence-aus-der-cloud/](#). Stand: 28.10.2015
- [TC01] Tecchannel:
http://www.tecchannel.de/server/cloud_computing/2030180/cloud_computing_das_muessen_sie_wissen_saas_paas_iaas/index5.html. Stand: 28.10.2015
- [SA01] SAP: SAP HANA Cloud Platform.
http://www.sap.com/bin/sapcom/en_us/downloadasset.2014-09-sep-16-23.build-new-and-extend-existing-applications-in-the-cloud-pdf.bypassReg.html Stand: 28.10.2015
- [SA02] SAP: A CLEAR PATH to Innovation with Platform as a Service.
http://hcp.sap.com/content/dam/website/skywalker/en_us/PDFs/Elevate%20your%20Business%20with%20SAP%20HANA%20Cloud%20Platform.pdf. Stand: 28.10.2015
- [SA03] SAP: Connected Cars: SAP, Shell und Volkswagen wollen Vernetzung vorantreiben.
<http://news.sap.com/germany/2014/11/11/connected-cars-sap-shell-und-volkswagen-wollen-vernetzung-vorantreiben/>. Stand: 28.10.2015
- [GO02] Google: <https://cloud.google.com/appengine/>. Stand: 02.11.2015
- [IS01] InetSoft:
https://www.inetsoft.com/solutions/dashboard_reporting_for_google_apps_engine/.
Stand: 02.11.2015
- [MS01] Microsoft Azure: <https://azure.microsoft.com/de-de/overview/what-is-azure/>. Stand: 02.11.2015
- [MS02] Microsoft Azure machine-learning: <https://azure.microsoft.com/de-de/services/machine-learning/>. Stand: 02.11.2015
- [MS03] Microsoft Power BI: <https://powerbi.microsoft.com/de/>. Stand: 03.11.2015
- [MS04] Microsoft Azure Stream Analytics: <https://azure.microsoft.com/de-de/services/stream-analytics/>. Stand: 03.11.2015

Business Intelligence aus der Cloud: Infrastructure-as-a-Service

Nils Lutz¹

Abstract: Auf Grund der mittlerweile breiten Verfügbarkeit in Deutschland und den rechtlich besser werdenden Rahmenbedingungen bekommt die Cloud immer mehr Relevanz in IT-Projekten. Eine Optimierung der Kostenstrukturen und erhebliche Kosteneinsparungen für kleine und mittelständische Unternehmen sind durch die Cloud möglich. Durch die verschiedensten Anforderungen sind jedoch individual Lösungen nötig im KMU-Bereich. Konzerne können einfacher Standardlösungen implementieren und besitzen die nötigen IT-Ressourcen, um mit Eigenentwicklungen die Performance noch weiter zu steigern. Diese IT-Ressourcen fehlen im Mittelstand. IaaS, PaaS und SaaS sind die einzelnen Bestandteile einer Cloud-Umgebung. In dieser Seminararbeit wird hauptsächlich auf IaaS eingegangen. Die technische Umsetzung erfolgt mit altbekannter Technologie, die durch enorm niedrige Hardwarekosten heute in entsprechender Größe gebündelt eingesetzt werden kann, um eine Cloud-Umgebung zur Verfügung zu stellen. Eine Marktanalyse des Gartner Instituts und eine Studie von Crisp Research machen den Markt ein wenig transparenter. So lassen sich einige Anbieter in klare Rollen im Markt einteilen. Letztendlich lässt sich ein klares Zugpferd des gesamten Marktes erkennen - namentlich ist das Amazon. Amazon bietet für jeden Einsatzzweck ein Produkt oder eine Lösung aus mehreren ihrer Produkte an und kann somit beinahe jeden Kundenwunsch erfüllen. Auch in Punkto Datenschutz und Datenstandort ist Amazon grade für den deutschen Markt sehr gut aufgestellt.

Keywords: IaaS, Cloud, Business Intelligence, Big Data, Storage, Compute, Network.

1 Motivation

Cloud Computing ist laut einer Studie von Crisp Research endgültig in Deutschland angekommen. In der Studie wurden 700 deutsche IT-Entscheider zu den einzelnen Cloud Computing Themen befragt. Daraus hat sich ergeben, dass bei gut einem fünftel der Befragten das Thema Cloud Computing ein fester Bestandteil der IT-Agenda und im produktiven IT-Betrieb ist. So zeigt sich, dass 63 Prozent der deutschen Unternehmen aktuell damit beschäftigt sind den Einsatz von Infrastructure-as-a-Service zu Planen- bzw. zu Implementieren. Sie setzen Cloud Computing bereits in ersten Projekten ein und bewältigen damit ihre Workflows.

Die Studie von Crisp Research prognostiziert ein Investitionsvolumen von etwa 6,1 Milliarden Euro für Cloud Technologien. Immer mehr Unternehmen prüfen daher den Einsatz von Infrastructure-as-a-Service, also die Nutzung von Infrastrukturkomponenten wie Server, Speicher, Datenbanken und anderen Komponenten aus der Cloud.(vgl. [St15])

Aus der beschriebenen Situation leitet sich die Motivation für diese Seminararbeit ab. Es

¹ Carl von Ossietzky Universität, Informationssysteme, Ammerländer Heerstraße 114-118, 26129 Oldenburg, nils.lutz@uni-oldenburg.de

soll ein Überblick über den Bereich Business Intelligence aus der Cloud und hier im speziellen zum Thema Infrastructure-as-a-Service gegeben werden. „Insbesondere das stetig steigende Datenaufkommen erfordert eine zuverlässige und vor allem stabile Konnektivität, um auf die Daten und Informationen zu jeder Zeit Zugriff zu erhalten und verlässlich zu speichern. Noch wichtiger wird es, wenn geschäftskritische Prozesse und Applikationen auf eine Cloud-Infrastruktur ausgelagert werden. Der Zugriff muss jederzeit und performant - mit einer geringen Latenz - sichergestellt werden und kann für Unternehmen im Fehlerfall zu maßgeblichen finanziellen als auch Schäden am Image führen. Die Qualität eines Cloud Service hängt somit maßgeblich von seiner Konnektivität und der Performance im Backend ab. Hierbei ist ein entscheidendes und wichtiges Merkmal die Konnektivität des Rechenzentrums, um den Kunden und deren Kunden einen stabilen und zuverlässigen Zugriff auf Cloud-Services zu jeder Zeit zu gewährleisten. Rechenzentren sind die Logistikzentren der Zukunft und erleben als logistisches Datenvehikel ihre Blütezeit.“ [Di15]

2 Begriffsdefinition

Um in den folgenden Kapiteln keine Verwirrung zu stiften wird eine kurze Erklärung der einzelnen Begriffe sowie eine Abgrenzung untereinander gegeben. Als schematische Darstellung des behandelten Themenbereichs dient Abbildung 1. Zu sehen ist der Begriffskomplex Cloud Computing mit seinen Berührungspunkten zur Außenwelt und einigen beispielhaften Applikation und Komponenten der jeweiligen Schichten. Für die in der Cloud bereitgestellten Services hat sich eine Einteilung in drei Schichten beziehungsweise Gruppen von Services etabliert. Bei dieser nutzen die höheren, abstrakten Schichten die Dienste der tieferen, konkreteren Schichten für ihre eigenen Dienstleistungen. Auf diese Art können auch die unterschiedlichen Typen von Clouds klassifiziert werden. (vgl. [Wa12])

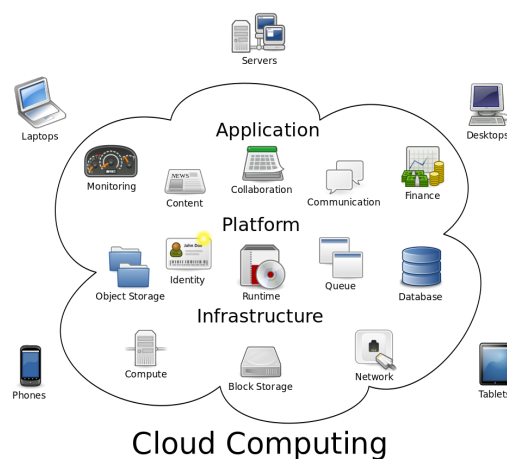


Abb. 1: Cloud Computing Übersicht mit typischen Anwendungen[CI15]

IaaS Cloud basiert auf virtuellem Zugang „Bei IaaS, in voller Form ausgeschrieben Infrastructure as a Service, handelt es sich um die grundsätzlichste Schicht der insgesamt drei Cloud-Optionen. Man geht dabei davon aus, dass hier der Anbieter einen virtuellen Zugang zu Hardwarekomponenten für den Nutzer bereit stellt.

Dies kann sowohl ein tatsächlicher Speicherplatz, aber auch eine Netzwerkstruktur oder entsprechende Prozessoren sein. Alles weitere wie die Installation sowie der Betrieb und die Wartung der Software muss der Nutzer selbst erledigen und trägt auch dafür die komplette Verantwortung. Simpel formuliert kann man die IaaS Cloud damit vergleichen, dass sich ein Bekannter einen Computer in seinem Zuhause aufstellt, auf den man über das Netz zugreifen kann. Was darauf installiert wird und ob das funktioniert ist alleinig dem Nutzer überlassen.“[C114] Somit dient IaaS als Fundament für das Cloud Computing und besteht aus den Grundkomponenten Compute, Storage und Network.

PaaS bietet grundlegende Software „Im Unterschied zur IaaS Cloud bietet die PaaS, also die Platform as a Service, durchaus grundlegende Software. Damit kann der Nutzer dann eigenständig die von ihm benötigte oder gewünschte [sic!] Softwareteile entwickeln, integrieren oder einfach nur testen. Der gravierendste Unterschied besteht also darin, dass nicht mehr die grundlegende Struktur zu verantworten ist, sondern lediglich die eigene Software.

Im Klartext bedeutet dies, dass der Bekannte nicht mehr nur den Computer aufstellt, sondern auch bereits ein Betriebssystem installiert. Er sorgt auch für die Wartung dieses Programms und die entsprechende Pflege. Der Nutzer selbst ist nur mehr für die Anwendungssoftware zuständig, die er tatsächlich nutzt oder benötigt.“[C114]

SaaS bietet Verantwortung für den Betreiber „Bei der SaaS, der Software as a Service, geht es darum, dass der Nutzer den größten Teil der Verantwortung an den Betreiber abgibt. Dieser ist damit nicht mehr nur für die Hardware und die grundlegende Softwareumgebung zuständig, sondern bietet dem Nutzer auch einen Zugang zur Anwendungssoftware. Dieser kann über das Netzwerk zugreifen und muss nur mehr seine einzelnen Daten verwalten.

Für den im Wohnzimmer installierten Computer bedeutet dies, dass nicht nur ein Betriebssystem, etwa OS x am Rechner ist, sondern auch noch Programme wie Word. Der Nutzer greift nicht mehr auf das Betriebssystem zu, sondern über das Netz direkt auf das gewünschte Programm.“[C114]

3 Realisierung

Nachdem die Begriffe klar von einander abgrenzt sind soll in diesem Kapitel die technische Umsetzung von IaaS erläutert werden. Wie in Abbildung 2 zu sehen ist platziert sich IaaS als unterste Ebene im Cloud-Stack. Auf dieser Ebene werden Ressourcen bereitgestellt auf denen die höheren Ebenen ihre Lösungen realisieren. Zu den drei Hauptressourcen gehören bei IaaS Compute, Storage und Network. Sprich die reine Prozessor oder CPU-Leistung, eine logische Speicherstruktur und ein entsprechendes Netzwerk um alles

miteinander zu vernetzen. In der Praxis wird eine IaaS aus einem Rechenzentrum her-

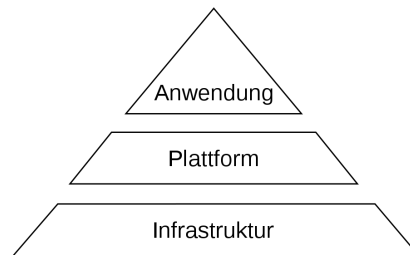


Abb. 2: Cloud-Architektur-Schichtenmodell

aus bereitgestellt. Für den Nutzer der IaaS zeigt sich eine logische Struktur, welche auf der Seite des Hosters aus mehreren einzelnen physischen Komponenten zusammensetzt. Der Hoster hat Zugriff auf Computercluster die in einem Rechenzentrum untergebracht sind. Diese Cluster sind über ein Netzwerk mit dem Internet verbunden. Will ein Kunde jetzt die IaaS in Anspruch nehmen wird ihm vom Hoster ein Fernzugriff über das Internet ermöglicht. Der Kunde braucht sich während der Nutzung des Dienstes keinerlei Gedanken über Wartung, Pflege oder Administration der darunterliegenden Hardware machen. Diese Aufgaben werden vom Hoster übernommen und sind mit dem Kunden über ein Service-Level-Agreement geregelt. Ein Vorteil für den Kunden ist die exakte Abrechnung der Nutzung. So wird ihm lediglich die Nutzung des Dienstes in Rechnung gestellt, jedoch nicht die interne Wartung oder Pflege der Hardware. Sicherlich sind auch diese Kosten mit im Nutzungspreis verrechnet, für den Kunden aber nicht ersichtlich.(vgl. [Ia13]) Damit ein Anbieter einer Vielzahl von Kunden seine Dienste zur Verfügung stellen kann wird auf Virtualisierungstechnik gesetzt.

3.1 Definition Virtualisierung

Zur Definition des Begriffs der Virtualisierung werden zwei Beispiele gezeigt. Zu erst folgt eine Definition, wie sie Wikipedia zu entnehmen ist:

„Virtualisierung bezeichnet in der Informatik die Erzeugung virtueller (d. h. nicht-physikalischer) Dinge wie einer emulierten Hardware, eines Betriebssystems, Datenspeichers oder einer Netzwerkressource. Dies erlaubt es etwa, Computer-Ressourcen (insbesondere im Server-Bereich) transparent zusammenzufassen oder aufzuteilen, oder ein Betriebssystem innerhalb eines anderen auszuführen.“[Vi15a]

Eine ähnliche Definition liefert das IT-Glossar der Firma Gartner. Hier wird die Virtualisierung beschrieben, als eine Abstraktion von IT-Ressourcen die ihre physikalischen Eigenschaften verbergen um andere physische Gegebenheiten vorzugeben als wirklich vorhanden sind. Eine IT-Ressource kann dabei ein Server, ein Client, ein Speicher, ein Netzwerk, eine Applikation oder ein Betriebssystem sein.(vgl.: [Vi15b])

3.2 Technischer Hintergrund

Allgemein spricht man von Host und Gast, wobei der Host die physikalische Maschine darstellt und ein Gast innerhalb dieser Host Maschine virtualisiert betrieben wird. Die Virtualisierung wird technisch grob auf zwei Arten umgesetzt. Es gibt die Host- und Hypervisor-basierte Virtualisierung. Manchmal auch Hypervisor Type 1 und Hypervisor Type 2 genannt. Beide Arten haben einen gemeinsamen Teil an Funktionen und Möglichkeiten, aber auch jeweils eigene Ansätze diese zur Verfügung zu stellen. Zur Verwaltung und Einrichtung von Virtuellen Maschinen, ganz gleich dem Einsatzzweck der Virtuellen Maschine selbst, wird ein Virtual Machine Monitor verwendet. Der Virtual Machine Monitor ist für die Zuteilung von Ressourcen an die einzelnen Virtuellen Maschinen zuständig. Er regelt Zugriffe auf die zu Grunde liegende Hardware. Der Virtual Machine Monitor stellt maßgeblich die Virtualisierungsschicht dar. An welcher Position der Virtual Machine Monitor innerhalb der System-Architektur liegt kann unterschiedlich je nach Art der Virtualisierung sein.(vgl. [VM15]) Virtualisierungslösungen müssen ohne Abhängigkeiten der verwendeten Architekturen folgende Anforderungen erfüllen(vgl. [PG74]):

- Äquivalenz: Das virtualisierte System muss exakt dasselbe Verhalten an den Tag legen wie bei Ausführung direkt auf der Hardware (nicht virtualisiert).
- Isolation: Es muss eine sichere Isolation der virtualisierten Systeme (VM) untereinander gewährleistet sein, so dass die Datensicherheit, Vertraulichkeit und Konsistenz von Daten und Zuständen garantiert ist. Eine instabile Virtual Machine (VM) darf eine andere VM nicht zum Absturz bringen.
- Kontrolle: Die Rechnerressourcen wie z.B. RAMs und CPUs (inkl. Core und Thread) müssen auf kontrollierte Weise den virtuellen Maschinen einzeln zugewiesen werden können.
- Effizienz: Die virtuelle Maschine darf aufgrund der Virtualisierung keinen unangemessenen Overhead produzieren, sondern sollte annähernd so schnell laufen wie auf der blanken Hardware.

Host & Hypervisor Virtualisierung Bei der Host-basierten Virtualisierung liegt der Virtual Machine Monitor (VMM) nicht direkt auf der Hardware, sondern innerhalb des Host Betriebssystems. Der VMM erzeugt eine standardisierte virtuelle Hardware Umgebung auf die die Virtuellen Maschinen zugreifen. Der VMM greift dabei selbst auf die Treiber des Host Betriebssystems zur Anbindung der Geräte zurück. Dadurch wird eine Vielzahl verschiedenster Geräte unterstützt, da die meisten Betriebssysteme weit mehr Treiber beinhalten als der VMM selbst. Bei dieser Art der Virtualisierung muss das Gast Betriebssystem nicht modifiziert werden. Es kann nativ innerhalb der Virtuellen Maschine installiert werden. Dadurch geht die Inbetriebnahme schnell und einfach von der Hand. Der Nachteil dieser komfortablen Lösung ist, dass der Management-Aufwand kaum geringer als mit physischen Servern ist. Außerdem wird sehr viel Overhead während der Verarbeitung erzeugt wodurch die Performance sinkt.

Das Gegenstück zur Host-basierten Virtualisierung wird durch die Hypervisor-basierten Virtualisierung gebildet. Bei dieser Form der Virtualisierung befindet sich der Virtual Machine Monitor direkt auf der physikalischen Hardware, wie in Abbildung 3 abgebildet. Der VMM hat dadurch mehr Privilegien als bei der Host-basierten Variante. Prozessoren

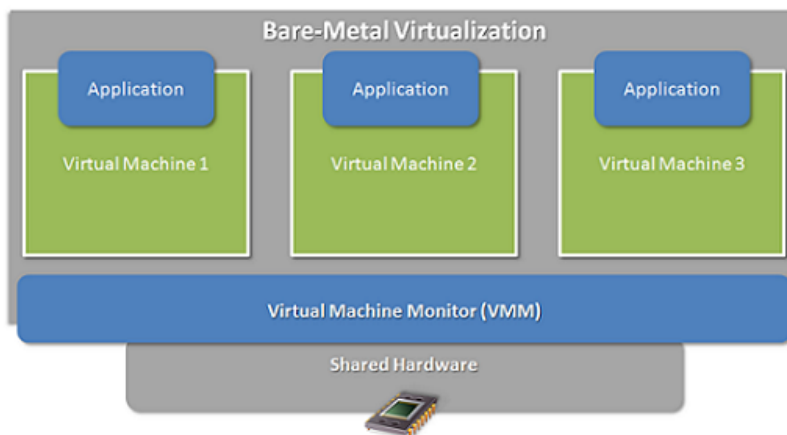


Abb. 3: Bare-Metal-Virtualisierung[Wi15]

bilden die vier Privilegienstufen durch ein Ringsystem ab. Ring 0 stellt dabei den niedrigsten Ring mit den höchsten Privilegien und Ring 3 den höchsten Ring mit den niedrigsten Privilegien dar. Normalerweise liegt das Betriebssystem in Ring 0 und die Anwendungssoftware in Ring 3. Bei einer Virtualisierungslösung befindet sich der VMM in Ring 0 und die Gast Betriebssysteme in Ring 1. Abbildung 4 zeigt die Privilegienstufen der CPU ohne(links) und mit(rechts) Virtualisierung.

„Der Ansatz ist performanter als die Hosted-Architecture und bietet mehr Sicherheit aufgrund höherer Isolation der Maschinen. Betriebssysteme lassen sich ohne Modifizierung betreiben. Nachteil: Die mit dem proprietären Betriebssystem mitgelieferten Treiber entsprechen meist nicht dem aktuellen technologischen Stand. Auch muss man innerhalb der Systemgrenzen bleiben. Auf ein 32-Bit-System lässt sich beispielsweise keine 64-Bit-Anwendung installieren.“[Vo07]

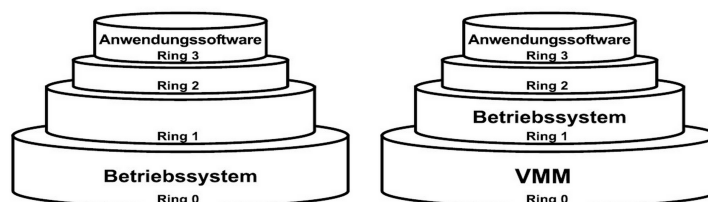


Abb. 4: Privilegienstufen der CPU ohne und mit Virtualisierung ([?])

4 Marktanalyse

Am Markt sind eine Vielzahl von Anbietern vertreten. Es werden die unterschiedlichsten Kombinationen und Varianten der verschiedenen Cloud Dienste angeboten. Um aus diesem Wust an Angeboten eine klare Richtung erkennen zu können wurde das Magic Quadrant Cloud IaaS (Abbildung 5) des Marktforschungsinstitut Gartner betrachtet und interpretiert. Grundlegend zur Erklärung des Magic Quadrants von Gartner ist zu sagen,

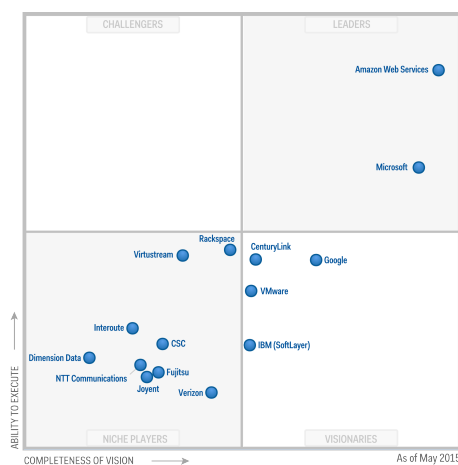


Abb. 5: Gartner 2015 Magic Quadrant Cloud IaaS[Ma15]

dass es immer in vier Quadranten aufgeteilt ist. Allen Quadranten liegen zwei Dimensionen zu Grunde. Da wäre zum einen die Fähigkeit zur Umsetzung an dem Anbieter gemessen werden und zum anderen die Vollständigkeit der Vision. Je nach dem wie die Anbieter in diesen beiden Kriterien abschneiden werden sie in den Quadranten positioniert. Die vier Quadranten selbst beschreiben (v.l.u.n.r.u) die Nieschenanbieter, Herausforderer, Marktführer und Vorreiter.

Amazon Web Services Gartner ordnet den Amazon Service im Quadranten der Marktführer ein mit großem Abstand zum nächsten Anbieter. Amazon dominiert weitestgehend den Markt von Cloud IaaS. Die Angebotene Dienste reichen von einem einfachen virtuellen Server bis hin zu einer hoch integrierten Lösung bestehend aus Server, Storage und Netzwerkinfrastruktur und den darauf aufbauenden Services. Amazon verwendet dabei ein gegliedertes Preismodell. Es besteht aus drei Komponenten - Datenverarbeitung, Speicherung und ausgehende Datenübertragung. Je nach verwendetem Dienst unterscheiden sich die Komponenten geringfügig voneinander. In Punkto Datenverarbeitung rechnet Amazon nach Stunden ab. Das heißt es wird nur die tatsächliche Nutzung der Ressourcen, wie Rechnerzeit, Arbeitsspeicher und lokalem Speicher, in Rechnung gestellt. Die Speicherung von Daten wird nach Gigabyte berechnet und skaliert je nach Grad der Redundanz. Eine Preisstaffelung ist möglich durch Speicherkontingente. Die Abrechnung der Komponenten Datenübertragung erfolgt nur anhand der ausgehenden Daten, für eingehenden

de Daten fallen keine Kosten an. Auch bei der Datenübertragung ist eine Preisstaffelung möglich. Damit bietet Amazon ein sehr flexibles Preismodell an mit dem kleine und auch große Kunden ihre Kosten im Blick behalten können und nur nach tatsächlicher Nutzung berechnet werden.(vgl. [Ho15])

Microsoft Azure Microsoft ist laut Gartner der größte Konkurrent von Amazon. Auch Microsoft wird im Quadranten der Marktführer eingeordnet durch Gartner. So scheinen die Vollständigkeit der Vision und ebenso die Fähigkeit zur Umsetzung ähnlich hoch wie bei Amazon. Microsoft nennt seine Cloud Microsoft Azure. Microsoft bietet eine Reihe von Diensten an in ihrer Cloud. Darunter zählen Virtuelle Maschinen, Datenbanken, Storage und Netzwerk Ressourcen sowie auch reine Compute Power für große parallele Datenverarbeitung aus der Cloud. Jeder ausgewählte Dienst lässt sich detailliert konfigurieren. Es lässt sich die Region, meistens ein Typ, ein Tarif und je nach Dienst noch eine Instanzgröße einstellen. Das Preismodell von Microsoft ist ähnlich dem von Amazon. Es wird jedoch Minuten genau abgerechnet, wobei die kleinste buchbare Option ein 10-Minuten Slot ist. Rabatte werden je nach Zahlungsart auch angeboten. Microsoft bietet mit seiner Cloud einen nahezu vollständigen Service an, der die meisten Einsatzgebiete abdeckt, aber auch für einige spezielle Anwendungen Lösungen im Portfolio hat.

Google Cloud Platform Google hat mit seinen ohnehin riesigen IT Ressourcen auf Grund ihres Kerngeschäfts eine ideale Voraussetzung um auch einen kommerziellen Cloud-betrieb anbieten und gewährleisten zu können. Wie die beiden Platzhirsche Amazon und Microsoft kann auch Google eine Reihe von unterschiedlichen Diensten vorweisen. Google bietet verschiedenste Storage und Compute Lösungen an die durch das starke eigene Netzwerk gestützt werden. Die Lösungen von Google unterscheiden sich von Amazon und Microsoft in dem Punkt, dass kein direkter Zugriff auf eine Virtuelle Maschine ermöglicht wird. Google stellt dafür entsprechende Software Development Kits bereit, die für die jeweilige Aufgabe vorgesehen und optimiert sind. Zudem bietet Google, wie auch die anderen Anbieter, Templates an die einen gesamten Cloud Stack zur Auslieferung eines Services verwenden. Die Abrechnung erfolgt Minuten genau und voll automatisch. Google kann unter den Visionären laut Gartner mit der höchsten Fähigkeit zur Umsetzung aufwarten.

VMware vCloud Air VMware gehört seit der Übernahme des Speicherherstellers EMC durch Dell auch zu Dell. Dadurch sind sie ein ganzes Stück näher an Google gekommen in Punkto Umsetzungsfähigkeit. Eigenen Aussagen zu Folge können sie mit der doppelten Rechenleistung von Microsoft Azure und der dreifachen Storage-Leistung von Amazon Web Service aufwarten. VMware kann mit hervorragendem Know-how im Bereich der Virtualisierung punkten. Dieses Know-how in Verbindung mit der Cloud ermöglicht ihnen einen guten Standpunkt unter den Visionären. Die Stärke von VMware liegt in der Virtualisierung von Windows Servern. Wohingegen die anderen Anbieter eher Unix/Linux basierte Lösungen anbieten. Sämtliche Bereiche einer IaaS Cloud werden von VMware mit entsprechenden Lösungen bedient. Alle Dienste lassen sich mit einander integrieren.

Zusätzlich zum Kernbereich einer IaaS Cloud bietet VMware Verfügbarkeits-, Sicherheits- und Managed-Services an. So lassen sich komplette SAP Landschaften geführt durch die VMware Cloud aufsetzen und betreiben. Das Preismodell von VMware unterscheidet zwischen einer dedizierten Cloud und einer virtuellen Cloud. Der Unterschied besteht lediglich in der Dimensionierung der Ressourcen und den Angeboten optionaler Dienste.

IBM SoftLayer IBM als Urgestein unter den Anbietern zählt wie Google und VMware zu den Visionären im Magic Quadrant von Gartner. Auch IBM unterscheidet wie VMware zwischen Bare-Metal Ressourcen und virtualisierten Ressourcen. Die Bare-Metal Server lassen hinsichtlich der gesamten verwendeten Hardware konfigurieren, wohingegen die virtualisierten Server nur bezüglich der Anzahl der Kerne, RAM und Speicher konfiguriert werden. Generell können die Bare-Metal Server mit erheblich mehr Leistung ausgestattet werden. Zusätzlich bietet IBM eine leistungsstarke GPU Cloud an für besondere rechenintensive Aufgaben. Im Bereich Storage und Network kann IBM mit den verschiedensten Diensten die selben Lösungen wie die anderen Anbieter zur Verfügung stellen. Abgerechnet wird bei IBM per Stunde oder Monat. IBM kann sich somit ebenso im Markt behaupten und stetig besser positionieren.

5 Datenschutz und Datensicherheit

„Datensicherheit bedeutet die technischen und organisatorischen Maßnahmen umzusetzen, um Vertraulichkeit, Verfügbarkeit und Integrität der IT-Systeme sicherzustellen. Public Cloud-Anbieter bieten weit mehr Sicherheit, als es sich ein deutsches mittelständisches Unternehmen leisten kann. Das hängt damit zusammen, dass Cloud-Anbieter gezielt in den Aufbau und die Wartung ihrer Cloud Infrastrukturen investieren und ebenfalls das dafür notwendige Personal beschäftigen und die entsprechenden organisatorischen Strukturen geschaffen haben. Hierzu werden jährlich Milliarden von US-Dollar in die Hand genommen. Es gibt nur wenige Unternehmen außerhalb der IT-Branche, die in ähnlicher Weise in IT-Sicherheit investieren können und wollen.“

Beim Datenschutz geht es um den Schutz der Persönlichkeitsrechte während der Datenverarbeitung und den Schutz der Privatsphäre. Dieses Thema sorgt bei den meisten Unternehmen für die echten Kopfschmerzen. Denn beim Verstoß gegen das Bundesdatenschutzgesetz macht der Gesetzgeber kurzen Prozess. Es geht zunächst also darum, den Cloud-Anbieter für die Einhaltung der im 9 festgehaltenen Regeln im Bundesdatenschutzgesetz in die Verantwortung zu nehmen und dies selbst auf Basis von 11 zu überprüfen. Für die Erfüllung von 11 empfiehlt es sich auf die Gutachten von Wirtschaftsprüfern zurückzugreifen, da kein Anbieter jeden Kunden einzeln ein Audit durchführen lassen kann. Der Datenschutz ist ein absolut wichtiges Thema, schließlich handelt es sich dabei um sensibles Datenmaterial. Es ist aber in erster Linie ein rechtliches Thema, was durch Maßnahmen der Datensicherheit gewährleistet werden muss.

Dass ein Rechenzentrum in Deutschland vor der Spionage befreundeter Staaten oder Angriffe durch Hacker einen besseren Schutz bietet ist und bleibt ein Märchen. Denn wo ein Wille ist, da ist auch ein Weg. Sprich, wenn ein Angreifer an Daten gelangen möchte, dann ist dies einzig und alleine mit der kriminellen Energie verbunden, die er bereit ist zu unternehmen und die finanziellen Mittel, die ihm dafür zur Verfügung stehen. Sind die technischen Herausforderungen zu hoch, dann bleibt immer noch der Faktor Mensch als Option und der ist bekanntlich käuflich.

Nichtsdestotrotz haben US-amerikanischen Cloud-Größen die Bedenken deutscher Unternehmen erkannt und haben angekündigt, ihre Services aus deutschen Rechenzentren anzubieten. Darunter Salesforce (Partnerschaft mit T-Systems), VMware und Oracle. Jüngst hat auch Amazon ein RZ in Deutschland eröffnet. Allerdings ist zu beachten, dass ein deutsches Rechenzentrum alleine nichts mit einer höheren Datensicherheit zu tun hat.“ [Di14]

6 Fazit

Cloud Computing und im besonderen IaaS ist heute ein ziemlich ausgereiftes Produkt. Jedoch ist immer noch jede Lösung eine individuelle Lösung. Google versucht das ganze über ihre fertigen Pakete und SDKs in eine Art Baukasten-Set zu pressen. Technisch ist das Cloud Computing keine Neuheit. Es werden alle Konzepte verwendet die schon lange bestehen, aber erst heute aufgrund der niedrigen Hardware Preise gerade für RAM und SSDs relevant. Nach Betrachtung der Marktanalyse von Gartner und der Studie von Crisp Research ist deutlich geworden das ein großer Markt für Cloud Computing vorhanden ist in dem noch viel Platz für neue Anbieter ist und damit auch weiterhin Wettbewerb und eine gewisse Dynamik aufrechterhalten wird. Global gesehen wird Cloud Computing immer bedeutsamer für Unternehmen. Auch in Deutschland auf dem Vormarsch, aber noch weit vom globalen Akzeptanzgrad entfernt. Oft ist fehlende Datensicherheit oder Datenhoheit einer der Hauptgründe Cloud Anwendungen nicht zu implementieren. Alle betrachteten Anbieter können den gesamten IaaS Bereich abdecken und bieten jeweils noch Spezialitäten Dienste für besondere Einsatzzwecke an - beispielsweise die GPU Cloud von IBM.

Literaturverzeichnis

- [Cl14] Cloud - IaaS, PaaS, SaaS, Public und Private einfach erklärt - Cloudstarter, <https://www.cloudstarter.info/cloud-iaas-paas-saas-public-und-private-einfach-erklart-104388>, Stand: 29.10.2015.
- [Cl15] Cloud Computing - Wikipedia, https://de.wikipedia.org/wiki/Cloud_Computing, Stand: 29.10.2015.
- [Di14] Die Suche nach dem heiligen IaaS-Gral - Crisp Research, <http://www.crisp-research.com/die-suche-nach-dem-heiligen-iaas-gral/>, Stand: 29.10.2015.
- [Di15] Die wichtigsten Cloud-Trends 2015 - Crisp Research, <http://www.crisp-research.com/die-wichtigsten-cloud-computing-trends-2015-teil-1/>, Stand: 29.10.2015.

- [Ho15] How AWS Pricing Works - Amazon Web Services, https://media.amazonwebservices.com/AWS_Pricing_Overview.pdf, Stand: 29.10.2015.
- [Ia13] IaaS Technical Guidance for HSP - Microsoft Corporation, <https://www.uk.insight.com/content/dam/insight/EMEA/uk/hosting/microsoft-tech-hosting-paper.pdf>, Stand: 29.10.2015.
- [Ma15] Magic Quadrant Cloud IaaS 2015 - Gartner Inc, <http://www.gartner.com>, Stand: 29.10.2015.
- [PG74] Popek, Gerald J.; Goldberg, Robert P.: Formal Requirements for Virtualizable Third Generation Architectures. Commun. ACM, 1974.
- [St15] Studie Digital Business Readiness - Crisp Research, <http://www.crisp-research.com/cloud-market-update-2015-wolkig-mit-aussichten-auf-digitale-unternehmen/>, Stand: 29.10.2015.
- [Vi15a] Virtualisierung (Informatik), [https://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](https://de.wikipedia.org/wiki/Virtualisierung_(Informatik)), Stand: 10.06.2015.
- [Vi15b] Virtualization - Gartner Inc., <http://www.gartner.com/it-glossary/virtualization>, Stand: 29.10.2015.
- [VM15] VMM - Virtual Machine Monitor - Klaus Lipinski, <http://www.itwissen.info/definition/lexikon/VMM-virtual-machine-monitor.html>.
- [Vo07] Vor- und Nachteile von Virtualisierungsarchitekturen - Tim Wolfinger, <http://www.it-business.de/software/betriebssysteme/articles/58217/index2.html>, Stand: 29.10.2015.
- [Wa12] Was hätten Sie denn gerne: IaaS, PaaS oder SaaS? - QSC AG, <http://blog.qsc.de/2012/12/was-hatten-sie-denn-gerne-iaas-paas-oder-saas/>, Stand: 29.10.2015.
- [Wi15] Wie funktioniert Virtualisierung? - National Instruments, <http://www.ni.com/white-paper/8709/de/>, Stand: 29.10.2015.

Lastenheft

Projektgruppe
Cloud-basierte analytische Informationssysteme

PG.CAIS

Version 1.5

Dokumentversionen

Version	Datum	Autor	Änderungsgrund/Bemerkung
0.1	24.11.2015	Marcel Kuchler	Template-Erstellung
0.2	02.12.2015	Marcel Kuchler	Anpassung des Templates gemäß letztem Jour fixe
0.3	02.12.2015	Nils Lutz	Seitenumbrüche und Anforderungstemplate eingefügt
1.0	09.12.2015	Dörthe Euhausen	Zusammenfügen der einzelnen Kapitel zur Version 1.0, an der alle Projektgruppenmitglieder gearbeitet haben
1.1	16.12.2015	Dörthe Euhausen	Inhaltliche und strukturelle Anpassungen nach Abstimmung mit dem Kunden
1.2	20.12.2015	Dennis Vorreiter	Use Case-Diagramm überarbeitet und Use Case-Erstellung im IEC 62559-Template
1.3	22.12.2015	Phillip Kramer	Anpassungen der Priorisierung der Anforderungen sowie eine Änderung bei der Zuordnung zu den Anforderungsarten
1.3	16.01.2016	Dennis Vorreiter	Überarbeitetes Use Case-Diagramm und zugehörige textuelle Beschreibung eingefügt
1.3	16.01.2016	Dörthe Euhausen	Mehrere kleinere Anpassung in der Formatierung und Formulierung
1.4	28.01.2016	Dörthe Euhausen	Überführung in das neue LaTeX-Template sowie erneute Korrekturen
1.5	22.02.2016	Dörthe Euhausen	Übernahme der letzten Anmerkungen des Kunden, finale Version

Autor des Dokumentes	Projektgruppe CAIS	Erstellt am	24.11.2015
Dateiname	Lastenheft_Version1-5_final		
Seitenanzahl	28	© 2016 PG.CAIS - Universität Oldenburg	Vertraulich!

Inhaltsverzeichnis

1 Einleitung	1
1.1 Zweck und Ziel dieses Lastenheftes	1
1.2 Projektbeschreibung	1
1.3 Bezüge zu anderen Dokumenten	2
1.4 Verteiler und Vorgehen	2
2 Ideen und Rahmenbedingungen	4
2.1 Ziele des Kunden	4
2.2 Nutzen des Anwenders	4
2.3 Benutzer/Zielgruppe	4
2.4 Systemvoraussetzungen	4
2.5 Ressourcen	5
3 Beschreibung der Anforderungen	6
3.1 Anwendungsfälle	6
3.2 Use-Case-Diagramm	12
3.3 Funktionale Anforderungen	13
3.4 Nicht-funktionale Anforderungen	14
3.5 Risiken	14
4 Freigabe/Genehmigung	17
Literaturverzeichnis	V
Tabellenverzeichnis	VI
Abbildungsverzeichnis	VII
Anhang	VII

1. Einleitung

Bei dem vorliegenden Dokument handelt es sich um das Lastenheft der Projektgruppe *Cloud-basierte analytische Informationssysteme*, kurz CAIS. Es enthält Informationen zur Projektidee und gegebenen Rahmenbedingungen sowie die Ergebnisse der Anforderungserhebung.

1.1 Zweck und Ziel dieses Lastenheftes

Der Inhalt des folgenden Lastenheftes soll eine strukturierte und eindeutige Übersicht über die Projektziele und die an das System genannten Anforderungen darstellen. Das Lastenheft fasst die inhaltliche Relevanz des Projektes zusammen und ordnet sie in den bestehenden Kontext ein. Es soll einen umfassenden Überblick über Problematik und Lösungsansatz bieten und das zugrunde liegende Projekt beschreiben.

Das vorliegende Lastenheft dient ebenfalls als Wiedergabe des Konsens innerhalb des Projekts zwischen Auftraggeber und Auftragnehmer und spiegelt somit die Erwartungen und Verpflichtungen an das Projekt wider. Es dient infolgedessen als eine verbindliche Grundlage für das Projekt.

1.2 Projektbeschreibung

Das zugrunde liegende Projekt dieses Lastenheftes umfasst eine Business Intelligence-Anwendung, die in ihrer Gesamtheit mittels einer cloud-basierten Plattform realisiert werden soll. Die Projektentwicklung wird im Rahmen der Projektgruppenveranstaltung (WS/SS 2015-2016) der Carl von Ossietzky Universität Oldenburg durchgeführt und beaufsichtigt. Der Projektzeitraum beträgt ein Jahr. Das Projektteam besteht aus sieben Mitgliedern. Als Domänenreferenz und Ansprechpartnern stellt sich Marco Haas von der Unternehmensberatung CeyonIQ zur Verfügung.

Das Projekt umfasst eine Anwendung zur Verarbeitung von Kennzahlen zur Gewinnanalyse von Energieverträgen, die den bestehenden Vorgang der Vertragsverhandlungen zwischen Energiekonzern und Kunde vereinfachen soll. Das Projekt vereint dafür die analytischen Vorteile einer Business Intelligence-Architektur mit der skalierbaren und dynamischen Nutzung von Cloud-Technologien. Dies umfasst auf Seiten des Business Intelligence-Bereichs sowohl

eine intelligente Verarbeitung von fremden Quelldaten, ein intelligentes Analyseverfahren sowie eine angemessene und anpassbare Darstellung der Analysen. Auf Seiten der Cloud-Technologien bietet das System ein Skalierungsverfahren, um Ressourcen je nach Anfragelast verteilen zu können, ein Accounting-Management, um eine verbrauchsorientierte Abrechnung bieten zu können, sowie eine hohe Verfügbarkeit auf allen gängigen Endgeräten.

Die Umsetzung des Projektes soll helfen, eine genaue Vorstellung über die Vor- und Nachteile bei der Implementierung eines solchen Systems in ein Unternehmen zu erhalten. So soll eine wissenschaftliche Grundlage für weitere Diskussionen geschaffen werden.

1.3 Bezüge zu anderen Dokumenten

Das vorliegende Lastenheft wurde unter Referenz folgender Dokumente erstellt:

- Projektdokumentation BICE - Business Intelligence in the Cloud for Energy
Genutzt als Übersicht zum gesamten Projektumfang, als Informationsquelle zu einzelnen Abläufen (Beispiel: Kostendeckungsbeitrag) und beispielhaften Vorgehen innerhalb eines solchen Projektes.
- Referenzarchitektur Oliver Norkus
Referenzarchitektur als Anhaltspunkt für notwendige Komponenten und Erstellung von Anforderungen und der Use-Cases.

1.4 Verteiler und Vorgehen

Zur Kommunikation innerhalb der Projektgruppe und auch zur Kommunikation mit dem Kunden wurden Mailverteiler eingerichtet. Das Lastenheft wird an den Kunden und an alle Mitglieder der Projektgruppe CAIS verteilt.

Die entsprechenden Mailadressen sind der untenstehenden Tabelle in Kapitel 1.2.1 zu entnehmen. Die Vorgehensweise der Anforderungserhebung wird im Unterkapitel 1.2.2 beschrieben.

1.4.1 Verteiler für dieses Lastenheft

Rolle/Rollen	Name	E-Mail	Bemerkung
Kunde	Oliver Norkus	oliver.norkus@offis.de bzw. pg-cais@informatik.uni-oldenburg.de	-
Projektgruppe	*	pg-cais@informatik.uni-oldenburg.de	-

1.4.2 Vorgehensweise

Um die Anforderungen des Kunden an das zu entwickelnde System zu ermitteln, hat sich die Projektgruppe für die Anforderungserhebung mittels Interviews entschieden.

Die Interviews fanden an zwei kurz aufeinander folgenden Terminen statt. Zur Vorbereitung für den Kunden und als Gesprächsleitfaden wurde im Voraus ein Fragebogen entwickelt, der dem Kunden vor den Interviews zugeschickt wurde. Das erste Interview diente dazu, grundlegende Anforderungen an die zu entwickelnde Business Intelligence-Lösung zu ermitteln. Das zweite Interview baute auf den Ergebnissen des ersten Interviews auf und diente dazu, die bereits aufgenommenen Anforderungen weiter zu spezifizieren.

Die Protokolle der beiden Interviews befinden sich im Anhang (Kapitel 4) dieses Dokuments.

2. Ideen und Rahmenbedingungen

Im Folgenden werden die Ideen und Rahmenbedingungen, die durch die Projektgruppe analysiert wurden, beschrieben.

2.1 Ziele des Kunden

Das Ziel des Projekts ist die prototypische Erstellung einer Software beziehungsweise eines Prototyps, der Nutzern anschauliche Reports über Kennzahlen liefert (Kennzahlen-Reports). Die Ergebnisse sollen visualisiert dargestellt werden. Zur Darstellung der Kennzahlen sollen verschiedene Möglichkeiten wie Balken- oder Kreisdiagramm zur Auswahl bereitstehen.

2.2 Nutzen des Anwenders

Der fertige Prototyp soll eine Cloud-BI-Lösung sein, auf die ein Vertriebsmitarbeiter während eines Kunden-Meetings mit seinem mobilen Endgerät zugreifen kann. Der Nutzen ist eine Echtzeitberechnung von Verbrauchswerten sowie eine Gewinn- und Verlustrechnung mit Prüfung des Gewinns, wodurch sich der Geschäftsprozess der Vertragsverhandlung optimieren lässt.

2.3 Benutzer/Zielgruppe

Die Zielgruppe sind Energieversorger (z.B. EWE) im B2B-Bereich, im Speziellen Mitarbeiter des Vertriebs, die Vertragsverhandlungen mit Großkunden (z.B. IKEA) durchführen. Das System soll neben einer Business User- und einer Power User-Rolle außerdem über eine Rolle für Administratoren verfügen. Die einzelnen Rollen werden später genauer spezifiziert.

2.4 Systemvoraussetzungen

Es soll das Produkt SAP HANA als In-Memory-Datenbank verwendet werden. Ansonsten sind die Systemvoraussetzungen hardwareunabhängig. Das System muss den Anforderungen an ein Cloud-BI-System entsprechen. Dazu muss das System die gemeinsame Nutzung physischer

Ressourcen durch unterschiedliche Mandanten ermöglichen. Außerdem muss im Rahmen des Load Balancing die unverzügliche Anpassung der Ressourcen an den tatsächlichen Bedarf stattfinden. Ebenso muss die Nutzung erfasst werden, um eine genaue Abrechnung der in Anspruch genommenen Leistungen zu gewährleisten. Auch muss ein allgemeiner Netzwerkzugriff durch die Bereitstellung einer ausreichenden Bandbreite ermöglicht werden. Der Anwender benötigt dazu User Self Services, um das System ohne den Anbieter oder einen Administrator konfigurieren zu können.

2.5 Ressourcen

Als personelle Ressourcen stehen der Projektgruppe die Projektgruppenmitglieder inklusive Oliver Norkus als Product Owner beziehungsweise Kunde sowie Marco Haas (Ceyoniq) zur Verfügung, der Branchenexpertise einbringt und Testdaten bereitstellt. Die benötigte technische Infrastruktur wird von der Carl von Ossietzky Universität Oldenburg und voraussichtlich außerdem vom Hasso-Plattner-Institut Potsdam bereitgestellt. Die Projektgruppe verfügt über kein Budget, jedoch kann kostenfreie Software verwendet und auch eigene Software entwickelt werden. Zusätzlich stehen der Projektgruppe die dokumentierten Ergebnisse des vorangegangenen Projekts *Business Intelligence in the Cloud for Energy* zur Verfügung.

3. Beschreibung der Anforderungen

Die Anforderungen enthalten alle an das zu entwickelnde System verbindlich geforderten Funktionalitäten. Kern des Lastenhefts sind die funktionalen und nicht-funktionalen Anforderungen an das System, sowie eine Skizze des Gesamtsystementwurfs. Der Entwurf berücksichtigt die zukünftige Umgebung und Infrastruktur, in der das System später betrieben wird, und gibt Richtlinien für Technologieentscheidungen. Zusätzlich werden die zu unterstützenden Phasen im Lebenszyklus des Systems identifiziert und als logistische Anforderungen aufgenommen. Ebenfalls Teil der Anforderungen ist die Festlegung von Lieferbedingungen und Abnahmekriterien.

Funktionale Anforderungen beschreiben die Fähigkeiten eines Systems, die ein Anwender erwartet, um mit Hilfe des Systems ein fachliches Problem zu lösen. Die Anforderungen werden aus den zu unterstützenden Geschäftsprozessen und den Ablaufbeschreibungen zur Nutzung des Systems abgeleitet. Nicht-funktionale Anforderungen beschreiben Anforderungen an das System, die nicht-fachlicher Natur sind, jedoch entscheidend zur Anwendbarkeit des Systems beitragen. Sie definieren beispielsweise Qualitätsanforderungen, Sicherheitsanforderungen oder Performanceanforderungen.

Zur ordentlichen Kategorisierung der Anforderungen wird folgende Tabelle verwendet.

Anforderungsart	Beschreibung
Funktion	Hierbei handelt es sich um eine funktionale Anforderung.
Schnittstelle	Die Anforderungsart Schnittstelle wird für funktionale Anforderungen, die für eine Schnittstelle gilt, genutzt.
Technologie	Technologische Anforderungen schränken den Lösungsraum für die Realisierung durch Vorgaben oder technische Rahmenbedingungen ein.
Benutzeroberfläche	Anforderungen an die Benutzeroberfläche beschreiben, wie sich das System dem Benutzer darstellen soll.
Lieferung	Anforderungen, die für die Auslieferung notwendig bzw. zu beachten sind.
Vertrag/Gesetzgebung	Nicht relevant im Rahmen der Projektgruppe

Tabelle 3.1: Beschreibung der Anforderungsarten

3.1 Anwendungsfälle

Das folgende Kapitel beschäftigt sich mit den Anwendungsfällen, in denen das System zum Einsatz kommen soll. Zunächst wird dazu beispielhaft der Anwendungsfall *Kostendeckungs-*

beitrag errechnen nach IEC 62559 dargestellt. Darauf folgt eine Übersicht über alle vom System unterstützten Anwendungsfälle.

3.1.1 Use Case Identification

Use Case Identification and Version Management

Use Case ID CAIS.REQ.UC.1
Area Domain(s)/Zones Energie, Kostenberechnung
Name of Use Case Kostendeckungsbeitrag errechnen

Version	Date	Author	Name of Use Case	Approval Status
0.1	19.12.2015	Dennis Vorreiter	Erstentwurf	Entwurf

3.1.2 Scope and Objectives

Scope Berechnung des Kostendeckungsbeitrags innerhalb einer Vertragsverhandlung. Optimierter Ablauf mittels schneller Darstellung der Kennziffern. Das System soll sowohl eine effizientere Verhandlung mit dem Kunden unterstützen als auch den Energielieferanten einen optimalen Gewinn berechnen.

Objective(s) Ziel des Use Case ist es, den Vorgang der Vertragsverhandlungen innerhalb des Kostendeckungsbeitrags zu optimieren. Am Ende einer Verhandlung soll so dem Kunden ein bestmöglicher Energiepreis errechnet werden können, der dem Energielieferanten aber noch Gewinn einbringt.

Related business case(s) Zusammenhang mit anderen Anwendungsfällen:

- Gesamtgewinn berechnen
- Teilgewinn einsehen
- Ergebnisse visuell einsehen

3.1.3 Narrative of Use Case

Short description Abwicklung einer Vertragsverhandlung zwischen Energielieferant und Kunde mittels Abruf der Kundendaten in dem BI-System. Berechnung des Deckungsbeitrags über Erfassung von aktuellen Arbeits-, Leistungs- und Netznutzungskosten mittels Abruf aus dem BI-Cloud-System. Optimierte Darstellung der Berechnung innerhalb des Interfaces.

Complete description Innerhalb einer Verhandlung zwischen Energielieferant und Kunde muss ein Energiepreis festgelegt werden, den der Kunde bereit zu zahlen ist, bei dem der Energielieferant allerdings keinen Verlust macht. Um einen geeigneten Preis zu finden, wird die sogenannte Deckungsbeitragsrechnung benutzt. Diese unterteilt sich in drei Unterbeiträge. Deckungsbeitrag 1 erfasst die Differenz aus dem Erlös und den Arbeitskosten. Deckungsbeitrag 2 erfasst die Differenz aus Deckungsbeitrag 1 und den Leistungskosten. Der dritte Beitrag erfasst die Differenz aus dem zweiten Beitrag und den Netznutzungskosten. Daraus setzt sich der gesamte Gewinn zusammen. Damit werden alle Kosten aus den eigenen Kraftwerken (Beitrag 1 und 2) sowie den regionalen Kosten (Beitrag 3) verrechnet. Diese Daten müssen zunächst eingeholt werden. Danach können dem Kunden Angebote unterbreitet werden.

3.1.4 Use Case Conditions

Assumptions Das System ist mit dem Cloud-System verbunden. Das System ist online.

Prerequisites Das System muss Anbindung an alle Kennziffern haben. Die entsprechenden Kundendaten müssen im System vorhanden sein. Der Nutzer muss eingeloggt sein.

3.1.5 Classification Information

Relations to other Use Cases

Gesamtgewinn errechnen: Der Gesamtgewinn kann entscheiden, inwieweit der Energielieferant bereit ist, den Angebotspreis zu reduzieren

Ergebnisse visuell darstellen: Gibt einen optimierten Überblick über für die Verhandlung wichtigen Kennziffern.

Level of Depth Detailed Use Case

Priorisation MUST

Generic, Regional, National Relation Netzgebiet Ceyoniq

Nature of Use Case Business Process

General Remarks Noch teils unvollständige Use Case-Beschreibung, für eine detailliertere Ausführung werden weitere Informationen von Marco Haas benötigt.

3.1.6 Diagram(s) of Use Case

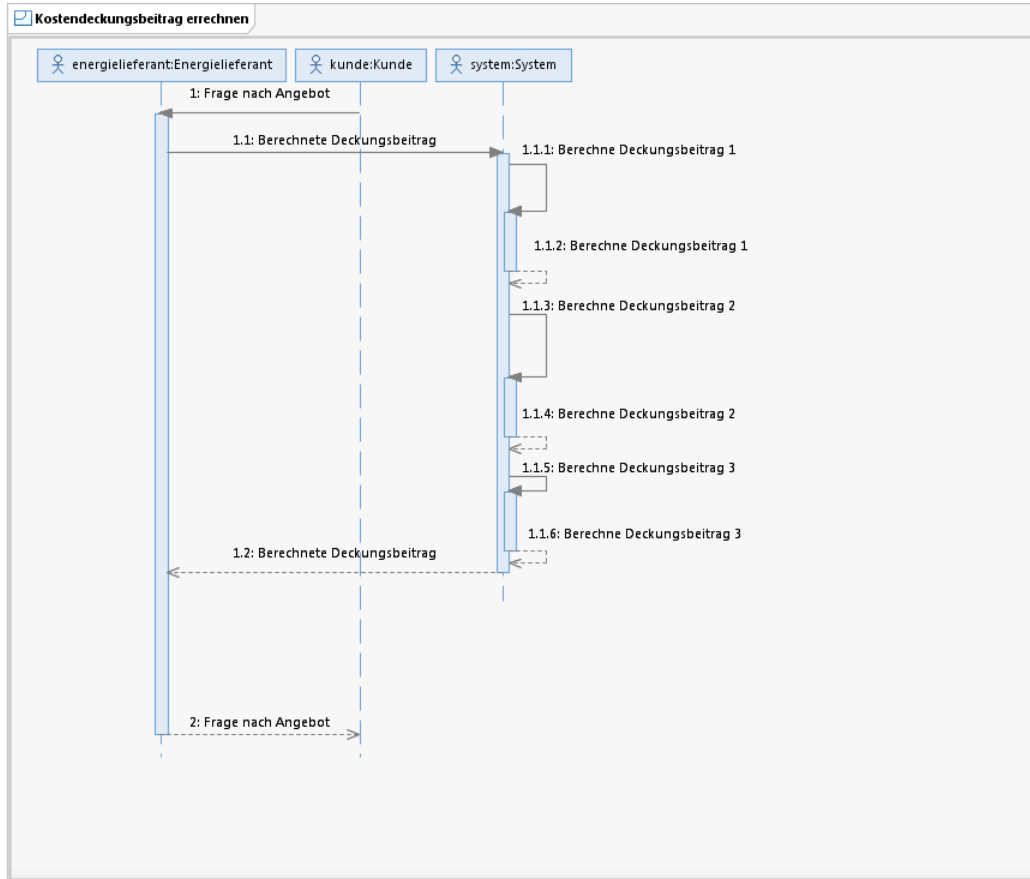


Abbildung 3.1: Sequenzdiagramm des Use Cases Kostendeckungsbeitrag errechnen

3.1.7 Actors

Grouping		Group Description
Actor Name	Actor Type	Actor Description
Energilieferant	Mensch	Vertreter des Energilieferants, dient als Mittelsmann für den Energieeinkauf
Kunde	Mensch	Energieverbraucher (Großkunde)
Berechnungssystem	System	verrechnet die entsprechenden Daten zu einem endgültigen Wert

3.1.8 References

No.	01
Reference Type	Präsentation
Reference	Vortrag über die Vertragsverhandlung in der Energiebranchen
Status	published
Impact on Use Case	Berechnungsschema zum Kostendeckungsbeitrag
Originator	Marco Haas/Ceyoniq

3.1.9 Scenario Conditions

No.	01
Scenario	Deckungsbeitrag festlegen
Scenario Description	Errechnung des Deckungsbeitrags
Primary Actor	Energielieferant
Triggering Event	Kunde möchte Strom kaufen
Pre-Condition	Der Kunde muss bereits im System bestehen
Post-Condition	Der Deckungsbeitrag wurde errechnet

3.1.10 Scenario Deckungsbeitrag festlegen

Step No.	01
Event	Angebot wird benötigt
Name of Process	Initialisierung
Description of Process	Der Energielieferant soll ein Angebot erstellen und muss nun einen Preis vorschlagen.
Information Producer	Energielieferant
Information Receiver	Berechnungssystem
Information Exchange ID	01

Step No.	02
Event	Deckungsbeitrag 1 errechnen
Name of Process	Berechnung
Description of Process	Differenz aus Erlös und Arbeitskosten
Information Producer	Berechnungssystem
Information Receiver	-
Information Exchange ID	-

Step No.	03
Event	Deckungsbeitrag 2 errechnen
Name of Process	Berechnung
Description of Process	Differenz aus Deckungsbeitrag 1 und Leistungskosten
Information Producer	Berechnungssystem
Information Receiver	-
Information Exchange ID	-

Step No.	04
Event	Deckungsbeitrag 3 errechnen
Name of Process	Berechnung
Description of Process	Differenz aus Deckungsbeitrag 1 und Leistungskosten
Information Producer	Berechnungssystem
Information Receiver	-
Information Exchange ID	02

Step No.	05
Event	Ergebnis dem Angebot hinzufügen
Name of Process	Rückmeldung
Description of Process	Gib dem Energielieferanten das Ergebnis zurück
Information Producer	Berechnungssystem
Information Receiver	Energielieferant
Information Exchange ID	-

3.1.11 Information Exchanged

ID	01
Name	Kundendaten
Description	Die Daten des Kunden über Verbrauch und Preisvorstellung

ID	02
Name	Deckungsbeitrag
Description	Der ermittelte Gewinn mit Abzug aller Kosten

3.2 Use-Case-Diagramm

Im Zuge der Anforderungsanalyse ergaben sich für das CAIS-System die im Folgenden genannten Use-Cases, die für die drei Akteure Admin, Power- und Business-User benötigt werden. Die Abbildung 3.2 gibt eine Übersicht über alle zu bearbeitenden Use-Cases.

Der Admin soll innerhalb der Datenbank, ein Subsystem des CAIS-Systems, Benutzer verwalten können, indem er neue Accounts anlegt, sie freischaltet und deren Rechte und Rollen verwalten kann. Der Business-User besteht aus einer rein konsumierenden Rolle. Er kann sich Kennzahlen anzeigen und Diagramme ausgeben lassen, auf der er selbst die Granularität der Dimensionen nach Belieben anpassen kann. Ebenso soll ihm eine übersichtliche Darstellung von Gewinnen/Kosten und Deckungsbeitrag angeboten werden. Für ihn ist über all diese Ansichten eine Offline-Ansicht verfügbar. Der Power-User interagiert zu einem Teil wie der Business-User, hat darüber hinaus aber Änderungsrechte in Bezug auf Kennzahlen und Ausgabefunktionen. Er ist somit in der Lage neue Funktionen für Ausgaben zu erstellen.

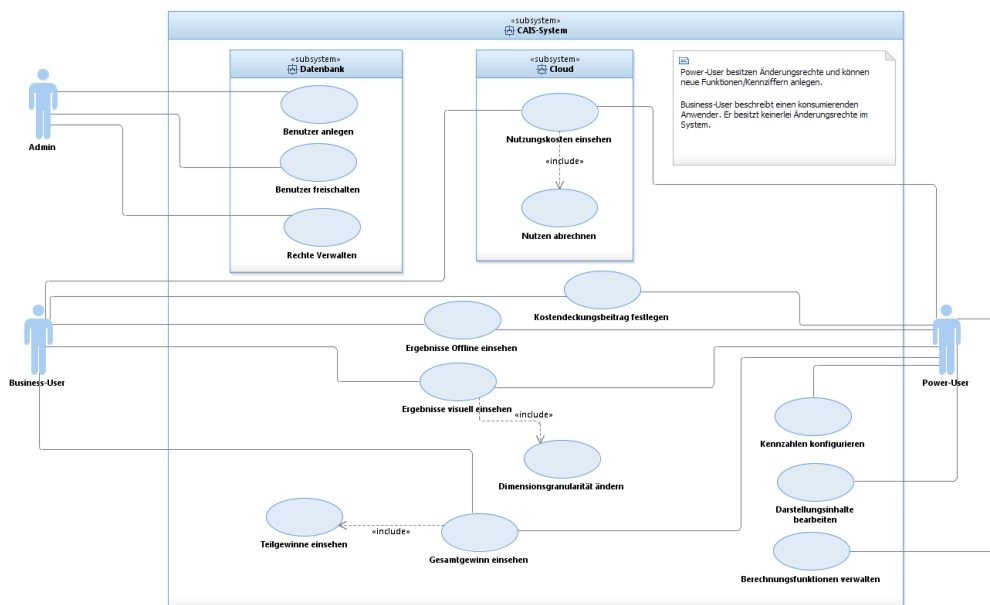


Abbildung 3.2: Use-Case-Diagramm

3.3 Funktionale Anforderungen

ID	Typ	Beschreibung	Priorität
CAIS_REQ101	Funktion	Das System muss Kennzahlen darstellen können.	MUST
CAIS_REQ102	Funktion	Das System muss die Kennzahlen in Abhängigkeit zu unterschiedlichen Dimensionen (Kunde, Quartal, ...) auswerten können.	MUST
CAIS_REQ103	Funktion	Das System muss einen Gesamtgewinn errechnen können.	MUST
CAIS_REQ104	Funktion	Das System muss die Profite pro Kraftwerk und pro Kunde auswerten können.	MUST
CAIS_REQ105	Funktion	Die Kennzahlen sollen über verschiedene Zeiträume angezeigt werden können. (stundengenau, tagesgenau, pro Quartal, ...)	MUST
CAIS_REQ106	Funktion	Das System muss Ergebnisse visuell darstellen können.	MUST
CAIS_REQ106-1	Funktion	Der User soll die Darstellung der Ergebnisse frei wählen können. (bspw. Kreis-/Balkendiagramm)	MUST
CAIS_REQ106-2	Funktion	Das System soll besonders gute Werte (Gewinnzone) grün und besonders schlechte Werte (Zerlustzone) rot markieren.	MUST
CAIS_REQ301	Schnittstelle	Das System muss Testdaten per Bulk Load einlesen können.	MUST
CAIS_REQ401	Funktion	Das System muss eine Accountverwaltung besitzen.	MUST
CAIS_REQ401-1	Funktion	Der User soll seinen eigenen Account nach dem Self Service-Prinzip verwalten können.	MUST
CAIS_REQ401-2	Funktion	Der User muss seinen Accountnamen ändern können.	MAY
CAIS_REQ401-3	Funktion	Der User muss seine Adresse angeben können.	MUST
CAIS_REQ401-4	Funktion	Der User muss seine Adresse ändern können.	MUST
CAIS_REQ401-5	Funktion	Der User muss sein Passwort ändern können.	MUST
CAIS_REQ401-6	Funktion	Das System muss falsche Zugangsdaten erkennen.	MUST
CAIS_REQ402	Funktion	Das System muss eine Rechteverwaltung besitzen.	MUST
CAIS_REQ403	Funktion	Das System muss eine Session-Verwaltung besitzen.	MUST
CAIS_REQ404	Funktion	Die Rechteverwaltung muss eine Admin-Rolle (User anlegen/freischalten) besitzen.	MUST
CAIS_REQ405	Funktion	Die Rechteverwaltung muss eine Power User-Rolle besitzen.	MUST
CAIS_REQ406	Funktion	Die Rechteverwaltung muss eine Business User-Rolle besitzen.	MUST

Tabelle 3.2: Übersicht über die funktionalen Anforderungen

3.4 Nicht-funktionale Anforderungen

ID	Typ	Beschreibung	Priorität
CAIS.REQ201	Technologie	Das System muss skalierbar sein. Dazu muss der mit dem Kunden vereinbarte Skalierungsalgorithmus implementiert werden.	MUST
CAIS.REQ202	Technologie	Das System soll eine Verfügbarkeit von 99,99% aufweisen.	SHOULD
CAIS.REQ203	Technologie	Die Daten eines zu bestimmenden Kunden sollen abgespeichert werden können.	MUST
CAIS.REQ204	Technologie	Das System muss über ein nutzungsorientiertes Abrechnungsverfahren verfügen.	MUST
CAIS.REQ205	Technologie	Das System muss einer Multi-Tenancy-Architektur zugrunde liegen. (Mandantenfähigkeit)	SHOULD
CAIS.REQ206	Technologie	Wartbarkeit	SHOULD
CAIS.REQ207	Technologie	Performance	MUST
CAIS.REQ208	Technologie	Erweiterbarkeit	MUST
CAIS.REQ209	Technologie	Daten eines bestimmten Kunden sollen offline zur Verfügung stehen.	MAY
CAIS.REQ501	Benutzeroberfläche	Die Anwendung muss über alle gängigen Browser erreichbar sein.	MUST
CAIS.REQ502	Benutzeroberfläche	Die grafische Oberfläche der Anwendung muss ein responsives Design aufweisen.	SHOULD

Tabelle 3.3: Übersicht über die nicht-funktionalen Anforderungen

3.5 Risiken

Laut der bekannten Chaos-Studie der Standish-Group [HW15] scheiterten 19% der erfassten IT-Projekte zwischen März 2014 und Februar 2015. Nur 29% Prozent lagen im finanziellen und zeitlichen Rahmen. Daraus ist zu erkennen, dass IT-Projekte zahlreichen Risiken ausgesetzt sind, die auch in der aktuellen Projektgruppe berücksichtigt werden müssen.

Im Folgenden sollen nun typische Risiken für IT-Projekte beschrieben und tabellenartig aufgelistet werden, wie mit diesen Risiken umgegangen werden kann. Ein finanzielles Risiko besteht für CAIS nicht, da es kein Budget gibt. Dennoch besteht die Gefahr, dass das Projekt scheitert oder im Rahmen des strikt festgelegten Zeitrahmens bis zum 31.09.2016 nicht die vom Auftraggeber gewünschte Funktionalität bietet.

Risiken in IT-Projekten lassen sich grob in drei Kategorien einteilen [BR10]. Es gibt planerische, technologische und personelle Risiken.

Planerische Risiken sind Risiken wie die Änderung der Unternehmenspolitik durch ein geändertes Produktportfolio oder interne oder externe regulatorische Maßnahmen wie

beispielsweise Corporate Governance und politische Entscheidungen. Weiterhin sind dies Budgetkürzungen, Prozessrisiken oder der Wegfall von Lieferanten z.B. externer Software. Außerdem könnten sich Kundenwünsche ändern [BR10].

Konkrete Risiken wie Absage durch den Kunden durch eine geänderte Firmenpolitik oder weggefallenen Bedarf bestehen nicht, da es sich um ein studentisches Projekt handelt und die Formalien durch die Universität bereits geregelt sind. Budgetkürzungen sind aufgrund eines fehlenden Budgets ausgeschlossen. Probleme mit Lieferanten sind durch Verwendung externer Systeme z.B. SAP HANA jedoch möglich. Auch Risiken durch die Änderung von Kundenwünschen sind möglich.

Technologische Risiken sind Risiken, die in Zusammenhang mit Hard- oder Software stehen. Das können die Geschwindigkeit von technischen Netzwerken und das Katastrophenmanagement sein. Hardwarerisiken, sowie Netzwerkrisiken bestehen durch die umfangreiche Infrastruktur des OFFIS, nach menschlichem Ermessen nicht. Softwarerisiken können zum jetzigen Zeitpunkt, aufgrund fehlender Auswahl, nicht eingeschätzt werden. Ein Katastrophenszenario ist z.B. bei Datenverlust denkbar [BR10]. Konkrete technologische Risiken könnten sich durch unvollständige oder sich im Laufe des Projektes ändernde Anforderungen ergeben [Hil15].

Personelle Risiken bestehen bei der Entscheidung zur Verwendung internen oder externen Personals, fehlende Akzeptanz des Produktes durch den Kunden, sowie widersprüchliche Prioritäten der Mitarbeiter [BR10]. Externes Personal wird zu Entwicklung im Rahmen des Projektes nicht eingesetzt, daher besteht hier kein Risiko. Das Risiko fehlender Akzeptanz durch den Kunden ist durch den Einsatz eines Product Owners im Team und eines Ansprechpartners auf Seiten des Auftraggebers gering. Widersprüchliche Prioritäten durch die Gruppenmitglieder sind durch die geringe Größe der Gruppe unwahrscheinlich. Es können jedoch Mitglieder aus der Projektgruppe ausscheiden.

In der folgenden Tabelle 3.4 werden die genannten Risiken noch einmal aufgeführt und nach ihrer Art sortiert in Wahrscheinlichkeiten eingeteilt, sowie mögliche Gegenmaßnahmen aufgezeigt und Auswirkungen eingeschätzt. Sie orientiert sich an „Understanding and Mitigating IT Project Risks“ [BR10] von Mike Bailey und Mike Riffel.

Ereignis	Risikoart	Wahrscheinlichkeit	Gegenmaßnahmen	Mögliche Auswirkungen
Probleme mit externen Zulieferern	organisatorisch	mittel	sorgfältige Auswahl	Zeitverlust durch Systemumbau
Änderung von Kundenwünschen	organisatorisch	gering	Product Owner bei Team, enge Absprache mit dem Kunden	komplette Änderung der Architektur

fehlendes Katastrophenmanagement	technisch	hoch	Sicherung von Artefakten durch geeignete Mittel wie z.B. Versionsverwaltung	Änderungen größerer Teile des Systems, geringere Funktionalität
unvollständige Anforderungen	technisch	hoch	Sorgfältige Erhebung der Anforderungen mit geeigneten Mitteln, enge Absprache mit dem Product Owner und dem Kunden	Änderungen größerer Teile des Systems, geringere Funktionalität
sich ändernde Anforderungen	technisch	gering	enge Absprache mit dem Product Owner und dem Kunden	Änderung größerer Teile des Systems, geringere Funktionalität
Ausscheiden eines PG-Mitglieds	personell	mittel	doppelte Besetzung von Funktionen, enger Austausch unter den Gruppenmitgliedern	Verlust von Know How, Verzögerungen, geringere Funktionalität, Qualität

Tabelle 3.4: Mögliche Risiken im Projekt

4. Freigabe/Genehmigung

Bei der Freigabe von Dokumenten, Entwicklungen oder anderen Objekten, die dem Kunden bereitgestellt werden, muss eine Qualitätssicherung erfolgen. Dabei wird zwischen Dokumenten und Code-Review unterschieden.

Wurde ein Dokument erstellt, beispielsweise eine Anleitung, wird diese zunächst vom Autor geprüft. Anschließend wird die Qualität des Dokuments von einem beauftragten Teammitglied sichergestellt. Nach Abnahme durch das Projektteam wird das Dokument dem Kunden bereitgestellt.

Beim Code-Review wird zunächst vom Entwickler geprüft, ob der Code fehlerfrei ist. Anschließend wird sichergestellt, dass ausreichend Kommentare vorhanden sind, die bei späteren Anpassungen hilfreich sind. Weiter ist sicherzustellen, dass der Code gut lesbar ist. Sind diese Kriterien erfüllt, wird der Code von einem beauftragten Teammitglied geprüft. Ist auch diese Prüfung erfolgreich, können Tests durchgeführt werden.

Oldenburg, _____

Unterschrift Auftraggeber

Unterschrift Projektleiter

Literaturverzeichnis

- [BR10] BAILEY, Mike ; RIFFEL, Mike: *Understanding and Mitigating IT Project Risks*. http://www.gfoa.org/sites/default/files/GFR_JUN_10_30.pdf, 2010. – zuletzt aufgerufen am 09.12.2015
- [Hil15] HILL, Peter: *The six big risks to your IT project, and how to avoid them*. http://www.brw.com.au/p/tech-gadgets/the_six_big_risks_avoid_your_them_0K3whbHhxA6GBvLNIWrzuL, 2015
- [HW15] HASTIE, Shane ; WOJEWODA, Stéphane: *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. <http://www.infoq.com/articles/standish-chaos-2015>, 2015. – zuletzt aufgerufen am 09.12.2015

Tabellenverzeichnis

3.1	Beschreibung der Anforderungsarten	6
3.2	Übersicht über die funktionalen Anforderungen	13
3.3	Übersicht über die nicht-funktionalen Anforderungen	14
3.4	Mögliche Risiken im Projekt	16

Abbildungsverzeichnis

3.1	Sequenzdiagramm Kostendeckungsbeitrag errechnen	9
3.2	Use-Case-Diagramm	12

Ergebnis-Protokoll: Interview vom 18.11.2015

Autor: Marcel Kuchler

Datum: 18.11.2015

Beginn: 10.15 Uhr

Ort: U64, OFFIS

Dauer: 35 Minuten

Anwesende: Oliver Norkus, Dennis Vorreiter, Marcel Kuchler

Projekt: Ein Produkt/eine Software, welches über Kennzahlen berichtet: Kennzahlen-Reports.
Die Ergebnisse sollen dabei visualisiert dargestellt werden; Die Darstellung kann frei wählbar sein (z.B. Kreisdiagramm, Balkendiagramm)

Nutzer: 12.000 User parallel; Gliederung in Admin, PowerUser und BusinessUser;
SelfService-User: Name der User änderbar; Adresse änderbar, Passwort änderbar
→ User-Account.
Identity-Management: Session-Verwaltung, Rechte, Rollen, nur Admin kann bspw. neue User hinzufügen

Das System soll auf einer Multi-Tenancy-Architektur basieren: Verschiedene Firmen, verschiedene User nutzen die Cloud;

Abrechnung: Wie lange wurde ein Report angeschaut, bzw. wie oft wurde ein Report angeschaut;
Kostenanalyse pro Benutzung -> Implementierung eines Abrechnungsmoduls
(Vorschläge, welche verschiedenen Möglichkeiten es gibt, anschließend gemeinsames Abstimmen)

Darstellung: Darstellung der Reports mittels Webanwendung über Browser (Responsive Darstellung), um eine Unabhängigkeit der Devices zu gewährleisten.

Ziel: Entwicklung einer Architektur, die standardisiert werden könnte; Referenz-Architektur: Oliver Norkus; Als Ausgangspunkt kann das Komponentendiagramm der Vorgänger-PG dienen.

Auswertungsmethoden: Immer verfügbar; Sensible Daten und weniger sensible Daten sollen verarbeitet und in den entsprechenden Storages gespeichert werden:
Sensible Daten: Private-Cloud-Storage; Weniger sensible Daten in den public-Cloud-Storage

Sicherheit: Datenspeicherung abhängig von der Datenstruktur und den Sicherheitsanforderungen selber: Dabei müssen die Kosten der Services beachtet werden. Dabei sollen die Sicherheitsverhältnisse jedoch in der PG erst einmal auf konzeptioneller Ebene betrachtet werden. (Metadaten der Datentypen beachten beim jeweiligen Einordnen der Speicher-Storages)

Kompatibilität des Produktes: Schnittstellen: Komponenten so definieren, dass die Schnittstellen zwischen den einzelnen Komponenten deutlich wird: z.B. liegt die Storage-Komponenten woanders als die Berechnungskomponente

- Daten:** Die Datenquellen und die Schnittstellen zu diesen sind sehr vielfältig, allerdings sollte vor allem auch eine AD-HOC-Anbindungsmöglichkeit vorhanden sein, sodass schnell Daten via USB-Stick oder HTTP eingespielt werden können.
- Reports:** Die Reports sollen auch mobil erreichbar und einsehbar sein. Entsprechend muss eine Auswertung vorhanden sein, in der deutlich wird, wie der User angebunden ist: Entsprechend werden bei Edge als Verbindung andere Daten aggregiert als im WLAN. Vorladen oder die Datenhaltung in Richtung User bei mobiler Anwendung sollte auch betrachtet werden.
- OLAP-Funktionen:** Roll-Up, Roll-Down, etc. -> Sortierung nach Tag/Monat/Jahr; Es soll möglich sein, neue Kennzahlen zu definieren; z.B. Kennzahl x und y sind vorhanden, nun soll ein Kreuzprodukt aus beiden Zahlen per Drag&Drop im Editor gefunden werden - Funktion nur bei PowerUsern und Admins vorhanden.
- Skalierbarkeit:** Bei 12.000 Usern muss das System skalierter sein: Rechenkapazität muss bei Bedarf automatisch dazu geschaltet oder entsprechend auch wieder abgeschaltet werden: Skalierungsalgorithmus muss gefunden werden
- Offline-Verfügbarkeit:** NOCH OFFEN
- Usability:** Die Funktionalität des Systems steht im Vordergrund!
- Nächste Schritte:** Accounting-Managing-Services finden - TOP5 als Vorschläge
Skalierungsalgorithmus finden - TOP5 als Vorschläge

Ergebnis-Protokoll: Interview vom 20.11.2015

Autor: Dörthe Euhäusen

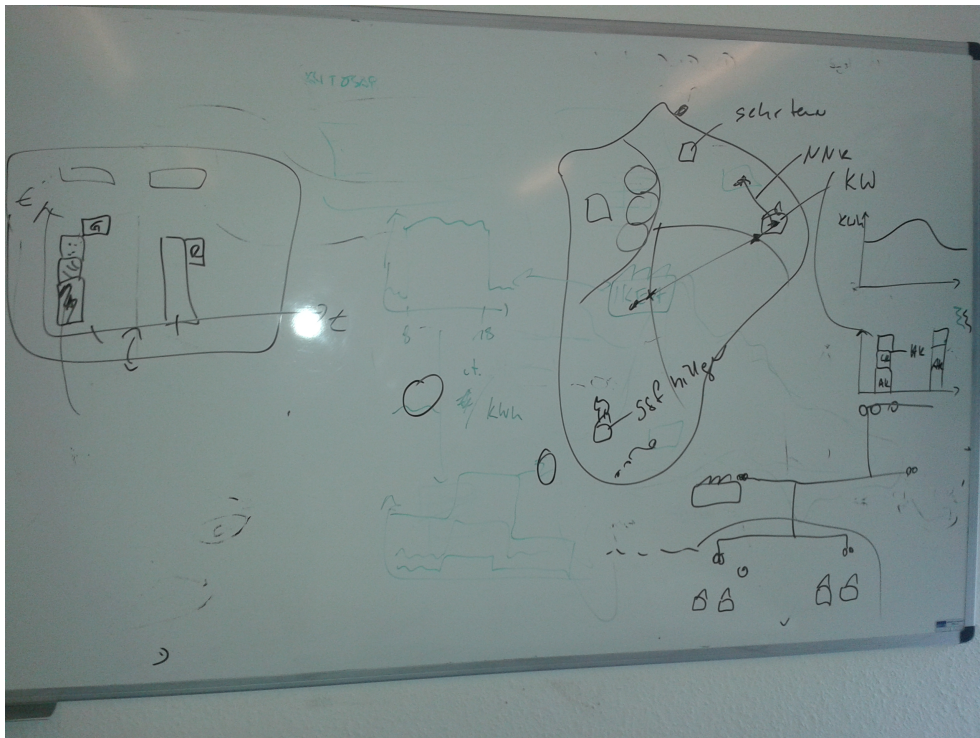
Datum: 18.11.2015

Beginn: 10.25 Uhr

Ort: O100, OFFIS

Dauer: 45 Minuten

Anwesende: Oliver Norkus, Jannick Asmuß, Dörthe Euhäusen, Phillip Kramer, Nils Lutz, Carsten Manske, Alexander Zeyßig



Externe Beteiligte

- Marco Haas (Ceyoniq) wird auch diese Projektgruppe unterstützen
- Eine persönliches Treffen und eine Firmenpräsentation wird stattfinden

Deadline

- 01.04.2016 alpha-Version des Prototypen für das Treffen mit neuem Kunde

Use Case

- Bei unserem Kunden handelt es sich um einen Energieversorger (bspw. EWE), der im B2B-Bereich tätig ist

- Mögliche Kunden des Energieversorgers sind beispielsweise IKEA (sämtl. Standorte Deutschlands)
- Verträge zwischen dem Energieversorger und seinen Kunden sind jeweils 2 Jahre gültig
- In den Verträgen wird ein einheitlicher Preis pro kWh angegeben (gilt für ganz Deutschland)
- Die Entstehungskosten pro kWh können sich jedoch je Region unterscheiden
- Die Entstehungskosten setzen sich zusammen aus Arbeitspreis, Leistungspreis und Netznutzungskosten bzw. Transportkosten
- Das System wird vom Vertrieb für den Kernprozess der Vertragsverhandlung/Preisfindung genutzt (sowohl Bestands- als auch Neukunden)

Der Use Case wird durch externe Ereignisse wie Vertragsverhandlungen und Neukundenakquise angestoßen.

Der Kernprozess lief bisher wie folgt ab:

Der Vertriebsmitarbeiter bereitet sich auf das Kundenmeeting vor, indem er mit Hilfe von diversen Excel-Tabellen und weiteren Informationen ein Angebot erstellt. Dieses Angebot präsentiert er im Meeting dem Kunden. Wenn dieser nicht einverstanden ist, sondern ein Gegenangebot abgibt, muss der Vertriebsmitarbeiter zurück in sein Büro, um ein neues Angebot zu erstellen und dieses zu einem späteren Termin dem Kunden vorstellen (ggf. mehrere Iterationen).
(→ vgl. Rollenspiel bei der Abschlusspräsentation der Vorgänger-PG)

Vom Kunden wird eine Cloud-BI-Lösung gewünscht, auf die der Vertriebsmitarbeiter während eines Kundenmeetings mit seinem mobilen Endgerät zugreifen kann. Es soll ihm anzeigen, ob der vom Kunden angegebene Verbrauch bedient werden kann und ob der vorgeschlagene Preis pro kWh machbar ist. Dabei ist wichtig, dass der verhandelte Preis insgesamt gewinnbringend für den Energieversorger ist. Es kann durchaus sein, dass ein Kraftwerk Verlust macht.

Schnittstellen

- Die Daten sollen nicht für weitere Systeme abgesehen natürlich vom Cloud-BI-System aufbereitet werden
- Die Daten stammen aus heterogenen Quellsystemen
 - SAP EDM
 - SAP PM
 - SAP ERP
 - SCADA
 - Geo-Informationssystem
- Quellsysteme sollen nicht an die Cloud-BI-Lösung angeschlossen werden, da der Kunde die Einführung von inMemory plant
- BI-System ist beim Kunden noch nicht vorhanden
- Webschnittstelle, auf die mobil zugegriffen werden kann, wird benötigt

Benutzer/Skalierbarkeit

- Die im ersten Interview genannte Angabe von 12.000 parallelen Benutzern dient als Anhaltspunkt
- Besonders wichtig ist es, einen passenden Skalieralgorithmus zu finden

Verfügbarkeit

- Verfügbarkeit von 99,999% wird angestrebt, ist aber von den bereitgestellten Ressourcen abhängig und kann daher nicht garantiert werden
- Ideen zur Mindestgeschwindigkeit beim mobilen Einsatz kann ins Konzept mit aufgenommen werden (Stichwort Zwei-Klassen-Internet)

- Backup-Lösung ist für den Prototypen nicht vorgesehen
- Offline-Verfügbarkeit: es sollen nicht alle Daten offline zur Verfügung stehen, aber das Speichern für die Offline-Verwendung von ausgewählten Daten (bspw. die des Kunden, den der Vertriebsmitarbeit zur Vertragsverhandlung trifft)

Nicht-funktionale Anforderungen

- Skalierbarkeit
- Accounting (Protokollierung pro Kostenstelle)
- Hohe Verfügbarkeit

Funktionale Anforderungen

- Entsprechende Kennzahlen sollen über verschiedene Zeiträume (stundengenau, tagesgenau, Woche, Monat, Quartal, ...) angezeigt werden können
- Auswertung pro Region, pro Kunde, ...
- Auswertung darüber, welches Kraftwerk am profitabelsten bzw. am wenigsten profitabel ist
- Auswertung über den Gesamtgewinn, Gewinn pro Kunde, Gewinn pro Kraftwerk, ...
- Darstellung tabellarisch und als Balkendiagramm, mit Highlighting in grün und rot für besonders gute und besonders schlechte Werte

Nächste Schritte

- Recherche und anschließende Vorstellung von fünf Accounting/Pay per use-Services, anschließend Abstimmung mit dem Kunden (Oliver Norkus)
- Recherche und anschließende Vorstellung von fünf Skalierungsalgorithmen, anschließend Abstimmung mit dem Kunden
- Dem Kunden das Lastenheft/aufgenommene Anforderungen vorstellen und gemeinsam eine Priorisierung vornehmen

Skalierungsalgorithmen

Projektgruppe: CAIS

May 12, 2016

1 Einleitung

Im vorliegenden Dokument soll das Problem der Lastverteilung dargelegt werden. Insbesondere die häufig verwendeten Verfahren und Algorithmen zur Lösung dieses Problems sollen bezüglich ihres Nutzens für die Projektgruppe CAIS diskutiert werden. Ziel dieses Dokuments ist es, einen der in diesem Dokument genannten Algorithmen auszuwählen und für die weitere Umsetzung des Projekts zu nutzen.

2 Grundlegende Algorithmen zur Lastverteilung

Im folgenden Kapitel werden die grundlegenden und gängigen Algorithmen zur Lastverteilung beschrieben. Die Problematik der Lastverteilung behandelt das Zusammenspiel von Jobs, bestimmten auszuführenden Aufgaben, und Maschinen, Ressourcen, die diese Jobs ausführen. In Bezug auf dieses Problem wird davon ausgegangen, dass jede Maschine in der Lage ist, jeden Job innerhalb des Problems behandeln zu können. Des Weiteren sei angenommen, dass innerhalb des Problems alle Maschinen und Jobs bereits bekannt sind.

Man unterscheidet zwischen folgenden Algorithmen (nach [F5]):

Random: Die Jobs werden mittels eines Zufallszahlengenerators auf die Maschinen verteilt.

Round Robin: Jeder Job bekommt nacheinander, entsprechend einer vorher festgelegten Reihenfolge, ein gewisses Zeitintervall auf einer bestimmten Maschine. Die Aufgaben werden also nacheinander auf die Maschinen verteilt. Abwandlungen sind die gewichtete Methode, in der die Reihenfolge der Verteilung an selbst definierte Gewichtungen angepasst wird, und die dynamische Methode, in der Realzeit-Probleme miteinbezogen werden, wie beispielsweise die momentane Last einer bestimmten Maschine.

Fastest: Jobs werden immer an die Maschine übergeben, die ihn am schnellsten bearbeiten kann (geringste Antwortzeit beim Server).

Least Connections: Jobs werden immer an die Maschine mit den wenigsten Jobs vermittelt.

3 Server Load Balancing Algorithmen

Im Folgenden sollen die gängigen Server Load Balancing (SLB) Algorithmen beschrieben werden (nach: [fac]). Für die Projektgruppe gelten die Lastverteilungsprobleme im Bezug auf Serveranfragen. Es wird also die Anfragelast durch Clients berücksichtigt.

3.1 DNS based SLB (DNS Round-Robin)

Dieser Algorithmus basiert auf dem Domain Name System (DNS)-Protokoll. Innerhalb dieses Algorithmus wird ein Load Balancer und ein Switch verwendet. Mehrere gleichartige Server werden mittels eines Namens (mit einer Domain) versehen. Die eingehenden Anfragen von Clients werden dann über einen bestimmten Lastverteilungsalgorithmus (beispielsweise Round Robin) an die Server weitergeleitet. Eingehende Verbindungen über `www.pgcais.de`, werden dann auf die Server `www1.pgcais.de`/`www2.pgcais.de` /`www3.pgcais.de` verwiesen.

Diese Variante ist sehr einfach gehalten und bietet somit keinen großen Leistungsumfang. Vorteile sind eben die einfach zu implementierende Struktur und kostengünstige Umsetzung. Dieses Verfahren wird beispielsweise innerhalb des Amazon Web Services verwendet [Amaa].

3.2 NAT based SLB

Ein aufwändigeres Verfahren geht über zwei Netze. Ein Load Balancer dient hier als Vermittler zwischen einem öffentlichen Netzwerk, das die Anfragen von Clients entgegen nimmt und einem privaten Netz, das die Serverarchitektur bildet. Mittels der Network Address Translation (NAT) -Technik wird dann das Kommunikationsprotokoll zwischen Server und Client so manipuliert, dass der Client vorgetäuscht bekommt, immer mit dem gleichen Server zu interagieren.

Um eine geeignete Lastverteilung zu erreichen, werden Serverinformationen vom Load Balancer erfragt. Sie beinhalten die vom Server zur Zeit verwaltende Last oder die Erreichbarkeit des Servers. Ausgefallene Server können so dynamisch aus der Verteilung genommen und nach Neustart wieder eingefügt werden.

Nachteil ist, dass der Load Balancer hier als Flaschenhals fungiert. Die komplette Kommunikation wird von ihm verarbeitet. Der Load Balancer dient

bei diesem Verfahren als "Schein"-Server. Anhand seiner Ressourcen wird also die Leistungsfähigkeit der Lastenverteilung gemessen. Dieses Verfahren ist leistungsfähiger als die "DNS based SLB"-Variante.

3.3 Flat based SLB

Dieses Verfahren bedient sich dem Direct Server Return (DSR) Prinzip, bei dem der Server die Antwort einer Anfrage direkt an den Client sendet und die Kommunikation so nur auf halbem Weg über den Load Balancer läuft, nämlich nur von Client zu Server. Die Anfragen werden also vom Load Balancer innerhalb eines Netzes an einen Server vergeben, der Server antwortet aber direkt dem Client. Dieses Verfahren ist durch die Manipulation des Ethernet-Frames ebenfalls aufwändiger als die "DNS based SLB"-Variante. Vorteil ist die Entlastung des Load Balancers, der auch hier als "Schein"-Server, durch DSR, dient.

4 Entscheidung der Projektgruppe

Zum Zeitpunkt der Erstellung dieses Dokuments kann noch keine eindeutige Entscheidung über den effizientesten Algorithmus zur Lastenverteilung getroffen werden, da die damit verbundene SAP HANA Unterstützung noch nicht vorhanden ist und somit nicht alle Aspekte des Systems genau betrachtet werden können.

In Bezug auf die reinen Anforderungen an das Projekt erscheint es aber durchaus sinnvoll folgende Umsetzung zu nutzen: Da sich die relevanten Antworten des Servers innerhalb dieses Projektes mit Kennziffern und Diagrammdateien beschäftigen, also größere Datenmengen übertragen werden, ist eine Entlastung des Load Balancers zu empfehlen. Die "Flat based"-Variante bietet hier durch die direkte Antwort des Servers an den Client diesen Vorteil. Um die Anfragelast zu verteilen, bietet sich damit eingeschlossen, die Fastest-Methode an, da das Projekt auf schnelle und performante Abfragen abzielt und die aufwändigen Berechnungen innerhalb der HANA durchgeführt werden.

5 Skalierung der Ressourcen

Neben der Verteilung von Anfragelasten, muss zudem auch die Belastungsfähigkeit der einzelnen Ressourcen bei der Skalierung berücksichtigt werden. Wie bereits erwähnt, kann es bei zu großer Anfragelast dazu kommen, dass der Load Balancer als Flaschenhals fungiert. Um dies zu umgehen steht die Überlegung einen weiteren Load Balancer zuzuschalten, auf den die Anfragelast verteilt werden kann. Für diese Art der Skalierung muss aber zunächst

ein geeignetes Vorgehen/ Regeln (Policies) und ein Skalierungsfaktor festgelegt werden, an denen entschieden werden kann, wann eine Ressource zur Einheit hinzugefügt oder abgeschaltet werden kann. Dazu wird ein Ressource Manager benötigt, der eine sogenannte *Auto Scaling Group* verwaltet [Gooa][Amab].

5.1 Skalierungsfaktoren

Skalierungsfaktoren sind Richtwerte, die einem Ressource Manager als Entscheidungskriterium zum Zu- oder Abschalten einer Ressource dienen soll. Im Zusammenhang mit dem Thema richten sich diese Skalierungsfaktoren an Lastinformationen der einzelnen Ressourcen. Es gibt eine Reihe möglicher zu überwachender Faktoren oder auch Metriken (*Metrics*), die das System zur Einschätzung benutzen kann (nach [GH]):

- CPU-Auslastung (%)
- RAM-Auslastung (%)
- Netzwerk-Ausgangslast (MB)
- RAM-Bedarf (MB)
- Verfügbarer Speicher (MB)
- Auslagerungs-Auslastung (%)
- Auslagerungsverbrauch (MB)
- Speicherplatz-Auslastung (%)
- Speicherplatz-Verbraucht (GB)
- Freier Speicherplatz (GB)

Die Nutzung dieser Metriken ist allerdings ebenfalls frei wählbar. Beispiel: Skalierung anhand der CPU-Auslastung ist eine gängige und einfache Variante. Hierbei dient die Auslastung der CPUs als Skalierungsfaktor. So können beispielsweise einzelnen Ressourcen berücksichtigt werden (starte eine neue Ressource, sobald die Auslastung der alten über 80% liegt) oder Aggregationswerte (starte eine neue Ressource, sobald die durchschnittliche Auslastung aller CPUs über 75% liegt) als Anhaltspunkt dienen. In Bezug auf den Load Balancer kann hier auch analog die HTTPS-Anfragelast berücksichtigt werden [Goob][Amab].

Die gängigsten Metriken sind CPU und Netzwerkauslastung, sowie der Speicherplatz. Letzteres ist im Falle der PG aber eher zu vernachlässigen [GH].

Quellen

- [Amaa] AMAZON: *How Elastic Load Balancing Works*. <http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/how-elb-works.html#request-routing>. – 12.4.2016
- [Amab] AMAZON: *Scaling Policy Types*. <http://docs.aws.amazon.com/autoscaling/latest/userguide/as-scale-based-on-demand.html#as-scaling-types>. – 1.5.2016
- [F5] F5, Dec C.: *Intro to Load Balancing for Developers – The Algorithms*. <https://devcentral.f5.com/articles/intro-to-load-balancing-for-developers-ndash-the-algorithms>. – 12.4.2016
- [fac] FACHADMIN.DE: *Server Based Load Balancing*. http://www.fachadmin.de/index.php/Server_Based_Load_Balancing. – 12.4.2016
- [GH] GEOFFREY HOFFMAN, Steven R.: *AUTO SCALING YOUR WEBSITE WITH AMAZON WEB SERVICES (AWS) – PART 2*. <http://www.cardinalpath.com/autoscaling-your-website-with-amazon-web-services-part-2/>. – 1.5.2016
- [Gooa] GOOGLE: *Load Balancing and Scaling*. <https://cloud.google.com/compute/docs/load-balancing-and-autoscaling>. – 1.5.2016
- [Goob] GOOGLE: *Scaling Based on CPU or Load Balancing Serving Capacity*. https://cloud.google.com/compute/docs/autoscaler/scaling-cpu-load-balancing#scaling_based_on_cpu_utilization. – 1.5.2016

Protokolle

Jedes Treffen der Projektgruppe wurde protokolliert, um Entscheidungen schriftlich festzuhalten. Auf den folgenden Seiten sind alle Protokolle in chronologischer Reihenfolge zu finden.

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	12.10.15
Anwesend:	O. Norkus, C. Manske, M. Kuchler, P. Kramer, D. Vorreiter, J. Asmuß, N. Lutz, D. Euhausen
Nicht Anwesend:	M. Tomovic
Protokollführer:	Dennis Vorreiter

Tagesordnungspunkte

1	Organisatorisches
2	Projektrelevantes
3	
4	
5	
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Der wöchentliche Sitzungstermin ist nach der Doodle-Umfrage auf Donnerstag 14.00-16.00 Uhr, beginnend nach der Seminarphase ab dem 12.11.15 festgelegt. - Für die Blockseminarphase wird ein weiterer Termin beschlossen. - Die Moderation der folgenden Sitzungen wird abwechselnd festgelegt. Moderator ist immer der Protokollant der vorherigen Woche. Die Rolle des Protokollanten wird ebenfalls abwechselnd vergeben. - Eine Protokollvorlage wird noch bearbeitet. - Das Protokoll soll maximal 3 Tage nach der jeweiligen Sitzung zur Verfügung stehen. - Die Agenda für die kommende Sitzung soll einen Tag vor Sitzung (also immer mittwochs) allen Teilnehmern zur Verfügung gestellt werden. - Ein E-Mail-Verteiler wird angelegt. 			
2	<ul style="list-style-type: none"> - Für das weitere Vorgehen soll sich jeder Teilnehmer Gedanken über geeignete Projektmanagement-Tools (Jira, Confluence, etc.) sowie eine mögliche Rollenverteilung machen. - Für das Projekt soll jeder Teilnehmer eine Zeiterfassung über geleistete Arbeitszeit führen. 			
3				
4				
5				
6				
7				
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	12.11.15
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter, A. Zeißig
Nicht Anwesend:	M. Tomovic
Protokollführer:	P. Kramer

Tagesordnungspunkte

1	Protokoll der letzten Sitzung
2	Organisatorisches
3	Projektrelevantes
4	
5	
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Terminfestlegung - Blockseminar für Präsentation der Seminararbeiten: Montag, der 23.11.2015, 13.00-17.00 Uhr - Es wurde ein Vorschlag bzgl. der Kommunikation innerhalb der PG gemacht: WhatsApp-Gruppe - Festlegung eines Projektmanagement-Tools: JIRA - Denkanstoß bzgl. Überlegungen zu Rollen, Projektleiter, etc. gegeben - Abschlusspräsentation (alte PG vom 06.11.2015) nochmal aufgegriffen - Überlegungen eines weiteren „Jour Fix Termins“ (wöchentlich) - ausschließlich PG-Mitglieder, ohne Kunde (Oliver) - Raum in OFFIS beantragen für einen längeren Zeitraum, falls frei - Klärungen bzgl. des Projektverlaufs <ul style="list-style-type: none"> o Serverzugang (wollte Oliver abklären) o Scrum-Einsatz, falls ja wie genau? - Anforderungserhebung – Interview mit Kunden (Oliver) <ul style="list-style-type: none"> o 1. Treffen (Mittwoch, der 18.11.2015, 10.15-11.15, U64 (OFFIS)) o 2. Treffen (Freitag, der 20.11.2015, 10.15-11.15, O100 (OFFIS)) o Die Anforderungserhebung mittels Interviews mündet noch vor Weihnachten in einem Lastenheft 			
2	<ul style="list-style-type: none"> - Kommunikation innerhalb der PG <ul style="list-style-type: none"> o Festlegung eines Kommunikationsmittels o Zustimmung des gemachten Vorschlags: WhatsApp-Gruppe? o Weitere Vorschläge? - Festlegung eines weiteren „Jour Fix Termins“ (wöchentlich) [Ohne Kunde, nur PG] <ul style="list-style-type: none"> o Auf einen festen Termin einigen + Abstimmung o Raum in OFFIS für längeren Zeitraum beantragen angelehnt an den zuvor festgelegten Termin - Weitere Punkte können nun angesprochen werden 			
3	<ul style="list-style-type: none"> - Überlegungen zu Rollen, Verantwortlichkeiten, Projektleiter etc. von Mitglieder einholen und diskutieren - Serverzugang (Aktuellen Stand erfragen - Oliver) 			

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

	<ul style="list-style-type: none"> - Scrum-Einsatz <ul style="list-style-type: none"> o Soll Scrum eingesetzt werden? Abstimmung o Falls Ja wie genau soll das aussehen? o Überlegungen der Mitglieder einholen (eine Woche Überlegungszeit gehabt) - Weitere Projektrelevante Punkte können nun angesprochen werden 			
4				
5				
6				
7				
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	19.11.15
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter, A. Zeißig
Nicht Anwesend:	M. Tomovic
Protokollführer:	D. Euhausen

Tagesordnungspunkte

1	Organisatorisches
2	Zweites Interview zur Anforderungsanalyse mit Oliver Norkus (Freitag, 20.11)
3	Rollenverteilung
4	Termine bis zum nächsten PG-Treffen
5	
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Kommunikation innerhalb der Projektgruppe - Offizielle Mitteilungen über den Mailverteiler pg-cais@informatik.uni-oldenburg.de - WhatsApp-Gruppe ausschließlich für kurze Nachfragen/Nebensächliches - Corporate Design/Website - Für Protokolle soll das Template von Marcel verwendet werden - Einheitliche Templates für weitere Dokumente wie beispielsweise Lastenheft und Abschlussbericht sollen erstellt werden - Templates werden auf dem Server bereitgestellt sobald dieser zur Verfügung steht - Eine Website für die Projektgruppe soll erstellt werden - Bereitstellung von Protokoll und Agenda - Das Ergebnisprotokoll soll bis spätestens 12.00 Uhr des folgenden Dienstags über den Verteiler verschickt bzw. auf den Server geladen werden - Die Agenda für das nächste Treffen soll mindestens einen Tag vor dem Termin bereitgestellt werden, damit Vorschläge für weitere Themen noch aufgenommen werden können - Terminfindung für PG-internen Jour fixe (ohne Kunde) - Nils hat bereits Einladungen zu einer Doodle-Umfrage verschickt - Es soll ein Termin freitags (oder alternativ donnerstags) gefunden werden, an dem nach Bedarf ein Treffen mit der Projektgruppe (ohne Kunde) stattfindet 			
2	<ul style="list-style-type: none"> - Das zweite Interview zur Anforderungsanalyse findet wie geplant am Freitag, 20.11. in Raum O100 (OFFIS) statt - Die aufgenommenen Anforderungen vom ersten Interviewtermin (s. Interview-Protokoll) sollen detailliert werden - Der Anwendungsfall für das System muss noch bestimmt werden (Vorschlag: Energie) - Skalierungsalgorithmen und Accounting-/Pay per Use-Methoden werden bereits von anderen Unternehmen eingesetzt, darüber soll sich die Projektgruppe informieren 			

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

	<p>Aufgabe bis zum nächsten PG-Treffen (26.11.): Die aufgenommenen Anforderungen aus beiden Interviews und die (vorläufige) Struktur und Aufbau des Lastenhefts sollen vorgestellt werden</p>			
3	<ul style="list-style-type: none"> - Folgende Bereiche/Rollen wurden benannt <ul style="list-style-type: none"> o Server (verantwortlich: Nils) o Projektleitung o SCRUM o Datenbank o Architektur o Dokumentation o Qualitätssicherung/Test o Front End o Back End - Vorschlag Projektleitung: Es gibt einen Projektleiter und zusätzlich eine weitere Person, die den Projektleiter unterstützt und ggf. vertritt - Vorschlag Rollenverteilung: Für jeden o. g. Themenbereich gibt es einen Experten, der nicht zwingend auch der Verantwortliche ist/die Rolle innehat, sodass das Tauschen der Rollen zwischen den PG-Mitgliedern möglich ist <p>Aufgabe bis zum nächsten PG-Treffen (26.11.): Ein Verantwortlicher für den Server und ein Verantwortlicher für die Projektleitung müssen Oliver genannt werden, auch die anderen Rollen sollen zugeteilt werden</p>			
4	<ul style="list-style-type: none"> - Freitag, 20.11., 10.15 – 11.15 Uhr, O100 (OFFIS) 2. Interviewtermin zur Anforderungsanalyse - Montag, 23.11., 13.00 – 17.00 Uhr, U64 (OFFIS) Blockseminar zur Präsentation der Seminararbeiten (Ablaufplan s. Mail von Oliver) 			
5				
6				
7				
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	26.11.15
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter, A. Zeißig
Nicht Anwesend:	M. Tomovic
Protokollführer:	C. Manske

Tagesordnungspunkte

1	Organisatorisches
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Terminfestlegung – interner Jour fix: Freitags 10:00 bis 12:00 Uhr, nur nach Absprache - Ein interner E-Mail-Verteiler (pg-cais-intern@informatik.uni-oldenburg.de) soll eingerichtet werden [Marcel] - Am Mittwoch den 9.12.2015 wird eine Team-Building-Maßnahme durchgeführt: 19:00 Uhr vor dem Vapiano [Alle] - Carsten und Marcel wurden zu Projektleitung eingeteilt - Vorstellung eines Templates für das Lasterhaft [Marcel] - Weitere Rollen für das Projekt wurden gefunden und wie folgt besetzt: <ul style="list-style-type: none"> o SCRUM (Phillip, Alexander) <ul style="list-style-type: none"> ▪ Admin in Jira; achtet auf Einhaltung der Vorgehensmethodik ▪ Sprintaufstellung ▪ Moderiert Reviews o Datenbank/Server (Nils, Dennis) <ul style="list-style-type: none"> ▪ Administration Datenbank + Server o Architektur (Jannick, Carsten) <ul style="list-style-type: none"> ▪ Überwachung der vorgeschriebenen Architektur in der Softwareentwicklung o Dokumentation (Dörthe, Marcel) <ul style="list-style-type: none"> ▪ Erstellung der Templates ▪ Admin Confluence ▪ Qualitätssicherung Dokumentation o Qualitätssicherung/Test (Dennis, Nils, Jannick) <ul style="list-style-type: none"> ▪ Code Review ▪ Auswahl Testframework ▪ Testfälle, Testszenarien ▪ Durchführung der Tests o PR (Marcel, Carsten) <ul style="list-style-type: none"> ▪ Website erstellen o Protokollant (abwechselnd) <ul style="list-style-type: none"> ▪ Protokoll schreiben ▪ Einladung, Moderation des nächsten PG-Treffens 			

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	10.12.15
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, A. Zeißig
Nicht Anwesend:	D. Vorreiter
Protokollführer:	M. Kuchler

Tagesordnungspunkte

1	Vorstellung Ceyoniq
2	To Dos
3	Seminararbeiten
4	Lastenheft
5	CAIS-Server
6	Sonstiges
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Vorstellung der Firma Ceyoniq Consulting durch Marco Haas: Schwerpunkt von Ceyoniq: Energie- und Informationssicherheit - Allgemeine Informationen über Stromnetze (Hoch-/Mittel-/Niederspannung) - Aktuell: Dezentrale Einspeisung durch PV-Anlagen o.a. - USE-CASE-Vorstellung: Erstellung eines Lastprofils in 15 Minuten-Takt: Es wird gemessen, wie viel Energie ein Wind-/PV-Park einspeist – entsprechend werden Just-in-Time-Berechnungen durchgeführt, dadurch wird auf dem Spot-Markt ein entsprechend guter Preis geboten - Bei der Berechnung der Preise werden virtuelle Kraftwerke (Zusammenschluss von verschiedenen Kraftwerken: PV; Kohle, Wind,...) verwendet: Vorteil: Besserer Preis (Mehr unter www.nextkraftwerke.de) - Kommunikationsweg: B2B mit Sender-/Empfängererkennung; Jedem Zählpunkt ist ein Lastprofil zugeordnet (VDW-Standardlastprofil) 			
2	Welche fachlichen Ausbaustufen gibt es noch: Deckungsbeitragsrechnung (Gewinn/Verlust); Nächste Ausbaustufe: Lastprofile; Was ist noch möglich? – Auch technischer Ebene: (Step-für-Step)			
3	Anregung: Eine weitere Seminararbeit in Q1/Q2-2016 mit Themen, die in den Kontext der Projektgruppe passen			
4	Offene Fragen: <ul style="list-style-type: none"> - In welchem System soll entwickelt werden? Freigestellt! - Standards einhalten, HPI nutzen, oder auf OpenSource setzen? - Korrektur der ersten Version mit Anregungen durch Oliver (Siehe E-Mail); Abgabe des Lastenheftes bis spätestens 23.12.2015 			
5	<ul style="list-style-type: none"> - Einrichtung ist erfolgt, Homepage ist im Aufbau; JIRA wird eingerichtet. - Projekt-Seite erreichbar unter http://cais.uni-oldenburg.de 			
6	Lesenswertes Paper: Einordnung der verschiedenen Dienstarten: Business Intelligence aus der Cloud (http://www.tdwi.eu/publikationen/bi-spektrum/archiv/archivdetailansicht?tx_mwjournal_pi1[pointer]=0&tx_mwjournal_pi1[mode]=1&tx_mwjournal_pi1[showUid]=583)			

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	16.12.15
Anwesend:	Prof. Sauer, O. Norkus, C. Manske, M. Kuchler, P. Kramer, D. Vorreiter, J. Asmuß, D. Euhouse, N. Lutz
Nicht Anwesend:	A. Zeyßig
Protokollführer:	Nils Lutz

Tagesordnungspunkte

1	Protokoll des letzten Jour Fixe
2	Vorstellung Anwendungsfall und Anforderungen
3	Abstimmung Lastenheft
4	Projektserver aktueller Stand
5	Zusätzliche Seminararbeit
6	Sonstiges
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	- Protokoll wurde genehmigt		I	
2	- DB Berechnung muss eingearbeitet werden - Power und Business User klar abgrenzen durch mehr Rechte in der Nutzung des Systems	23.12.15	H	DV
3	- Priorisierung anpassen (1-4 Skala wird entfernt) - Anforderungstypen anpassen - Verhandlungsszenario erarbeiten - EPKs für die Geschäftsprozesse erarbeiten - Oliver liefert Template für Use-Case Beschreibung - Weitere Fachliche Ausbaustufen definieren	23.12.15 15.01.16	H	PK
4	- Server ist online - JIRA teilweise eingerichtet (Task fehlt noch als Issue) - Website fast fertig		I	NL, MK
5	- generell kein Problem - können auch zu unterschiedlichen Zeitpunkten geschrieben werden - wenn die PG Bedarf am Thema hat		I	
6	- Anmeldung zur Klausur im StudIP erst nächstes Semester nötig			
7				
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	11.01.2016
Anwesend:	O. Norkus, C. Manske, M. Kuchler, P. Kramer, J. Asmuß, N. Lutz, D. Euhausen
Nicht Anwesend:	A. Zeyßig, D. Vorreiter
Protokollführer:	Jannick Asmuß

Tagesordnungspunkte

1	Protokoll des letzten Jour Fix
2	Abstimmung Anwendungsfall
3	Skalierungsalgorithmen
4	Testdaten
5	Form der PG Dokumente
6	Projektserver
7	Sonstige
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	-Protokoll wurde genehmigt		I	
2	-Abstimmung wurde vertagt -Power und Businessuser sollen durch unterschiedliche Berechtigungen abgegrenzt werden	14.01	H	DV
3	-Festlegung als Seminarthema		I	NL
4	-Als CSV verfügbar -per Mail von Oli anfordern		I	CM
5	-Anforderungen ca. 8 Seiten -Dokumentation kann nach Präsentation eingereicht werden -Abschlussbericht via Latex anfertigen		I	
6	-Issues angelegt		I	NL
7	-Diskussion Nomenklatur -Abstimmung vertagt		H	JA
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	14.01.16
Anwesend:	Prof. Sauer, O. Norkus, C. Manske, M. Kuchler, P. Kramer, D. Vorreiter, J. Asmuß, N. Lutz, M. Haas
Nicht Anwesend:	A. Zeyßig
Protokollführer:	D. Vorreiter

Tagesordnungspunkte

1	Genehmigung des letzten Protokolls
2	Abstimmung über Use-Case Diagramm
3	Abstimmung über Nomenklatur
4	Testdaten
5	Aktueller Zustand Projekt-Server
6	Sonstiges
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Das Protokoll der letzten Sitzung wurde besprochen und genehmigt. 		I	
2	<ul style="list-style-type: none"> - Eine klare Trennung zwischen Business-User und Power-User innerhalb des Use-Case Diagramm wird zur nächsten Sitzung eingepflegt. - Änderungen des Diagramms wird mit weiteren Änderung in das Lastenheft eingepflegt und zum Dienstag an Oliver übergeben. - 	12.01.16	H	DV
3	<ul style="list-style-type: none"> - Nomenklatur wird beim Internen Jour Fixe besprochen. 		H	
4	<ul style="list-style-type: none"> - Die Testdaten wurden von Oliver bereitgestellt und besprochen. - Die Testdaten sollen noch einmal in Bezug zu den geführten Interview gesetzt werden, damit die Cases klar gemacht werden können. 		I	
5	<ul style="list-style-type: none"> - Die Webseite wurde von Marcel veröffentlicht. Sie ist unter cais.uni-oldenburg.de zu erreichen. - Die Inhalte der Seite werden zur nächsten Woche ausgearbeitet. - Alle Arbeitspakete wurden in Jira angelegt und können dort eingesehen werden. 	21.01.16	I	MK
6	<ul style="list-style-type: none"> - Die Abgabetermine für die nächsten Seminararbeiten (Skalierungsalgorithmen & Ticketsystem in Jira) wurde auf den 4.2.2016 gelegt. - Erster Architekturansatz wurde von Nils vorgestellt und besprochen. Eine ausgearbeitete Form mit Positionspapier wird zur nächsten Sitzung eingereicht. - Für eine tiefer eingehende Architektur wird von Oliver ein Architektur-Workshop angeboten, zu dem ein Termin abgestimmt werden muss. - Die Sprints sollen skizziert werden (bspw. in einer Roadmap) - VMs, Cloud und SAP HANA-Instanz werden bereit gestellt. Die Cloud Instanz voraussichtlich erst ab dem 4.1.2016 zur Verfügung stehen. - Alpha-Version wird somit zunächst mit einer simplen Datenbank realisiert. - Die Vorstellung zu den Zielen der Alpha-Version werden auf die nächste Sitzung verschoben. 	21.01.16	I	NL, JA
7				

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	O100, OFFIS
Datum:	05.02.16
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	-
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Genehmigung des letzten Protokolls
2	Lastenheft
3	Architektur / Pflichtenheft
4	Mockups
5	Roadmap und Ausbaustufen
6	Notwendige Hardware
7	Seminararbeiten
8	Sonstiges
9	
10	
11	

	<ul style="list-style-type: none"> + Weitere Seminarthemen wurden aufgestellt (Themen sind in Confluence nachlesbar) 1) Abrechnungsalgorithmus (Ende Beta / Anfang Final) 2) Sicherheitskonzepte / Firewall (Final) 3) SAP U5 (noch unbekannt) 4) HANA-Plattform (Ende Beta) 5) Vergleich und Zusammenfassung aktuelle und vorherige Projektgruppe (Final) 6) Integration von Datenbanken (Konzept) (Release-unabhängig) 		I	
8	<ul style="list-style-type: none"> - Sonstiges + Was für Parameter muss eine BI-Cloud mitbringen, um ... (Kontext fehlt) <p>Momentan wird dazu eine Masterarbeit erarbeitet, angedacht ist die Teilnahme an deren Abschlusspräsentation. Der Termin ist noch zu erfragen.</p>	11.02.16	H	

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U61, OFFIS
Datum:	25.02.2016
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	A. Zeyßig
Protokollführer:	Dörthe Euhausen

Tagesordnungspunkte

1	Statusbericht
2	Abschlussbericht
3	Verteidigung der Masterarbeit von Simon Küspert
4	Antrag HPI Future SOC Lab
5	Seminararbeiten
6	Vorstellung alpha Version
7	Nächstes Jour fixe
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Vorstellung des aktuellen Projektstands <ul style="list-style-type: none"> o Oliver hat keine weiteren Anmerkungen mehr zum Lastenheft o alle Teams arbeiten sich zurzeit in ihre Arbeitsgebiete ein und können bereits Ergebnisse vorweisen o Login Screen wurde Oliver präsentiert 	-	I	-
2	<ul style="list-style-type: none"> - im Abschlussbericht soll kein Grundlagenwissen vermittelt werden (bspw. „Was ist REST?“), sondern Entscheidungen begründet werden (bspw. Vor- und Nachteile von REST, Alternativen zu REST, Begründung, weshalb wir uns für REST entschieden haben) - die Gliederung des Abschlussberichts kann sich an der der Vorgänger-PG orientieren 	-	I	-
3	<ul style="list-style-type: none"> - Diskussion der Verteidigung zur Masterarbeit „Regelbasierte, automatisierte Verteilung von Anfragen an Cloud Storage-Dienste zur Optimierung von Kosten-, Leistungs- und Sicherheitsfaktoren unter Verwendung einer Abstraktionsschicht“ (23.02.16) - im Ausblick (→ Abschlussbericht) können wir darauf eingehen, wie das vorgestellte Konzept auf unser Projektthema angewendet werden kann 	-	I	-
4	<ul style="list-style-type: none"> - Oliver wird seinen Anteil zum Antrag an das HPI vorbereiten - der Antrag muss bis zum 16.03. an das HPI gesendet werden, als interne Deadline wurde der 05.03. gesetzt 	-	C	O. Norkus
		05.03.16	C	PG CAIS, O. Norkus
5	<ul style="list-style-type: none"> - die Seminararbeiten von Nils (Skalierungsalgorithmen) und von Jannick (Issue Tracking-System, JIRA und Nomenklatur) werden am 14.04. vorgestellt 	14.04.16	I	N. Lutz, J. Asmuß
6	<ul style="list-style-type: none"> - die Vorstellung der alpha-Version findet am 30.03. statt, Uhrzeit und Raum werden noch bekanntgegeben 	30.03.16	I	PG CAIS
7	<ul style="list-style-type: none"> - die Jour fixe am 03.03. und am 10.03. fallen aus, der nächste Jour fixe findet am 07.03. ab 14.00 Uhr in O100 statt 	07.03.	I	PG CAIS
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U61, OFFIS
Datum:	25.02.2016
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	A. Zeyßig
Protokollführer:	Dörthe Euhausen

Tagesordnungspunkte

1	Statusbericht
2	Abschlussbericht
3	Verteidigung der Masterarbeit von Simon Küspert
4	Antrag HPI Future SOC Lab
5	Seminararbeiten
6	Vorstellung alpha Version
7	Nächstes Jour fixe
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Vorstellung des aktuellen Projektstands <ul style="list-style-type: none"> o Oliver hat keine weiteren Anmerkungen mehr zum Lastenheft o alle Teams arbeiten sich zurzeit in ihre Arbeitsgebiete ein und können bereits Ergebnisse vorweisen o Login Screen wurde Oliver präsentiert 	-	I	-
2	<ul style="list-style-type: none"> - im Abschlussbericht soll kein Grundlagenwissen vermittelt werden (bspw. „Was ist REST?“), sondern Entscheidungen begründet werden (bspw. Vor- und Nachteile von REST, Alternativen zu REST, Begründung, weshalb wir uns für REST entschieden haben) - die Gliederung des Abschlussberichts kann sich an der der Vorgänger-PG orientieren 	-	I	-
3	<ul style="list-style-type: none"> - Diskussion der Verteidigung zur Masterarbeit „Regelbasierte, automatisierte Verteilung von Anfragen an Cloud Storage-Dienste zur Optimierung von Kosten-, Leistungs- und Sicherheitsfaktoren unter Verwendung einer Abstraktionsschicht“ (23.02.16) - im Ausblick (→ Abschlussbericht) können wir darauf eingehen, wie das vorgestellte Konzept auf unser Projektthema angewendet werden kann 	-	I	-
4	<ul style="list-style-type: none"> - Oliver wird seinen Anteil zum Antrag an das HPI vorbereiten - der Antrag muss bis zum 16.03. an das HPI gesendet werden, als interne Deadline wurde der 05.03. gesetzt 	-	C	O. Norkus
		05.03.16	C	PG CAIS, O. Norkus
5	<ul style="list-style-type: none"> - die Seminararbeiten von Nils (Skalierungsalgorithmen) und von Jannick (Issue Tracking-System, JIRA und Nomenklatur) werden am 14.04. vorgestellt 	14.04.16	I	N. Lutz, J. Asmuß
6	<ul style="list-style-type: none"> - die Vorstellung der alpha-Version findet am 30.03. statt, Uhrzeit und Raum werden noch bekanntgegeben 	30.03.16	I	PG CAIS
7	<ul style="list-style-type: none"> - die Jour fixe am 03.03. und am 10.03. fallen aus, der nächste Jour fixe findet am 07.03. ab 14.00 Uhr in O100 statt 	07.03.	I	PG CAIS
8				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	O100, OFFIS
Datum:	07.03.16
Anwesend:	O. Norkus, J. Asmuß, D. Enhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	A. Zeyßig
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Genehmigung des letzten Protokolls
2	Antrag Future SOC Lab
3	Aktueller Projektstand
4	Nächsten Arbeitsschritte
5	Statusbericht / Reporting
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	– Das Protokoll der letzten Sitzung genehmigt.	-	I	
2	– Die Deadline vom 05.03.16 wurde eingehalten. Oliver hat den Antrag nochmals angepasst. Es gab zu dessen Anpassung keine Anregungen bzw. Änderungswünsche seitens der PG. PG hat den Text von Oliver geprüft. Das Absenden unter Einhaltung der Abgabefrist, den 16.03.16, wird in den kommenden Tagen erledigt.	16.03.16	C	PG CAIS, O. Norkus
3	– Alle Teams (Frontend, Backend, Architektur) arbeiten ihre to-do's ab und können weitere Informationen zu ihren Tätigkeiten bzw. Ergebnisse vorweisen. – Unsere Lösung muss Daten auch ohne SAP HANA verarbeiten können	-	I I	
4	– Alle Teams schilderten ihr weiteres Vorgehen bzw. Arbeitsschritte.	-	I	
5	– Für den zukünftigen Jour-Fixe hat jedes Team einen Statusbericht abzugeben. Dieser wird dem Reporting dienen und Auskunft darüber geben, was bei den Teams gut bzw. schlecht läuft und wird für den darauffolgenden Jour-Fixe als Vergleichsgegenstand herangezogen. Jedes Team hat seinen Bericht (elektronisch) bis spätestens den vorherigen Tag des jeweiligen Jour-Fixe bei Phillip Kramer abzugeben.	-	H	PG CAIS, P. Kramer

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U61, OFFIS
Datum:	24.03.16
Anwesend:	O. Norkus, Marco Haas, J. Asmuß, D. Enhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Genehmigung des letzten Protokolls
2	Statusbericht / Reporting
3	Alpha-Vorstellung - erste Überlegungen bzgl. Agenda und Planung
4	Rückfragen zu dem jeweiligen Report (Skalierungsalgorithmus und Billing-Model)
5	Sonstiges
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Das Protokoll der letzten Sitzung wurde genehmigt. 	-	I	
2	<ul style="list-style-type: none"> - Darstellung der Fortschritte der jeweiligen Teams durch Berichtsampeln wurde als gelungen empfunden und es sind folgende Anregungen einzupflegen bzw. zu berücksichtigen: <ul style="list-style-type: none"> • Es soll zukünftig erkennbar sein, ob kritische Arbeitsschritte der jeweiligen Teams, auch Auswirkungen auf andere Teams haben. Eine Ergebnisdarstellung der Fortschritte soll nicht nur für einzelne Teams sondern zusätzlich auch Team-übergreifend für die Projektgruppe als Ganzes (Statusampel für gesamte Team) umgesetzt werden. • Einfachheitshalber soll der Animations-Umfang bei der Vorstellung der Team-Fortschritte verringert werden. - Nach aktuellem Stand gibt es keine gesetzlichen Anforderungen bzw. internationale Gesetzgebungen, die wir bei der Entwicklung der Software einzuhalten haben. 	-	I	PG CAIS, P.Kramer
3	<ul style="list-style-type: none"> - <u>Anregungen zum Inhalt & Ablauf der Vorstellung:</u> <ul style="list-style-type: none"> • Die Verwendung von Slides und Storyboards bietet sich optimal an, um die Vorstellung zu erleichtern. Über die gesamte Vorstellung hinweg muss ein roter Faden zu erkennen sein. Folgende Inhalte wurden uns angeraten in die Vorstellung einzubringen: <ul style="list-style-type: none"> ○ Motivation (Worum geht es? Wie haben wir uns dem Thema genähert?) ○ Problembeschreibung (Mehrwert) ○ Umsetzung ○ Zielsetzung ○ Fehlwege und Entscheidungen während des Projektverlaufs ○ Lernkurve des Projektteams (Ergebnisse unserer Sprint-Retrospektive einbringen) ○ Demo (Live Präsentation) ○ Ausblick • Die Vortragsdauer ist auf maximal 45 Minuten zu beschränken. Anschließend folgt eine 15 minütige Frage-Antwort Runde. - <u>Organisatorisches:</u> <ul style="list-style-type: none"> • Es werden <u>zwei</u> Beamer für die Vorstellung verwendet. <u>Ein</u> Beamer ist noch zu besorgen/beantragen. Die Beantragung kann ausschließlich durch Oliver erfolgen. Oliver ist hinsichtlich der Besorgung eines zweiten Beamers zu informieren. 	30.03.16	I	PG CAIS
		30.03.16	C	O.Norkus, PG CAIS

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

4	– Für den jeweiligen Report (1)„Skalierungsalgorithmus“ & (2)„Billing-Modell“ soll kein Grundlagenkapitel verfasst werden. Es soll bspw. keine Definition dazu geben wie ein Algorithmus im Allgemeinen definiert ist. Jedoch ist eine Definition unseres Algorithmus zu liefern.	(1)15.04.16 (2)15.05.16	H	PG CAIS
5				

Ergebnisprotokoll

Thema:	Wöchentlicher Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	21.04.16
Anwesend:	O. Norkus, P. Kramer, M. Kuchler, D. Vorreiter,
Nicht Anwesend:	J. Asmuß, D. Euhausen, N. Lutz, C. Manske
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Präsentation Skalierungsalgorithmus
2	Statusbericht / Reporting
3	Anforderungen Product Backlog
4	Terminfindung - Workshop „Mockups“ & „Geschäftsprozess“
5	Sonstiges
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	– Entscheidung: Fastest – Flat based – <u>Die Ausarbeitung ist zu überarbeiten und zeitnah nachzureichen</u> <ul style="list-style-type: none"> • Über einen Master-Loadbalancer nachdenken • Ausarbeitung soll Verteilungskonzept sowie Virtualisierungskonzept enthalten • Beantwortung der Frage „Was passiert wenn der Load-Balancer überlastet ist, d.h. zu viele Anfragen gestellt werden?“ • Mehr Quellen angeben • Weitere Quellen werden von Oliver N. nachgereicht 	-	I H	PG CAIS, O. Norkus
2	-	-	-	-
3	– Termin für Verhandlungsworkshop „Anforderungen/Product Backlog“ nur mit Oliver N. festlegen <ul style="list-style-type: none"> • Per Doodle-Umfrage Terminvorschlag machen 	-	C	PG CAIS
4	– Termin für Workshop „Mockups“ & „Deckungsbeitrag“ mit Oliver N. und eventuell Marco H. festlegen <ul style="list-style-type: none"> • Per Doodle-Umfrage Terminvorschlag machen – Deckungsbeitrag ist nur mit Marco zu klären	-	C	PG CAIS
5	– Test-Daten mit Marco H. klären – BETA-Vorstellung erfolgt nur intern mit Marco H. und Oliver N. – Bzgl. der HPI Ressourcenverteilung ist der Verzicht auf SSDs entschieden worden – Laut Oliver N. steht uns die SAP HANA vor dem 15.05 zur Verfügung – Überlegen ob ein gesondertes Treffen mit Marco stattfinden soll	-	I	PG CAIS

Ergebnisprotokoll

Thema:	Jour-Fixe mit Product Owner und Stakeholder
Ort:	OFFIS
Datum:	04.05.16
Anwesend:	O. Norkus, M. Haas, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, D. Vorreiter,
Nicht Anwesend:	C. Manske
Protokollführer:	Phillip Kramer

Tagesordnungspunkte


1	Workshop - Deckungsbeitragsrechnung
2	Workshop - Mockups
3	Sonstiges
4	
5	
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Für die Berechnung des Deckungsbeitrags sind die Testdaten der uns zur Verfügung gestellten CSV-Dateien (vier) zu verwenden - Arbeit wird in kwh gemessen (15min Wert) - Abgerechnet wird in kw (kwh multiplizieren) - Ein Lastgang hat eine Dauer von einem Monat <ul style="list-style-type: none"> o Nach diesem Monat erfolgt ein weiterer Lastgang und dazu ein Preis-Lastgang - Vier Lastgänge die multipliziert werden müssen (deren 15 min Werte) <ul style="list-style-type: none"> o Aus dieser Berechnung kann die Marge erkannt werden - Der Preis ist für das ganze Jahr anzubieten, da die Vertragslaufzeit des Kunden ein Jahr beträgt <ul style="list-style-type: none"> o Es erfolgt unabhängig von der Vertragslaufzeit eine monatliche Abrechnung mit dem Kunden - Weitere Informationen sind von den im Jour-Fixe erarbeiteten Skizzen/Notizblätter aus dem Anhang zu entnehmen 	-	I	PG CAIS
2	<ul style="list-style-type: none"> - Die Grafik, welche die Lastgänge und Marge etc. anzeigt, ist anzupassen. Sie soll größer dargestellt werden - Es ist die Funktion „filtern“ zu implementieren - Die Anwendung ist sowohl Mobil- als auch Desktopfähig - Die Grafik muss kennzeichnen, was es aktuell anzeigt <ul style="list-style-type: none"> o Hinzunahme einer Legende - Es ist ein Self-Service zu implementieren, welcher es ermöglicht die Diagrammart zu ändern - Es müssen OLAP-Funktionen implementiert sein - Bei Bestandskunden ist die Anzeige der Marge sichtbar zu machen - Bei Neukunden ist anzuzeigen, wie viel maximal auf den aktuellen Preis aufgeschlagen werden kann 	-	I	PG CAIS
3	<ul style="list-style-type: none"> - Zusätzliche Komplexität einbringen, die mit dem Szenario weniger zu tun hat, jedoch die SAP HANA fordert <ul style="list-style-type: none"> o Jahreswerte, 10-Jahreswerte o Testdaten erweitern (u.a. Anzahl erhöhen) - Für den Beta-Prototypen muss der Verteilungsalgorithmus laufen und der Ressource(n) Manager enthalten sein 	-	I	PG CAIS

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

	<ul style="list-style-type: none">- Für die Abschlusspräsentation ist zu überlegen, wie das Publikum Zugriff z.B. über WLAN auf unsere vorgestellte Anwendung erhält (evtl. über PublicIP)- Es ist zukünftig zu berücksichtigen wie sich die Antwortzeiten verhalten- Es sind drei Ebenen gedanklich zu berücksichtigen<ul style="list-style-type: none">o Technische Umsetzungo Kundensichto Vertriebsicht- Optionaler Ausblick<ul style="list-style-type: none">o Pay-per-usageo Gastaccount, der nur den eigenen Verbrauch des Kunden anzeigt, damit dieser seine Last verlagern kann um keine hohen Lastspitzen (Umkosten/Aufpreis) zu habeno Überlegung, wie Daten von Neukunden erfasst werden können (Gewerbe etc.)o Statt einen Lastgang pro Kunde (bisher) soll zukünftig eine Szenario-basierte Evaluation erfolgeno Es wird ein Abnahmeszenario durch Marco H. und Oliver N. geben			
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--

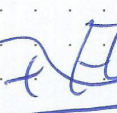
KDBS offers

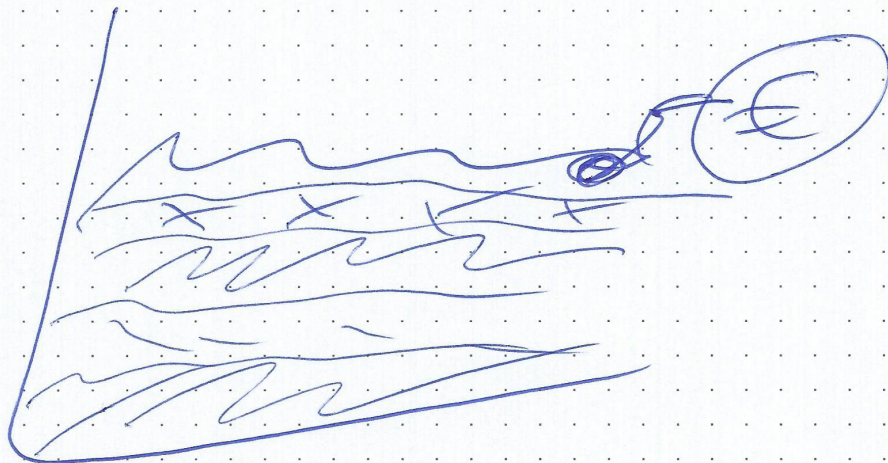
①  kur kWh 15 min

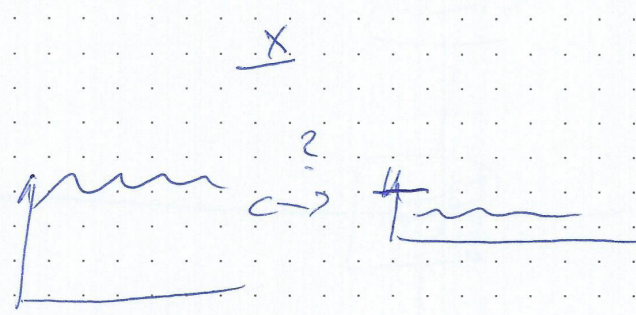
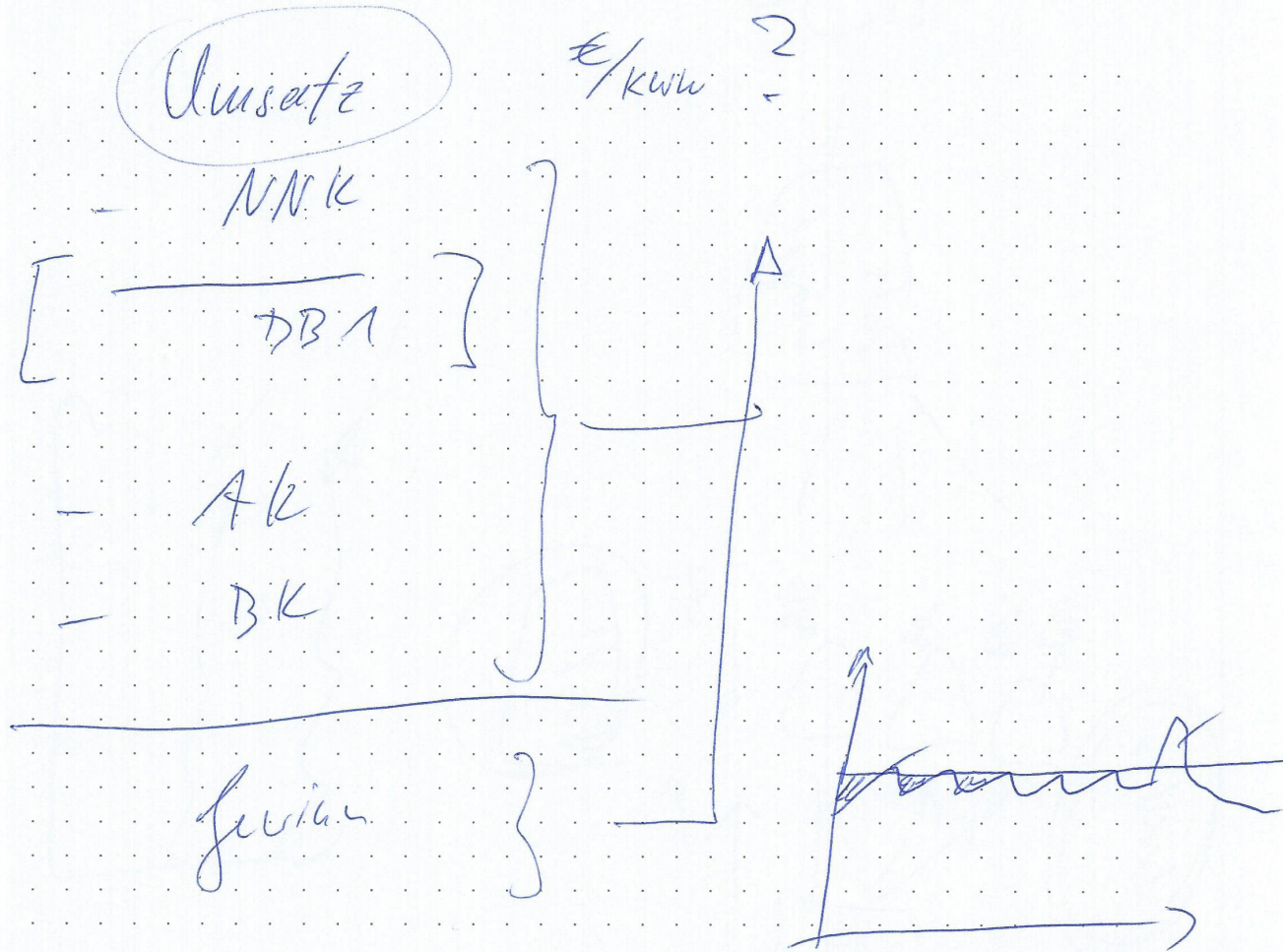
② kur + NVA 15 min

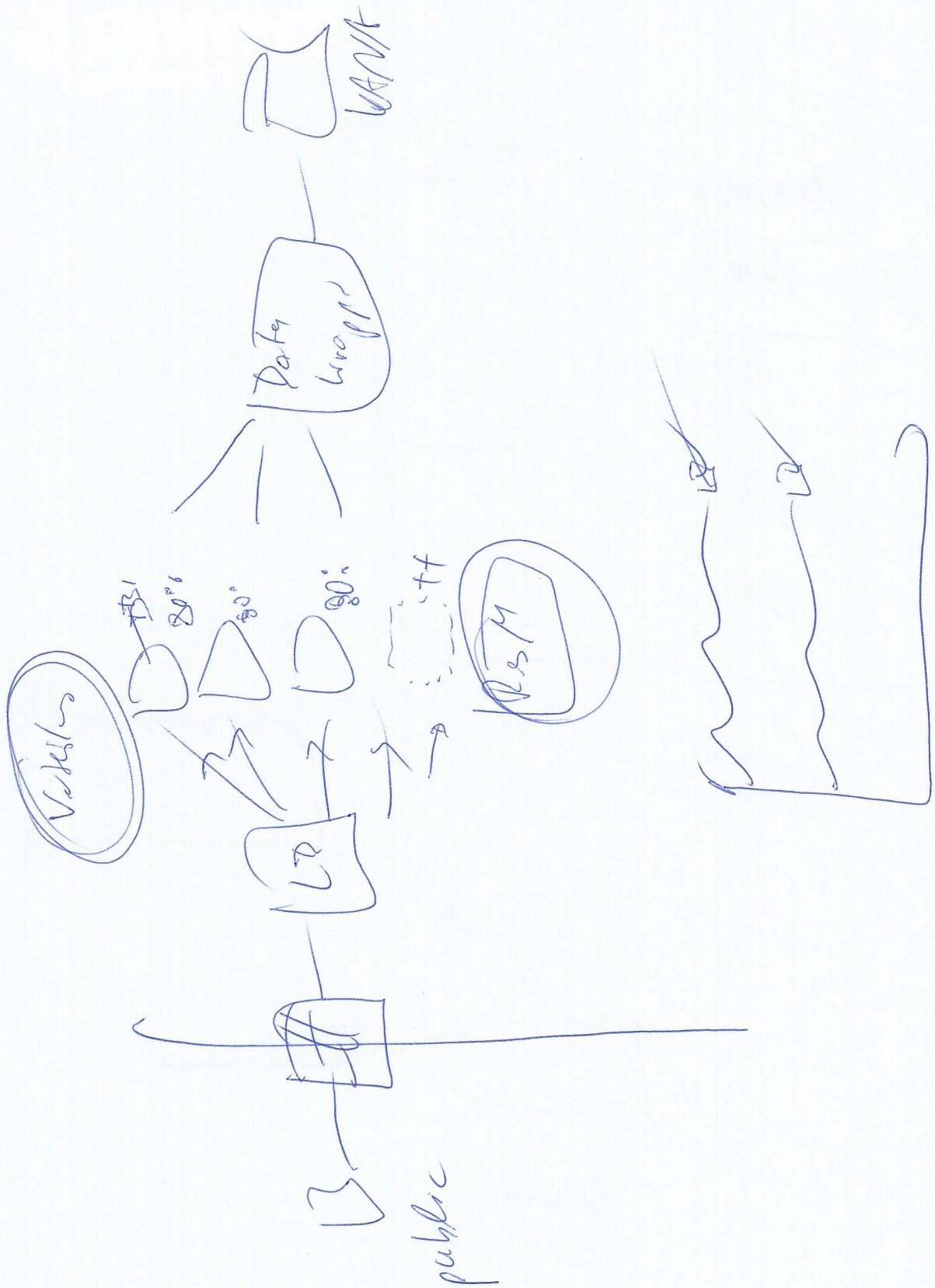
③ kur + Betrieb Krefpakt 15 min

④ kur + Produktiv Krefpakt 15 min

~~④b kur +  15 min~~

~~~~





Ergebnisprotokoll

Thema:	Jour Fixe mit Product Owner
Ort:	U64, OFFIS
Datum:	13.07.16
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Vorstellung & Besprechung - Prototyp Beta-Version
2	Projektmanagement
3	Besprechung - Entwurf Abschlussdokumentation
4	Besprechung - Entwurf Abschlusspräsentation
5	Sonstiges
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - <u>Backend:</u> <ul style="list-style-type: none"> o Derzeit ist ein Round Robin-Algorithmus zur Lastverteilung implementiert o Die Lastverteilung nutzt einen Algorithmus der auf gewichtete Verbindungen aufbaut o Web-Tokens (stateless) statt Session-Management - <u>Frontend:</u> <ul style="list-style-type: none"> o Schwerpunkt bei der Entwicklung von der alpha zur beta lag auf „Visualisierung“ und „Usability“ o Das Launchpad samt Kacheln für den Zugriff auf Apps wurde verworfen o Alle Funktionen befinden sich nun kompakt in einer einzelnen Applikation o Einbau des Self-Services o Deckungsbeitrag ist noch nicht implementiert o Deckungsbeitrag wird im Backend berechnet o Anbindung vom Backend an das Frontend wird vorgenommen - <u>API:</u> <ul style="list-style-type: none"> o API stellt alles bereit und ist lauffähig o Das System kann beide Möglichkeiten, sowohl die HANA als auch Mongo-DB, nutzen o Es sind Anfragen zu stellen und die Daten in die HANA zu integrieren 	-	I	PG CAIS
2	<ul style="list-style-type: none"> - Wechsel von Scrum zum Wasserfallmodell <ul style="list-style-type: none"> o Modifikation: Weg von Scrum und hin zum Wasserfallmodell o Elemente des Scrum-Prozesses wie der Jour-Fixe werden beibehalten o Es ist eine Liste, die alle noch zu erledigenden Schritte bzw. Aspekte enthält, ist zu erstellen und abzuarbeiten - Die Begründung für den Rückgang ist in der Abschlusspräsentation zu erwähnen und im Abschlussdokument zu verarbeiten <ul style="list-style-type: none"> o Warum Wechsel? o Empfehlung für die nächste Projektgruppe? 	- 31.08.16	I H	PG CAIS
3	<ul style="list-style-type: none"> - Besprechungsgrundlage ist der Vorschlag unserer Gliederung der Abschlussdokumentation und Abschlusspräsentation (An das Protokoll angehängt sind die besprochene vorläufige Gliederung des Abschlussdokuments (Bild 1) sowie die der Abschluss- 	31.08.16	H	PG CAIS

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

	<ul style="list-style-type: none"> präsentation (Bild 2)) – Kapitel 1.4: „Aufbau der Arbeit“ hinzufügen – Kapitel 4: Umbenennung in „Cloud“ oder ähnliches – Kapitel 4.1: Umbenennung in „Grundlagen zur Cloud“ oder ähnliches – Kapitel 4 und 5 sind vor 3 anzuordnen – Kapitel 8: „Lessons Learned“ hinzufügen – Kapitel 2: Internetauftritt der PG (Webseite) ist in einem bereits bestehenden oder neuen Unterkapitel einzubringen – Kapitel 6: Unterteilung zwischen technischer- (6.1) und fachlicher (6.2) Lösung – Es ist ein Kapitel/Unterkapitel einzubauen (evtl. im Kapitel 8), indem Erweiterungen/Ausbaustufen enthalten sind, worauf nachfolgende Arbeiten wie Projektgruppen oder Masterarbeiten aufgebaut werden können – Es ist ein Deployment-Diagramm zu erstellen – Es ist ein UML-Diagramm der Architektur inkl. Methoden und Klassen zu erstellen – Statt den Load-Balancer als Abbildung zu veranschaulichen, ist die formale Lösung in Form von Pseudo-Code zu wählen – Abgrenzung zu der vorherigen Projektgruppe <ul style="list-style-type: none"> ○ Haben wir Ergebnisse der vorherigen Projektgruppe genutzt? ○ Falls ja, was und warum haben wir es genutzt? ○ Welche bzw. was von unseren Anforderungen geht über die Anforderungen der vorherigen Projektgruppe hinaus? 			
4	<ul style="list-style-type: none"> – Es ist eine Präsentation inklusive der Beantwortung von Fragen vom Umfang 90 Minuten vorgesehen – Wir haben uns auf Fragen, die im Zuge der Alpha-Vorstellung aufkamen, vorzubereiten – Die Abschlusspräsentation wird im Zeitraum drei bis sechs Wochen nach offizieller Abgabe (30.09.2016) stattfinden. PG-Mitglieder haben sich auf einen möglichen Termin zu einigen. 	- 03.08.16	I H	PG CAIS
5	<ul style="list-style-type: none"> – Für das HPI sind folgende Aufgaben zu erledigen <ul style="list-style-type: none"> ○ Poster designen ○ Paper vom Umfang drei bis fünf Seiten schreiben – Vorstellung beim HPI (Freiwillig) <ul style="list-style-type: none"> ○ Termin im November Zusage Anfang Oktober ○ Entscheidung ist zeitnah zu treffen 	30.09.16 03.08.16?	H I, H	PG CAIS

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

	<ul style="list-style-type: none">- Nächster Jour Fixe findet am 03.08.2016 um 09:00 Uhr im OFFIS Raum U64 statt<ul style="list-style-type: none">o Es ist der Stand der Architektur zu zeigeno Sämtliche Diagramme sind in eine sinnvollen Reihenfolge zu bringen (anschließend Identifizierung der fehlenden Diagramme)	03.08.16	I, H	
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------	------	--

Inhaltsverzeichnis

1	Einleitung	2
1.1	Motivation	2
1.2	Problemstellung	2
1.3	Lösungsansatz	2
2	Organisation	3
2.1	Projektgruppe	3
2.2	Ressourcen	3
3	Analyse der Anforderung	4
3.1	Geschäftsprozess	4
3.2	Anwendungsfälle	4
3.3	Funktionale Anforderungen	4
3.4	Qualitative Anforderungen	4
4	Stand der Technik	5
4.1	Cloud-Computing	5
4.2	X-as-a-Service	5
4.3	Public/Private & Hybrid	5
5	Analytische Informationssysteme	6
5.1	In-Memory Databases	6
5.2	Algorithmen	6
5.3	Open Standards & Open Source	6
5.3.1	SAP UI5 Framework	6
5.3.2	OData Protokoll	6
6	Konzeption	7
6.1	Billing-Model	7
6.2	System Architektur	7
6.3	Datenmodell	7
6.4	Oberflächen Mockups	7
6.5	Lastverteilung	7
7	Implementierung des Prototypen	8
7.1	Entwicklungsumgebung	8
7.2	Backend	8
7.2.1	Docker	8
7.2.2	NodeJS REST-API	8
7.3	Frontend	8
7.3.1	MVC	8
7.3.2	Sädsf	8
7.4	Testing	8
8	Fazit und Ausblick	9
8.1	Sdsdsdsfsd	9
8.2	Sdsdd	9

Bild 1: Vorläufige Gliederung unserer Abschlussdokumentation

Gliederung

1. Aufgabenstellung
2. Projekt-Management
 1. Soll-Planung
 2. Ist-Verlauf
3. Geschäftsprozess
 1. Gesamtkontext des GP (WKD)
 2. Beschreibung (EKP)
 3. DB-Rechnung
4. Prototyp
 1. Anforderungen //Was wir machen sollten
 2. Architektur //Wie wir es gemacht haben + UML Diagramme
 3. ETL & OLAP // Wie wir Daten integriert und ausgewertet haben
 4. Anwendungsfall //Wie der PT benutzt wird + UML
5. Demonstration des Prototypen // Wir führen ihn vor
6. Fazit
 1. Scrum
 2. Prototyp

Bild 2: Vorläufige Gliederung unserer Abschlusspräsentation

Ergebnisprotokoll

Thema:	Jour Fixe mit Product Owner und Stakeholder
Ort:	OFFIS
Datum:	03.08.16
Anwesend:	J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	O. Norkus
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Projektmanagement
2	Abschlussdokumentation
3	Sonstiges
4	
5	
6	
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	<ul style="list-style-type: none"> - Noch zu erledigende Entwicklungsarbeiten für das Frontend sind abzuschließen - Anschließend ist das Backend mit dem Frontend zusammenzuführen / zu verknüpfen - Wir haben uns auf einen Entwicklungsstopp geeignet <ul style="list-style-type: none"> o Bestehende Entwicklungsarbeiten (zu implementierende Funktionen) sind abzuschließen o Es werden keine neuen Entwicklungsarbeiten (Implementierung neuer/weiterer Funktionen) begonnen o Genehmigung durch Oliver ist offen - Es wird der Fokus auf die Abschlussdokumentation gelegt <ul style="list-style-type: none"> o Teammitglieder, die nicht aktiv entwickeln, arbeiten an der Abschlussdokumentation weiter o Teammitglieder, die aktiv entwickeln, schließen ihre Entwicklungsarbeiten ab, bevor sie an der Erstellung der Abschlusspräsentation mitwirken o Die Abschlussdokumentation ist <u>drei</u> Wochen vor Abgabe fertig zu stellen, damit QS-Maßnahmen und eine Korrektur erfolgen können 	<ul style="list-style-type: none"> 17.08.16 24.08.16 - 10.08.16 - 03.08.16 09.09.16 	<ul style="list-style-type: none"> C H I H I H C 	<ul style="list-style-type: none"> PG CAIS PG CAIS PG CAIS O.Norkus PG CAIS PG CAIS PG CAIS
2	<ul style="list-style-type: none"> - Es sind die Anmerkungen zur Abschlusspräsentation aus dem letzten Jour Fixe eingepflegt worden. Die angepasste Version inkl. einer Schätzung der Seitenzahlen sind dem Anhang zu entnehmen <ul style="list-style-type: none"> o Feedback von Oliver bis zum nächsten Jour Fixe 	<ul style="list-style-type: none"> - 10.08.16 	<ul style="list-style-type: none"> I H 	<ul style="list-style-type: none"> PG CAIS O.Norkus
3	<ul style="list-style-type: none"> - Wenn nicht zwingend notwendig, entscheiden wir uns gegen eine Vorstellung unseres Projektes beim HPI - Vorschlag für den nächsten Jour Fixe ist der 10.08.2016 um 09:00 Uhr im OFFIS Raum U64 <ul style="list-style-type: none"> o Eine Einladung inkl. der Agenda wird zeitnah erfolgen 	<ul style="list-style-type: none"> - 10.08.16? 	<ul style="list-style-type: none"> I I 	<ul style="list-style-type: none"> PG CAIS O.Norkus PG CAIS O.Norkus

Inhaltsverzeichnis

1	Einleitung (1-3)	3
1.1	Motivation	3
1.2	Problemstellung	3
1.3	Lösungsansatz	3
1.4	Struktur der Dokumentation	3
2	Organisation (1-3)	4
2.1	Projektgruppe	4
2.2	Internetauftritt	4
2.3	Ressourcen	4
3	Cloud-Computing (2-4)	5
3.1	Grundlagen	5
3.1.1	Public	5
3.1.2	Private	5
3.1.3	Hybrid	5
3.2	X-as-a-Service	5
3.2.1	Infrastruktur-as-a-Service	5
3.2.2	Plattform-as-a-Service	5
3.2.3	Software-as-a-Service	5
4	Analytische Informationssysteme (2-4)	6
4.1	Definition	6
4.2	In-Memory Databases	6
4.3	Algorithmen	6
5	Analyse der Anforderung (5-8)	7
5.1	Geschäftsprozess	7
5.2	Anwendungsfälle	7
5.3	Funktionale Anforderungen	7
5.4	Qualitative Anforderungen	7
6	Konzeption (5-10)	8
6.1	Technisches Konzept	8
6.1.1	System Architektur	8
6.1.2	Datenmodell	8
6.1.3	Oberflächen Mockups	8
6.1.4	Lastverteilung	8
6.1.5	SAP UI5 Framework	8
6.1.6	Kommunikationsprotokolle	8
6.2	Fachliches Konzept	8
6.2.1	Skalierungsalgorithmus	8
6.2.2	Billing-Model	8
7	Implementierung des Prototyps (5-10)	9
7.1	Entwicklungsumgebung	9
7.2	Infrastruktur	9
7.2.1	Cloud	9
7.2.2	Docker	9
7.3	Backend	9
7.4	Frontend	9
7.5	Testing	9
8	Fazit und Ausblick (1-3)	10
8.1	Lessons Learned	10
8.2	Mögliche Ausbaustufen	10
9	Anhang	11

Minimum: 22 Seiten + Anhang

Maximum: 45 + Anhang

Bild 1: Gliederung inkl. Schätzung der Seitenzahlen unserer Abschlussdokumentation

Prioritäten: C: Critical H: High M: Medium L: Low I: Information

Ergebnisprotokoll

Thema:	Jour Fixe mit Product Owner und Stakeholder
Ort:	OFFIS
Datum:	10.08.16
Anwesend:	O. Norkus, J. Asmuß, D. Euhausen, P. Kramer, M. Kuchler, N. Lutz, C. Manske, D. Vorreiter
Nicht Anwesend:	
Protokollführer:	Phillip Kramer

Tagesordnungspunkte

1	Genehmigung des letzten Protokolls
2	Stand der Architektur
3	Identifizierung fehlender Diagramme
4	Projektmanagement
5	Abschlussdokumentation
6	Sonstiges
7	
8	
9	
10	
11	

TOP	Anmerkungen zum TOP:	Termin:	Prio.:	Bearbeiter:
1	– Das Protokoll der letzten Sitzung wurde genehmigt	-	I	-
2	– Der Stand der Architektur ist unverändert	-	I	-
3	– Es ist ein Deployment-Diagramm zu erstellen	31.08.16	H	PG CAIS
4	– Der Entwicklungsstopp ist genehmigt, solange die wichtigsten bzw. höchst priorisierten Funktionen implementiert wurden	-	I	-
5	– Die Gliederung aus dem Anhang des Protokolls vom 03.08.2016 wurde erfolgreich abgenommen	-	I	-
	– Verwendete Algorithmen sollen im Abschlussdokument ersichtlich werden (Bsp: Quellcode, Pseudocode)	31.08.16	H	PG CAIS
	– Der gesamte Quellcode sowie die Installations- und Bedienungsanleitung müssen auf einer CD-ROM vorhanden sein, die neben der Abschlussdokumentation beim Projektende abzugeben sind	30.09.16	C	PG CAIS
	– Der Anhang muss sämtliche Protokolle und Seminararbeiten enthalten	31.08.16	H	PG CAIS
6	– Wir haben uns <u>gegen</u> eine Vorstellung unseres Projektes beim HPI entschieden	-	I	-
	– Abgabe/Projektende: 30.09.16	-	I	-
	– Abschlusspräsentation: Voraussichtlich: Mitte Oktober	-	I	-
	– Der nächste Jour Fixe wird nicht fest vereinbart. Bei aufkommende Fragen oder Feedback zum Zwischenstand der Abschlusspräsentation ist eine Mail aufzusetzen. <ul style="list-style-type: none"> ○ Bei gravierenden Rückfragen wird spontan ein Treffen vereinbart 	-	I	-

Benutzerhandbuch

Die Applikation soll Vertriebsmitarbeiter im Bereich der Energieversorgung bei Vertragsverhandlungen mit ihren Kunden unterstützen. Dazu ist es notwendig, dem Benutzer wichtige Informationen zum jeweiligen Kunden anzuzeigen und ihn beispielsweise durch die Berechnung und Anzeige von Deckungsbeitragsrechnungen in den Verhandlungen zu unterstützen.

Das vorliegende Benutzerhandbuch erläutert die Funktionalitäten und Benutzungsoberflächen der Applikation.

Kundeninformationen anzeigen

Das Layout der Applikation, wie es in der Abbildung 1 zu sehen ist, nennt sich Split App und besteht aus der Masterliste auf der linken Seite, in der sämtliche Kunden aufgeführt sind, und der Detailseite auf der rechten Seite, auf der detaillierte Informationen zum ausgewählten Kunden angezeigt werden. Im oberen Bereich der Masterliste befindet sich eine Suchleiste, mit der die Kundenliste durchsucht werden kann.

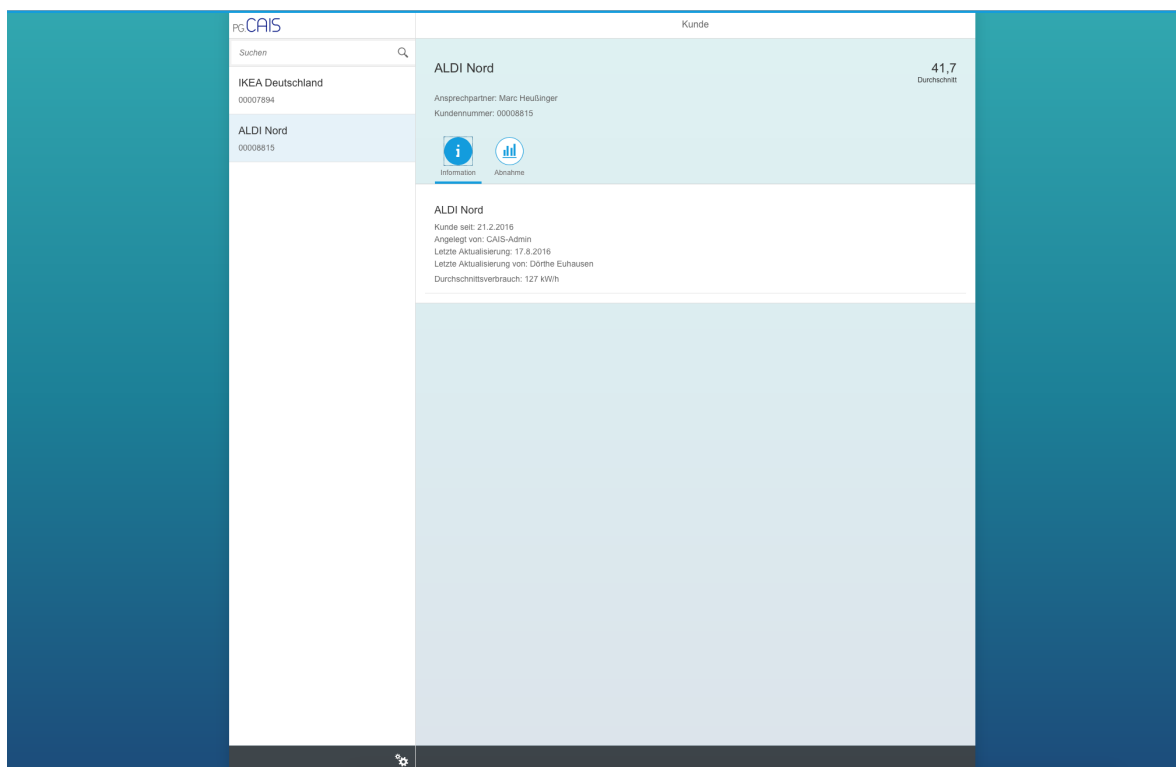


Abbildung 1: Kundeninformationen anzeigen

Die runden Objekte im oberen Bereich der Detailseite nennen sich Icon Tabs und dienen zur Navigation durch mehrere Detailsichten. Der erste Icon Tab zeigt dem Anwender die wichtigsten Informationen zum jeweiligen Kunden und zur Bearbeitung des Datensatzes.

Deckungsbeitrag anzeigen

Durch Klicken auf den zweiten Icon Tab auf der Detailseite wird die grafische Deckungsbeitragsrechnung zum ausgewählten Kunden angezeigt.

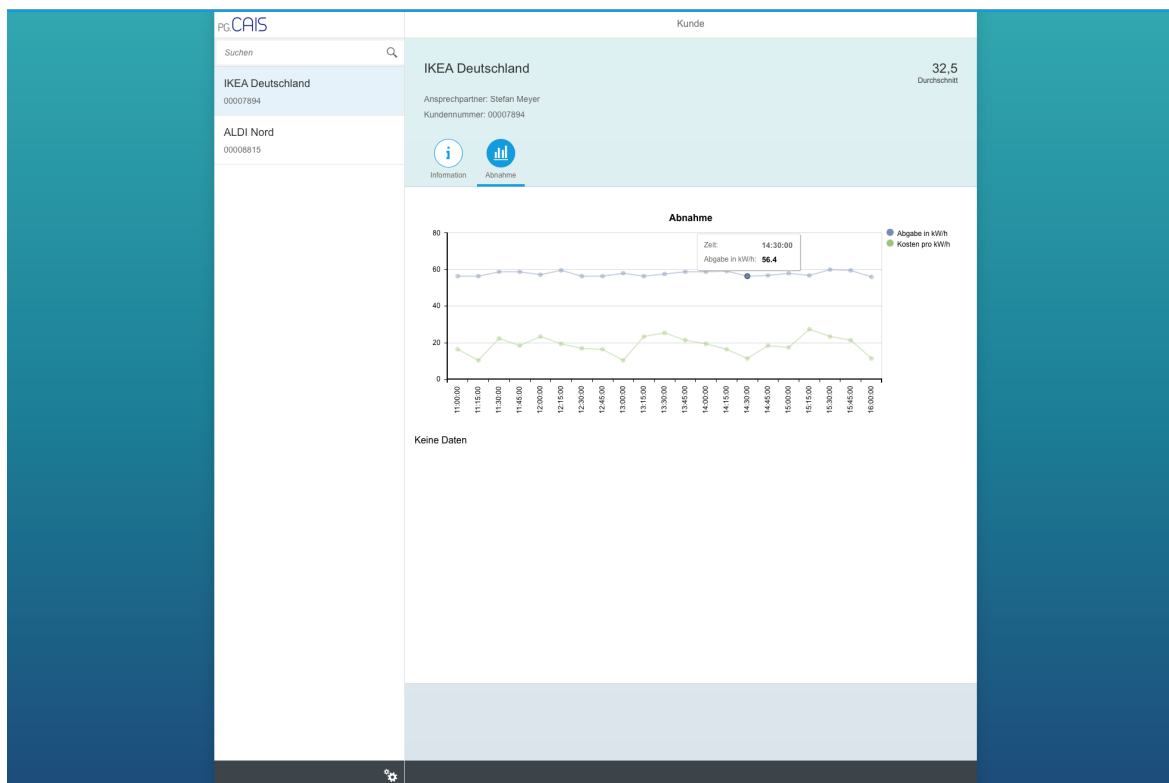


Abbildung 2: Deckungsbeitrag anzeigen

Wie in der Abbildung 2 dargestellt ist, werden durch Auswahl eines Datenpunktes die genauen Werte angezeigt.

Self Service

Das Zahnrad-Icon im unteren rechten Bereich der Masterliste führt den Anwender zu einem Self Service-Popup, das in Abbildung 3 gezeigt ist. Dort können die Kontoeinstellungen eingesehen und die üblichen Self Services, wie beispielsweise Adressänderung, durchgeführt werden.

Mobile Support

Das responsive Design der Applikation erlaubt die Verwendung auf einem Tablet oder einem Smartphone. Wie die Abbildungen 4 und 5 zeigen, passen sich die Benutzungsoberflächen automatisch an unterschiedliche Displaygrößen an.

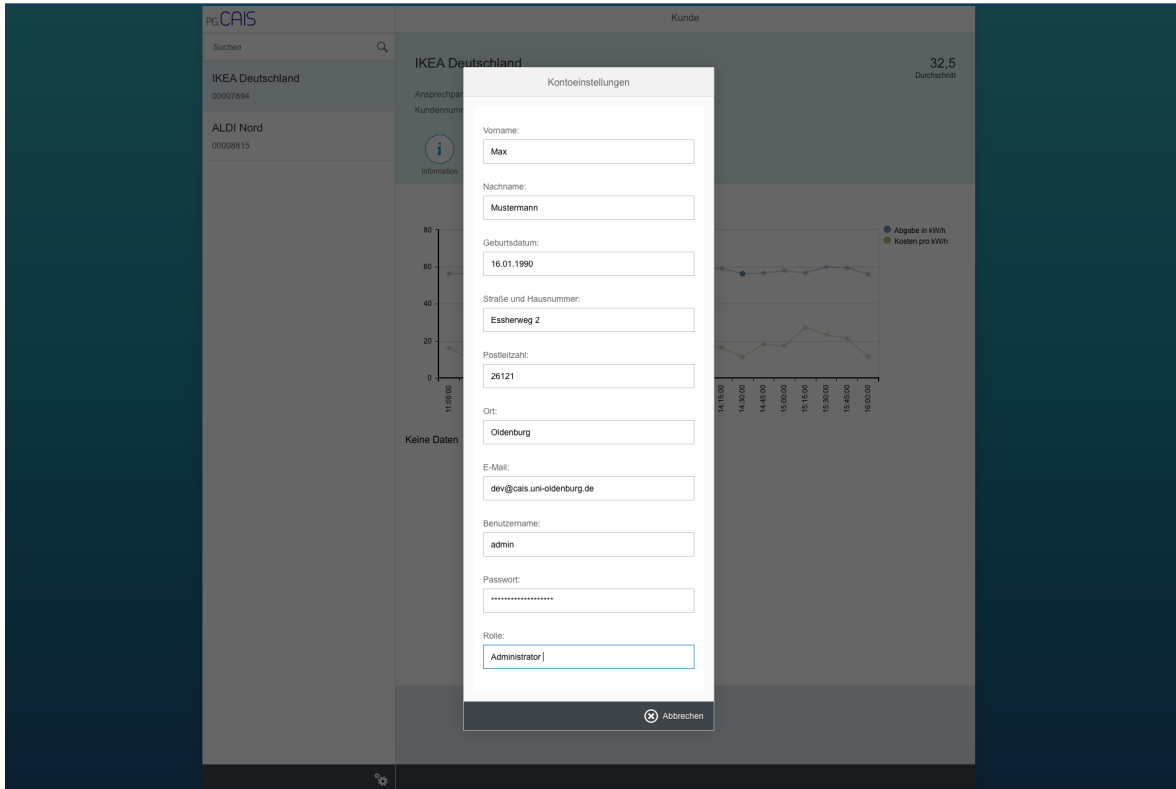


Abbildung 3: Self Service

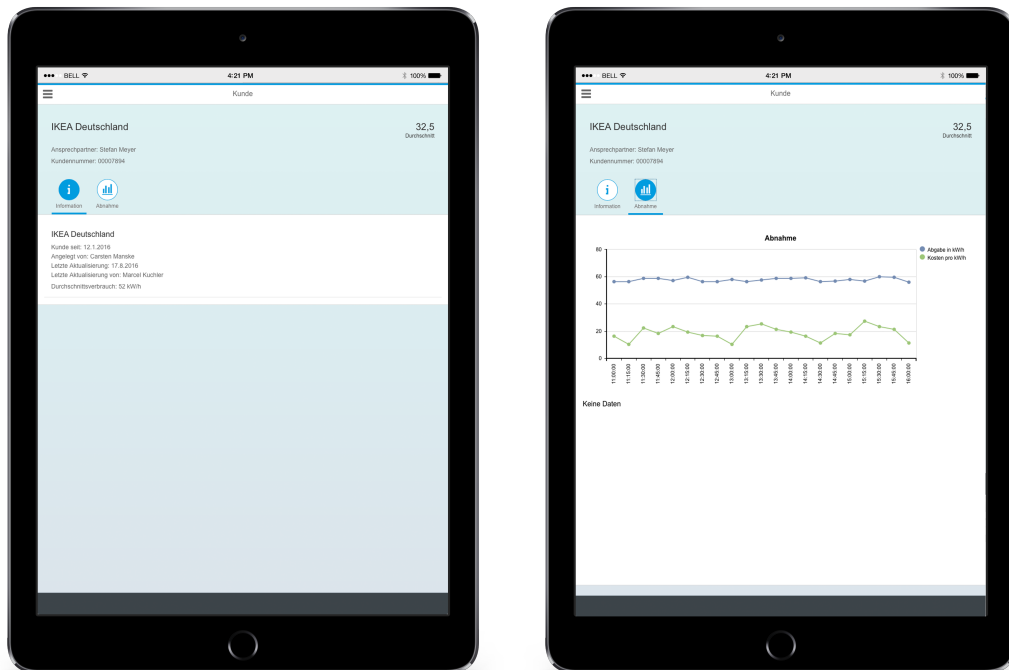


Abbildung 4: Verwendung der Applikation auf einem Tablet

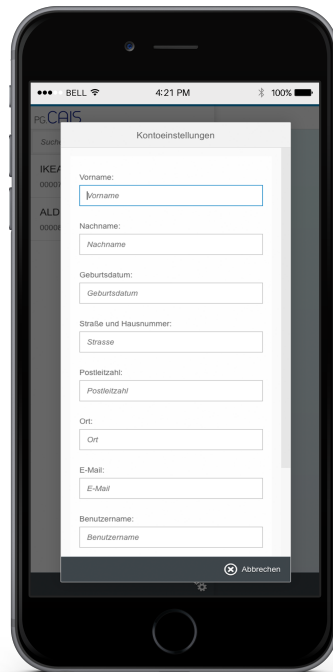
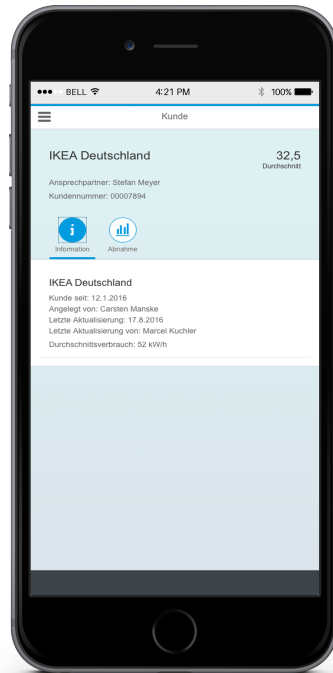
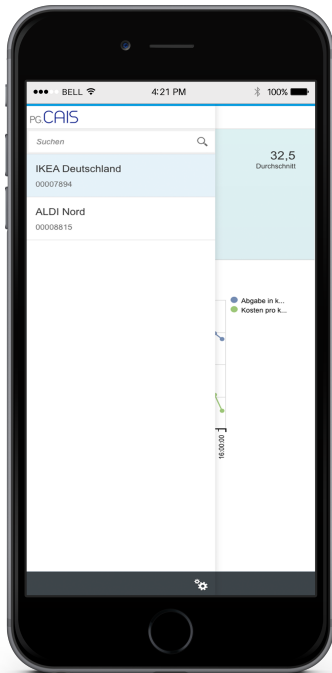


Abbildung 5: Verwendung der Applikation auf einem Smartphone

Testdokumentation

CAIS | Prototyp

Final-Version / 1.0

PG. CAIS

Testdokumentation

letztes Update 26.09.2016
durch Phillip Kramer

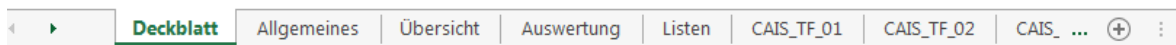


Abbildung 6: Testdokumentation Deckblatt

Allgemeines

1.1 Organisatorisches und Dokumentstatus:

Status Testddrehbuch: Fertiggestellt

Grundlage:

Die Erstellung der Testfälle erfolgte auf Basis der folgenden Dokumente:

- Anforderungen, Epics, User-Stories, Geschäftsprozesse , Lastenheft, etc.

Testdauer ca. 10 Stunden

Testmanager: Phillip Kramer

Ablage: lokal, bei Bedarf ändern

Dateiname

1.2 Änderungsnachweis

Version	Seiten	Gegenstand der Änderung	Autor	Datum
0.1	Alle	Erstellung	Phillip Kramer	31.03.2016
0.2	Übersicht, Auswertung	Testfälle hinzugefügt, durchgeführt und ausgewertet	Phillip Kramer	26.09.2016

QS durch	Rolle	Name	Datum
Projektgruppe	alle	alle PG-Mitglieder	31.03.2016
Projektgruppe	alle	alle PG-Mitglieder	26.09.2016

Abbildung 7: Testdokumentation Allgemeines

Testfall-ID	Testfallname	Testfallkurzbeschreibung	Testgrundlage	Getestete Prototyp-Version	Priorität	Vorbedingung	Status Testdefinition	Tester	Testdatum	Sollverhalten	Testergebnis	Jira Ticket	Bemerkungen	Ausprägung
CAIS_TF_01	Kennzahlen darstellen	Das System muss sicherstellen, dass Kennzahlen dargestellt werden.	CAIS_REQ_101	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_02	Kennzahlen auswerten	Das System muss sicherstellen, dass die Kennzahlen in Abhängigkeit zu unterschiedlichen Dimensionen (Kunde, Quartal,...) ausgewertet werden können.	CAIS_REQ_102	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_03	Gesamtgewinn	Das System muss sicherstellen, dass der Gesamtgewinn errechnet wird.	CAIS_REQ_103	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_04	Profitauswertung	Das System muss sicherstellen, dass die Profite pro Kratwerk und pro Kunde ausgewertet werden.	CAIS_REQ_104	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_05	Kennzahlen Zeiträume	Das System muss sicherstellen, dass die Kennzahlen über verschiedene Zeiträume angezeigt werden (stundengenau, tagessgenau, pro Quartal).	CAIS_REQ_105	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_06	Visuelle Darstellung der Ergebnisse	Das System muss sicherstellen, dass durch den User im Rahmen der visuellen Darstellung der Ergebnisse folgende Schritte durchgeführt werden können: - Filtern der Ergebnisse frei wählbar (bspw. Kreis-/Balkendiagramm) - Markierung von besonders guten Werten (Gewinnzone) grün und besonders schlechte Werte (Verlustzone) rot	CAIS_REQ_106 CAIS_REQ_106-1 CAIS_REQ_106-2	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Fehlerhaft	-	Darstellung der Ergebnisse sind nicht frei wählbar (nur Liniendiagramm). Verlustzone (rote Gewinn- und Verlustzone) ist nicht frei wählbar (nur die obere in grün und die Kosten pro kWh).	Core
CAIS_TF_07	Testdaten per Bulk Load	Das System muss sicherstellen, dass Testdaten per Bulk Load eingelesen werden können.	CAIS_REQ_301	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Fehlerhaft	-	Fehlverhalten ist an den fehlenden Rechten innerhalb der HANA zurückzuführen	Core
CAIS_TF_08	Accountverwaltung	Das System muss sicherstellen, dass durch den User im Rahmen der Accountverwaltung folgende Schritte durchgeführt werden können: - Änderung des Accountnamen - Angabe der Adresse - Änderung der Adresse - Änderung des Passworts	CAIS_REQ_401 CAIS_REQ_401-1 CAIS_REQ_401-2 CAIS_REQ_401-3 CAIS_REQ_401-4 CAIS_REQ_401-5	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_09	Falsche Zugangsdaten	Das System muss sicherstellen, dass im Rahmen der Accountverwaltung falsche Zugangsdaten eintreten können.	CAIS_REQ_401-6	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_10	Rechteverwaltung	Das System muss sicherstellen, dass eine Rechteverwaltung vorhanden ist.	CAIS_REQ_402	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_11	Session-Verwaltung	Das System muss sicherstellen, dass eine Session-Verwaltung vorhanden ist.	CAIS_REQ_403	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_12	Rollen-Prüfung	Das System muss sicherstellen, dass im Rahmen der Rechteverwaltung folgende Rollen vorhanden sind: - Admin-Rolle (User anlegen/freischalten) - Power-User-Rolle - Business-User-Rolle	CAIS_REQ_404 CAIS_REQ_405 CAIS_REQ_406	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_13	Speicherung Kundendaten	Das System muss sicherstellen, dass Daten eines bestimmten Kunden abgespeichert werden können.	CAIS_REQ_203	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	-	Core
CAIS_TF_14	Kundendaten offline verfügbar	Das System muss sicherstellen, dass Daten eines bestimmten Kunden offline zur Verfügung stehen können.	CAIS_REQ_114	1.0	Mittel	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Nicht ausfüllbar	-	Nicht implementiert und kann als zukünftige Ausbaustufe betrachtet werden	Core
CAIS_TF_15	Browsierfähigkeit	Das System muss sicherstellen, dass die Anwendung über alle gängigen Browser erreichbar ist.	CAIS_REQ_115 CAIS_REQ_XXX	1.0	Hoch	n.e.	OK	Philip	26.09.2016	Bei Bedarf ausfüllen	Erfolgreich	-	Das System lief in Safari, Chrome, IE, Firefox und auf dem iPhone 7 Plus, iPhone 7, iPhone 6 Plus, iPhone 5s, iPhone 4s, iPad Air 2, iPad Pro und iPad mini	Core

Abbildung 8: Testdokumentation Übersicht

Hier sind Beispiele angegeben, wie eine Auswertung über die Testfälle erfolgen kann. Diese Lasche kann bei Nichtverwendung gelöscht werden.

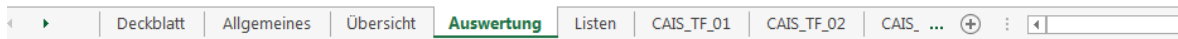
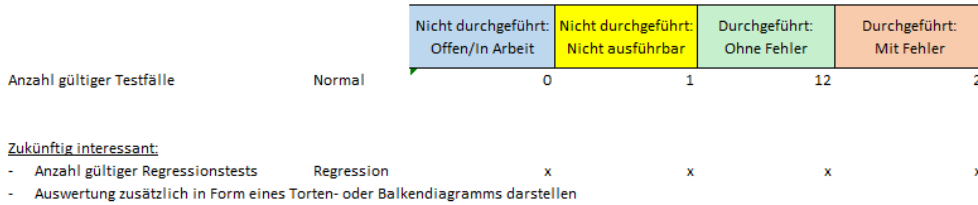


Abbildung 9: Testdokumentation Auswertung

Priorität	Status Testdefinition	Testergebnis	Produkt	Zukünftig sinnvoll: Regression	Zukünftig sinnvoll: Ausführung	Getestete Version	Ausprägung	Tester
Hoch	OK	Fehlerhaft	CAIS Prototyp	Ja	manuell	0.1	Core	Carsten
Mittel	In Arbeit	Erfolgreich		Nein	automatisiert	0.5	Business User	Dennis
Niedrig	Zur Prüfung	Nicht ausführbar			automatisch	1.0	Power User	Dörthe
offen	Zur Überarbeitung	In Ausführung					Administrator	Jannick
	Offen	Offen						Marcel
		Gestrichen						Nils
								Phillip

Bei Erweiterung der Einträge über die jeweilige Markierung hinaus ist die Datenüberprüfung in den Zellen anzupassen.

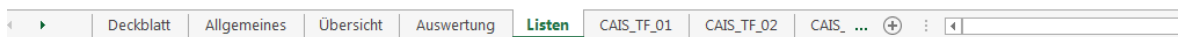


Abbildung 10: Testdokumentation Listen