

Duale Hochschule Baden-Württemberg  
Mannheim

## **Bachelorthesis**

**Integration einer Container-Umgebung in einen automatisierten  
Deployment-Prozess und die Untersuchung ihrer Effekte auf diesen**

### **Studiengang Wirtschaftsinformatik**

Studienrichtung Software Engineering

### **Sperrvermerk**

Verfasser/in:	Yves Torsten Staudenmaier
Matrikelnummer:	7146590
Firma:	SV Informatik GmbH
Abteilung:	IE2 – Deployment
Kurs:	WWI17SEC
Studiengangsleiter:	Prof. Dr.-Ing. habil. Dennis Pfisterer
Wissenschaftlicher Betreuer:	Marius Ebel info@mariusebel.net +49 176 / 473 45452
Firmenbetreuer:	Thomas Teske thomas.teske@sv-informatik.de +49 621 / 454 44096
Bearbeitungszeitraum:	17.02.—08.05.2020

# Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte vorliegt. Die Bachelorarbeit enthält unternehmensinterne Architektur- und Prozessmodellierung und deren Dokumentation. Es ist zum Zeitpunkt der Anmeldung nicht sicher, ob interne Schnittstellen in der Anwendungslandschaft offen gelegt werden.

Mannheim, 31.10.2019

Nadja Haumbach, Ausbildungsverantwortliche

# Lesehinweise

Die folgenden Hinweise sollen das Lesen dieser Projektarbeit erleichtern und spezielle Formatierung definieren:

- Im Sinne der Gleichberechtigung wird in dieser Arbeit entweder die Form „*die Entwickler\*in*“ oder die grammatikalisch korrekte Form „*die/der Entwickler/-in*“ verwendet werden. Bei der Kurzform mit der Sternnotation wird auf Grund der Lesbarkeit der weibliche Artikel benutzt.
- Produkt- oder Eigennamen werden in KAPITÄLCHEN gesetzt, wie beispielsweise NODE.JS.
- Hochgestellte Ziffern weisen auf Fußnoten am Seitenende hin.

# Kurzfassung

Titel	Integration einer Container-Umgebung in einen automatisierten Deployment-Prozess und die Untersuchung ihrer Effekte auf diesen
Verfasser/in:	Yves Torsten Staudenmaier
Kurs:	WWI17SEC
Ausbildungsstätte:	SV Informatik GmbH

# Inhaltsverzeichnis

<b>Abstract</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Quelltextverzeichnis</b>	<b>VIII</b>
<b>Abkürzungsverzeichnis</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Wie können Container-Anwendungen den Prozess des automatisierten „Deployments“ unterstützen?</b>	<b>4</b>
2.1 Grundlagen: Definieren der Begrifflichkeiten zur Forschungsfrage eins .	4
2.1.1 Methodik der Anforderungsanalyse . . . . .	4
2.1.2 Cloud Computing . . . . .	7
2.1.3 Container . . . . .	7
2.1.4 „Deployment“ . . . . .	7
2.2 Ist-Analyse des jetzigen „Deployment“-Prozesses . . . . .	7
2.3 Konzeption eines container-basierten, automatisierten „Deployments“ .	7
<b>3 Welche wirtschaftlichen Vorteile hat der Einsatz von Container auf den Prozess des automatisierten „Deployments“?</b>	<b>8</b>
<b>4 Welche besonderen sicherheitstechnischen Aspekte muss ein solcher Prozess im Bereich der Versicherung erfüllen?</b>	<b>9</b>
<b>5 kritische Betrachtung</b>	<b>10</b>
5.1 Zusammenfassung der Erkenntnisse . . . . .	10
5.2 Fazit . . . . .	10
5.3 Ausblick . . . . .	10
<b>Literaturverzeichnis</b>	<b>X</b>
<b>Anhang</b>	<b>XI</b>
<b>A Ergänzungen zur Forschungsfrage eins</b>	<b>XI</b>
A.1 Anforderungsdokument . . . . .	XI

<b>B</b>	<b>Ergänzungen zur Forschungsfrage zwei</b>	<b>XIV</b>
<b>C</b>	<b>Ergänzungen zur Forschungsfrage drei</b>	<b>XV</b>
	<b>Ehrenwörtliche Erklärung</b>	<b>XVI</b>

# Abbildungsverzeichnis

Abbildung 1.1	Dilbert Comic zu KUBERNETES . . . . .	1
Abbildung 2.1	Entwicklungsprozess der Anforderungen . . . . .	5
Abbildung A.1	Volere Snow Card . . . . .	XIII

# Tabellenverzeichnis



# Quelltextverzeichnis

# Abkürzungsverzeichnis

<b>AWL</b>	Anwendungslandschaft
<b>BaFin</b>	Bundesanstalt für Finanzdienstleistungsaufsicht
<b>IE2</b>	IE2 – Deployment
<b>IE</b>	IE – Entwicklungs- und Betriebsunterstützung
<b>CAB</b>	„Change Advisory Board“
<b>ITIL</b>	Information Technology Infrastructure Library
<b>SV</b>	SV SparkassenVersicherung
<b>SVI</b>	SV Informatik GmbH

# 1 Einleitung

**Motivation der Arbeit** irgendwas Originelles...

Solved all your problems. You're welcome.



Abbildung 1.1: Dilbert Comic zu KUBERNETES

Quelle: *Dilbert on Kubernetes* 2017

Redaktionelle Anmerkung: Abbildung nur als komprimiertes Format verfügbar (Qualitätseinbuße)

## Problemstellung/-abgrenzung

## Zielstellung der Arbeit

**Forschungsfragen/-design** Die Forschungsfragen mit der sich diese Bachelorarbeit beschäftigen wird, sind eine direkte Konsequenz aus der Zielstellung und aus den unternehmensinternen Anforderungen an einen möglichen automatisierten Prozess. Dabei liegt der Fokus auf der Betrachtung beider Teildisziplinen der Wirtschaftsinformatik, nämlich der Informatik und der Wirtschaft – jedoch wird der größere Teil dieser Arbeit einen informationstechnischen Fokus besitzen. Die folgende Aufzählung nennt die einzelnen Forschungsfragen, die im weiteren Verlauf ein gemeinsames Ergebnis erbringen werden. Dieses ist in Kapitel 5 auf Seite 10 zu finden.

1. Wie können Container-Anwendungen den Prozess des automatisierten „Deployments<sup>1</sup>“ unterstützen?
2. Welche wirtschaftlichen Vorteile hat der Einsatz von Container auf den Prozess des automatisierten „Deployments“?

<sup>1</sup>die Definition dieses Begriffes ist in Kapitel 2.1.4 auf Seite 7 zu finden

3. Welche besonderen sicherheitstechnischen Aspekte muss ein solcher Prozess im Bereich der Versicherung erfüllen?

Die Forschungsfrage eins wird einen Ist-Zustand analysieren. Dieser enthält eine Prozessanalyse, eine identifizierte Technologie-Wertekette<sup>2</sup> sowie einen Anforderungskatalog der Entwicklungsabteilungen an den zu konzeptionierenden „Deployment“-Prozess für die Container-Anwendungen. Danach wird ein Konzept eines container-basierten, automatisierten „Deployment“-Prozesses erstellt, dabei wird die Methodologie und das eigentliche Konzept erläutert. Die Forschungsfrage eins schließt mit einem Teilergebnis ab.

Die Forschungsfrage zwei beschäftigt sich mit den wirtschaftlichen Vorteilen eines Einsatzes der Container auf den Prozess des automatisierten „Deployment“-Prozesses. Dabei werden die Erstellung eines „Business Case“<sup>3</sup>, die Prüfung der Übereinstimmung der Ziele dieser Arbeit mit der Geschäftsstrategie der SV Informatik GmbH (SVI) und mögliche Disharmonien dieser identifiziert. Außerdem entsteht eine Konzeption eines verbesserten Geschäftsszenarios, das die Kosteneinsparpotentiale und die Zielharmonisierung enthalten wird. Ein Ausblick schließt die Forschungsfrage zwei ab.

Die Forschungsfrage drei identifiziert sicherheitsrelevante Anforderungen, die nicht nur die funktionalen/nicht-funktionalen Anforderungen einer Anwendung betreffen, sondern auch die komplette Anwendungslandschaft (AWL). Dabei beeinflusst die Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) und auch verschiedene DIN/ISO-Normen diese Anforderungen. Außerdem soll analysiert werden, wie bei der Beschaffung von „open source“- bzw. „closed source“-Anwendungen mögliche Schwachstellen identifiziert werden, die potentielle Angriffsvektoren in der AWL eröffnen würden, und wie mit diesen verfahren wird. Dabei soll versucht werden Rückschlüsse auf die Anwendung OPENSIFT<sup>4</sup> von RED HAT<sup>5</sup> zu ziehen. Auch hier wird ein Teilergebnis diese Forschungsfrage abschließen

**Einordnung der Abteilung in den Geschäftsprozess** Die Abteilung IE2 – Deployment (IE2), die sich im Bereich der Organisationseinheit IE – Entwicklungs- und Betriebsunterstützung (IE) befindet, befasst sich in erster Linie mit dem Transport („Deployment“) von Software-Artefakten der einzelnen Software-Produkte der SVI. Diese werden für die SV SparkassenVersicherung (SV) entwickelt, betrieben und gewartet. Zu den zentralen Aufgaben der Abteilung gehören die Planung, Durchführung und Überwachung der „Build/Deployment“-Prozesse auf den verschiedenen Serverumgebungen. Des weiteren stellt IE2 die Einspielung von datenbank-relevanten Objekten

---

<sup>2</sup>Definition: <Defintion/>

<sup>3</sup>engl. Geschäftsszenario

<sup>4</sup><Definition/>

<sup>5</sup><Definition/>

sicher. Auch entwickelt sie die Bau- und Transportprozesse kontinuierlich weiter und passt diese an die sich ständig veränderten Anforderung der Entwicklungsabteilungen an. Von zentraler Bedeutung ist die Planung und Durchführung der Veröffentlichungen der neuen Versionen einer zu betreuenden Anwendung. Zu dieser Aufgabe gehören auch Aufbau und Bereitstellung der Systemtest-, Releasetest- und Produktions-Umgebungen. Eine weitere zentrale Aufgabe, die nach der Organisationsumstrukturierung am 01.01.2020 in der Abteilung IE2 angesiedelt wurde, ist das Umgebungsmanagement. Die Aufgaben dieses Teilbereichs befasst sich mit folgenden Inhalten: Planung von Aktivitäten in der Produktionsumgebung, Planung und Koordination der Infrastruktur und Notfall-„Fixe“ der Produktion, der allgemeinen „Patch“-Planung; Beratung zur Erweiterung, Koordination und Planung von verschiedenen Testumgebungen. Außerdem ist das Umgebungsmanagement Teil des „Change Advisory Board“ (CAB), das ein Gremium nach der Sammlung Information Technology Infrastructure Library (ITIL) darstellt. Dieses ist für die Freigabe von „Changes“ verantwortlich und hat ständige, wie auch der Situation angepasste, Mitglieder.

**Aufbau der Arbeit** In Kapitel 2 auf der nächsten Seite

In Kapitel 3 auf Seite 8

In Kapitel 4 auf Seite 9

In Kapitel 5 auf Seite 10

# 2 Wie können Container-Anwendungen den Prozess des automatisierten „Deployments“ unterstützen?

Dieses Kapitel ...

## 2.1 Grundlagen: Definieren der Begrifflichkeiten zur Forschungsfrage eins

Dieses Teilkapitel soll grundlegende Begrifflichkeiten, die im weiteren Verlauf dieser Arbeit verwendet werden, definieren, um so eine einheitliche Terminologie der Begriffe zu entwickeln. Dadurch wird ein gemeinsames Verständnis erzeugt.

### 2.1.1 Methodik der Anforderungsanalyse

Die Anforderungsanalyse leitet sich aus dem thematischen Komplex des „Requirements-Engineering“ ab, die verschiedene Bedeutungsvarianten besitzt – dabei „[...] steht [es] einmal für alle konkreten Aktivitäten am Beginn einer Systementwicklung, die auf eine Präzisierung der Problemstellung abzielen. Ebenso steht es aber auch für eine ganze Teildisziplin im Grenzbereich zwischen Systems-Engineering, Informatik und Anwendungswissenschaften.“<sup>6</sup> Diese Analyse soll, laut der herrschenden Meinung der Wissenschaft, am Anfang jeder Systementwicklung stehen, um so bestimmte Vorgehensweise anzuwenden. Dabei entstehen, wenn der später weiter definierte Prozess verfolgt wird, viele systematisch verbundene Dokumente, die Anforderungen enthalten. So ist jede Anforderung wieder ein Cluster von kleineren Anforderungen, die miteinander verbunden sind. Diese werden durch den IEEE-Standard 1220 definiert als „a statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines).“<sup>7</sup> Dieser

---

<sup>6</sup>Partsch 2010, S.19.

<sup>7</sup>IEEE 2005, S.9.

Standard legt mit höchster Priorität den Fokus auf die Formulierung einer Anforderung als elementar wichtig für das Produkt bzw. für das Erreichen der Akzeptanz des Produktes. Ziel der Analyse ist es, funktionale und nicht-funktionale Anforderungen zu identifizieren und diese testbar zu dokumentieren. Funktionale Anforderungen definieren genau, was ein System später erfüllen muss, sie ergeben sich aus der Fragestellung „Was tut das System?/Was soll es aufgrund der Aufgabenstellung können?“<sup>8</sup> Nicht-funktionale Anforderungen konkretisieren die Qualitätsansprüche an das System, die Forderung an das zu implementierende System als Ganzes, sowie Randbedingungen, die aus Projekt-/Prozess-/Unternehmensbedingungen resultieren können.<sup>9</sup>

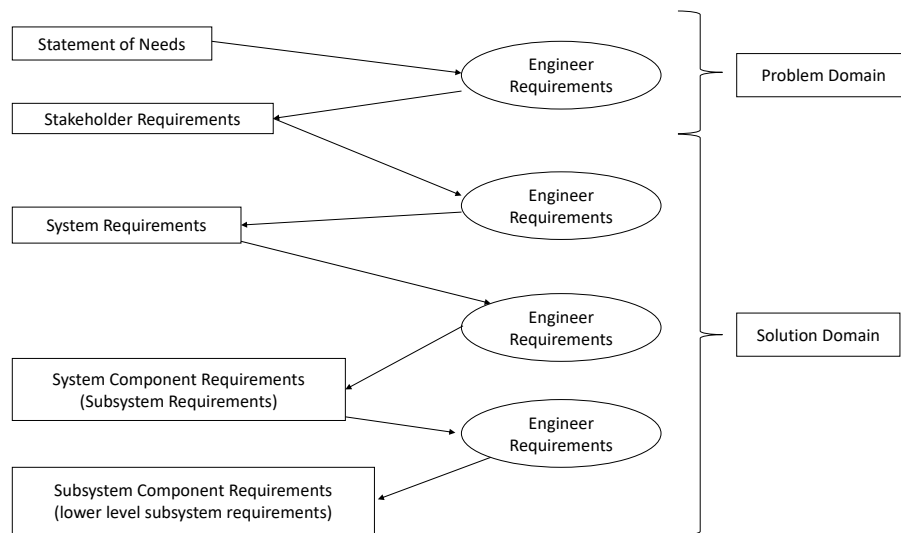


Abbildung 2.1: Entwicklungsprozess der Anforderungen

Quelle: in Anlehnung an Hull, Jackson und Dick 2011, S.28

Das „statement of needs“ ist der Startpunkt für die Entwicklung einer Anforderung die am Ende des Prozesses, der in Abbildung 2.1 dargestellt ist, präzise dokumentiert sein wird. Dieses ist am Anfang immer ein Ausdruck eines Anspruchs oder Wunsches an das zu entwerfende System; dabei bildet das „statement“ und die „stakeholder requirements“ die „problem domain“. Diese definiert grundständige Methodik, wie auch eine nicht-technische Herangehensweise, die auf die Projektbeteiligten („stakeholder“) angepasst ist. Nachfolgend werden die Projektbeteiligten als „stakeholder“ bezeichnet, dabei ist die Rolle beschrieben als „(Stakeholder) sind Personen oder Organisationen, die ein potenzielles Interesse an einem zukünftigen System haben und somit in der Regel auch Anforderungen an das System stellen.“<sup>10</sup> Später definiert die „problem domain“ den Zweck des Systems – dadurch ist bei der Ermittlung der Anforderungen

<sup>8</sup>Partsch 2010, S.27.

<sup>9</sup>vgl. Partsch 2010, S.27-29.

<sup>10</sup>Partsch 2010, S.8.

die Frage „Was ist der Zweck des Systems?“ anstelle „Was soll das System ihrer Meinung nach tun?“. Dies soll die „stakeholder“ extrinsisch motivieren über den Zweck des zu entwerfenden Systems und nicht über einen möglichen Lösungsweg (das Wie) nachzudenken. Durch diesen Ansatz folgen Antworten nach dem Muster „Ich möchte etwas tun können ...“ – wissenschaftlich bzw. literarisch betrachtet sind diese Form der Anforderungen als „capability requirement(s)“<sup>11</sup> bekannt. Sie stellen die wichtigsten Erkenntnisse in der „problem domain“ dar. Nun wird im weiteren Verlauf ein Modell konstruiert, das den Projektbeteiligten, den „stakeholder“, präsentiert wird. Dies unterliegt der Einschränkung, dass es jede/jedem Projektbeteiligte/n versteht. Denn sie validieren das konstruierte Modell in jedem weiteren Schritt, der in Abbildung 2.1 auf der vorherigen Seite, ersichtlich ist. Die Anforderungen an das Modell sind quantitativ gering: es muss nicht-technisch sein und es muss geeignet sein die Anforderungen an das System abzubilden. Eine solche Darstellung ist dann geeignet, wenn sie den gewünschten Zweck an das System abbildet, das heißt, dass sie keine technischen Details zeigt, sondern einen Überblick bietet. Ein „use scenario“<sup>12</sup> wird meist verwendet, da es sich eignet menschliche Aktionen bzw. Ziele darzustellen. Abschließend müssen die „stakeholder“-Anforderungen folgende Kriterien erfüllen:

- kurz und prägnant formulierte Beschreibung, jedoch einfach zu verstehen und
- gleichzeitig sollten sie nicht-technisch aber realistisch formuliert sein.

Die „solutions domain“, die auf Abbildung 2.1 auf der vorherigen Seite zu sehen ist, ist die Nachfolgerin von der „problem domain“. Der Hauptunterschied zwischen den beiden Bereichen ist, dass die „solution domain“ idealtypisch qualitativ hochwertig beschriebene Anforderungen als „Input“ bekommt. Dazu konträr erhält die „problem domain“ vage formulierte Wunschliste oder einem nicht klar definierten Ziel als initialen „Input“. Ausgehend von der Aussage von E. Hull, „in an ideal world, all the requirements would be clearly articulated, individual test able requirements“,<sup>13</sup> ist zu deduzieren, dass viele Ebenen zu erforschen gibt, um dieser Aufforderung zu entsprechen. So muss iterativ in jeder Ebene eine neue Analyse des „Inputs“ erfolgen, um einen Ausgangspunkt für das weitere Vorgehen zu initialisieren. Die Komplexität dieser Ebenen ist anhängig von dem Grad der Innovation sowie vom Kontext des zu entwickelnden Systems. Jede Entscheidung während des Prozess kann mögliche Entscheidungspfade in einer anderen Ebene verhindern. Ziel des Prozesses ist es, ein Anforderungsdokument/-katalog zu entwerfen, das laut der gesichteten Literatur in verschiedenen Repräsentationen vorliegen kann. Dennoch sollten primäre Bestandteile, wie die Rahmenbedingungen, die Projektbeteiligten, die Projektaspekte und die funktionale/nicht-funktionale Anforderungen, enthalten sein. Ein Beispiel dieses Katalogs ist im Anhang A.1 auf Seite XI zur Ansicht enthalten.

<sup>11</sup>vgl. Hull, Jackson und Dick 2011, S.94.

<sup>12</sup>vgl. Hull, Jackson und Dick 2011, S.94.

<sup>13</sup>Hull, Jackson und Dick 2011, S.115.



### **2.1.2 Cloud Computing**

### **2.1.3 Container**

### **2.1.4 „Deployment“**

## **2.2 Ist-Analyse des jetzigen „Deployment“-Prozesses**

## **2.3 Konzeption eines container-basierten, automatisierten „Deployments“**

### **3 Welche wirtschaftlichen Vorteile hat der Einsatz von Container auf den Prozess des automatisierten „Deployments“?**

**4 Welche besonderen  
sicherheitstechnischen Aspekte  
muss ein solcher Prozess im  
Bereich der Versicherung erfüllen?**

# **5 kritische Betrachtung**

## **5.1 Zusammenfassung der Erkenntnisse**

## **5.2 Fazit**

## **5.3 Ausblick**

# Literaturverzeichnis

*Atomic Requirement Download* (19. Aug. 2019). URL: <https://www.volere.org/atomic-requirement-download/>.

*Dilbert on Kubernetes* (11. Aug. 2017). URL: [https://miro.medium.com/max/1024/1\\*R0DEnf\\_7sjswuBHouioQFg.jpeg](https://miro.medium.com/max/1024/1*R0DEnf_7sjswuBHouioQFg.jpeg).

Hull, Elizabeth, Ken Jackson und Jeremy Dick (2011). *Requirements engineering*. 3rd ed. London ; New York: Springer. ISBN: 9781849964043 9781849964050.

IEEE (2005). „IEEE Standard for Application and Management of the Systems Engineering Process“. In: *IEEE Std 1220-2005 (Revision of IEEE Std 1220-1998)*, S. 1–96. DOI: 10.1109/IEEESTD.2005.96469.

Partsch, Helmuth (2010). *Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme*. ger. 2., überarb. und erw. Aufl. eXamen.press. OCLC: 845656932. Berlin: Springer. ISBN: 9783642053580 9783642053573. URL: <http://dx.doi.org/10.1007/978-3-642-05358-0>.

*Volere Requirements Specification Template* (19. Aug. 2019). URL: <https://www.volere.org/templates/volere-requirements-specification-template/>.

# A Ergänzungen zur Forschungsfrage eins

## A.1 Anforderungsdokument

Ein Anforderungskatalog hat bestimmte Anforderungen, die an den Katalog gestellt werden. Neben der Forderung nach Einhaltung der Qualitätskriterien, definiert nach dem ISO-Standard 9000/9001, sind noch folgende Forderungen in der Literatur beschrieben:<sup>14</sup>

- vollständig (inhaltlich – d. h., alle Anforderungen sind erfasst –, formal, Normkonform)
- konsistent (keine Widersprüche zwischen den Bestandteilen des Dokuments, insbesondere keine Konflikte zwischen verschiedenen Anforderungen)
- lokal änderbar (Änderungen an einer Stelle sollten keine Einflüsse auf Konsistenz und Vollständigkeit des Gesamtdokuments haben)
- verfolgbar (ursprüngliche Stakeholderwünsche und Zusammenhänge zwischen Anforderungen sind leicht zu finden)
- klar strukturiert
- umfangsmäßig angemessen
- sortierbar/projezierbar (nach verschiedenen Kriterien, für verschiedene Stakeholder).

Die folgende Aufzählung beschreibt eine Vorlage für das Anforderungsdokument nach Quelle: Sie nutzt die Hilfsmittelsammlung „Volere“. Diese bietet im Themenbereich „requirements engineering“ kostenpflichtig Dokumentenvorlagen an. Die beiden Bekanntesten sind die hier gezeigte „Volere Requirements Specification Template“ und das kostenlose „Volere Atomic Requirement Template“, das umgangssprachlich „Snow Card“ genannt wird. Die „Snow Card“ ( A.1 auf Seite XIII) ist eine Karteikarte, die benutzt wird, um eine vollständige Aufnahme aller Informationen einer einzelnen Anforderung zu gewährleisten.<sup>15</sup> Die folgende Liste wurde in Anlehnung an die Quelle *Volere Requirements Specification Template* 2019 erstellt.

---

<sup>14</sup>sig. Partsch 2010, S.34.

<sup>15</sup>vgl. *Atomic Requirement Download* 2019.

- Projekt-Treiber
  1. Zweck des Projekts
  2. Auftraggeber, Kunde und andere Stakeholder
  3. Nutzer des Produkts
- Projekt-Randbedingungen
  1. Einschränkungen
  2. Namenskonventionen und Definitionen
  3. Relevante Fakten und Annahmen
- Funktionale Anforderungen
  1. Arbeitsrahmen
  2. Systemgrenzen
  3. Funktionale und Daten-Anforderungen
- Nicht-funktionale Anforderungen
  1. Look-and-Feel-Anforderungen
  2. Usability-Anforderungen
  3. Performanz-Anforderungen
  4. Operationale und Umfeld-Anforderungen
  5. Wartungs- und Unterstützungsanforderungen
  6. Sicherheitsanforderungen
  7. Kulturelle und politische Anforderungen
  8. Rechtliche Anforderungen
- Projekt-Aspekte
  1. Offene Punkte
  2. Standardlösungen
  3. Neu aufgetretene Probleme
  4. Installationsaufgaben
  5. Migrationstätigkeiten
  6. Risiken
  7. Kosten
  8. Nutzerdokumentation
  9. Zurückgestellte Anforderungen
  10. Lösungsideen

Requirement #:	Requirement Type:	Event/BUC/PUC #:
Description:		
Rationale:		
Originator:		
Fit Criterion:		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority:	Conflicts:	
Supporting Materials:		
History:		

**Volere**  
Copyright © Atlantic Systems Guild

Abbildung A.1: Volere Snow Card  
Quelle: *Atomic Requirement Download* 2019



## **B Ergänzungen zur Forschungsfrage zwei**

## **C Ergänzungen zur Forschungsfrage drei**

# Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *Integration einer Container-Umgebung in einen automatisierten Deployment-Prozess und die Untersuchung ihrer Effekte auf diesen* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

---

Ort, Datum

---

Yves Torsten Staudenmaier