

Technische Universität Dresden  
Fakultät Informatik  
Institut für Software- und Multimediatechnik  
Professur Softwaretechnologie

Großer Beleg

# Organisierte Software-Startups mit kollaborativen Canvases

Christina Korger  
geb. am 16.03.1991 in Regensburg  
Matrikelnr. 3703541

Betreuer:  
**Dipl.-Inf. Georg Püschel**  
Hochschullehrer:  
**Prof. Dr. rer. nat. habil. Uwe Aßmann**



# Aufgabenstellung für einen Großen Beleg

**Thema:**

## Organisierte Software-Startups mit kollaborativen Canvases

**Zielstellung:**

Zur Verbesserung der Kommunikation in Projektteams können verschiedene Aspekte eines Diskurses durch einen Canvas veranschaulicht werden. Ein Canvas unterteilt eine Visualisierungsfläche thematisch in Felder, so dass der Diskurs gesteuert werden kann und die Kollaboration zwischen den Kommunikationspartnern verbessert wird. Diese Vorgehensweise kann sowohl bei der Geschäftsmodellentwicklung, der Softwareentwicklung oder zum Austausch mit verschiedenen Stakeholdern in einer Start-Up-Inkubation eingesetzt werden. Es kommt in diesen Bereichen eine Vielzahl von unterschiedlich strukturierten und themenspezifischer Canvases in Frage, zwischen denen Abhängigkeiten auftreten können. So können beispielsweise bei der Anforderungsanalyse und dem Softwareentwurf spezifische Canvases aufeinander Bezug nehmen.

Zur optimalen Unterstützung agiler und potentiell verwertbarer Softwareprojekte in Start-Ups kann ein digitales Canvas-Framework angeboten werden, mit dem Canvases für bestimmte Einsatzszenarien automatisch konfiguriert und ausgewertet werden können. Dafür ist es notwendig, eine Canvas-Produktlinie sowie Konfigurationskriterien zu definieren, die dann als Modell im Framework verarbeitet werden können. Die Analyse der Canvases kann zudem durch Constraints und Metriken erfolgen, die ebenfalls modelliert werden müssen.

**Angestrebte Arbeitsergebnisse:**

1. Recherche und Vergleich bekannter Canvases aus den Wissensgebieten Geschäftsmodellentwicklung und Softwareprojektsteuerung
2. Entwicklung von Canvases zur Steuerung von Inkubationsprozessen, Softwareprojekten und Geschäftsmodellierung
3. Variabilitätsanalyse der gefunden Canvases und Entwicklung einer konfigurierbaren Canvas-Produktlinie
4. Definition von Kriterien zur Konfiguration der Canvas-Produktlinie

5. Untersuchung und Abhängigkeiten in Canvases und zwischen Canvases; sowie deren Definition als formale Constraints
6. Definition einer Metrik zur Vollständigkeitsanalyse und Guidance für Canvas-basierte Startup-Steuerung

Betreuer: Dipl.-Inf. Georg Püschel

Institut für Software- und Multimediatechnik  
Lehrstuhl Softwaretechnologie





# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Verwandte Literatur . . . . .	2
1.3 Konzept . . . . .	3
<b>2 Das Canvas als kollaboratives Werkzeug</b>	<b>5</b>
2.1 Was ist ein Canvas? . . . . .	5
2.2 Welche Vorzüge bietet die Verwendung eines Canvas als Werkzeug? . . . . .	6
<b>3 Einsatz von Canvases in Geschäftsmodellentwicklung und Softwareprojektsteuerung</b>	<b>9</b>
3.1 Canvases für die Entwicklung von Geschäftsmodellen . . . . .	9
3.1.1 Business Model Canvas . . . . .	9
3.1.2 Lean Canvas . . . . .	11
3.1.3 Business Model Canvas for User Experience . . . . .	13
3.2 Canvases für die Steuerung von Softwareprojekten . . . . .	14
3.2.1 Lean Software Engineering Canvas . . . . .	15
<b>4 Erarbeitung von Kundenwünschen mit Canvases</b>	<b>17</b>
4.1 Die Einteilung von Kundenwünschen nach dem Kano-Modell . . . . .	17
4.2 Das Customer Interview Canvas . . . . .	19
<b>5 Iterationsplanung im Softwareprojekt mit Canvases</b>	<b>21</b>
5.1 Das Planning Game . . . . .	21
5.2 Das Iteration Planning Canvas . . . . .	22
<b>6 Studie zum Einsatz von Canvases in Softwareprojekten</b>	<b>25</b>
6.1 Entwicklung eines Fragebogens . . . . .	25
6.2 Durchführung der Studie . . . . .	27
6.3 Auswertung der Studie . . . . .	28

6.4	Fazit . . . . .	31
<b>7</b>	<b>Variabilitätsanalyse</b>	<b>33</b>
7.1	Feature-Modelle . . . . .	33
7.2	Kategorisierung der Canvas-Felder im Zusammenhang mit Softwareprojektsteuerung . . . . .	36
7.2.1	Felder für eine erste Vorstellung des Produkts . . . . .	36
7.2.2	Felder für die Projektplanung in Iterationsschritten . . . . .	39
7.2.3	Felder für separate Kundengespräche . . . . .	42
7.3	Konfigurationsmöglichkeiten von Canvases im Softwareentwicklungskontext . .	43
7.4	Konfigurationsbeispiel . . . . .	45
<b>8</b>	<b>Möglichkeiten zur Bewertung der Vollständigkeit von Canvases</b>	<b>49</b>
8.1	Bewertungsansätze . . . . .	49
8.2	Diskussion geeigneter Bewertungskriterien . . . . .	50
<b>9</b>	<b>Schlussbetrachtung</b>	<b>55</b>
9.1	Zusammenfassung . . . . .	55
9.2	Bewertung und Ausblick . . . . .	56
	<b>Anhang</b>	<b>59</b>
	<b>Literaturverzeichnis</b>	<b>71</b>



# Abbildungsverzeichnis

2.1	Canvas-Beispiel für die Planung und Organisation eines Essens . . . . .	6
3.1	Business Model Canvas von Alexander Osterwalder . . . . .	10
3.2	Lean Canvas (Startup) von Ash Maurya . . . . .	12
3.3	Business Model Canvas for User Experience von Tristan Kromer [17] . . . . .	14
3.4	Lean Software Engineering Canvas von Manuel Oddoy . . . . .	15
4.1	Die Felder des Customer Interview Canvas . . . . .	19
5.1	Die Felder des Iteration Planning Canvas . . . . .	23
6.1	Die „10 Gebote der Frageformulierung“ [25] . . . . .	26
6.2	Anwendungsbeispiel des Customer Interview Canvas aus der Studie . . . . .	29
7.1	Feature-Diagramm für die Zusammenstellung eines Sandwiches . . . . .	36
7.2	Feature-Diagramm für die Konfiguration eines Canvas . . . . .	44
7.3	Canvas-Vorlagen zum Konfigurationsbeispiel . . . . .	47



# Tabellenverzeichnis

7.1	Verbindungen zwischen Features gemäß FODA [15] . . . . .	34
7.2	Erweiterte Feature-Notation nach Czarnecki und Eisenecker [10] . . . . .	35
7.3	Canvas-Felder für den Softwareentwicklungskontext, kategorisiert . . . . .	41



# 1 Einführung

Canvases gelten als leichtgewichtiges Werkzeug für die Darstellung komplizierter Sachverhalte. Die Arbeit „Organisierte Software-Startups mit kollaborativen Canvases“ beschäftigt sich mit der Konzeption eines Canvas-Frameworks für die Unterstützung von Softwareprojekten. Ein Überblick über die Eigenschaften des Werkzeugs sowie existierende Canvases aus den Bereichen der Geschäftsmodellierung und der Softwareprojektsteuerung bilden die Einführung in das Thema. Im Anschluss werden zwei weitere Canvases für Projektplanung und Kundengespräche eingeführt, deren Eignung im Rahmen einer universitär durchgeführten Studie evaluiert wird. Aus den Feldern der insgesamt drei Canvases für den Kontext der Softwareentwicklung werden in einer Variabilitätsanalyse Kriterien herausgearbeitet, die die Anpassung des Werkzeugs für den individuellen Projektkontext ermöglichen. Den Abschluss bildet die Diskussion möglicher Ansätze zur Vollständigkeitsbewertung ausgefüllter Canvases.

## 1.1 Motivation

Die Bearbeitung eines Canvas beginnt mit einer leeren Fläche, unterteilt in mehrere Felder. Im Verlauf kommen Notizen in Form verschiedenfarbiger Zettel hinzu, frei formuliert in Zusammenarbeit mit allen Beteiligten. Das von Alexander Osterwalder veröffentlichte Business Model Canvas hat sich über die letzten Jahre zu einem wichtigen Werkzeug für die Erarbeitung von Geschäftsmodellen entwickelt. Als Erfolgsrezept gilt die vereinfachte, visuelle Erfassung komplexer Sachverhalte mit Konzentration aufs Wesentliche. Die vollständige Erfassung der Sachverhalte wird unterstützt durch die Zuweisung der Felder zu entsprechenden Themenaspekten und weitere Ausfüllhinweise. Einen umfassenden Leitfaden bietet das Buch [23]. Begünstigt durch die Regelung zur Wiederverwendung und Adaption<sup>1</sup> sind viele Variationen von Canvases entstanden, die in unterschiedlichen Bereichen zur Anwendung kommen. Zu letzteren zählt auch die Softwareentwicklung.

Viele Softwareprojekte scheitern an unzureichender Kommunikation, sei es innerhalb des Projektteams oder mit dem Kunden. Ein Beispiel, wie dieses Risiko durch die Nutzung von Canvases reduziert werden kann, zeigt die Arbeit „Softwareentwicklung mit Natürlicher Sprache“

---

<sup>1</sup>Veröffentlichung des Business Model Canvas unter der Creative Commons Attribution-Share Alike 3.0 Unported License (s. <http://creativecommons.org/licenses/by-sa/3.0/>)

von Manuel Oddoy [21]. Während diese das „Lean Software Engineering Canvas“ als Werkzeug speziell für die Anforderungsanalyse im vorgestellten Ansatz „Lean Modelling“ propagiert, ist ebenso die Erweiterung auf andere Teile des Softwareentwicklungsprozesses denkbar. Dazu kommen Unterschiede in der Planung und Durchführung von Projekten, die individuelle Ansprüche an ein Werkzeug stellen.

Mit der Abdeckung weiterer Teile des Softwareentwicklungsprozesses durch Canvases rückt die Vision einer organisationsumfassenden Unterstützung näher. Auf welche Weise diese sich in einem Framework realisieren lässt, wird in dieser Arbeit erstmals für den Teilbereich der Softwareprojektsteuerung untersucht. Projektspezifische Anforderungen und die dementsprechenden Anpassungen der Canvas-Vorlagen lassen sich auf Konfigurationskriterien und Produktvarianten übertragen. Diese Zusammenhänge können in ein entsprechendes Framework integriert werden und so bei einer Vielzahl von Projekten zugeschnittene Unterstützung bieten.

## 1.2 Verwandte Literatur

Den Einstieg in die Arbeit mit Canvases bietet das Buch „Business Model Generation“ [23] von Alexander Osterwalder und Yves Pigneur, den Erfindern des Canvas. Darin werden mit und um das „Business Model Canvas“ verschiedene Werkzeuge und Praktiken vorgestellt, die bei der Entwicklung von Geschäftsmodellen unterstützen. Der Schwerpunkt liegt auf der Ausrichtung am Markt und den Bedürfnissen der unterschiedlichen Kundensegmente sowie, im Gegensatz zu detaillierten Businessplänen, einer flexiblen und leichtgewichtigen Dokumentation.

Neben dem Original hat auch das „Lean Canvas“ von Ash Maurya als Variante des „Business Model Canvas“ einige Aufmerksamkeit erhalten. In seinem Buch [19] argumentiert er auf der Vorlage von Osterwalders Canvas und führt mit der Begründung des risikoreichen Umfelds von Startups einige Modifikationen durch.

Weitere Teile der Arbeit über Businesspläne und Startup-Gründung stützen sich auf Steve Blank, amerikanischen Unternehmer, Autor und Dozent unter anderem an der University of Stanford und Berkeley<sup>2</sup> [8,9]. Mit „The Four Steps to Epiphany“ beschreibt Blank den Weg zum erfolgreichen Startup. Die von ihm ins Leben gerufene Plattform „LaunchPad Central“<sup>3</sup> bietet Existenzgründern die Möglichkeit, ihre Geschäftsidee Schritt für Schritt über externes Feedback zu überprüfen und verbessern. Die Risiken eines Fehlstarts sollen durch Kundengespräche sowie durch iterative Entwicklung und Prototyping minimiert werden. Gleichzeitig dient die Plattform Investoren dazu, evidenzbasiert über eine eventuelle Förderung zu entscheiden.

Eine wichtige und aktuelle Informationsquelle zu den Themen *Lean Startup*, *Entrepreneurship*, *Minimum Viable Product* und *Customer Development* findet sich außerdem in den Blogs einzelner Unternehmer und Berater [6, 16, 18, 20, 22, 27].

---

<sup>2</sup>vgl. <http://steveblank.com/about/>

<sup>3</sup><https://www.launchpadcentral.com/>

Von Seiten der Softwareentwicklung setzt sich Manuel Oddoy [21] mit der Verwendung von Canvases für die Anforderungsanalyse auseinander. Er konstruiert aus Ansätzen wie Model-Driven, Agile und Lean Software Development das sogenannte „Lean Modelling“. Dieses ist mit dem „Lean Software Engineering Canvas“ und einem Werkzeug zur Generierung von Code und Modellen aus natürlichsprachlichen Spezifikationen in hohem Maß auf eine gute Kommunikation zwischen Kunden und Entwicklern ausgerichtet. Bei der Auswahl der Themen für die Canvas-Felder orientiert er sich am Template „arc42“ von Dr. Gernot Starke und Dr. Peter Hruschka [32].

Rupp und Schüpferling [28–31] schreiben in einer Artikelreihe über die Einteilung von Kundenwünschen nach dem Kano-Modell und stellen für den jeweiligen Typ geeignete Ermittlungstechniken vor. Diese bilden die Basis für die Felder des in Kapitel 4 eingeführten *Customer Interview Canvas*.

Das Eclipse Process Framework [12] stellt Informationen zum Softwareprozessen wie Open Unified Process, Scrum und Extreme Programming bereit. Die Prinzipien der agilen Softwareentwicklung sind offiziell einsehbar als „Manifesto for Agile Software Development“ [5]. Weitere Details speziell zum Vorgehen nach dem Extreme Programming bietet Beck mit seinem Buch „Extreme Programming Explained: Embrace Change“ [4]. Das in diesem Prozess zur Anwendung kommende Planning Game stellt die Grundlage für die iterative Projektplanung mit dem *Iteration Planning Canvas* (s. Kapitel 5) dar.

Unterstützend zur Planung und Durchführung der Studie zum Einsatz von Canvases (s. Kapitel 6) liefert Porst [26] praktische Hinweise und eine Übersicht über die Möglichkeiten zur Erstellung von Fragebögen.

Feature-Modelle, eingeführt von Kang et al. [15] im Kontext der „Feature-Oriented Domain Analysis“ (FODA) und erweitert von Czarnecki und Eisenecker [10], kommen bei der Entwicklung eines Konfigurationsmodells für Canvases (s. Kapitel 7) zum Einsatz. Weitere Literatur zur Entwicklung von Softwareproduktlinien mit Hilfe von Feature-Modellierung bieten Pohl et al. [24] sowie Apel et al. [1].

## 1.3 Konzept

Kapitel 2 und 3 der Arbeit geben eine Einführung in das Thema Canvases und vergleichen unterschiedliche Vorlagen aus der Literatur, sowohl aus dem Themengebiet Geschäftsmodellentwicklung als auch aus dem Bereich der Softwareentwicklung.

Angesichts der Heterogenität in der Planung und Durchführung von Softwareprojekten erscheint es sinnvoll, auf die unterschiedlichen Anforderungen mit unterschiedlichen Werkzeugen einzugehen. Für eine umfassende Unterstützung von Software-Startups in Softwareprojekten werden in den Kapiteln 4 und 5 weitere Canvases eingeführt, das *Customer Interview Canvas* sowie das *Iteration Planning Canvas*. Während sich ersteres auf die Ermittlung von Kun-

denwünschen in separaten Gesprächen konzentriert, basiert das Iteration Planning Canvas auf den Grundlagen des Planning Game aus dem Extreme Programming, in dem zusammen mit einem Kundenvertreter vor Ort eine flexible Aufgabenplanung praktiziert wird. Es konzentriert sich damit vornehmlich auf die iterative Projektplanung.

Um den Nutzen der eingeführten Canvases für den Einsatz in Softwareprojekten zu evaluieren, wird im Rahmen dieser Arbeit eine Studie durchgeführt. Dazu setzen mehrere studentische Teams in der Anfangsphase eines universitären Projekts eines der beiden Canvases ein. Ein eigens entwickelter Fragebogen dient der Erhebung des individuellen Nutzungsverhaltens und dessen Wirkung. Die Planung, Durchführung und Evaluation sind Inhalt von Kapitel 6.

Ähnlich der abweichenden Einbindung des Kunden ins Projekt gibt es weitere Unterschiede zwischen Softwareprozessen, Arbeitsmodellen und Projekttypen, die die Belegung einzelner oder mehrerer Felder in einem Canvas beeinflussen. Kapitel 7 befasst sich mit der Analyse von Variabilitätsfaktoren. Darauf aufbauend wird ein Konfigurationsmodell entworfen, nach dessen Vorgabe durch Auswahl geeigneter Kriterien Canvases für den entsprechenden Anwendungsfall erstellt werden können.

Ziel der Arbeit ist, zum Konzept eines Framework beizutragen, das Canvases für die Prozesse eines Software-Startups bereitstellt. Ebenso, wie Unternehmen nach unterschiedlichen Abläufen arbeiten, sollen sich auch die jeweiligen Canvas-Vorlagen flexibel anpassen lassen. Ergänzend dazu wird in Kapitel 8 diskutiert, nach welchen Maßstäben die Vollständigkeit eines ausgefüllten Canvas bewertet werden kann. Dies birgt die Option einer Rückmeldung an Startups, einzelne Mitarbeiter und Teams oder Inkubatoren.

Abschließend werden in Kapitel 9 die Ergebnisse betrachtet, welchen Beitrag sie zu einer besseren Organisation von Startup-Unternehmen mit Canvases darstellen und wo mögliche Ansatzpunkte für weiterführende Arbeiten liegen.



## 2 Das Canvas als kollaboratives Werkzeug

Bevor es an die Betrachtung spezifischer Canvases geht, sollen zunächst die Eigenschaften des Werkzeugs dargestellt werden. Die folgenden Abschnitte beschäftigen sich mit der grundlegenden Struktur und dem Potenzial bzw. den vielseitigen Einsatzmöglichkeiten von Canvases. Ein Beispiel veranschaulicht die Anwendung.

### 2.1 Was ist ein Canvas?

Ein Canvas (engl. für Leinwand) ist grundsätzlich ein Diagramm, bestehend aus mehreren Feldern. Wie eine Leinwand stellt es selbst nur die Unterlage für eine individuelle Gestaltung dar. Im Fall des Canvas bedeutet das: Es werden in den einzelnen Feldern Stichpunkte zu Aspekten eines übergeordneten Themas gesammelt. Dabei kann nach Belieben mit bunten Zetteln oder Stiften gearbeitet werden. Zusätzlich zur Felddefinition sind häufig noch einzelne Fragen formuliert, die Hilfestellung beim Ausfüllen geben.

Der Vorteil dieses Werkzeugs liegt vor allem in der Visualisierung komplexer Sachverhalte. So können beispielsweise kollaborativ Anforderungsanalysen erstellt oder Geschäftsmodelle erarbeitet werden. Das Canvas unterstützt in seiner Eigenschaft die Kommunikation und Verständigung der Beteiligten. Durch intelligente Anordnung der Felder kann, ähnlich wie bei Mind-Maps, der natürliche Gedankengang unterstützt werden. Die Visualisierung in dieser vordefinierten Struktur hilft, den Überblick zu behalten und nichts unberücksichtigt zu lassen.

Ein Canvas wird nach Bedarf über mehrere Iterationsschritte ergänzt, korrigiert und verbessert. Es spiegelt den aktuellen Wissensstand wieder und bietet eine Alternative zu statischen, hoch formalisierten Dokumenten, in die Inhalte häufig erst in ausgereiftem Zustand übernommen werden.

Ein kleines Beispiel, wie ein Canvas für die Planung und Organisation eines Essens verwendet werden kann, bietet Abbildung 2.1. Die Felder betreffen die grundlegende Idee der kleinen Veranstaltung, die Teilnehmer und eine Auflistung der Ressourcen, darunter auch eine Liste

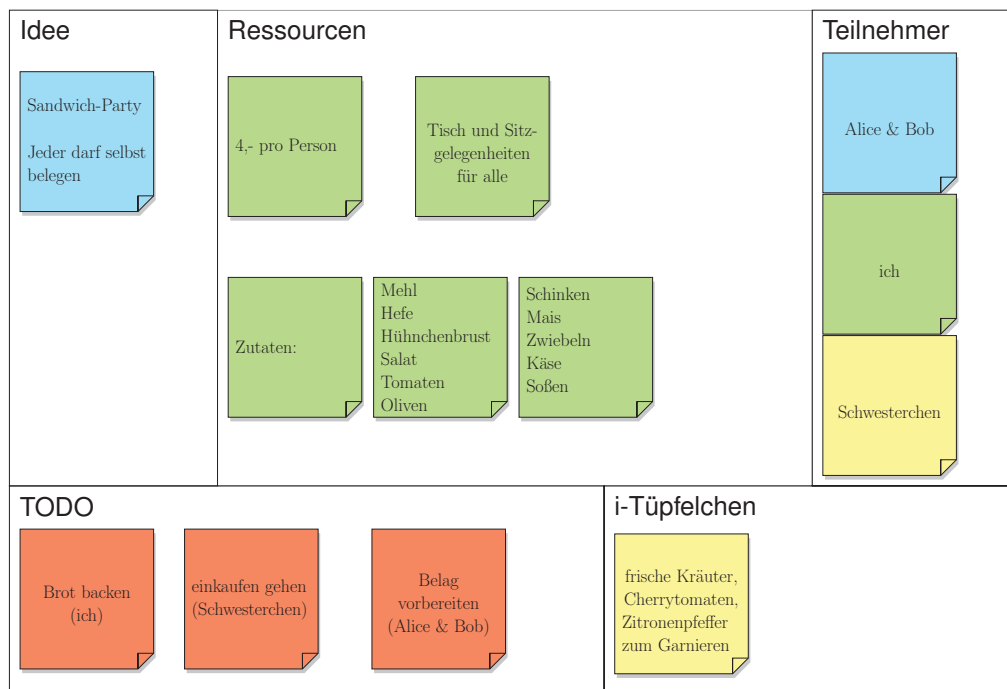


Abbildung 2.1: Canvas-Beispiel für die Planung und Organisation eines Essens

der benötigten Zutaten. Die weiteren zwei Felder sind für „das eine, besondere Detail“ - schließlich soll das Abendessen ein Erfolg werden, und die Aufgaben, die bis zur Veranstaltung noch anstehen. Als konkrete Fragestellungen für das Feld *Ressourcen* kommen beispielsweise in Betracht: Was kostet das Essen? Welche Ausstattung und welche Zutaten werden benötigt?

## 2.2 Welche Vorzüge bietet die Verwendung eines Canvas als Werkzeug?

Großes Potenzial liegt bei der Verwendung eines Canvas vor allem in der Unterstützung der Kommunikation zwischen unterschiedlichen Stakeholdern. Es fungiert als Brücke bei der gemeinsamen Entwicklung und Organisation von Ideen. Die Felder werden in natürlicher Sprache und mit weitestgehendem Verzicht auf Fachbegriffe ausgefüllt, sodass eine optimale Verständigung zwischen den Disziplinen gewährleistet ist. Notwendiges Fachvokabular wird im Zuge der Bearbeitung eingeführt.

Unabhängig von der Bearbeitung im Team, mit dem Kunden oder auch allein unterstützt ein Canvas mit seinen vordefinierten Feldern und Fragen z.B. die Analyse der Ausgangssituation, von Zielen, Voraussetzungen oder Anforderungen. Die Vorlage dient als Leitfaden für die Diskussion. Durch die Konzentration der wichtigsten Aspekte einer Themenstellung auf einer zusammenhängenden Fläche (im Gegensatz zu mehrseitigen Dokumentationen) lässt sich das

## 2.2 Welche Vorzüge bietet die Verwendung eines Canvas als Werkzeug?

---

Gesamtbild leichter erfassen.

Über den Bearbeitungsvorgang hinaus ist ein Canvas in mehrfacher Hinsicht von weiterem Nutzen. Einerseits eignet es sich als Vorlage für weiteres Vorgehen, zum Anderen als Grundlage für eine Bewertung. Außerdem ist es hilfreich für die Veranschaulichung von Konzepten und inhaltlichen Zusammenhängen gegenüber Dritten.

Eine weitere Verwendungsmöglichkeit ist die Erstellung mehrerer Canvases als Vergleichsmodelle zur Abwägung zwischen verschiedenen Alternativen.



# 3 Einsatz von Canvases in Geschäftsmodellentwicklung und Softwareprojektsteuerung

Bekannt geworden ist das Canvas durch das *Business Model Canvas* (BMC) von Alexander Osterwalder [23]. Unter der Creative Commons-Lizenz ist neben diesem eine Vielfalt an weiteren Canvases entstanden, darunter sowohl Varianten des BMC als auch Canvases für andere Einsatzszenarien. Neben der Geschäftsmodellierung wird hier ausschließlich auf den Bereich der Softwareprojektsteuerung eingegangen. Eine an der TU Dresden angefertigte Arbeit beschäftigt sich in diesem Zusammenhang mit dem *Lean Software Engineering Canvas* [21]. Im Folgenden werden einige Canvases genauer erläutert.

## 3.1 Canvases für die Entwicklung von Geschäftsmodellen

Als ein Beispiel für Canvases im Bereich Geschäftsmodellierung wird das BMC angeführt. Varianten des Originals sind einerseits das *Lean Canvas* von Ash Maurya [19], das speziell auf Startup-Gründer zugeschnitten ist, und andererseits das *Business Model Canvas for User Experience*. Weitere entstandene Canvases, auf die hier jedoch nicht näher eingegangen werden soll, sind das *Business Solution Canvas*<sup>1</sup>, das *Product Canvas*<sup>2</sup> und das *Customer Journey Canvas*<sup>3</sup> für die kundenorientierte Gestaltung von Dienstleistungen.

### 3.1.1 Business Model Canvas

Das BMC ist ein Werkzeug zur Erarbeitung von Geschäftsmodellen. Es deckt die vier wichtigsten Bereiche eines Unternehmens ab: Kunden, Angebot, Infrastruktur und finanzielle Tragfähigkeit [23]. Strukturell ist das Canvas in neun Felder aufgeteilt. Die genaue Zusammensetzung

---

<sup>1</sup>[http://technology-map.biz/index.php/Business\\_Solution\\_Canvas](http://technology-map.biz/index.php/Business_Solution_Canvas)

<sup>2</sup><http://www.romanpichler.com/tools/product-canvas/>

<sup>3</sup><http://www.thisisservicedesignthinking.com>

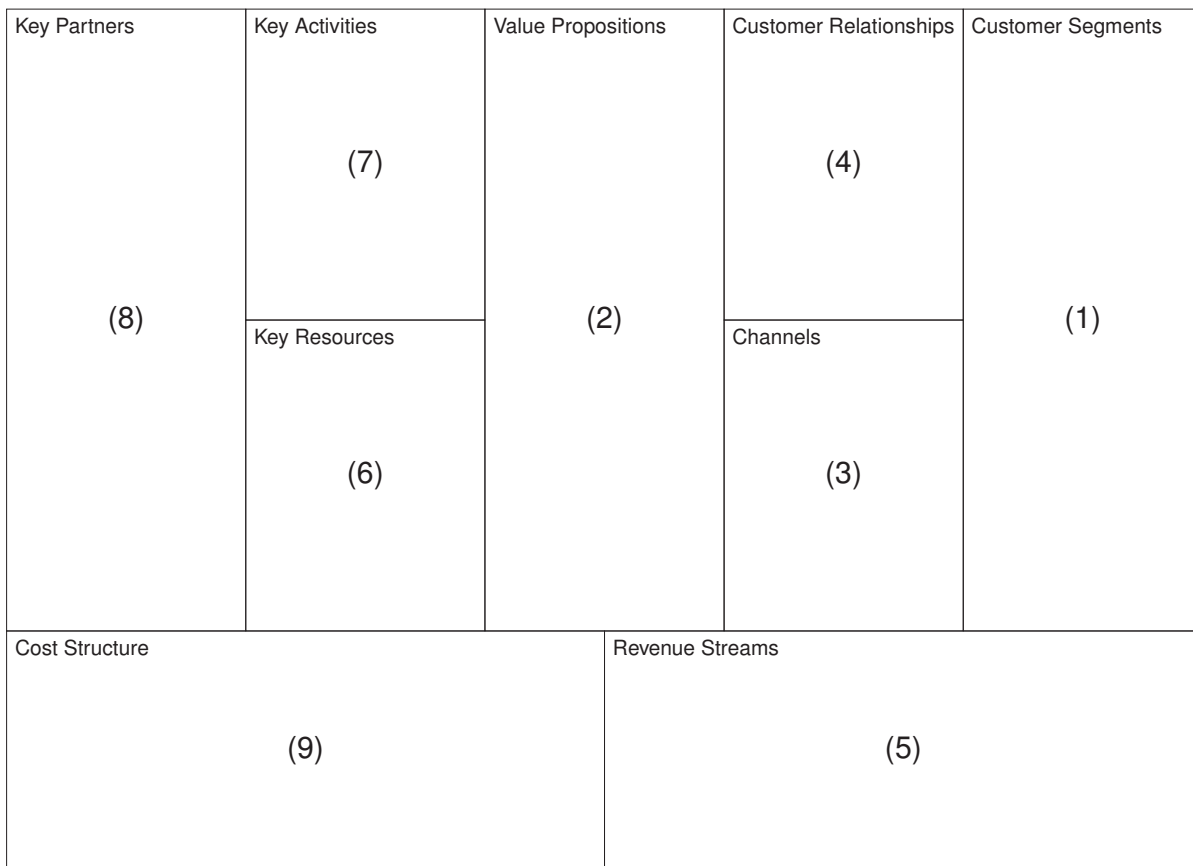


Abbildung 3.1: Business Model Canvas von Alexander Osterwalder

der Felder und ihre Anordnung sind aus Abbildung 3.1 ersichtlich. Die Reihenfolge der nachstehenden Aufzählung entspricht der vorgesehenen Ausfüllreihenfolge.

**Customer Segments** Als Kern jedes Geschäftsmodells werden die Kundensegmente genannt. Wichtig ist, nach Bedürfnissen und Verhalten zu unterteilen und von Beginn an zu definieren, an welche Zielgruppe sich das Produkt oder die Dienstleistung richten soll. Die speziellen Ansprüche der Zielgruppe können dann genau analysiert und das Geschäftsmodell ihnen angepasst werden.

**Value Propositions** Der nächste Schritt ist eine Auflistung der Vorteile, die dem Kunden geboten werden. Dies umfasst den Nutzen der Geschäftsidee, die Probleme, die sie lösen kann und wie sich die Idee quantitativ oder qualitativ vom konkurrierenden Markt abhebt.

**Channels** Fokus des dritten Feldes ist die Kommunikation zwischen Unternehmen und Kunde. Hier wird aufgeführt, über welche Medien der Kunde auf das Produkt oder die Dienstleistung aufmerksam gemacht wird, auf welchem Weg Feedback zurückkommt, wie die Beschaffung und Auslieferung zu Stande kommt und auf welche Weise Support gewährleistet wird.

**Customer Relationships** Im Feld Customer Relationships wird festgehalten, welche Art von Beziehungen zu den einzelnen Kundensegmenten gepflegt werden sollen. Die unterschiedlichen Möglichkeiten reichen dabei von persönlicher Betreuung über automatisierte Services hin zur Förderung von Online Communities und der direkten Einbindung von Kunden in die Wertschöpfung, beispielsweise durch Bewertungen und Empfehlungen.

**Revenue Streams** Für ein erfolgreiches Geschäftsmodell ist eine genaue Abschätzung der Einnahmequellen wichtig. Diese werden im Feld Revenue Streams festgehalten. Es kann sich dabei entweder um einmalige Zahlungen oder um regelmäßige Zahlungen für die Beziehung einer Dienstleistung handeln.

**Key Resources** Im nächsten Schritt werden die wichtigsten Mittel gesammelt, die für die Verwirklichung der Geschäftsidee notwendig sind. Darunter fallen sowohl materielle, als auch immaterielle Ressourcen. Sie müssen nicht notwendigerweise im Unternehmen vorhanden sein, sondern können ebenso gemietet oder von Partnern bereitgestellt werden.

**Key Activities** Das Feld Key Activities gibt eine Übersicht über die Tätigkeiten, die bei der Umsetzung der Geschäftsidee im Mittelpunkt stehen.

**Key Partnerships** Das achte Feld beschreibt die Geschäftsbeziehungen, die für den Erfolg des Geschäftsmodells wichtig sind. Es enthält ebenfalls, welche Rolle die Kontakte im Einzelnen spielen, beispielsweise die eines Zulieferers, eines Servicedienstleisters oder eines Geschäftspartners.

**Cost Structure** Nachdem die benötigten Mittel, Tätigkeiten und Geschäftsbeziehungen feststehen, werden im letzten Feld die geschätzten Kosten akkumuliert.

Die Autoren des BMC betonen, dass die Bearbeitung mit dem einmaligen Ausfüllen nicht abgeschlossen ist [23]. So kann das Canvas und das dadurch definierte Geschäftsmodell iterativ verbessert und aktualisiert werden.

### 3.1.2 Lean Canvas

Basierend auf dem BMC wurde von Ash Maurya ein weiteres Canvas entwickelt: das Lean Canvas [19]. Maurya konzentriert sich auf Risiken und Unsicherheiten als primäre Faktoren einer Startup-Gründung und präsentiert mit seiner angepassten Fassung ein Werkzeug speziell für dieses Anwendungsfeld. Erhalten bleibt dabei die Anordnung in neun Feldern. Der genaue Aufbau im Vergleich zum BMC erschließt sich aus Abbildung 3.2. Im Folgenden wird besonders auf die Unterschiede und deren Begründungen eingegangen<sup>4</sup>. Zu beachten ist, dass nach Mauryas Auffassung ein separates Canvas je Kundensegment entstehen soll. Die Reihenfolge der Aufzählung entspricht der vorgesehenen Ausfüllreihenfolge.

---

<sup>4</sup>vgl. <http://practicetrumpesttheory.com/2012/02/why-lean-canvas/>





**Cost Structure** Vgl. BMC *Cost Structure*. Zusätzlich zu den Kosten enthält das Feld gemäß Mauryas Entwurf auch die Angabe der Gewinnschwelle unter Berücksichtigung der Einnahmequellen.

**Revenue Streams** Vgl. BMC *Revenue Streams*.

**Key Metrics** Vgl. BMC Key Activities. Gemäß der abweichenden Benennung geht es Maurya bei diesem Feld nicht nur um die essentiellen Tätigkeiten, die mit der Umsetzung des Geschäftsmodells verbunden sind, sondern darum, woran letztendlich der Erfolg gemessen werden kann.

**Unfair Advantage** Um hervorzuheben, wie wichtig ein echter Wettbewerbsvorteil für die erfolgreiche Startup-Gründung ist, sieht Mauryas Entwurf dafür ein eigenes Feld vor.

An die Erstellung mehrerer Lean Canvases für die einzelnen Kundensegmente schließt sich die Auswahl an, welches Modell sich für den Start am besten eignet. Als Kriterien sind dafür genannt: Welche Zielgruppe benötigt das Produkt oder die Dienstleistung am dringendsten? Welches Kundensegment lässt sich am leichtesten ansprechen? Mit welcher Zielgruppe kann der höchste Gewinn erzielt werden? Welches Kundensegment hat den größten Marktanteil bzw. einen Marktanteil, groß genug für den erfolgreichen Vertrieb des Produkts oder der Dienstleistung?

Für die Bearbeitung des Lean Canvas betont Maurya, dass nicht die „richtige“ Antwort wichtig ist, sondern eine zügige Bearbeitung nach bestem Wissen [19]. Können einige Felder bzw. Fragen noch nicht beantwortet werden, lässt sich dieses auch zu einem späteren Zeitpunkt nachholen. Die Bearbeitung ist nicht mit dem ersten Ausfüllen abgeschlossen.

### 3.1.3 Business Model Canvas for User Experience

Neben Mauryas Lean Canvas sind aus dem BMC weitere Varianten entstanden. Im Folgenden wird das Business Model Canvas for User Experience [17] vorgestellt. Bei dieser Variante ist im Gegensatz zum Lean Canvas nicht die Spezialisierung für einen bestimmten Anwendungsbereich die Grundlage, sondern der Versuch, die Verwendung des Canvas durch andere Anordnung und kleine inhaltliche Änderungen intuitiver zu gestalten. Abbildung 3.3 zeigt die entstandene Vorlage.

Kromer kritisiert am BMC in erster Linie die Nutzbarkeit und möchte mit seiner Variante ein Werkzeug teilen, das sich ohne mehrfachen Bezug auf eine Anleitung bearbeiten lässt. Er listet dazu die konkreten Verbesserungsvorschläge mit ihrer Begründung auf und wendet diese Schritt für Schritt auf die Vorlage an.

1. Ausfüllreihenfolge vs. Aufbau: So wie die Analyse der Kundensegmente der erste Schritt beim Ausfüllen des BMC ist, soll auch das entsprechende Feld gleich am „Anfang“, also links, stehen. Das gesamte Canvas wird einmal gespiegelt.

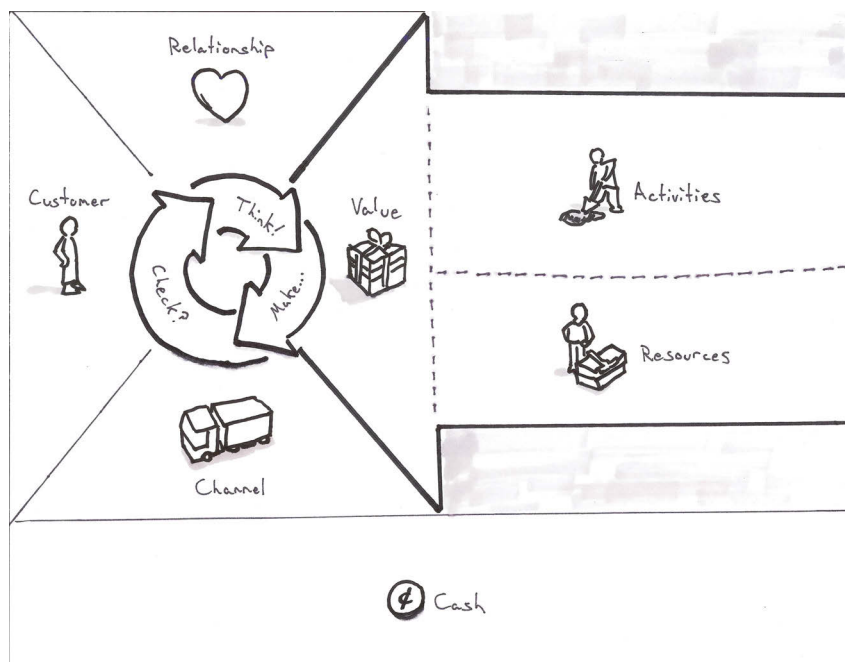


Abbildung 3.3: Business Model Canvas for User Experience von Tristan Kromer [17]

2. Beziehung zwischen Kunde und Produkt: Kromer zitiert Eric Ries' „Build-Measure-Learn loop“ zur iterativen Verbesserung eines Produkts auf der Basis von Nutzer-Feedback. Um diese im Canvas zu berücksichtigen, fügt er den Feldern *Customer*, *Relationship*, *Value* und *Channel* einen geschlossenen Kreis aus Pfeilen mit den Beschriftungen *Think!*, *Make...* und *Check?* hinzu.
3. Ressourcen und Partner: Kromer stellt fest, dass das Feld *Key Partners* im Allgemeinen bei der Nutzung des BMC-Werkzeugs zu groß ausfällt. Seine weitergehende Überlegung diesbezüglich ist, dass auch ein Geschäftspartner letztendlich eine Ressource darstellt. Der Entwurf ist deshalb um das entsprechende Feld reduziert.
4. Ausgaben und Einnahmen: Kromer argumentiert, dass sich Kosten und Einnahmen am besten gemeinsam betrachten lassen und führt die beiden Felder in seinem Entwurf zusammen.

## 3.2 Canvases für die Steuerung von Softwareprojekten

Neben unterschiedlichen Canvases für die Geschäftsmodellentwicklung sind Canvases entstanden, die speziell auf den Softwarebereich zugeschnitten sind. Ein an der TU Dresden entstandenes Werkzeug ist das *Lean Software Engineering Canvas* von Manuel Oddoy [21].



**Use Cases** Im nächsten Schritt werden unterschiedliche Handlungsmöglichkeiten der zuvor definierten Stakeholder abstrakt beschrieben. Ergeben sich dabei weitere Gruppen oder Gemeinsamkeiten zwischen Gruppen, wird das vorherige Feld entsprechend aktualisiert.

**Crosscutting** Hier stehen die Anforderungen an das System. Diese können sowohl technischer, als auch architektonischer oder organisatorischer Art sein.

**Scenarios** Dieses Feld enthält die konkrete Beschreibung von Handlungsabläufen bezogen auf den jeweiligen Use Case, welche im Projektverlauf als Abnahmetests fungieren sollen.

**Domain Model** Das Domänenmodell beschreibt die Objekte der Domäne und ihre Abhängigkeiten untereinander. Die vorgeschlagene Art der Darstellung ist ein UML-Klassendiagramm.

**Business Rules** Das Feld *Business Rules* enthält eine genaue Beschreibung der Regeln, denen das System entsprechen muss. Darunter versteht sich auch das Verhalten in Fehlerfällen.

**Components** Im letzten Feld wird die Systemarchitektur und die Aufteilung der Funktionalitäten unter den einzelnen Komponenten beschrieben.

Entscheidend ist, dass aus allen Stakeholdergruppen Mitarbeiter an der Bearbeitung teilhaben und sich so ein allgemeines Verständnis des dargestellten Sachverhalts etabliert. Bei der kollaborativen Erstellung des Lean Software Engineering Canvas sollen durch Diskussion und Rückfragen möglichst viele Unklarheiten beseitigt werden.

## 4 Erarbeitung von Kundenwünschen mit Canvases

Wie bei der Entwicklung eines Geschäftsmodells stehen auch bei der Entwicklung eines Softwareprodukts die Bedürfnisse des Kunden im Vordergrund. Entsteht das Produkt im direkten Auftrag des Kunden, bedeutet das viel Zeit und Verhandlungen über gewünschte Funktionalitäten. Unterstützend zur Erarbeitung von Kundenwünschen im Gespräch wird in diesem Abschnitt das *Customer Interview Canvas* eingeführt. Es stellt zum einen eine Form der Dokumentation dar und dient zum anderen der Entwicklung eines besseren Verständnisses zwischen Kunde und Dienstleister.

### 4.1 Die Einteilung von Kundenwünschen nach dem Kano-Modell

Auch hier gilt, dass mithilfe verschiedener Strategien mit dem Kunden seine Vorstellungen und Anforderungen erörtert werden müssen. Rupp und Schüpferling [28–31] beziehen sich dabei auf das 1978 von Noriaki Kano eingeführte *Kano-Modell* und unterteilen in Basis-, Leistungs- und begeisternde Faktoren, die es zu erfüllen gilt. Dem gegenüber werden unterschiedliche Techniken propagiert, die auf die Erfassung der jeweiligen Faktoren abzielen. Während sich der Kunde beispielsweise der Leistungsfaktoren sehr bewusst ist und diese auf Nachfrage nennen kann, sind Basisfaktoren für ihn selbstverständlich und können leicht übersehen werden. Das Entwicklerteam, das sich in der Domäne nicht vollständig auskennt, setzt diese Faktoren jedoch nicht selbstverständlich voraus und ist auf die Spezifikation angewiesen. Die begeisternden Faktoren wiederum sind für den Kunden ebenso wenig formulierbar, denn es handelt sich dabei um Lösungen von Problemen, denen sich der Kunde noch gar nicht bewusst ist.

Die Erfüllung der Basisfaktoren macht ein Produkt für den Kunden überhaupt erst nutzbar. Die Leistungsfaktoren müssen erfüllt sein, damit der Kunde sich für dieses Produkt und gegen die Konkurrenz entscheidet. Die Begeisterungsfaktoren allerdings bedingen am Ende die Zufriedenheit und langfristige Freude am Produkt.

Um Basisanforderungen zu erheben, ist es sinnvoll, sich mit den Altsystemen zu befassen. Man kann den Kunden beim Gebrauch des Systems beobachten, das System selbst ausprobieren oder sich ihm mithilfe entsprechender Dokumentation nähern. Sind die Entwickler mit dem System und der Domäne schließlich vertraut, können sie Verbesserungsvorschläge machen, den Bedarf einzelner Funktionalitäten im Rahmen eines Kundengesprächs abklären und es wird weniger wahrscheinlich, dass sie etwas implementieren, das dem Anwendungszweck gänzlich widerspricht. Ist kein Altsystem vorhanden, können allenfalls Systeme betrachtet werden, die ähnliche Anwendungsfelder abdecken. Dann kann es außerdem ratsam sein, verstärkt mit Prototypen zu arbeiten, an denen der Kunde Vorzüge und Defizite klarer erkennen kann.

Für die Besprechung der Leistungsfaktoren eignet sich gemäß Rupp et al. [28–31] das Interview. Hier beantwortet der Kunde direkte Fragen. Diese müssen jedoch vorbereitet sein. Auf Basis bereits vorliegender Informationen sollte vor dem tatsächlichen Gespräch ein „grobe[r] Leitfaden mit einem Katalog aus Fragen“ [29] ausgearbeitet sein. Als wichtiger Punkt für das Gespräch ist vorausgesetzt, dass der Kunde über den aktuellen Projektstand informiert ist. Um nach Besprechung einer Sache zu überprüfen, ob die Antwort korrekt verstanden wurde, sollte diese vom Fragesteller noch einmal selbst formuliert werden. Ein Eintrag wird also ggf. mehrmals diskutiert und erst vorgenommen, wenn er vom Kunden bestätigt wurde. Außerdem kann die jeweilige Frage mit Folgefragen zu weiteren Details oder Aspekten erweitert werden. Abschließend nach dem Gespräch werden die Erkenntnisse nach Bedarf noch einmal strukturiert und gegebenenfalls Teile aussortiert oder für ein weiteres Gespräch vorgemerkt.

Als dritte Herausforderung gilt es, die begeisternden Faktoren zu erfassen. Rupp und Schüpferling bewerben für diese Aufgabe sogenannte „Kreativitätstechniken“, die geeignet sind, innovative Lösungen zu entwickeln. Eine davon ist das Brainstorming. Ideen werden dabei gesammelt und erst nach Abschluss dieses Prozesses diskutiert und weiter ausgearbeitet oder aber verworfen. Umgekehrt werden beim „Brainstorming paradox“ Ideen gesammelt, gegen die sich das Produkt schließlich einmal abgrenzen soll. Damit werden sowohl die Ziele als auch die zu vermeidenden Risiken definiert.

Wie lassen sich diese Aspekte in ein geeignetes Werkzeug umsetzen? Aus dem Beschriebenen können folgende Elemente extrahiert werden:

1. Das Entwicklerteam benötigt Vorwissen über die Domäne und die grobe Funktionalität der Anwendung durch Auseinandersetzung mit Altsystemen oder Anwendungen, die auf den selben Bereich abzielen
2. Im Gespräch werden die beobachteten Eigenschaften der Software diskutiert und gegebenenfalls an Verbesserungswünsche angepasst oder berichtigt
3. Für das Gespräch sind entsprechende Fragen vorzubereiten, auf welche im Verlauf Antworten formuliert und verfeinert werden
4. Der Kunde erhält einen Überblick über den gegenwärtigen Projektstand
5. Das Gespräch bietet Raum für ein Brainstorming über die Ziele sowie die gegensätzlichen Ziele des Projekts, wodurch das Projekt von mehreren Seiten abgegrenzt werden kann

Goal/Paradox Inverse Goal	System Users	Project Status Within Schedule	Domain Model	Success Indicators/ Criteria
	Resources		Risks	
Questions		Answers		

Abbildung 4.1: Die Felder des Customer Interview Canvas

## 4.2 Das Customer Interview Canvas

Um diesen Anforderungen zu entsprechen, wurde das Customer Interview Canvas (s. Abbildung 4.1) entwickelt. Die Annahmen über Domäne und Anwendung findet sich in den Feldern *Domain Model* und *Risks* wieder und werden zusammen mit dem Kunden validiert. Weiterführende Fragen können in das Feld *Questions* übernommen werden. Dieses enthält auch die vorbereiteten Fragen und kann im Gespräch nach Bedarf ergänzt werden. Überblick über die wichtigsten Punkte zum Projekt geben die Felder *Project Status Within Schedule*, *System Users* und *Resources*. Eine Annäherung an die Anforderungen von kreativer Seite ermöglicht das weitgefaste Feld *Goal/Paradox Inverse Goal*. Die gesammelten Punkte aus diesem Feld können im weiteren Verlauf die Feststellung der Erfolgskriterien (*Success Indicators/Criteria*) sowie die Identifikation weiterer Risiken beeinflussen. Nachstehend sind die Felder des Customer Interview Canvas näher erläutert.

**Project Status within Schedule** Als Ausgangsgrundlage sollte der Kunde über den aktuellen Status des Softwareprojekts im Bild sein. Dazu gehört ein grober Zeitplan mit wichtigen Terminen, ähnlich wie bei einem ein Gantt-Diagramm, und der aktuelle Stand in Relation zu den festgelegten Meilensteinen.

**Goal/Paradox Inverse Goal** Sowohl für das Projektteam als auch für den Kunden müssen die Projektziele klar definiert sein. In manchen Fällen ist es der Eindeutigkeit förderlich, auch

komplementäre Ziele zu nennen, insbesondere wenn das Ziel schwer festzumachen ist, aber sich durch letztere gut abgrenzen lässt. Dabei sollte soweit wie nötig, aber nicht zu sehr ins Detail gegangen werden.

**System Users** Um erfolgreich zu sein, muss ein Produkt die Anforderungen all seiner Systemnutzer erfüllen. Hier werden die Nutzergruppen aufgeführt, denen jeweils bestimmte Funktionalitäten der Software zur Verfügung stehen.

**Resources** Zu den Anforderungen eines Softwareprodukts gehören nicht nur die Funktionalitäten, die von der fertigen Software abgedeckt sein müssen, sondern auch, dass die vereinbarten Ressourcen eingehalten werden. Letztere werden hier festgehalten - zu ihnen zählen Zeit und Geld sowie das Projektteam mit seinen Erfahrungen und Kompetenzen.

**Domain Model** Als Diskussionsgrundlage sollte Klarheit über die Grundstruktur und Objekte innerhalb der Projektdomäne herrschen. Diese werden in einem Domänenmodell dargestellt. Muss das Modell im Verlauf des Projekts erweitert werden, kann anhand der Übersicht über die strukturelle Integration und möglicherweise damit zusammenhängende Einzelheiten der Funktionalität diskutiert werden.

**Risks** Um die Vorstellungen und Ansprüche des Kunden zu treffen, sollte eine eindeutige Gewichtung von Risiken vorliegen. Wenn ein System beispielsweise maximale Sicherheit bieten soll und diese bedingt Einbußen bei der Benutzerfreundlichkeit und Performance, können die Einbußen durch eine gegebene hohe Gewichtung von Sicherheitsrisiken gerechtfertigt sein. Andererseits ist der Kunde mit der fachlichen Seite der Domäne besser vertraut und kann wertvolle Hinweise liefern, worauf im Projekt Rücksicht zu nehmen ist.

**Success Indicators/Criteria** Beide Parteien sollten sich darüber im Klaren sein, worüber sich der Erfolg des Projekts definiert. Um dem Kunden einen Einblick zu ermöglichen, inwieweit die entsprechenden Kriterien erfüllt sind, einigen sich Kunde und Projektteam auf Maße, die Fortschritt oder Erfüllungsgrad widerspiegeln.

**Questions** Vor und beim Ausfüllen des Canvas ist es natürlich, dass auf beiden Seiten weitere Fragen aufkommen. Diese sollten hier gewissenhaft festgehalten werden. Da nicht alles rund um das Projekt in einem Gespräch geklärt werden kann, sollten die Fragen in einen Bereich fallen, der für das Gespräch vereinbart wurde.

**Answers** Offene Fragen sollten im Kundengespräch eindeutig geklärt werden. Das Projektteam sollte von sich aus die Antwort des Kunden bzw. wie sie verstanden wurde formulieren können. Eine Antwort wird erst aufgenommen, wenn der Kunde sie als korrekt verstanden bestätigt. Dabei wird ein erweiterter Diskussionsbedarf in vielen Fällen aufgedeckt und Unklarheiten können frühzeitig beseitigt werden.



# 5 Iterationsplanung im Softwareprojekt mit Canvases

Das Lean Software Engineering Canvas stellt zwar ein gutes Werkzeug für eine erste Anforderungsanalyse dar, es werden damit aber nicht speziell die agilen Prozessen eigenen Iterationsschritte unterstützt. Diese Funktion soll deshalb das sogenannte *Iteration Planning Canvas* erfüllen. Es ist angelehnt an das im Extreme Programming verankerte *Planning Game* [11] und bietet sowohl die Basis für eine gute Kommunikation bzw. Kollaboration zwischen Entwicklerteam und Kunden als auch eine persönliche Referenz für jeden Entwickler während der gesamten Iteration. Dazu ist das Canvas aufgeteilt in Iterationen übergreifende Felder, die den größeren Zusammenhang abbilden sollen, iterationspezifische Felder, die sich detaillierter mit den für die entsprechende Iteration anstehenden Aufgaben befassen, sowie persönliche Felder, die von jedem Entwickler individuell ausgefüllt werden (da sie vom Detailgrad speziell für ihn relevant sind) und die die Übersicht über die Iteration vervollständigen sollen. Stellt man nun die Canvases aller Iterationsschritte in eine Reihe, ergibt sich eine übersichtliche Dokumentation des Entwicklungsprozesses.

## 5.1 Das Planning Game

Das Planning Game unterliegt der Überzeugung, dass eine gute Planung nicht ohne fortlaufende Verfeinerung auskommt. Es stellt eine Praktik des Extreme Programmings dar und wird hier gemäß [12] dargestellt.

Der Softwareentwicklungsprozess wird unterteilt in mehrere Zyklen. *Releases* gehen über einen Zeitraum von zwei bis drei Monaten und sind wiederum in *Iterations* der Länge von zwei bis drei Wochen gegliedert. Die Zeitspannen stellen hier eine Empfehlung über die jeweilige Länge dar, welche über das Projekt konstant bleiben soll. Grund dafür ist, dass für die Aufgabenplanung die *Team Velocity* (engl. für Team-Geschwindigkeit) gemessen werden muss. Diese wird über den Entwicklungsaufwand bestimmt, welchen das Projektteam innerhalb der festen Zyklen bewältigt. Die Zeitachse fungiert demnach als Konstante für vergleichbare Messwerte.

Die ermittelte Geschwindigkeit aus dem letzten Zyklus dient der Festlegung des Aufwands für

den nächsten Zyklus. Hierfür werden im Planning Game *Story Points* verwendet, die von den Programmierern für den geschätzten Aufwand der vom Kunden erstellten *User Stories* vergeben werden. *User Stories* beschreiben Aktionsmöglichkeiten in Bezug auf einen Systemnutzer und die daraus hervorgehende Wirkung. Das Konzept erlaubt sehr unterschiedliche Detailgrade. Bei besonders umfangreichen *User Stories* ist es häufig möglich und auch nötig, diese noch einmal in kleinere Geschichten zu zerlegen.

Die Erstellung von *User Stories* kann jederzeit erfolgen, die Aufwandsschätzung und Auswahl von *Stories* für den nächsten Zyklus entsprechend der Team Velocity sind Bestandteil des *Release Plannings*. Zum *Release Plan* gehört eine grobe Beschreibung der Funktionalitäten des Produkts, die innerhalb des Release implementiert werden sollen (und können). Er wird im Verlauf nach Bedarf aktualisiert.

Die *Iteration Plans* beinhalten die kurzfristigen, taktischen Ziele des Projekts und liefern Feedback in kurzen Abständen. Im *Iteration Planning* zu Beginn jeder Iteration wählt der Kunde aus den *Stories* des aktuellen Release eine Teilmenge mit entsprechendem Aufwand für den anstehenden Zyklus. Die *User Stories* werden von den Entwicklern in *Tasks*, d.h. technisch trennbare Aufgaben, gespalten. Die Aufgaben werden selbstständig im Team verteilt, bis jeder gemäß eigener Schätzung sich für einen der Iteration angemessenen Stundenaufwand eingetragen hat. Dabei spielt die persönliche Erfahrung und Qualifikation für das jeweilige Aufgabengebiet eine Rolle. Es ist üblich, dass der Stundenaufwand für eine Aufgabe individuell unterschiedlich geschätzt wird. Für die Flexibilität des Teams ist prinzipiell wichtig, dass die Entwickler die Möglichkeit haben, in unterschiedlichen Bereichen Erfahrungen und Expertise zu gewinnen.

*User Stories*, die am Ende einer Iteration nicht vollständig sind, werden abgebrochen und bei der Messung der Team Velocity nicht gewertet. Ziel ist, am Ende jedes Zyklus an den Kunden ein funktionierendes Produkt auszuliefern. Ob die geforderte Funktionalität erfüllt ist, wird anhand von *Acceptance Tests* überprüft. Diese sind vom Kunden zu spezifizieren und bieten dem Entwicklerteam Auskunft über die Details, die in den *User Stories* keine Berücksichtigung finden.

## 5.2 Das Iteration Planning Canvas

Das Iteration Planning Canvas (s. Abbildung 5.1) hilft dem Projektteam bei der Planung eines Iterationszyklus. Für den größeren Überblick dient das Feld *Release Plan*. Daneben werden im Feld *Previous Achievements* die für die Aufwandsplanung benötigten Zahlen aus vorherigen Iterationen gesammelt. Die Funktionalitäten werden vom Kunden in *User Stories* verpackt. Diese werden in *Tasks* aufgeteilt und mit einer individuellen Aufwandsschätzung (*Personal Estimates*) versehen. Während der Kunde für die Formulierung von *Acceptance Tests* zuständig ist, tragen die Entwickler ihre ausgewählten Aufgaben in ihr Feld *Personal Subscriptions* ein und ergänzen das Canvas damit zu ihrer persönlichen Referenz. Bei der Verteilung der Aufgaben mit Blick auf die Flexibilität des Teams helfen außerdem die Felder *Special Qualifications* und

Acceptance Tests	Previous Achievements	Release Plan	Special Qualifications	Personal Subscriptions
	Stories		Sought Experience	
Tasks			Personal Estimates	

Abbildung 5.1: Die Felder des Iteration Planning Canvas

*Sought Experience.* In diesen finden die Erfahrungen und Qualifikationsziele der Entwickler Berücksichtigung. Die einzelnen Felder sind nachfolgend genauer spezifiziert.

**Release Plan** Beim Extreme Programming umfasst ein Release mehrere Iterationen. In diesem Feld werden Informationen über die geplanten Features, den Umfang in Story Points (gemessen an früheren Releases) und ggf. bereits verfasste Stories gesammelt, aus denen für neue Iterationen geschöpft werden kann.

**Previous Achievements #** Das Feld enthält die Zahlen zur vorherigen Iteration: Wie viele Story Points wurden gesamt absolviert? Wie viele Stunden Arbeit am Projekt wurden nach Schätzung geleistet? Die Werte bestimmen das Aufgabenpensum für die neue Iteration. Zu Beginn, wenn noch nicht auf vergangene Iterationen zurückgegriffen werden kann, werden die Werte vom Entwicklerteam geschätzt.

**Stories** Im Planning Game bzw. im Extreme Programming werden User Stories verwendet, um Programmfunktionalitäten zu formulieren. Eine User Story ist von der Form „Als [Systemnutzer (Typ)] kann ich [Aktion], sodass [Wirkung].“ Das Feld enthält die Auswahl für die aktuelle Iteration, abgestimmt auf das aus früheren Iterationen ermittelte Pensum. Die Auswahl wird beeinflusst von der für den Kunden geschätzten Wichtigkeit und vom Entwicklerteam geschätzten Aufwand.

**Acceptance Tests** Das Mittel zur Verifikation der Auftragserfüllung sind im Extreme Pro-

gramming Akzeptanztests. Der Kunde ist verpflichtet, die geforderte Funktionalität in User Stories durch Akzeptanztests genauer zu spezifizieren. Dies hilft dabei, die Details den Entwicklern gegenüber zu kommunizieren und schafft eine Grundlage für automatisiertes Testen.

**Special Qualifications** Ein wesentliches Merkmal des Extreme Programming-Ansatzes ist, dass einzelne Teammitglieder flexibel Aufgaben übernehmen können. Es gibt keine Experten für die einzelnen Aufgaben, sondern es kann prinzipiell jeder beliebig Aufgaben übernehmen. Das Team bleibt damit flexibel einsetzbar, es kann zu jedem Problem einen fachlichen Austausch geben, und einzelne Teammitglieder können sich leichter auch neue Techniken aneignen, wenn es für ein Projekt erforderlich ist. Dabei wird auch der Wahl schlechter Designentscheidungen auf Grund von Gewohnheiten vorgebeugt. Das Feld dient der Einschätzung der aktuellen Kompetenzen der Teammitglieder.

**Sought Experience** Das Feld stellt das Gegenstück zu den aktuellen Kompetenzen im Team dar und enthält zu jedem Mitglied einen oder mehrere Bereiche, in denen dieses bereit ist, sich fachlich zu vertiefen, um weitere Kompetenzen zu erlangen bzw. schwächere Kompetenzen auszugleichen. Diese Gesichtspunkte werden nach Möglichkeit bei der Aufteilung von Aufgaben berücksichtigt. Ziel ist ein flexibel einsetzbares, gut qualifiziertes Team. Es werden jedoch keine Aufgaben vorgeschrieben sondern auf Respekt und Selbstorganisation gesetzt mit dem Standpunkt, dass freiwillig gewählte Aufgaben auch besser erledigt werden.

**Tasks** Die für die Iteration ausgewählten User Stories werden vom Entwicklerteam auf technische Aufgaben der Größenordnung von einem Arbeitstag heruntergebrochen. Das kann zum Beispiel die GUI für ein Funktionselement oder das Datenbankmodell für eine größere Komponente sein. Unter Umständen können auch Teilaufgaben aus mehreren Stories in einem Task zusammengefasst werden.

**Personal Estimates** Der Aufwand der einzelnen Tasks wird von jedem Entwickler für sich geschätzt. Zusätzlich sollten sich alle darüber im Klaren sein, wie viel Zeit sie in der geplanten Iteration für das Projekt zur Verfügung haben werden. Basierend auf den Schätzungen, den aktuellen Kompetenzen und der gesuchten Erfahrung werden die Aufgaben untereinander aufgeteilt.

**Personal Subscriptions** Die bei der Aufteilung persönlich ausgewählten Tasks werden in das Feld *Personal Subscriptions* des jeweiligen Entwicklers übernommen. Die Aufteilung selbst geschieht möglichst so, dass neue Erfahrungen gesammelt, das angesetzte Pensum erfüllt und die Budgets der einzelnen Entwickler eingehalten werden können. Sind die ausgewählten Aufgaben noch nicht verteilt und alle Budgets ausgeschöpft, können Aufgaben so umverteilt werden, dass sie schneller erledigt werden können und Kapazitäten für die übrigen Aufgaben verbleiben. Es sollte jedoch für jedes Teammitglied regelmäßig die Möglichkeit bestehen, an der Erweiterung ihrer Kompetenzen zu arbeiten (vgl. Sought Experience). Ist dies nicht der Fall, kann der Kunde gebeten werden, eine User Story für eine spätere Iteration zurückzuhalten.

# 6 Studie zum Einsatz von Canvases in Softwareprojekten

Der folgende Teil der Arbeit befasst sich mit einer Studie im Rahmen des universitär durchgeführten Softwarepraktikums. Um die Eignung und den Nutzen des Customer Interview Canvas für die Planung von Softwareprojekten zu evaluieren, wird dieses von mehreren Studentengruppen in der Anfangsphase eines Projekts eingesetzt und anschließend ausgewertet. Bei einer schriftlichen Befragung bzw. der Erstellung eines Evaluationsbogens ist es wichtig, formale sowie psychologische Aspekte zu beachten. Der erste Abschnitt gibt einen Überblick über Möglichkeiten und Regeln zur Erstellung als Basis, welche die Basis für die Entwicklung eines Fragebogens für den gegebenen Kontext bilden. Nachfolgend stehen die Beschreibung der Durchführung und die Diskussion der Ergebnisse.

## 6.1 Entwicklung eines Fragebogens

Der Aufbau eines Fragebogens sollte gewissen Regeln folgen, um ein valides und aussagekräftiges Ergebnis zu erzielen. Eine gute Anleitung und Praxisbeispiele hierfür bietet Porst [26]. Einerseits ist darauf zu achten, durch die formale Gestaltung das Ausfüllen möglichst unmissverständlich und intuitiv zu machen, andererseits sollten die Fragen inhaltlich verständlich und zielführend sein. Als Orientierung für die Formulierung der Fragen im Einzelnen verweist Porst auf die „10 Gebote der Frageformulierung“ [25, 26] (s. Abbildung 6.1).

Es stehen drei Optionen für die Gestaltung von Fragen zur Verfügung:

**Offene Fragen** Der Befragte ist bei der Beantwortung der Frage frei in der Formulierung. Einerseits wird er nicht durch Antwortmöglichkeiten beeinflusst, andererseits werden die Reichweite und das Abzielen einer Frage häufig erst durch Antwortbeispiele richtig klar. Die Auswertung ist etwas schwieriger. Offene Fragen sind sinnvoll, wenn es ein breites Spektrum an möglichen Antworten gibt oder die Antwortmöglichkeiten nicht hinreichend bekannt sind.

**Geschlossene Fragen** Hier sind Antworten vorformuliert und es wird vom Befragten nur noch die Passende ausgewählt. Als Ausweichmöglichkeit, wenn der Befragte keine der Ant-

1. Du sollst einfache, unzweideutige Begriffe verwenden, die von allen Befragten in gleicher Weise verstanden werden!
2. Du sollst lange und komplexe Fragen vermeiden!
3. Du sollst hypothetische Fragen vermeiden!
4. Du sollst doppelte Stimuli und Verneinungen vermeiden!
5. Du sollst Unterstellungen und suggestive Fragen vermeiden!
6. Du sollst Fragen vermeiden, die auf Informationen abzielen, über die viele Befragte mutmaßlich nicht verfügen!
7. Du sollst Fragen mit eindeutigem zeitlichen Bezug verwenden!
8. Du sollst Antwortkategorien verwenden, die erschöpfend und disjunkt (überschneidungsfrei) sind!
9. Du sollst sicherstellen, dass der Kontext einer Frage sich nicht (unkontrolliert) auf deren Beantwortung auswirkt!
10. Du sollst unklare Begriffe definieren!

Abbildung 6.1: Die „10 Gebote der Frageformulierung“ [25]

worten als passend erachtet, können zum Teil Antworten der Art „keines davon“ oder „kann ich nicht beurteilen“ gewählt werden. Damit wird der Befragte nicht gezwungen, eine falsche Antwort zu geben.

**Halboffene Fragen** Ähnlich zur geschlossenen Frage sind Antworten vorformuliert. Es besteht jedoch die Möglichkeit, eine eigene Antwort hinzuzufügen, sollte sich die Befragungsperson mit keinem der Vorschläge identifizieren können.

Bei geschlossenen Fragen lässt sich weiterhin zwischen *Einfachnennungen* und *Mehrfachnennungen* unterscheiden, die sich entsprechend in ihrer Präsentation unterscheiden sollten. Wird eine Frage geschlossen formuliert, gibt es mehrere Möglichkeiten, den Antwortbereich aufzuschlüsseln:

**Nominal-Skala** Bei der Nominal-Skala hat der Befragte zwei oder mehr Antwortmöglichkeiten zur Auswahl, die sich gegenseitig ausschließen.

**Ordinal-Skala** Die Ordinal-Skala besteht ebenfalls aus disjunkten Antwortmöglichkeiten, zwischen ihnen besteht allerdings eine klare Ordnung (z.B. „ja“ – „eher ja“ – „eher nein“ – „nein“).

**Intervall-Skala** Bei der Intervall-Skala sind die Antwortmöglichkeiten geordnet und befinden sich zusätzlich in gleichen Abständen zueinander.

**Ratio-Skala** Ratio-Skalen gleichen in ihren Eigenschaften Intervall-Skalen, haben aber zusätzlich einen *echten Skalennullpunkt* (z.B. „-1“ – „0“ – „1“).

Die Auswertung hängt in vielen Fällen von der Unterteilung in Intervalle ab. Diese lassen sich bei einer Ordinal-Skala wie der im Beispiel angegebenen am besten konstruieren, indem nur die Endpunkte benannt werden. Das ist insbesondere dann sinnvoll, wenn die Skala breiter ist, da eine sinnvolle Benennung der einzelnen Punkte mit zunehmender Breite schwieriger wird. Grundsätzlich empfiehlt Porst [26] eine Skalenbreite zwischen 5 und 7 für endpunktbenannte Skalen bzw. zwischen 4 und maximal 6 für Skalen mit Benennung der einzelnen Punkte. Ziel ist, die Befragungsperson nicht zu überfordern aber gleichzeitig ausreichend Möglichkeiten zur Differenzierung zu bieten. Bei der Skalenbreite ist ebenfalls abzuwägen zwischen einer geraden bzw. einer ungeraden Anzahl an Skalenpunkten mit einem Mittelpunkt. Handelt es sich um eine gerade Anzahl, ist die Befragungsperson an eine Entscheidung für die eine oder andere Tendenz gebunden. Gibt es allerdings einen Mittelpunkt, kann dieser auch genutzt werden, um einer tatsächlichen Entscheidung auszuweichen. Besteht die Möglichkeit, dass Befragte zum entsprechenden Thema wirklich keine Aussage machen können, kann man dies bei der Konstruktion mit einer wie zuvor beschriebenen Ausweichmöglichkeit (s. Geschlossene Fragen) berücksichtigen. Bei der Benennung von End- und Skalenpunkten ist es für eine neutrale Darstellung sinnvoll, auf negative Formulierungen zu verzichten. Die einheitliche Verwendung von Skalen innerhalb eines Fragebogens erleichtert das Ausfüllen und beugt Lesefehlern vor.

Grundsätzlich gelten ähnliche Regeln wie in der Kommunikation<sup>1</sup> – so sollte der Befragte weder durch unnötige Komplexität, überflüssige Informationen, noch durch die Verwendung nicht eingeführter Fachbegriffe überfordert werden. Um eine gewissenhafte Beantwortung zu fördern, sollte der Befragte außerdem nach Möglichkeit angesprochen und motiviert werden. Ein guter Ansatz ist dafür die logische Ordnung und eine Durchmischung offener und geschlossener Fragen. Porst [26] rät ebenfalls dazu, für die Analyse notwendige, aber nicht themenspezifische, Angaben wie die Angabe von Alter, Geschlecht etc. erst zum Schluss zu erfragen, um den Befragten nicht gleich zu Beginn zu langweilen und damit willkürliche Antworten herauszufordern.

Der Fragebogen für die Evaluation zur Nutzung von Canvases in Softwareprojekten wurde unter weitestgehender Berücksichtigung der angeführten Hinweise entwickelt und ist im Anhang auf Seite 66 abgebildet. Um eventuelle strukturelle Zusammenhänge zu berücksichtigen, wurde die Anordnung der Zettel weitmöglichst aus dem Original übernommen.

## 6.2 Durchführung der Studie

In den Studiengängen Informatik, Medieninformatik, Informationssystemtechnik und dem Ergänzungsstudiengang Softwaretechnologie der TU Dresden ist nach aktueller Studienordnung ein auf die Einführungsveranstaltung Softwaretechnologie folgendes Projekt vorgesehen. Gemäß dem Ablaufplan fällt das Projekt damit auf das dritte Semester. Die Studenten haben mit Abschluss des zweiten Semesters eine Einführung in die objektorientierte Programmierung erhalten und ihre Fähigkeiten in einer Klausur gezeigt. In Gruppen von fünf Leuten bearbeiten

---

<sup>1</sup>vgl. Grice [13]



die Studenten unter Betreuung durch einen Tutor über 12 Wochen ein vorgegebenes Thema.

Für die Studie wurden fünf Gruppen ausgewählt. Eine Gruppe führt ein Projekt bei einer externen Firma durch, die anderen vier Gruppen bearbeiten ein vom Lehrstuhl gestelltes Thema. In diesem Fall vertritt der Betreuer die Position des Kunden. Die Gruppen werden nachfolgend nach Bedarf als „Gruppe 1“ (externe Gruppe) und von „Gruppe 2“ bis einschließlich „Gruppe 5“ (interne Gruppen) referenziert. Die Aufgabenstellung ist für die vier internen Praktikumsgruppen gleich und bezieht sich auf die Entwicklung einer Organisations- und Verwaltungs-Software für die Abwicklung von Bestattungen. Dazu ist unter Verwendung der Frameworks Salespoint<sup>2</sup> und Spring<sup>3</sup> eine Java-Anwendung zu erstellen, die die Mitarbeiter des (imaginären) Bestattungsinstituts in ihren unterschiedlichen Rollen bei der Arbeit unterstützt. Neben der Anwendung des Canvas werden u.a. ein Pflichtenheft erstellt sowie CRC-Karten, diverse UML-Diagramme und Gesprächsprotokolle eingesetzt.

Die Gruppen erhalten zu Beginn des Projekts einen Auszug aus der vorliegenden Arbeit zur Einarbeitung: das Einführungskapitel zum Canvas als kollaborativem Werkzeug sowie das Kapitel zum Customer Interview Canvas mit der entsprechenden Vorlage. Außerdem werden ihnen optional für die Anwendung Plakate und bunte Haftnotizen zur Verfügung gestellt. Gegenstand der Untersuchung ist vordergründig die Eignung des Canvas für den Einsatz in der Softwareentwicklung als solches und die Bewertung des Nutzens der einzelnen Felder sowie der Ausfüllhinweise und unterstützenden Fragen. Ein weiterer Gesichtspunkt ist der Grad, inwieweit sich das Canvas als leichtgewichtiges, intuitives Werkzeug beweisen kann. Auf zusätzliche Hilfestellungen im Sinne einer persönlichen Betreuung wird aus letztgenanntem Grund verzichtet. Die Auswertung der Ergebnisse folgt im nächsten Abschnitt.

### 6.3 Auswertung der Studie

Die Evaluation erfolgt auf Grundlage des Fragebogens, formlosen Feedbacks und konkreter Arbeitsergebnisse. Leider liegt nicht von allen teilnehmenden Gruppen eine abschließende Beurteilung vor. Von einer detaillierten Auswertung der Fragen wird deshalb zugunsten eines Vergleichs der ausgefüllten Canvases abgesehen. Von allen Gruppen wurde das Customer Interview Canvas (siehe Anhang S.62) bearbeitet. Bis auf Gruppe 1, die das Canvas digital ausfüllte, wurde dazu das bereitgestellte Material genutzt. Abbildung 6.2 zeigt das digitalisierte Ergebnis von Gruppe 5.

Auf den ersten Blick fallen die beiden leeren Felder *Domain Model* und *Answers* auf, welche auch von den anderen Gruppen nicht oder nur teilweise genutzt wurden. Eine mögliche Begründung dieser Auslassungen ist die Überschneidung mit anderen Methoden (s.a. Durchführung der Studie). Da die Studenten bereits vor Einsatz des Canvas CRC-Karten eingesetzt hatten, bot das Feld *Domain Model* in diesem Fall keinen signifikanten Nutzen. Gruppe 3

---

<sup>2</sup>Version 6, s. <http://st.inf.tu-dresden.de/SalesPoint/> [Aufruf 30.11.14]

<sup>3</sup>Version 4.1 mit Spring Boot 1.2, s. <http://spring.io/> [Aufruf 30.11.14]



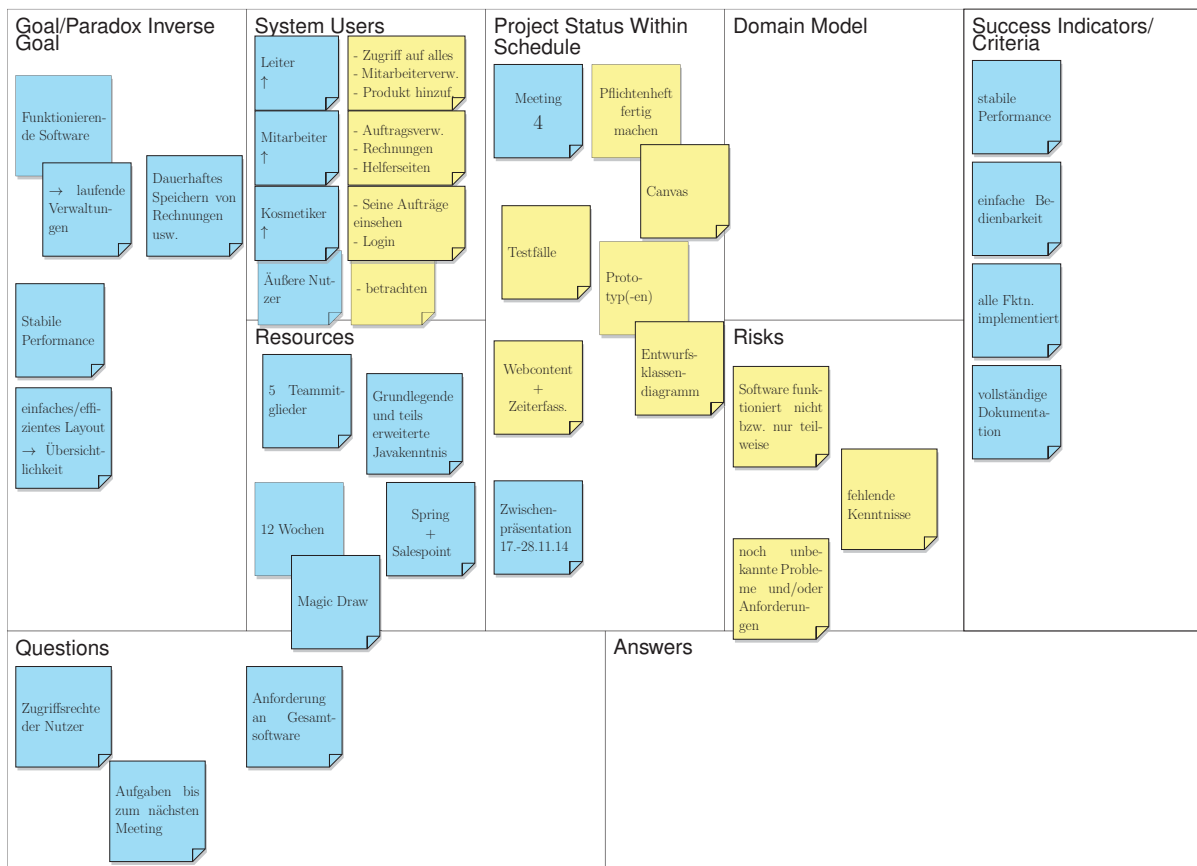


Abbildung 6.2: Anwendungsbeispiel des Customer Interview Canvas aus der Studie

macht in ihrem Canvas mit einer Notiz „siehe CRC-Karten“ in einem direkten Verweis darauf aufmerksam, das entsprechende Feld von Gruppe 1 markieren Auslassungspunkte. Die Felder *Questions* und *Answers* dienen der Vorbereitung, Verwaltung und Protokollierung der Ansprechpunkte eines Gesprächs. In diesem Fall könnte die teilweise oder vollständige Auslassung auf die zusätzliche Anfertigung eines klassischen Gesprächsprotokolls zurückzuführen sein. Die Bewertung der genannten Felder im Fragebogen auf einer Skala von 1 (kein erkennbarer Nutzen) bis 10 (sehr hilfreich) beträgt 5 für das Feld *Domain Model* und 8 bzw. 7 für die Felder *Questions* und *Answers*. Zu dem in den Ausfüllhinweisen (s. 4.2) beschriebenen Rückfragemechanismus liegt außerdem eine positive Rückmeldung vor. Diese Diskrepanz festigt die Annahme, dass sich die Auslassung vorwiegend auf den Kontext des studentischen Projekts mit parallel verwendeten Methoden gründet.

Die Einträge bzw. Notizen zum Feld *Goal/Paradox Inverse Goal* weisen zusammen mit denen des Felds *Success Indicators/Criteria* die größten Unterschiede zwischen den verschiedenen Gruppen auf. Dabei reichen die Stichpunkte von unspezifischen Zielstellungen wie „Kundeninteraktion“, „Einbeziehung relevanter Zielgruppen“ und „Funktionierende Software“ bis zu konkreten Formulierungen wie „Dauerhaftes Speichern von Rechnungen usw.“ oder „Übersicht über laufende Beauftragung“. Während einige Punkte direkt die Funktionalität betreffen, wie

auch die beiden letztgenannten Formulierungen, zielen andere auf die Projektdurchführung ab. Beispiele hierfür sind die „Abstraktion techn. Details in Kommunikation mit Experte“ oder „Zugriff auf Git“. Die Möglichkeit einer invertierten Darstellung wurde von einer Gruppe als gutes Werkzeug gelobt, insgesamt allerdings nur bedingt genutzt. Bei der Bewertung durch den Fragebogen erhielt das Feld mit 3 von 10 Punkten die schlechteste Bewertung. Dies kann zum einen mit dem augenscheinlich vernachlässigten Zusammenhang zum Feld *Success Indicators/Criteria* in Verbindung stehen, der den ermittelten Zielen konkret messbare Kriterien gegenüberstellen soll. Dadurch, dass die Inhalte dieses Felds ähnlich denen des Felds *Goal/Paradox Inverse Goal* in vielen Fällen wenig spezifisch sind, geht dieser Nutzen verloren. Zum anderen lässt sich anhand des Feedbacks nicht beurteilen, inwieweit die Hinweise oder die Fragen zu den Feldern für die Bewertung eine Rolle spielen. Interessant ist, dass gerade in der Bewertung der Gruppe 5, in deren Feldern z.B. „stabile Performance“ wortwörtlich in beiden Komponenten steht, das Feld *Success Indicators/Criteria* mit einer Punktzahl von 8 gegenüber *Goal/Paradox Inverse Goal* um 5 Punkte besser abschneidet.

Das Feld *Success Indicators/Criteria* enthält überwiegend durchaus relevante Erfolgsfaktoren. Dazu gehören unter anderem eine „wartungsfähige“, „erweiterbare“ Software, eine „hohe Nutzung“, ein „guter Workflow“ sowie die „weitestgehende Automatisierung“ der Arbeitsabläufe im Unternehmen. Was fehlt, sind hier - wie in den Hinweisen genannt - Maße, über die sich der Erfolg definieren bzw. quantifizieren lässt. Wo sich direkt keine aussagekräftigen Zahlen ableiten lassen, wie vielleicht im Fall einer guten Bedienbarkeit, sollte diese Lücke mithilfe des Felds aufgedeckt werden und gemeinsam ein Kriterium vereinbart werden, z.B. min. 80% Zustimmung einer Testgruppe bei Bedienung eines Prototyps mit der geplanten Nutzeroberfläche. Gerade für die Messung von Softwarequalität in Bezug auf Wartungsfähigkeit oder Erweiterbarkeit gibt es jedoch eine Vielzahl potenziell geeigneter Metriken. Die Kenntnis solcher Metriken lässt sich bei Studenten des 3. Semesters im Gegensatz zur Wirtschaft allerdings sicher nur sehr eingeschränkt voraussetzen.

Im Feld *System Users* wurden von allen Projektteams die Nutzergruppen des Systems festgehalten. Im Vergleich zur ausführlichen Beschreibung im Beispiel der Gruppe 5 (Abbildung 6.2) wurden von den Gruppen, die den Nutzergruppen keine Funktionalität zugeordnet haben, tendenziell weniger Rollen unterschieden. Andererseits zeichnet sich ab, dass es bei einer gewissen Menge Nutzergruppen, Funktionalitäten und/oder Überschneidungen von Funktionalitäten unterschiedlicher Nutzergruppen dieser Darstellungsweise an Übersichtlichkeit mangeln würde. Der Inhalt des Felds deckt sich nicht vollständig mit dem Titel. Gruppe 1 wiederum kommentiert die Überschneidung mit dem Feld *Goal/Paradox Inverse Goal* und begründet damit die Aufführung der Funktionalitäten ausschließlich in letzterem Feld.

Das Feld *Resources* wurde überwiegend zum vorgegebenen Zweck genutzt, wobei im Kontext des studentischen Projekts eine Festlegung oder Verständigung mit dem Kunden geringe Bedeutung hatte. Damit begründet Gruppe 1 auch die Auslassung bei der Bearbeitung. Die Notizen der übrigen Gruppe mit Angaben wie „Transportwagen“, „Autos“ und „Urnen“ lassen darauf schließen, dass diese das Feld falsch aufgefasst hat. Da unterstützend als Beispiele explizit die Erfahrung und Kompetenzen des Entwicklerteams sowie die Ressourcen Zeit und Geld genannt sind, und von den vier anderen teilnehmenden Gruppen korrekt verstanden wurde,

kommt hier eher Nachlässigkeit als Fehlerquelle in Betracht.

Das Feld *Project Status Within Schedule* wurde von allen Gruppen sinnvoll bearbeitet. Angegeben wurden einerseits konkrete Meilensteine sowie andererseits auch erledigte bzw. geplante Aufgaben, die in ihrer Anordnung mutmaßlich der zeitlichen Einteilung entsprechen. Der aktuelle Stand wurde explizit markiert oder z.B. durch Anordnung der Zettel visuell hervorgehoben. Von Gruppe 5 erhielt das Feld zusammen mit dem Feld *Goal/Paradox Inverse Goal* mit 3 von 10 Punkten dennoch die niedrigste Bewertung. Als mögliche Begründung kommen dafür mehrere Faktoren in Frage: Zu Beginn eines Projekts gibt noch keine großen Fortschritte, oder die System- und Teamgröße ist jeweils klein genug, dass der Überblick über den Zeitplan und den Projektstand auch ohne formale Hilfe aufrecht erhalten werden kann. Außerdem ist die ursprüngliche Bedeutung des Felds durch die Doppelrolle von Betreuer bzw. Kunde im Fall der Gruppen 2 bis 5 deutlich abgeschwächt. Gruppe 1 bewertet das Feld als bedingt geeignet für ihren agilen Entwicklungsprozess.

Das beste Ausfüllbeispiel für das Feld *Risks* bietet das Canvas der Gruppe 2. Hier werden Risiken wie Datenbank-Inkonsistenzen oder eine unzureichende Skalierbarkeit der Anwendung genannt. Von anderen Gruppen finden Risiken wie das einer geringen Nutzerfreundlichkeit, die erschwerte Organisation von Teamtreffen im Hinblick auf unterschiedliche Stundenpläne sowie zu wenig Einarbeitungszeit für die Frameworks Erwähnung. Eine explizite Gewichtung wurde von keiner der Gruppen vorgenommen, spiegelt sich aber möglicherweise in der Anordnung wieder. So wurde das Risiko „Meteor destroys everything“ von einer Gruppe am unteren Ende des Felds platziert. Durch den Fragebogen erhielt das Feld mit 6 von 10 Punkten eine mittlere Wertung, Gruppe 1 kritisiert eine Überschneidung mit dem inversen Ziel. Die Aufteilung des Inversen Ziels und der Verfeinerung im Feld *Risks* basiert auf der Artikelreihe zur Ermittlung von Kundenwünschen nach dem Kano-Modell [28–31], könnte aber auf Grundlage eines umfassenderen Feedbacks überdacht werden.

## 6.4 Fazit

Auch wenn die Studie über die Eignung des Customer Interview Canvas für den Einsatz im Startup-Kontext wenig spezifische Rückschlüsse erlaubt, konnten einige Erkenntnisse gewonnen werden, die in die Verbesserung des Werkzeugs einfließen können. Eine konkrete Maßnahme wäre eine intelligentere Anordnung der beiden Felder „Goal/Paradox Inverse Goal“ und „Success Indicators/Criteria“ für eine bessere Hervorhebung des Zusammenhangs. Auch eine Überarbeitung der unterstützenden Ausfüllhinweise könnte für eine bessere Bewertung einzelner Felder von Nutzen sein. Ein weiterer Punkt ist die Zuordnung der Funktionalitäten zu den einzelnen Nutzergruppen. Zwar zeigt sich bei Gruppe 5, die die Funktionalitäten im Feld *System Users* mit aufgelistet hat, gegenüber den anderen Teilnehmern ein deutlich besseres Verständnis der Rollen und Aufgaben. Es gilt jedoch zu überdenken, ob die Art der Darstellung bzw. Aufteilung gerade für komplexere Systeme geeignet ist. Dabei sollte vor allem auf Eindeutigkeit, Übersichtlichkeit und Vermeidung von Redundanz geachtet werden. Dazu kommt, dass

die aktuelle Bezeichnung des Feldes die Funktionalitäten nicht mit einschließt. Die Einflussfaktoren für die Beurteilung des Felds „Project Status within Schedule“ lassen sich aufgrund des geringen Feedbacks nicht eindeutig identifizieren. Da die Studienbedingungen in diesem Fall, wie für die Felder *Domain Model*, *Questions* und *Answers*, ausschlaggebend gewesen sein können, sollten für eine gesicherte Aussage weitere Praxiserfahrungen verglichen werden. Auf eine persönliche Betreuung und ein Übermaß an Vorgaben zur Bearbeitung der einzelnen Felder wurde ausdrücklich verzichtet. Die Ergebnisse sollten zeigen, inwieweit sich das Canvas in der Studie als leichtgewichtiges, intuitives Werkzeug bestätigt. Die direkte Beurteilung durch den Fragebogen auf einer Skala mit 6 Punkten (beginnend bei „sehr intuitiv“ und endend bei „gar nicht intuitiv“) fällt mit dem 2. Punkt sehr positiv aus. Dies bestätigen zum Teil auch die Arbeitsergebnisse, wenn man von einer Bewertung der drei Felder *Domain Model*, *Questions* und *Answers* absieht. Die eher mäßige Bereitschaft zur Wiederverwendung (3. Punkt) zusammen mit den vagen oder unvollständigen Inhalten, wie sie ebenfalls in den Ergebnissen vertreten sind, spricht aber auch dafür, dass gerade die einfache Struktur des Werkzeugs und die Knappheit der Dokumentation besondere Klarheit und Eindeutigkeit fordert. Positiv hervorgehoben wurden besonders der Rückfrage-Kontroll-Mechanismus und die Abgrenzung des Systems durch die Formulierung inverser Ziele. Dementsprechend könnten weitere Vorschläge und Konzepte dieser Art in den Hinweisen<sup>4</sup> ebenfalls den Wert des Canvas-Werkzeugs steigern.

---

<sup>4</sup>vergleichbar mit dem Stil des Buchs „Business Model Generation“ zum Business Model Canvas [23]

# 7 Variabilitätsanalyse

Innerhalb einer Gruppe verwandter Produkte kann mithilfe unterschiedlicher Methoden ermittelt werden, welche Merkmale variieren und welche Merkmale allen Produkten gemein sind. Dieser Vorgang wird als Variabilitätsanalyse bezeichnet. Werden die Informationen daraus genutzt, um den Produktionsaufwand zu verringern und dabei zu einem gewissen Grad die Anpassbarkeit an unterschiedliche Anforderungen zu erhalten, entsteht eine Produktlinie. Im Rahmen der Arbeit wird die Variabilitätsanalyse genutzt, um Konfigurationsmöglichkeiten für Canvases herauszuarbeiten.

Eine Möglichkeit, die Variabilität innerhalb einer Domäne darzustellen, bieten Feature-Modelle. Das Konzept und die gebräuchlichen Notationen sind in Abschnitt 7.1 dargestellt. In Abschnitt 7.2 werden die Felder der besprochenen Canvases für den Softwareentwicklungskontext in Vorbereitung auf eine Zuordnung von Konfigurationsmöglichkeiten zu Canvas-Konfigurationen kategorisiert und in eine klarere Struktur überführt. Abschnitt 7.3 dokumentiert die Entwicklung der Konfigurationsmöglichkeiten mit dem resultierenden Feature-Diagramm. Ein abschließendes Beispiel einer möglichen Canvas-Konfiguration zeigt Abschnitt 7.4.

## 7.1 Feature-Modelle

*Feature-Modelle* haben ihren Ursprung in der *Feature-Oriented Domain Analysis* (FODA) von Kang et al. [15], einer Methode zur Untersuchung und Aufbereitung von Unterschieden und Gemeinsamkeiten zwischen Softwaresystemen derselben Domäne. FODA wurde vorgestellt als Mittel, um die Wiederverwendbarkeit von Software zu sichern und hat sich zu einem Standard in der Planung von Softwareproduktlinien entwickelt.

Ein *Feature-Diagramm* ist als hierarchische Baumstruktur aus *Features* (Knoten) und deren Beziehungen untereinander (Kanten) aufgebaut. Als Features werden die charakteristischen Merkmale von Systemen innerhalb der betrachteten Domäne bezeichnet. Gemäß Batory et al. [3] stehen sie für ein „increment in product functionality“, d.h. die Auswahl eines Features für eine Konfiguration bringt immer eine Erweiterung der Funktionalitäten eines Systems. Features können allen Systemen gemein sein oder variieren. Es wird zwischen obligatorischen, optionalen und alternativen Features unterschieden. Aus einer zulässigen Auswahl an Features


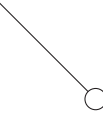
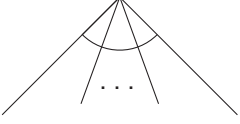
Kanten-Notation	Art des/der Subfeatures
	<b>Obligatorisches Feature</b> Jedes Produkt muss dieses Merkmal haben
	<b>Optionales Feature</b> Ein Produkt kann, muss aber nicht dieses Merkmal haben
	<b>Gruppe alternativer Features</b> Ein Produkt muss bei gewähltem Feature genau eines der Subfeatures haben

Tabelle 7.1: Verbindungen zwischen Features gemäß FODA [15]

lassen sich letztendlich Softwaresysteme bzw. die Produkte der Produktlinie konfigurieren. Die entsprechenden Kanten-Notationen sind in Tabelle 7.1 dargestellt. Darüber hinaus gibt es für erweiterte Ausdrucksmöglichkeiten die folgenden zwei Kompositionsregeln, die über die hierarchische Struktur hinweg gelten :

- **mutually exclusive with** – sich gegenseitig ausschließend
- **mandatory with** – sich gegenseitig bedingend

Die Notation wurde von Czarnecki und Eisenecker [10] um die Differenzierung zwischen Exklusiv-Oder und Inklusiv-Oder erweitert. Es ist damit möglich, zusätzlich zur Auswahl genau eines (aus beliebig vielen) Unter-Features die Kardinalitäten *zwischen 0 und 1* sowie *mindestens 1* auszudrücken. Eine Übersicht ist in Tabelle 7.2 dargestellt. Für die folgenden Feature-Diagramme wird der EMF Feature Diagram Editor<sup>1</sup> genutzt, der diese erweiterte Notation in vereinfachter Form verwendet. Es gelten die eingeführten Notationen für OR- und XOR-Featuregruppen, die Differenzierung zwischen der Auswahl genau eines oder maximal eines Features wird jedoch über das übergeordnete Feature realisiert. Aus einer XOR-Featuregruppe muss immer genau ein Feature gewählt werden. Das Feature, das in der Hierarchie über der Gruppe steht, kann entsprechend obligatorisch oder optional sein. An den unteren Kantenenden innerhalb einer Featuregruppe kann so auf die Markierung mit leeren oder ausgefüllten Kreisen verzichtet werden.

Abbildung 7.1 zeigt ein beispielhaftes Diagramm der Kompositionsmöglichkeiten von Sandwiches. Verbindungen zwischen Eltern- und Kindknoten stehen für „besteht aus“-Beziehungen. Als Voraussetzung gilt, dass jedes Sandwich aus genau einem Brot, mindestens einer Soße und mindestens einem sonstigen Belag besteht. Wählbar, aber nicht verpflichtend ist ein Fisch-

<sup>1</sup>[http://wiki.eclipse.org/EMF\\_Feature\\_Model/Feature\\_Diagram\\_Editor](http://wiki.eclipse.org/EMF_Feature_Model/Feature_Diagram_Editor) [Aufruf 25.10.14]

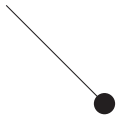
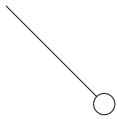
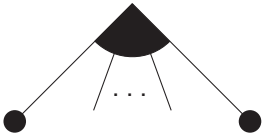
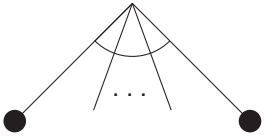
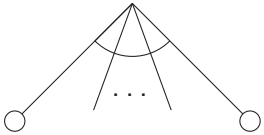
Kanten-Notation	Art des/der Subfeatures
	<b>Obligatorisches Feature</b> vgl. FODA
	<b>Optionales Feature</b> vgl. FODA
	<b>OR-Featuregruppe</b> Ein Produkt mit dem ausgewählten Merkmal muss mindestens eines der Subfeatures aus der Gruppe haben
	<b>XOR-Featuregruppe</b> Ein Produkt mit dem ausgewählten Merkmal muss genau eines der Subfeatures aus der Gruppe haben
	<b>XOR-Featuregruppe</b> Ein Produkt mit dem ausgewählten Merkmal kann kein oder maximal eines der Subfeatures aus der Gruppe haben

Tabelle 7.2: Erweiterte Feature-Notation nach Czarnecki und Eisenecker [10]

/Fleischbelag, Brot, Soße und Sonstiges (Belag) sind somit durch einen ausgefüllten Kreis gekennzeichnet, Fisch/Fleisch durch einen leeren Kreis. Bei Brot kann zwischen den Alternativen Weizen und Vollkorn gewählt werden. Auch aus der Auswahl an Fisch und Fleisch kann genau eine Sache ausgewählt werden, z.B. Hühnchen. Beides ist entsprechend gekennzeichnet durch den leeren Kreisbogen. Der Erfinder des Sandwich-Konfigurators möchte dem Kunden allerdings die Möglichkeit bieten, durch die Kombination mehrerer Soßen neue Geschmackserlebnisse zu entdecken. Es können aus den verschiedenen Soßen deshalb eine oder auch mehrere ausgewählt werden, dargestellt durch den ausgefüllten Kreisbogen. Das gleiche ist selbstverständlich auch mit der Auswahl an vegetarischem Belag möglich. Ein „gültiges“ Sandwich wäre Vollkornbrot mit Hühnchen, Barbecuesoße, Salat und Tomaten. Die Kombination ohne Salat und Tomaten wäre aber nicht möglich, da zu Beginn festgelegt wurde, dass ein Sandwich einen vegetarischen Belag enthalten muss.

An genanntem Beispiel zeigt sich, dass es keine eindeutige Lösung gibt. Ob sich Soßen kombinieren lassen sollen oder Fisch/Fleisch von vegetarischem Belag getrennt sein muss, sind subjektive Entscheidungen. Die Struktur des Feature-Baums hängt im Wesentlichen davon ab, wie die realen Objekte und Zusammenhänge modelliert werden. Dabei kann selten jedes Detail berücksichtigt werden. Vielmehr geht es darum, welches Modell unter vertretbarem Auf-

wand und Komplexität die größtmögliche Übersichtlichkeit und Flexibilität bietet. So bietet das besprochene Feature-Diagramm mit 1050 Möglichkeiten für die Zusammenstellung von Sandwiches bereits große Vielfalt unter Inkaufnahme einer gesteigerten Komplexität.

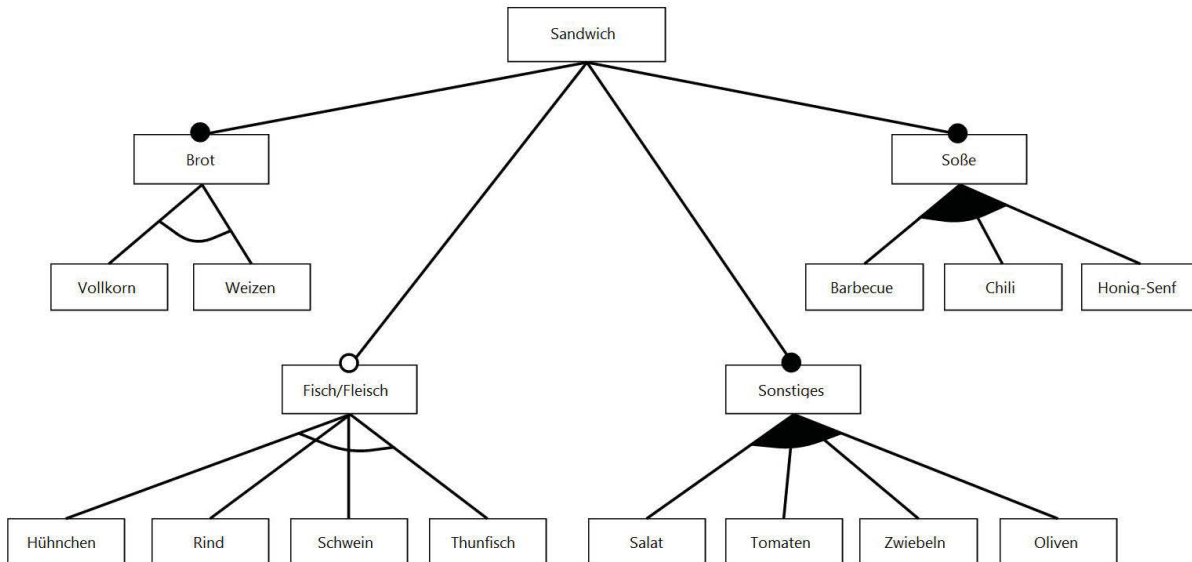


Abbildung 7.1: Feature-Diagramm für die Zusammenstellung eines Sandwiches

## 7.2 Kategorisierung der Canvas-Felder im Zusammenhang mit Softwareprojektsteuerung

Als erster Schritt hin zur Formulierung einzelner Konfigurationskriterien und Konfigurationen werden die Felder aus Oddoys Lean Software Engineering Canvas (LSEC) [21] und den beiden neu eingeführten Canvases, dem Customer Interview Canvas (CIC) und dem Iteration Planning Canvas (IPC), an dieser Stelle gesammelt betrachtet. Damit stehen 27 Felder für die Darstellung verschiedener Aspekte der Softwareentwicklung zur Diskussion. Durch eine inhaltliche Kategorisierung und geeignete Modifikationen werden die Canvas-Felder in eine klarere Struktur überführt. Das Resultat zeigt Tabelle 7.3. Die Ergebnisse der Studie zeigen, dass die unmissverständliche Benennung und Hilfestellung zum Ausfüllen der Felder eine wichtige Rolle spielt. Einige Feldbezeichnungen werden deshalb angepasst.

### 7.2.1 Felder für eine erste Vorstellung des Produkts

Das Feld *Domain Model* markiert im LSEC wie im CIC den Platzhalter für die Grundstruktur aus fachlichen Objekten und Zusammenhängen. Diese herauszuarbeiten und ein klares



Verständnis der Domäne zwischen Kunde und Projektteam zu etablieren ist von essentieller Bedeutung für jedes Projekt. Das Feld wird in dieser Form beibehalten.

Eine starke Parallele besteht zwischen den Feldern *Stakeholders* und *System Users*. Im Allgemeinen sind Stakeholder alle, die ein Interesse am Verhalten oder der Gestaltung des Objekts haben. Damit wären nicht nur Systemnutzer, sondern auch der Auftraggeber, das Entwicklungsteam oder andere Institutionen, Individuen oder Personengruppen gemeint, die selbst nicht mit dem System interagieren<sup>2</sup>. Mit dem Vorsatz, einen Bezug zu den Funktionalitäten (*Key Functions*) zu schaffen, beschreibt Oddoy in seiner Arbeit [21] an anderer Stelle aber auch, dass es ihm rein um die Nutzergruppen des Systems geht. Da in diesem Sinn Einigkeit über die Relevanz der Frage „Wer hat Zugriff auf das System bzw. auf bestimmte Funktionalitäten?“ herrscht, werden die Felder *Stakeholders* und *System Users* zusammengelegt und im Weiteren unter dem Namen *System Users* geführt.

Das Feld *Key Functions* im LSEC ist für die Beschreibung der groben Funktionalität der geplanten Software bestimmt. Bei dem Feld *System Users* im CIC wird davon ausgegangen, dass die Nutzergruppen sich bereits über die Funktionalitäten, die das System für sie bietet, definieren, d.h. es fasst die Systemnutzer und die ihnen offen stehenden Funktionalitäten in sich zusammen. Da durch die getrennten Felder im LSEC mehr Klarheit geschaffen wird, wird diese Trennung auch für mögliche Konfigurationen beibehalten. Das Feld *Key Functions* wird dementsprechend weitergeführt.

Im Feld *Components* aus dem LSEC wird gemäß Oddoy [21] die Systemarchitektur, d.h. vor allem dessen Komponenten und zugeordnete Funktionalitäten, beschrieben. Im ersten Schritt, in dem eine gemeinsame Vorstellung des Produkts entwickelt wird, können noch keine solchen Entscheidungen getroffen werden. Eine zu zeitige und mehr oder weniger statische Zuordnung entspräche nicht dem dynamischen Charakter eines Startup-Projekts. Die genauere Ermittlung und Umsetzung der (funktionalen) Anforderungen erfolgt in Iterationen und bietet zu Beginn keine Vorlage für eine Systemarchitektur. Für den Kunden wiederum ist die interne Aufteilung der Funktionalitäten zweitrangig, sie sollte sich nach den von ihm spezifizierten Anforderungen richten. Die Planung und ggf. Anpassung der Systemarchitektur kann vom Entwicklerteam am ehesten zu Beginn eines Release verfolgt werden. Dabei ist aber aufgrund des Umfangs und der Nähe zur Implementierung nicht unbedingt ein Canvas-Feld sinnvoll. Im Rahmen dieses Konzepts wird auf die Darstellung der Systemarchitektur in einem Feld *Components* verzichtet.

Das Feld *Cross-Cutting* (auch *Cross-Cutting Requirements*) zielt auf technische, organisatorische und architektonische Rahmenbedingungen ab. Das Template zum LSEC enthält außerdem die Frage nach den Qualitätsanforderungen an das System. Ein Beispiel in Oddoys Arbeit [21] beinhaltet in diesem Feld als nicht-funktionale Anforderungen die Umsetzung mit Java und .NET als Tablet- und Desktopanwendung und die Beschränkung auf Endgeräte, die die JBoss Standard-Konfiguration unterstützen. Diese betreffen zwar die konkrete Implementierung, können aber zu den Anforderungen gezählt werden, wenn sie vom Kunden explizit gefordert sind. Da die durch das Feld *Cross-Cutting* abgedeckten Anforderungen und Rand-

---

<sup>2</sup>s.a. <http://www.viewpoints-and-perspectives.info/home/stakeholders/> [Aufruf 01.11.14]

bedingungen sehr zahlreich sein können und inhaltlich stark auseinandergehen, wird das Feld im Folgenden weiter aufgeteilt.

- Zu den organisatorischen Bedingungen zählen z.B. zeitliche Vorgaben oder das Budget. Diese sind bereits separat im Feld *Resources* berücksichtigt.
- Die technischen Randbedingungen umfassen Plattformen, Frameworks und die Schnittstellen zu bestehenden Systemen. Theoretisch wird die Technologie den funktionalen und qualitativen Anforderungen nachgeordnet, besteht allerdings eine explizite derartige Forderung, so kann diese als Ressource festgelegt werden. Soll es Schnittstellen zu anderen Systemen geben, müssen diese klar formuliert werden. Das geschieht getrennt im Feld *Interfaces*.
- Einschränkungen betreffend die Systemarchitektur werden im Allgemeinen nicht als gültige Anforderung angesehen, sollen doch die geforderte Qualität und Funktionalität die Architektur bestimmen. Einzelne Bedingungen wie die oben genannte Umsetzung als Tablet- und Desktopanwendung können allerdings in Ausnahmefällen der groben Beschreibung des Systems beigelegt werden.
- Einen großen Teil des Feldinhalts können die Qualitätsanforderungen ausmachen. Der folgende Paragraph begründet die Einführung dreier Felder zur Qualitätssicherung.

Software wird in großem Umfang und einem hohen Tempo entwickelt. Viele bestehende Systeme werden nicht ausreichend gepflegt und unterliegen einem schleichenden Qualitätsverfall. Dafür verantwortlich sind einerseits veraltende Technologien, andererseits werden ehemals gut geplante Architekturen durch vielfache Erweiterung erodiert. Je überholter die Technologien, desto schwieriger die Integration mit neuen Komponenten und desto höher die Kosten für geeignetes Fachpersonal. Die Erweiterung um neue Funktionalitäten ohne die entsprechende Anpassung der bestehenden Architektur wird immer schwieriger, die ursprüngliche Struktur wird an einzelnen Stellen umgangen und geht nach und nach verloren. Um dem entgegenzusteuern und Qualitätsmängel bei Neusystemen von Anfang an zu vermeiden, kann ein Teil der Ressourcen eines Entwicklungsprojekts auf die Qualitätssicherung und -verbesserung verwendet werden. Im Gegensatz zu Oddoy [21] wird eine klare Spezifikation der qualitativen Anforderungen bereits in der Anfangsphase eines Projekts als sinnvoll angesehen, da sie das Verständnis zwischen Kunde und Projektteam und damit grundlegende Designentscheidungen beeinflusst. Zum Qualitätsmanagement in Softwareprojekten gibt es in Literatur und Praxis zahlreiche Ansätze. Einer davon ist die Bestimmung und Gewichtung von Qualitätsfaktoren. Beispiele für Qualitätsfaktoren eines Softwaresystems geben Normen wie die ISO 25010<sup>3</sup>, Barbacci et al. [2] sowie Kan [14]. Dazu gehören etwa Sicherheit, Performanz, Verlässlichkeit, Wartbarkeit, Portabilität oder Nutzerfreundlichkeit. Je nach Bedarf ist auch eine genauere Spezifikation möglich, beispielsweise Kapazität, Zeitverhalten, Modularität, Anpassbarkeit, Code-Komplexität oder Wiederverwendbarkeit. Viele dieser Faktoren beeinflussen sich gegenseitig. So erleichtert ein ungeschützter Zugang die Nutzung eines Systems, genügt aber keinen hohen Sicherheitsansprüchen. Für das jeweilige System spielt deshalb vor allem die Gewichtung der Faktoren eine Rolle. Diese sollte als Grundlage für Designentscheidungen dienen. Zusammen

---

<sup>3</sup>[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=35733](http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733) [Aufruf 04.11.14]

mit jeweils geeigneten Metriken und einer konkreten Zielstellung ergänzen die Qualitätsfaktoren das Canvas-Werkzeug um die Möglichkeit der Qualitätssicherung. Als Canvas-Felder werden dementsprechend *Quality Factors*, *Quality Metrics* und *Quality Objectives* eingeführt.

Im Feld *Resources* werden die Ressourcen für das ganze Projekt festgelegt. Im Extreme Programming spricht man von vier Variablen, von denen drei von außen bestimmt werden können [4]: Zeitrahmen, Budget, Qualität und Funktionsumfang. Die vierte Variable festzulegen obliegt dem Entwicklerteam. Wird eine Qualitätsmessung durchgeführt, kann nur noch eine der drei anderen Variablen vom Entwicklerteam bestimmt werden. Es können im Feld *Resources* für Zeit oder Kosten flexible Rahmen angegeben werden oder auch bewusst eine Variable offen gelassen werden. Dementsprechend gestaltet sich im späteren Verlauf das Projekt. Alle Projektbeteiligten sollten sich über die Einschränkungen im Sinne verfügbarer Ressourcen im Klaren sein. Das Feld bleibt deshalb ein wichtiger Bestandteil für die Steuerung von Softwareprojekten mit Canvases.

Als besonderer Zusatz im Feld *Resources* wurden oben die in einzelnen Fällen geforderte Verwendung zur Verfügung gestellter Frameworks, Plattformen und Technologien genannt. Da die Entscheidung über die Verwendung von Technologien für die Neuentwicklung eines Systems einen deutlich größeren Stellenwert einnimmt, wird im Weiteren ein optionales Feld mit der Bezeichnung *Technologies* geführt.

Das Feld *Scope and Context* bietet Platz für eine Beschreibung von Schnittstellen, die für die Integration mit bestehenden Systemen von hoher Wichtigkeit sind. Für eine klarere Zuordnung wird das Feld im Folgenden unter dem Namen *Interfaces* geführt.

### 7.2.2 Felder für die Projektplanung in Iterationsschritten

Eine Projektplanung in Iterationsschritten entspricht dem dynamischen Charakter von Canvases im Startup-Kontext. Im IPC umfasst das Feld *Release Plan* eine kurze Beschreibung der geplanten Features für die Iterationsschritte eines Release, die Anzahl der geplanten Story Points (gemessen am letzten Release) und bereits gesammelte Stories, die für die Iteration mit zur Verfügung stehen. Es bildet damit die Basis für eine aufwandsorientierte Taskplanung, wie sie im Planning Game bzw. im Extreme Programming praktiziert wird. Wie in Kapitel 5 besprochen, ist es für die Taskplanung notwendig, die Länge der Releases und Iterationen über das Projekt hinweg konstant zu halten. Mit einer individuellen Festlegung dieser beiden Zykluslängen kann aber auf die projektspezifischen Anforderungen reagiert werden. Eine Rolle spielen dabei der Projektumfang, der Fertigstellungstermin bzw. die Projektdauer und der eher statische oder dynamische Charakter des Projekts (d.h., sind die Anforderungen für das System zu Beginn bereits sehr klar oder nicht). Das Feld bleibt als wichtiger inhaltlicher Bestandteil der Projektplanung erhalten.

Die Felder *Use Cases* und *Stories* (auch *User Stories*) beschreiben beide die Handlungsmöglichkeiten von Systemnutzern und deren Wirkung. Use Cases oder Anwendungsfälle kommen klas-

Feld	Abhängigkeiten	Übergeordnete Kategorie	Betreff
Domain Model		Basiskomponenten für Produktvision	Erste Vorstellungen des Produkts bzw. der Domäne
System Users			
Key Functions			
Quality Factors			
Resources			
Quality Metrics	bedingt: <i>Quality Factors, Quality Objectives</i>	Qualitätsmessung	
Quality Objectives	bedingt: <i>Quality Factors, Quality Metrics</i>		
Technologies		Technologien (separat) bei Neuentwicklung	
Interfaces		Schnittstellen bei Erweiterung eines Systems	
Release Plan		Basiskomponenten für iterative Projekt- bzw. Taskplanung	Projektplanung, Aufgabenverteilung je Iteration
Tasks	bedingt: <i>User Stories</i> oder <i>Use Cases</i>		
Previous Achievements #			
Personal Estimates	bedingt: <i>Tasks</i>		
Personal Subscriptions	bedingt: <i>Tasks</i>	Beschreibung von Handlungsmöglichkeiten im Fall kundenbegleiteter Projekte	
User Stories	gegenseitiger Ausschluss mit: <i>Use Cases</i> ; bedingt: <i>Acceptance Tests</i>		
Acceptance Tests	gegenseitiger Ausschluss mit: <i>Scenarios</i> ; bedingt: <i>User Stories</i>		

Feld	Abhängigkeiten	Übergeordnete Kategorie	Betreff
Special Qualifications		Qualifikationsförderung der Teammitglieder	Projektplanung, Aufgabenverteilung je Iteration
Sought Experience			
Use Cases	gegenseitiger Ausschluss mit: <i>User Stories</i> ; bedingt: <i>Scenarios</i>	Basiskomponenten für Kundengespräche	Kundengespräch Status-Update, Abklärung der Implementierungsaufgaben für die nächste Iteration
Scenarios	gegenseitiger Ausschluss mit: <i>Acceptance Tests</i> ; bedingt: <i>Use Cases</i>		
Project Status within Schedule			
Goal/Inverse Goal			
Success Indicators			
Risks			
Questions			
Answers			

Tabelle 7.3: Canvas-Felder für den Softwareentwicklungskontext, kategorisiert

sicherweise zum Einsatz, wenn Einzelheiten nicht zwischendurch mit dem Kunden abgeklärt werden können, sondern möglichst vollständig in einer Sitzung besprochen werden, d.h. sie sind abhängig von der Art der Kundenbeteiligung. Bei Use Cases liegt häufig die höhere Verantwortung beim Softwaredienstleister. Dieser muss sicherstellen, dass die Use Cases vollständig und eindeutig formuliert sind, bevor die Umsetzung in Angriff genommen wird. Die Form der Dokumentation von Anforderungen als Stories ist vor allem aus dem Extreme Programming und anderen agilen Entwicklungsmethoden bekannt. User Stories können ohne große Hilfe vom Kunden formuliert werden. Das Hinzufügen und Anpassen von User Stories ist flexibler als von detailreichen Anwendungsfällen. Gibt es einen Kundenvertreter vor Ort, ist der Einsatz von User Stories sinnvoll. Anderenfalls werden diese durch die Formulierung von Anwendungsfällen im separaten Kundengespräch ersetzt. Beide Felder bleiben erhalten und kommen in alternativen Konfigurationen zum Einsatz.

Das Feld *Acceptance Tests* hängt konzeptionell eng mit dem Feld *User Stories* zusammen. Im Feld *Acceptance Tests* sind die Anforderungen hinter einer User Story in Form eindeutiger Akzeptanzkriterien zum Verhalten des Systems im Einzelnen dargestellt. Diese Form von Details ist nötig für eine anforderungsgerechte Umsetzung. Das Feld bleibt in gleicher Form erhalten.

Das Feld *Tasks* enthält im IPC die auf technische Aufgaben von überschaubarer Größe heruntergebrochenen User Stories. Es kann jedoch auch ein etwas komplexerer Anwendungsfall in ähnlicher Weise in Tasks zerlegt werden. Da die Definition klar getrennter Aufgaben wichtige Voraussetzung für eine flexible Verteilung ist, wird das Feld für alle möglichen Konfigurationen beibehalten.

Die Felder *Previous Achievements #*, *Personal Estimates* und *Personal Subscriptions* ermöglichen in ihrer in Kapitel 5 dargelegten Funktion die Aufgabenverteilung anhand des gemessenen bzw. geschätzten Aufwands und bleiben in dieser Weise als Grundbestandteil der Projektplanung bestehen.

Die Felder *Special Qualifications* und *Sought Experience* bilden zusammen die Grundlage für eine gezielte Förderung von Qualifikationen der einzelnen Teammitglieder und bieten Unterstützung bei der Verteilung von Implementierungsaufgaben. Sie bleiben in dieser Form als optionale Konfigurationsmerkmale erhalten.

### 7.2.3 Felder für separate Kundengespräche

Das Feld *Use Cases* wurde bereits im Zusammenhang mit dem Feld *Stories* besprochen. Es stellt im Kundengespräch die Alternative zu *Stories* bei direkter Beteiligung des Kunden am Projekt dar. Um in Kombination mit Anwendungsfällen eine aufwandsbasierte Iterationsplanung durchzuführen (s.o.), müssen auch diese im Zusammenhang mit der Taskplanung mit einer Aufwandsschätzung versehen werden. Eine Möglichkeit ist es, bei der Iterationsplanung jedem Anwendungsfall einen geschätzten Aufwand zuzuweisen. Ist die Abschätzung eines kompletten Anwendungsfalls zu schwierig, kann dieser vor der endgültigen Zuweisung zu einer

Iteration von den Entwicklern in einem Zwischenschritt unterteilt werden.

Das Feld *Scenarios* ist konzeptionell eng mit dem Feld *Use Cases* verbunden. Es bietet Platz für die Beschreibung der verschiedenen möglichen Szenarien zu einem Anwendungsfall. Die Szenarien sollen gemäß Oddoy [21] so formuliert sein, dass sie sich als Abnahmetests eignen. Damit bildet das Feld *Scenarios* zusammen mit dem Feld *Use Cases* auch ein ähnliches Konstrukt zu den Feldern *Acceptance Tests* und *User Stories*. Die beiden Kombinationen werden im Weiteren als zueinander alternative Konfigurationsmerkmale behandelt.

Das Feld *Business Rules* umfasst bereits von der fachlichen Seite bestimmte Regeln für das Verhalten des Systems. Da solche Regeln auch Bestandteil der funktionalen Anforderungen sind, finden sie bereits in einer der Kombinationen *Use Cases* und *Scenarios* bzw. *User Stories* und *Acceptance Tests* Berücksichtigung. Auf das separate Feld *Business Rules* wird verzichtet.

Das Feld *Project Status within Schedule* umfasst den aktuellen Status des Projekts auf der Basis eines durch die Ressourcen festgelegten Zeitplans. Da der Kunde durch seine externe Beteiligung nicht umfassend informiert ist, bildet das Feld in dieser Form weiterhin die Basis für ein Gespräch über die weitere Kundenwünsche, deren Umsetzung noch offen steht.

Die Felder *Questions* und *Answers* unterstützen, basierend auf dem Kano-Modell, wie in Kapitel 4 erläutert die Ermittlung der Leistungsfaktoren. Diese machen einen wesentlichen Teil der Anforderungsermittlung im Kundengespräch aus. Die Felder bleiben in ihrer Form bestehen.

Die Felder *Goal/Paradox Inverse Goal*, *Success Indicators/Criteria* und *Risks* helfen entsprechend ihrer Darlegung in Kapitel 4 bei der Ermittlung der begeisternden Faktoren. Im Vergleich zum CIC leitet sich die dritte Form von Kundenwünschen, die Basisfaktoren, bereits in einem initialen Schritt aus der ersten Vorstellung des Produkts ab. Alle drei Faktoren werden noch einmal durch die Formulierung von Anwendungsfällen und Szenarien konkretisiert. Mit dem Ziel einer knappen, eindeutigen Beschriftung werden die beiden ersten Bezeichnungen zu *Goal/Inverse Goal* und *Success Indicators* gekürzt. Die Felder für separate Kundengespräche sind damit vollständig.

## 7.3 Konfigurationsmöglichkeiten von Canvases im Softwareentwicklungskontext

Aus der Kategorisierung der zur Verfügung stehenden Canvas-Felder für die Softwareentwicklung gehen Unterschiede in der Planung und Durchführung von Projekten hervor. Diese werden wie folgt in vier Bereiche unterteilt.

**Einbindung des Kunden ins Projekt** Einigen Einfluss auf die Gestaltung eines Projekts hat die Art der Kundenbeziehung. Zur Wahl stehen dafür zwei Alternativen. Eine davon ist die aktive Teilnahme am Projekt, auf die vor allem im Extreme Programming (vgl. Einführung des



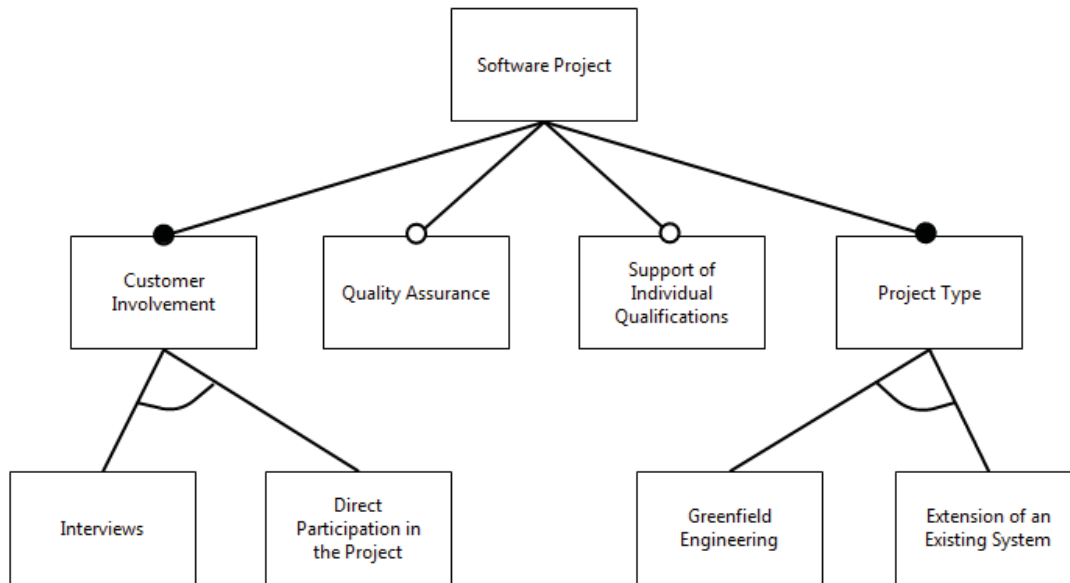


Abbildung 7.2: Feature-Diagramm für die Konfiguration eines Canvas

Iteration Planning Canvas, Kapitel 5) besonderer Wert gelegt wird. Gleichzeitig ermöglichen die äußeren Umstände in einigen Fällen keine Kundenbetreuung in dieser Intensität. Potenzielle Kundenvertreter haben kein ausreichendes Wissen zu den Systemvoraussetzungen oder sind schwer entbehrlich. Wo eine direkte Teilnahme am Projekt nicht möglich ist, besteht der Kundenkontakt in regelmäßigen Gesprächen zur Abklärung von Anforderungen und Absprache des weiteren Vorgehens. Diese bilden die Alternative zur Einbeziehung ins Projektgeschehen und werden mehr oder weniger abseits davon geplant und durchgeführt. Diesem Konstrukt wird Rechnung getragen, indem für die separaten Gespräche auch ein separates Canvas bereitgestellt wird. Da der Kunde nicht mit der gleichen Flexibilität agieren und Einfluss nehmen kann wie im ersten Fall, ist eine besonders sorgfältige Planung und Abklärung der Anforderungen erforderlich. Zu diesem Zweck kommen die Felder des CIC zum Einsatz, die für die Ermittlung der Kundenwünsche nach dem Kano-Modell entwickelt wurden (vgl. Einführung des Customer Interview Canvas, Kapitel 4). Als weiterer Unterschied wird in Kombination mit der Projektbegleitung das eine gute Ergänzung bildende Konzept aus *User Stories* und *Acceptance Tests* genutzt, während für separate Gespräche die etwas umfangreichere Spezifikation von *Use Cases* und *Scenarios* vom Kunden gefordert wird.

**Art des Projekts** Ein weiterer Bereich, in dem sich Projekte voneinander differenzieren, ist der Projekttyp. Es lässt sich unterscheiden zwischen der Renovation von Altsystemen, der Neuentwicklung und der Erweiterung von bestehenden Systemen. Da es in dieser Arbeit vor allem um die Unterstützung von Startups bei der Softwareentwicklung geht und in diesem Stadium ersteres keine entscheidende Rolle spielt, wird dieser Projekttyp im folgenden Modell nicht berücksichtigt. Je nachdem, ob es also um ein komplett neues System oder um eine Erweiterung geht, kommen bei der Planung bestimmte Aspekte hinzu. Bei der Neuentwicklung liegt ein zusätzlicher Schwerpunkt auf der Vereinbarung zu verwendender Technologien. Bei der



Systemerweiterung dagegen müssen die Schnittstellen zu bestehenden Komponenten eindeutig abgeklärt werden. Hierfür wird je nach Konfiguration das notwendige Feld bereitgestellt.

**Qualitätsmessung** Auch die Qualitätsanforderungen sind bei manchen Projekten strenger als bei anderen. Wo dies der Fall ist, empfiehlt es sich besonders, die Anforderungen nicht nur als grobe Richtlinie zu nennen und zu gewichten, sondern für das ganze Projekt Metriken und Zielwerte zu bestimmen. Die Erfüllung der Qualitätsanforderungen wird zu einer vertraglichen Bestimmung. Die Qualitätsmessung wird als optionales Feature behandelt.

**Förderung individueller Qualifikationen** Ein weiterer Unterschied in Softwareprojekten ist die Gestaltung der Verantwortlichkeiten unter Entwicklern. Ist jeder für einen speziellen Aufgabenbereich verantwortlich oder kann ein Teammitglied flexibel Aufgaben übernehmen, sei es nun die Implementierung des Backends oder die Gestaltung des Web Interface? Egal nach welchem Konzept gearbeitet wird, ist es möglich, die Qualifikationen des Entwicklerteams zu berücksichtigen und auf lange Sicht positiv zu beeinflussen. Dazu kommt, dass eine selbstgewählte Aufgabe häufig mit mehr Elan angegangen wird und damit schneller erledigt ist. Im Idealfall profitieren alle Parteien davon, die persönlichen Kompetenzen und Entwicklungsziele der Teammitglieder bei der Aufgabenverteilung zu berücksichtigen. Dass dies vielleicht nicht in allen Projekten möglich oder sinnvoll ist, wird insofern berücksichtigt, dass die beiden betreffenden Felder für die Projektplanung, *Special Qualifications* und *Sought Experience*, optional gehalten werden.

Die beschriebenen Bereiche finden sich umgesetzt in einem Feature-Diagramm in Abbildung 7.2 wieder. Im Anhang auf Seite 68ff. befindet sich eine Tabelle mit den 16 gültigen Konfigurationen und daraus resultierenden Varianten von Canvases.

## 7.4 Konfigurationsbeispiel

Zur Veranschaulichung wird im Folgenden ein Konfigurationsbeispiel betrachtet. Als obligatorische Features werden die Einbindung des Kunden in Form von separaten Kundengesprächen und der Projekttyp mit der Erweiterung eines bestehenden Systems gewählt. Außerdem werden beide optionalen Features ausgewählt. Dies bedingt zum Einen Felder für eine verstärkte Qualitätssicherung und zum Anderen Felder für die Förderung individueller Qualifikationen im Entwicklerteam.

Die Durchführung von Kundengesprächen abgespalten vom Projektgeschehen hat die Konfiguration dreier Canvas-Vorlagen zur Folge. Allen Konfigurationen gemein sind zwei Canvases für die erste Vorstellung eines Produkts und die Iterationsplanung, im Folgenden als *Product Vision Canvas* und *Iteration Planning Canvas* (vgl. Kapitel 5) bezeichnet. Spezifisch für die gewählte Art der Kundeneinbindung ist eine weitere Vorlage für die Unterstützung von Kundengesprächen. Diese soll im Folgenden als *Customer Interview Canvas* referenziert werden, weicht aber an mehreren Stellen von dem unter diesem Namen eingeführten Canvas in Kapitel 4 ab. So sind einerseits die Felder für die Ermittlung der Ressourcen, der Systemnutzer

und der domänenspezifischen Objekte ausgelagert in das Product Vision Canvas. Andererseits sind die beiden Felder für Anwendungsfälle und Szenarien zusätzlicher Bestandteil der aktualisierten Vorlage. Es erfolgt entsprechend die Beschreibung von Handlungsmöglichkeiten im Rahmen eines Gesprächs. In diesem Fall sind die Felder für die Erfassung von User Stories und Acceptance Tests nicht Bestandteil des Iteration Planning Canvas.

Bei der Erweiterung eines bestehenden Systems kommt ergänzend zu den Basis-Komponenten des Product Vision Canvas ein Feld für die Spezifikation der Schnittstellen hinzu.

Durch die Auswahl des Features *Quality Assurance* erhält die Vorlage für das Product Vision Canvas zwei weitere Felder für Metriken und Zielstellungen entsprechend der jeweiligen Qualitätsfaktoren.

Die Auswahl des Features *Support of Individual Qualifications* bedingt die Erweiterung des Iteration Planning Canvas um die Felder für bestehende und angestrebte Qualifikationen des Entwicklerteams. Diese sind in der eingeführten Fassung standardmäßig vorhanden, bei der Kategorisierung in Abschnitt 7.2 allerdings zu optionalen Feldern ernannt worden.

Die exemplarische Konfiguration ergibt drei Vorlagen zur Unterstützung der auftretenden Prozesse, die in Abbildung 7.3 zu sehen sind. Die Anordnung der Felder ist dabei eine mögliche Lösung, in der versucht wurde, Zusammenhänge zwischen den Feldern und den individuellen Platzbedarf mit zu berücksichtigen. Zu beachten ist, dass das Customer Interview Canvas sowie das Iteration Planning Canvas im Gegensatz zum Product Vision Canvas im Laufe eines Projekts mehrfach ausgefüllt werden müssen. Dies ist durch die Pfeile angedeutet. Für alle Canvases gilt das klassische Prinzip, dass sie in mehreren Schritten und so oft wie nötig ausgefüllt, überarbeitet und zur Rate gezogen werden.

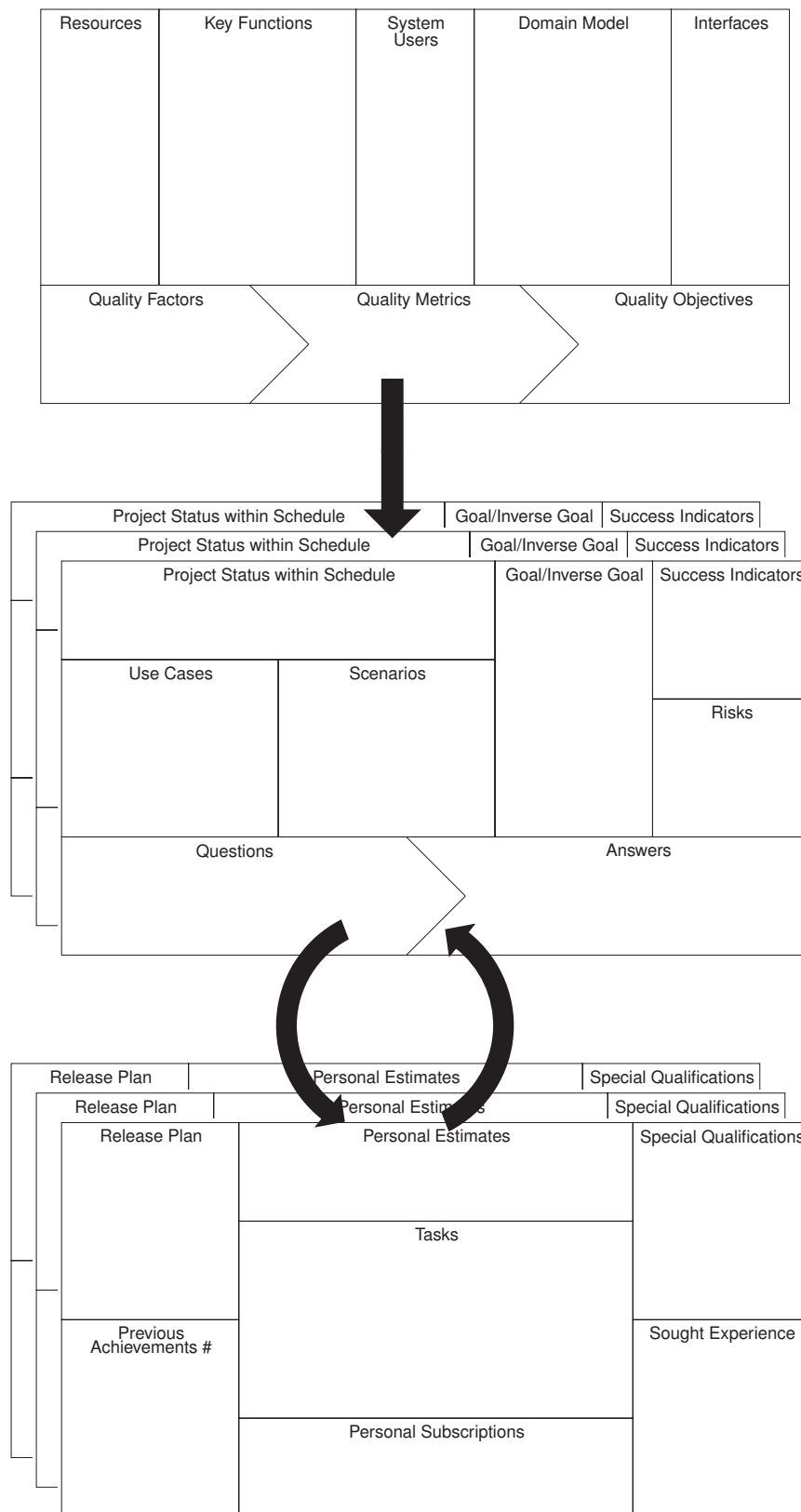


Abbildung 7.3: Canvas-Vorlagen zum Konfigurationsbeispiel



## 8 Möglichkeiten zur Bewertung der Vollständigkeit von Canvases

Canvases erfüllen während und nach der Bearbeitung mehrere wichtige Funktionen. Als eine davon bieten sie Basis für eine Bewertung hinsichtlich qualitativer und struktureller Vollständigkeit. Die strukturelle Vollständigkeit eines Canvas lässt sich teilweise anhand der Abhängigkeiten zwischen Feldern oder der Definition einer minimal geforderten Anzahl an Stichpunkten zu einzelnen Feldern überprüfen. Qualitative Vollständigkeit lässt sich allerdings schwer definieren und automatisiert kontrollieren. Im Folgenden werden Bewertungsansätze vorgestellt und mögliche Lösungen für die Integration von Feedback-Mechanismen in ein digitales Canvas-Framework diskutiert.

### 8.1 Bewertungsansätze

Ein Ansatz zur Bewertung der Förderungswürdigkeit eines Startups und dessen Geschäftsmodell - unter Einbeziehung des BMC - findet sich bei Steve Blank [7]. Auf der von ihm konzipierten Plattform LaunchPad Central<sup>1</sup> können Startup-Teams verschiedene Daten über den Prozess der Geschäftsmodellentwicklung hinweg sammeln und ihren Fortschritt verfolgen.

Mit dem BMC sind die Aspekte vorgeschrieben, zu denen die Teams in einem ersten Schritt Hypothesen aufstellen. Diese werden in Kundengesprächen und Mentoren-Treffen schrittweise validiert bzw. korrigiert. Komponenten wie eine „Activity Map“ und Erfahrungsberichte zu den Kundengesprächen protokollieren zusammen mit dem wöchentlich aktualisierten BMC die Aktivitäten der Unternehmensgründer. Ein weiterer Bestandteil der Plattform, das *Investment Readiness Level* (IRL), zeigt einen aus den Daten errechneten Bewertungsstand an. Die Plattform bietet damit sowohl potenziellen Investoren Unterstützung bei der Beurteilung als auch den Unternehmensgründern selbst ein umfassendes Feedback.

Zum IRL gehören wie zu seinem Vorbild (NASA Technology Readiness Level aus den 70er und 80er Jahren) neun Stufen, die auf einem Thermometer angezeichnet sind. Gemäß Blank<sup>2</sup> ist

---

<sup>1</sup><https://www.launchpadcentral.com>

<sup>2</sup><http://steveblank.com/2014/02/25/is-this-startup-ready-for-investment/> [Aufruf 08.10.14]

es flexibel genug, um es auf unterschiedliche Geschäftsbereiche anpassen zu können. Die neun Stufen sind wie folgt:

1. Erstes vollständig skizziertes BMC
2. Marktanalyse
3. Validierung der Abstimmung zwischen Problem und Lösung
4. Prototyp Low-Fidelity Minimal Viable Product<sup>3</sup>
5. Validierung der Abstimmung zwischen Produkt und Markt
6. Validierung des Ertragsmodells (rechte Seite des BMC)
7. Prototyp High-Fidelity Minimum Viable Product<sup>4</sup>
8. Validierung des Wertbeitrags (linke Seite des BMC)
9. Identifikation und Validierung aussagekräftiger Metriken

Ist auch das neunte Level absolviert, gilt das Geschäftsmodell von Seiten des IRL als förderungswürdig. Natürlich spielen für eine Entscheidung abgesehen von der Subjektivität der Bewertungsskala noch andere Faktoren eine Rolle.

Blank macht sich für sein Konzept das Element der *Gamification* zu Nutze. Gemäß Definition ist darunter „die Übertragung von spieltypischen Elementen und Vorgängen in spielfremde Zusammenhänge mit dem Ziel der Verhaltensänderung und Motivationssteigerung bei Anwenderinnen und Anwendern“<sup>5</sup> zu verstehen. Das IRL dient der Visualisierung von Erfolg und zunehmendem Fortschritt. Als Belohnung steht die Förderung des Startups in Aussicht. Dieser Ansatz findet sich auch an anderen Stellen wieder<sup>6</sup> und kann die Motivation der Anwender und den Nutzen eines Systems deutlich erhöhen.

## 8.2 Diskussion geeigneter Bewertungskriterien

Die Berechnung des IRL für die auf der Plattform vertretenen Startups findet basierend auf den gesammelten Informationen wie validierten und invalidierten Hypothesen, Kundengesprächen und anderen Aktivitäten statt. Geht man für eine Bewertung der Vollständigkeit von den im Konfigurationsmodell (vgl. Abschnitt 7.3) enthaltenen Canvases davon aus, dass außer den Canvases selbst keine weiteren Informationen verfügbar sind, ist eine automatisierte Einstufung nur strukturell möglich. Zwar unterscheiden sie sich je nach Konfiguration in der Zusammensetzung der Felder, es können jedoch einige Bewertungskriterien aufgestellt werden.

---

<sup>3</sup>prototypische Darstellung des Problems mit den minimal erforderlichen Eigenschaften für die Erhebung von Kundenfeedback (vgl. [9])

<sup>4</sup>prototypische Lösung mit den minimal erforderlichen Eigenschaften für die Erhebung von Kundenfeedback (vgl. [9])

<sup>5</sup><http://wirtschaftslexikon.gabler.de/Definition/gamification.html> [Aufruf 08.10.14]

<sup>6</sup>vgl. 5-Stufenmodell für Produktmarketing und -management von Dave McClure: <http://500hats.typepad.com/500blogs/2007/09/startup-metrics.html> [Aufruf 22.11.14]

Für das *Product Vision Canvas* lassen sich folgende Bedingungen überprüfen:

- Grundvoraussetzung ist mindestens eine Notiz pro Feld.
- Die Anzahl der grundlegenden Funktionalitäten muss mindestens gleich der Anzahl der Systemnutzer sein. Dies lässt sich damit begründen, dass sich unterschiedliche Nutzergruppen über den Zugang zu unterschiedlichen Funktionalitäten definieren.
- Unabhängig vom optionalen Schwerpunkt auf der Qualitätssicherung sollten wenigstens drei wichtige Qualitätsfaktoren genannt und gewichtet sein.
- Falls die Qualitätssicherung ausgewählt wurde, gilt, dass mindestens eine Metrik für einen Qualitätsfaktor angegeben sein muss. Dabei ist es üblich, auch mehrere Metriken zu verwenden. Zur Bestimmung der Komplexität einer Software können beispielsweise die Schnittstellen betrachtet werden oder die Anzahl von Methoden. Würden bei zwei Programmen mit unterschiedlicher durchschnittlicher Größe einer Methode nur die Anzahl der Methoden berücksichtigt werden, wäre die abgeleitete Aussage über die Komplexität falsch.
- Zusätzlich zur Relation zwischen Qualitätsfaktor und Metrik muss die Anzahl der Metriken mit der Anzahl der konkreten Zielsetzungen übereinstimmen.

Zu den übrigen Feldern lassen sich strukturell keine allgemeingültigen Aussagen treffen.

Für das *Customer Interview Canvas* kommen folgende Bedingungen in Frage:

- Zu jedem Feld gibt es mindestens eine Notiz.
- Zum Feld *Project Status within Schedule* existieren wenigstens drei Notizen - der Projektbeginn, der nächste Meilenstein und der jetzige Projektstand. Wird der Arbeitsstand in funktionaler Dimension mit berücksichtigt, handelt es sich im Normalfall um eine wesentlich größere Anzahl Notizen.
- Die Ziele und inversen Ziele stimmen in ihrer Anzahl mit den Erfolgsindikatoren und den Risiken überein. Dies ist eine sehr strenge Bedingung, da sich die erfolgreiche Umsetzung eines Ziels unter Umständen auch in mehreren Auswirkungen zeigt oder umgekehrt. Es besteht ein klarer kausaler Zusammenhang zwischen den drei Feldern *Goal/Inverse Goal*, *Success Indicators* und *Risks*, es ist aber nicht sicher, ob der Nutzen der Bedingung die Einschränkung rechtfertigt.
- Zu jedem Anwendungsfall sollte mindestens ein Szenario beschrieben sein.
- Zu jeder Frage sollte am Ende des Kundengesprächs eine Antwort auf dem Canvas festgehalten sein. In Anbetracht der Platzbeschränkung auf einem Zettel (oder dem digitalen Äquivalent) wäre es allerdings denkbar, dass sich die Frage-Antwort-Relation nicht in der Relation der Zettelanzahl widerspiegelt.

Durch die eingeführten Bedingungen kann die Bearbeitung eines Canvas ungleich komplizierter werden. Um dies zu vermeiden, könnten die kritischen Bedingungen als Grundlage für Ausfüllhinweise dienen, ohne dass die Befolgung verpflichtend wird.

Für das *Iteration Planning Canvas* können die nachstehenden strukturellen Bedingungen gestellt werden:

- In jedem Feld steht mindestens eine Notiz.
- Falls es sich um eine Variante mit einem Kundenvertreter vor Ort handelt, gilt, dass die Anzahl an Acceptance Tests mindestens der der User Stories entsprechen muss.
- Im selben Fall muss die Anzahl der Tasks mindestens so groß sein wie die Anzahl der Stories. Umfasst die durchschnittliche Größe einer User Story beispielsweise vier Arbeitseinheiten, wird die Anzahl an Tasks entsprechend viermal so hoch. Als Ausnahme gilt, wenn z.B. ein Datenmodell mehreren Stories zugrunde liegt und die Erstellung eines Schemas entsprechend in eine Arbeitseinheit passt.
- Falls der Kunde nicht ins Projekt mit eingebunden ist, müssen die Arbeitseinheiten (Tasks) aus den im Customer Interview Canvas spezifizierten Anwendungsfällen abgeleitet werden. Die entsprechen im Umfang erfahrungsgemäß mindestens einer Story, deshalb kann die gleiche Relation ggf. zwischen den Canvases berücksichtigt werden. Das hängt allerdings davon ab, ob tatsächlich immer genau ein Gespräch im Laufe einer Projektiteration stattfindet.
- Abhängig davon, ob die Schätzung der Tasks in einer individuellen Sicht des Canvas oder offen im gleichen Canvas erfolgt, ist die Anzahl der Notizen für das Feld *Personal Estimates* gleich oder die Anzahl der Entwickler mal der Anzahl Tasks. Gibt es eine einfache Möglichkeit, die Anzahl der Entwickler aus den Angaben im Feld Resources zu gewinnen, kann auch diese Relation überprüft werden.
- Die Anzahl der persönlich angenommenen Tasks muss am Ende insgesamt gleich der Anzahl der Tasks sein bzw. für jeden persönlich kleiner als die Anzahl der Tasks. Ebenfalls möglich wäre ein Verschieben der Elemente.
- Ist das Canvas für die Förderung individueller Qualifikationen konfiguriert, entspricht die Anzahl der Notizen im Feld *Special Qualifications* der Anzahl Entwicklern und der Anzahl Notizen im Feld *Sought Experience*. Dies trifft nur zu, wenn eine Notiz pro Person verwendet wird. Anderenfalls ist sind die Anzahlen nicht bestimm- bzw. vergleichbar.

Zu den aufgelisteten Bedingungen zu den einzelnen Canvases besteht kein Anspruch auf Vollständigkeit. Allerdings bieten sie für die Bearbeitung eine gute Starthilfe. Können die Story Points aus den Notizen eindeutig zugeordnet werden, lässt sich für das Iteration Planning Canvas auch die Einhaltung und Berechnung aus vorherigen Iterationen automatisch überprüfen.

Sind all diese strukturellen Vorgaben erfüllt, kann trotzdem noch keine qualitative Aussage über



den Bearbeitungsstand gemacht werden. Weitere Kriterien können nur unter Berücksichtigung der individuellen Projektdetails geprüft werden. Da diese Informationen nicht für den Vergleich vorliegen, ist eine automatische Überprüfung nicht möglich. Eine Option wäre allerdings die Bereitstellung einer Bewertungsfunktion an Systemnutzer. Auf diese Weise könnte, entweder in gemeinsamer Abstimmung oder von speziell Zugriffsberechtigten eine manuelle Bewertung vorgenommen werden. Ein Vorschlag für die Bewertungsstufen zum *Product Vision Canvas*, angelehnt an das IRL, ist nachfolgend abgebildet.

1. Erstes vollständig skizziertes Product Vision Canvas
2. Vollständige Erfassung der Systemnutzer und grundlegenden -funktionalitäten
3. Einigung auf die Ressourcen und kundenseitige Festlegung dreier Variablen<sup>7</sup> (Zeitraumen, Budget, Qualität, Funktionsumfang)
4. Vollständige Erfassung der Domäne
5. Vereinbarung der Qualitätsfaktoren und deren Gewichtung sowie abhängig von der Konfiguration ggf. Metriken und Zielstellungen
6. Abhängig von der Konfiguration eine begründete Entscheidung über die Verwendung von Technologien bzw. eine klare Identifizierung und Definition der Schnittstellen

Zusätzlich zur Bewertung der Bearbeitungsvollständigkeit könnte eine Framework-seitige Unterstützung der digital erfassten Kriterien wie Zeitraumen, Meilensteine, und Erfolgskriterien für die in den einzelnen Kundengesprächen gesetzten Ziele in Betracht gezogen werden. Wichtiger ist aber eine gute Bedienbarkeit des Canvas-Tools.

---

<sup>7</sup>vgl. Abschnitt 7.2 - Resources bzw. [4]



## 9 Schlussbetrachtung

Zielstellung des Großen Belegs war ein konzeptioneller Beitrag zur Organisation von Software-Startups mit Canvases. Dazu wurden Canvases, als flexible, leichtgewichtige Werkzeuge für die Unterstützung der kollaborativen Erschließung komplexer Themen, analysiert, konstruiert und eine Produktlinie für den Einsatz in Softwareprojekten entwickelt. Die folgenden Abschnitte bilden mit einer Zusammenfassung der Ergebnisse, der anschließenden Auswertung und einem Ausblick auf mögliche Weiterführung den Abschluss der Arbeit.

### 9.1 Zusammenfassung

Zunächst wurden bekannte Canvases aus den Wissensgebieten Geschäftsmodellentwicklung und Softwareprojektsteuerung recherchiert. Die Gemeinsamkeiten, die Form der Anwendung und die Vorzüge des Werkzeugs, wurden in Kapitel 2 erläutert. Eine Gegenüberstellung ausgewählter Canvases wurde in Kapitel 3 durchgeführt. Dabei zeigte sich als hervorstechende Eigenschaft die Dynamik des Werkzeugs. Zum einen ist gemäß den jeweiligen Autoren eine wiederholte Bearbeitung der Felder nicht nur erlaubt sondern auch erwünscht - es gilt das Prinzip der iterativen Verbesserung. Zum anderen wird zwar zu jeder Vorlage eine Hilfestellung in Form vorgeschlagener Ausfüllreihenfolgen gegeben. Im Zusammenhang wird jedoch deutlich gemacht, dass eine Bearbeitung letztendlich nach eigenem Ermessen erfolgt. Gegenüber anderen Werkzeugen besteht der große Vorteil von Canvases in einer übersichtlichen visuellen Darstellung.

Als Erweiterung zu den existierenden Canvases sollten eigene Vorlagen entwickelt werden. Das erstere der beiden in diesem Zusammenhang entstandenen Werkzeuge wurde in Kapitel 4 unter dem Namen „Customer Interview Canvas“ eingeführt. Grundlage dazu boten die Einteilung von Kundenwünschen und geeignete Ermittlungstechniken in einer Artikelreihe von Rupp und Schüpferling [28–31]. Motiviert durch die iterative Projekt- und Aufgabenplanung im Extreme Programming mit dem Planning Game wurde in Kapitel 5 das „Iteration Planning Canvas“ eingeführt. Damit wurde das bisher bekannte „Lean Software Engineering Canvas“ [21], das in Teilen der Anforderungsspezifikation dient und andererseits Aspekte der technischen Umsetzung enthält, um zwei Vorlagen ergänzt: Eine Vorlage, spezialisiert auf die Spezifikation und gezielte Ermittlung der Anforderungen, sowie eine Vorlage, die sich auf die Projektplanung

konzentriert.

Im Zuge eines jährlichen universitären Projektkurses bot sich die Gelegenheit, die neu konzipierten Canvases in der Praxis zu evaluieren. Für die Studie wurden in Kapitel 6 ein Fragebogen erstellt sowie das entsprechende Material für fünf Studententeams bereitgestellt. Ausgewertet wurden neben direktem Feedback auch die konkreten Arbeitsergebnisse. Zu den wichtigsten Erkenntnissen gehören ein besonderer Anspruch an Klarheit und Eindeutigkeit sowie die Identifikation einiger struktureller Verbesserungsmöglichkeiten.

In einem nächsten Schritt hin zu einem Konfigurationsmodell wurde in Kapitel 7 eine Variabilitätsanalyse auf Basis der Felder aus den insgesamt drei Canvases mit Bezug zur Softwareentwicklung durchgeführt. Dazu wurden die Felder kategorisiert und mit Hilfe geeigneter Modifikationen in eine klarere Struktur überführt. Aus den ermittelten Kategorien entstand die Aufteilung in das „Product Vision Canvas“, das „Iteration Planning Canvas“ und das „Customer Interview Canvas“ (in neuer Zusammensetzung). Ebenso daraus abgeleitet werden konnten vier Konfigurationsmerkmale: Die Einbindung des Kunden ins Projekt, der Projekttyp und die beiden optionalen Schwerpunkte auf Qualitätsmessung und Förderung individueller Qualifikationen im Team. Das resultierende Feature-Modell ermöglicht 16 gültige Konfigurationen in Abhängigkeit dieser Parameter. Dabei variiert die Anzahl bereitgestellter Canvases zwischen zwei und drei um das zusätzliche Customer Interview Canvas<sup>1</sup>. Die Anzahl der Felder pro Canvas wurde mit maximal neun Feldern<sup>2</sup> bewusst klein gehalten, um die Übersichtlichkeit zu erhalten.

Ergänzend zum entstandenen Konfigurationsmodell wurde im Hinblick auf die primäre Zielstellung in Kapitel 8 diskutiert, inwieweit ein digitales Canvas-Framework Unterstützung bei der Bewertung der Bearbeitungs-Vollständigkeit bieten kann. Als mögliche Konzepte wurde hierfür neben einigen strukturellen Bewertungskriterien ein Maßstab für die Bewertung qualitativer Faktoren in Betracht gezogen. Potenzielle Profiteure der Integration solcher Bewertungshilfen sind sowohl das Startup-Unternehmen mit seinen Mitarbeitern wie auch Inkubatoren.

## 9.2 Bewertung und Ausblick

Die Studie mit den Studentengruppen wurde durchgeführt, um eine erste Aussage über die tatsächliche Anwendbarkeit von Canvases im Kontext von Softwareprojekten treffen zu können. Es wurde einiger Aufwand betrieben, um mit Hilfe einer schriftlichen Befragung unbeeinflusste Ergebnisse zu bekommen. Gleichzeitig wurde absichtlich auf eine persönliche Betreuung verzichtet. Dies geschah vor allem auch aus dem Grund heraus, dass sich das Canvas in seiner Eigenschaft als intuitives, leichtgewichtiges Werkzeug beweisen sollte. Leider wurde durch diese Entscheidung auch eine mögliche Einflussnahme auf ungünstige Studienbedingungen verhindert. So hätten durch die verstärkte Betreuung und Moderation unter Umständen der

---

<sup>1</sup>im Fall einer schwachen Kundeneinbindung in Form separater Kundengespräche

<sup>2</sup>vgl. Iteration Planning Canvas für die Konfigurationsmöglichkeiten 10,12,14 und 16, s. Anhang S. 68

Verzicht auf teilweise sich mit dem Canvas überschneidende Werkzeuge vereinbart und die Feedbackquote verbessert werden können. Als positiver Effekt konnten durch das formlose Feedback einer der Gruppen qualitative Erkenntnisse gewonnen werden, die durch den Fragebogen nicht berücksichtigt waren. Die niedrige Feedbackquote konnte durch eine Analyse der Arbeitsergebnisse ausgeglichen werden. Um eine repräsentative Aussage über die Nutzbarkeit des im Anschluss präsentierten konfigurierbaren Canvas-Sets zu bekommen, wäre eine größer angelegte Studie unter passenden Voraussetzungen denkbar.

Es gibt zu jedem Sachverhalt zahlreiche Aspekte, die als Komponenten für ein Canvas in Frage kommen. In dieser Arbeit wurde durch die Aufteilung und die Zusammensetzung der einzelnen Canvases eine Kombination angestrebt, die für die jeweiligen Kommunikationspartner unter Berücksichtigung der Geltungsdauer den größten Nutzen bietet. Die Arbeit liefert mit dem Canvas-Konfigurationsmodell für die Unterstützung von Softwareentwicklungsprojekten ein anpassungsfähiges Werkzeug, das einen wichtigen Teil der Organisation von Software-Startups abdeckt. Unsicherheiten, die in der Entwicklungsphase auftraten, unterstreichen, dass es zum einen keine kompromissfreie Lösung gibt und zum anderen Anwender-Feedback ein wesentlicher Schritt auf dem Weg zu einem guten Design ist. Der nächste Schritt wäre eine Erweiterung auf die wirtschaftlichen Prozesse eines Startups. Gute Anhaltspunkte dazu bieten die in Kapitel 3 vorgestellten Canvases für die Geschäftsmodellentwicklung und die von Steve Blank geschaffene Plattform „LeanLaunchpad“ (vgl. Kapitel 8), die Startup-Gründern bereits erfolgreich bei der Entwicklung, Validierung und Umsetzung ihrer Ideen unterstützt. Besondere Aufmerksamkeit gilt dabei der nahtlosen Verbindung und Berücksichtigung entstehender Zusammenhänge zu den integrierten Canvases. Zudem bieten sich Forschungsmöglichkeiten bezüglich weiterer auftretender Variabilitäten und Konfigurationsmöglichkeiten in den hinzukommenden Bereichen.

Im Hinblick auf die Integration in ein digitales Framework sind neben den offensichtlichen Vorteilen einer verbesserten Lesbarkeit, erleichtertem Zugriff, Vervielfältigung und Modifizierung eine Anzahl unterstützender Features denkbar. In der Diskussion geeigneter Metriken zur Bewertung der Vollständigkeit werden dafür verschiedene Ansätze erwähnt. Interessant wäre im Zusammenhang mit der angesprochenen Erweiterung des Frameworks die Ausführung, auf welche Weise diese sich mit ein oder mehreren Feedback-Mechanismen verbinden lässt.



# Anhang





# **Anhang A**

## **Customer Interview Canvas**

<p><b>Goal/Paradox Inverted Goal</b>  <i>What is the goal of the project?          What has to be avoided?</i></p>	<p><b>System Users</b>  <i>Who has access to the system?          For what type of user does the system offer a certain functionality?</i></p>	<p><b>Project Status Within Schedule</b>  <i>What is the schedule for the project (important dates)?          What is the current state of the project?</i></p>	<p><b>Domain Model</b>  <i>What are the objects identified within the project domain?          How do the objects relate to each other?</i></p>	<p><b>Success Indicators/Criteria</b>  <i>What measures are suitable to verify the success of the project?          What criteria does the project have to meet?</i></p>
<p><b>Resources</b>  <i>What are the resources for this project, e.g. developer team, experience &amp; expertise, money, time</i></p>	<p><b>Risks</b>  <i>What are the main risks to be addressed?</i></p>			
<p><b>Questions</b>  <i>What aspects need further clarification?</i></p>	<p><b>Answers</b>  <i>What are the answers as the interviewer has understood them?          Have they been understood correctly from the point of the customer?</i></p>			

# **Anhang B**

## **Iteration Planning Canvas**

<p><b>Acceptance Tests</b>  <i>What criteria does a story implementation have to meet to get accepted?          What effect/output is expected for a specific action/input?</i></p>	<p><b>Previous Achievements #</b>  <i>How many story points were done in the last release/iteration by the whole team?          How many estimated hours of work did you complete in the last iteration?</i></p>	<p><b>Release Plan</b>  <i>What features can/should be implemented?          How many story points are scheduled (based on previous releases)?</i></p>	<p><b>Special Qualifications</b>  <i>In which field is who the expert/has who currently a lot of practice?</i></p>	<p><b>Personal Subscriptions</b>  <i>What tasks have you subscribed for?</i></p>
<p><b>Stories</b>  <i>What type of user can execute what kind of action for what reason?          What is the estimated workload in story points?          (a single story should not exceed the workload for two persons for the whole iteration)</i></p>	<p><b>Sought Experience</b>  <i>In which field does who seek to gain more experience?</i></p>			
<p><b>Tasks</b>  <i>How are the stories subdivided into "simple units of work", e.g. db schema, html page servlet</i></p>		<p><b>Personal Estimates</b>  <i>What is your personal estimate for this task?</i></p>		

# **Anhang C**

## **Fragebogen zur Canvas-Studie**

# Evaluation zur Nutzung von Canvases in Softwareprojekten

Projektgruppe: \_\_\_\_\_ männlich  weiblich

## Verwendung des Canvas in diesem Projekt

1. Das Canvas war hilfreich bei der Anforderungsermittlung.      sehr       gar nicht
2. Die Fragen boten eine gute Hilfestellung für die Diskussion.      sehr       gar nicht
3. Durch die gemeinsame Bearbeitung des Canvas konnten Missverständnisse verhindert/beseitigt werden.      sehr       gar nicht

4. Welche Felder waren für dich besonders hilfreich, welche waren von geringem Nutzen? Bitte vergib zu jedem Feld eine Punktzahl zwischen 1 (kein erkennbarer Nutzen) und 10 (sehr hilfreich).

Feld 1      \_\_\_\_\_      Feld 4      \_\_\_\_\_      Feld 7      \_\_\_\_\_  
 Feld 2      \_\_\_\_\_      Feld 5      \_\_\_\_\_      Feld 8      \_\_\_\_\_  
 Feld 3      \_\_\_\_\_      Feld 6      \_\_\_\_\_      Feld 9      \_\_\_\_\_

1	2	4	5	7
	3		6	
8			9	

5. Wie oft habt ihr das Canvas überarbeitet?      0  1  2   $\geq 3$
6. Welche Hilfsmittel (Plakat, Whiteboard, bunte Zettel, Stifte,...) habt ihr für euer Canvas verwendet?

\_\_\_\_\_

\_\_\_\_\_

7. Falls du zuvor bereits mit anderen (vergleichbaren) Methoden gearbeitet hast, welche waren das? Mit welcher Methode kommst du am besten zurecht?

\_\_\_\_\_

\_\_\_\_\_

## Allgemeine Einschätzung des Tools

8. Die Ausfüllhinweise zu den Feldern sind ausreichend.      sehr       gar nicht
9. Das Canvas bietet eine gute Grundlage für das Gespräch mit dem Kunden.      sehr       gar nicht
10. Das Canvas bietet eine gute Grundlage für die Kommunikation im Team.      sehr       gar nicht
11. Die Bearbeitung des Canvas ist intuitiv.      sehr       gar nicht
12. Die Verwendung des Canvas-Tools macht Spaß.      sehr       gar nicht
13. Das Ausfüllen des Canvas ist Zeitverschwendung.      sehr       gar nicht
14. Das Canvas fördert die gemeinsame Entwicklung und Organisation von Ideen.      sehr       gar nicht
15. Das Canvas hilft bei der Erfassung des „großen Ganzen“.      sehr       gar nicht
16. Ich könnte mir vorstellen, das Canvas für andere Projekte wieder zu verwenden.      sehr       gar nicht
17. Ich könnte mir vorstellen, das Canvas in Form eines Software-Tools für andere Projekte zu verwenden.      sehr       gar nicht
18. Anmerkungen und Verbesserungsvorschläge:

\_\_\_\_\_

\_\_\_\_\_

Vielen Dank fürs Mitmachen!

# **Anhang D**

## **Mögliche Canvas-Konfigurationen**

Nr.	Kunden- Einbindung	Projekttyp	Qualitäts- sicherung	Förderung individueller Qualifikat.	Canvases
1	Separate Kundengespräche	Neuentwicklung	nein	nein	Produktions-Canvas mit dem zusätzlichen Feld <i>Technologies</i> ; Iterationsplanungs-Canvas; Kundengesprächs-Canvas
2	Separate Kundengespräche	Neuentwicklung	nein	ja	Produktions-Canvas mit dem zusätzlichen Feld <i>Technologies</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>Special Qualifications</i> , <i>Sought Experience</i> ; Kundengesprächs-Canvas
3	Separate Kundengespräche	Neuentwicklung	ja	nein	Produktions-Canvas mit den zusätzlichen Feldern <i>Technologies</i> , <i>Quality Metrics</i> , <i>Quality Objectives</i> ; Iterationsplanungs-Canvas; Kundengesprächs-Canvas
4	Separate Kundengespräche	Neuentwicklung	ja	ja	Produktions-Canvas mit den zusätzlichen Feldern <i>Technologies</i> , <i>Quality Metrics</i> , <i>Quality Objectives</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>Special Qualifications</i> , <i>Sought Experience</i> ; Kundengesprächs-Canvas
5	Separate Kundengespräche	Erweiterung eines Systems	nein	nein	Produktions-Canvas mit dem zusätzlichen Feld <i>Interfaces</i> ; Iterationsplanungs-Canvas; Kundengesprächs-Canvas
6	Separate Kundengespräche	Erweiterung eines Systems	nein	ja	Produktions-Canvas mit dem zusätzlichen Feld <i>Interfaces</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>Special Qualifications</i> , <i>Sought Experience</i> ; Kundengesprächs-Canvas
7	Separate Kundengespräche	Erweiterung eines Systems	ja	nein	Produktions-Canvas mit den zusätzlichen Feldern <i>Interfaces</i> , <i>Quality Metrics</i> , <i>Quality Objectives</i> ; Iterationsplanungs-Canvas; Kundengesprächs-Canvas



Nr.	Kunden- Einbindung	Projekttyp	Qualitäts- sicherung	Förderung individueller Qualifikat.	Canvases
8	Separate Kundengespräche	Erweiterung eines Systems	ja	ja	Produktions-Canvas mit den zusätzlichen Feldern <i>Interfaces, Quality Metrics, Quality Objectives</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>Special Qualifications, Sought Experience</i> ; Kundensprache-Canvas
9	Aktive Teilnahme am Projekt	Neuentwicklung	nein	nein	Produktions-Canvas mit dem zusätzlichen Feld <i>Technologies</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories, Acceptance Tests</i>
10	Aktive Teilnahme am Projekt	Neuentwicklung	nein	ja	Produktions-Canvas mit dem zusätzlichen Feld <i>Technologies</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories, Acceptance Tests, Special Qualifications, Sought Experience</i>
11	Aktive Teilnahme am Projekt	Neuentwicklung	ja	nein	Produktions-Canvas mit den zusätzlichen Feldern <i>Technologies, Quality Metrics, Quality Objectives</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories, Acceptance Tests</i>
12	Aktive Teilnahme am Projekt	Neuentwicklung	ja	ja	Produktions-Canvas mit den zusätzlichen Feldern <i>Technologies, Quality Metrics, Quality Objectives</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories, Acceptance Tests, Special Qualifications, Sought Experience</i>
13	Aktive Teilnahme am Projekt	Erweiterung eines Systems	nein	nein	Produktions-Canvas mit dem zusätzlichen Feld <i>Interfaces</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories, Acceptance Tests</i>

Nr.	Kunden- Einbindung	Projekttyp	Qualitäts- sicherung	Förderung individueller Qualifikat.	Canvases
14	Aktive Teilnahme am Projekt	Erweiterung eines Systems	nein	ja	Produktions-Canvas mit dem zusätzlichen Feld <i>Interfaces</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories</i> , <i>Acceptance Tests</i> , <i>Special Qualifications</i> , <i>Sought Experience</i>
15	Aktive Teilnahme am Projekt	Erweiterung eines Systems	ja	nein	Produktions-Canvas mit den zusätzlichen Feldern <i>Interfaces</i> , <i>Quality Metrics</i> , <i>Quality Objectives</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories</i> , <i>Acceptance Tests</i>
16	Aktive Teilnahme am Projekt	Erweiterung eines Systems	ja	ja	Produktions-Canvas mit den zusätzlichen Feldern <i>Interfaces</i> , <i>Quality Metrics</i> , <i>Quality Objectives</i> ; Iterationsplanungs-Canvas mit den zusätzlichen Feldern <i>User Stories</i> , <i>Acceptance Tests</i> , <i>Special Qualifications</i> , <i>Sought Experience</i>

# Literaturverzeichnis

- [1] APEL, S., D. BATORY, C. KÄSTNER und G. SAAKE: *Feature-Oriented Software Product Lines: Concepts and implementation*. Springer, 2013.
- [2] BARBACCI, M., M. H. KLEIN, T. A. LONGSTAFF und C. B. WEINSTOCK: *Quality Attributes*. Techn. Ber., SEI Carnegie Mellon University, 1995.
- [3] BATORY, D., D. BENAVIDES und A. RUIZ-CORTES: *Automated analysis of feature models*. Communications of the ACM, 49(12):45, 2006.
- [4] BECK, K.: *Extreme Programming Explained: Embrace Change*. An Alan R. Apt Book Series. Addison-Wesley, 2000.
- [5] BECK, K., M. BEEDLE, A. VAN BENNEKUM, A. COCKBURN, W. CUNNINGHAM, M. FOWLER, J. GRENNING, J. HIGHSMITH, A. HUNT, R. JEFFRIES, J. KERN, B. MARICK, R. C. MARTIN, S. MELLOR, K. SCHWABER, J. SUTHERLAND und D. THOMAS: *Manifesto for Agile Software Development*. Online verfügbar unter <http://agilemanifesto.org/>. [Aufruf 30.07.2014].
- [6] BLANK, S. G.: *Blog on Entrepreneurship and Conservation*. Online verfügbar unter <http://steveblank.com/>. [Aufruf 11.08.2014].
- [7] BLANK, S. G.: *It's Time To Play Moneyball: The Investment Readiness Level*. Online verfügbar unter <http://steveblank.com/2013/11/25/its-time-to-play-moneyball-the-investment-readiness-level/>. [Aufruf 11.08.2014].
- [8] BLANK, S. G.: *The Four Steps to the Epiphany: Successful strategies for products that win*. K&S Ranch, Inc., 2. Aufl., 2013.
- [9] BLANK, S. G. und B. DORF: *The Startup Owner's Manual: The step-by-step guide for building a great company*. K&S Ranch, Inc., 1. Aufl., 2012.
- [10] CZARNECKI, K. und U. W. EISENECKER: *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, New York, 2000.
- [11] ECLIPSE FOUNDATION AND MEMBERS: *Guideline: Planning Game*. Online verfügbar

- unter [http://epf.eclipse.org/wikis/xp/xp/guidances/guidelines/planning\\_game\\_3C5849C8.html](http://epf.eclipse.org/wikis/xp/xp/guidances/guidelines/planning_game_3C5849C8.html), 2007. [Aufruf 27.07.2014].
- [12] ECLIPSE FOUNDATION AND MEMBERS: *Eclipse Process Framework*. Online verfügbar unter <http://epf.eclipse.org/>, 2014. [Aufruf 27.07.2014].
- [13] GRICE, H. P.: *Logic and Conversation*. *Syntax and Semantics*, 3:41–58, 1975.
- [14] KAN, S. H.: *Metrics and Models in Software Quality Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd Aufl., 2002.
- [15] KANG, K. C., S. G. COHEN, J. A. HESS, W. E. NOVAK und A. S. PETERSON: *Feature-oriented domain analysis (FODA) feasibility study*. Techn. Ber., DTIC Document, 1990.
- [16] KROMER, T.: *Lean Startup Blog*. Online verfügbar unter <http://grasshopperherder.com/>. [Aufruf 11.08.2014].
- [17] KROMER, T.: *Business Model Canvas for User Experience*. Online verfügbar unter <http://grasshopperherder.com/business-model-canvas-for-user-experience/>, 2012. [Aufruf 20.07.2014].
- [18] MAURYA, A.: *A Blog on Entrepreneurship by Ash Maurya*. Online verfügbar unter <http://practicetrumpstheory.com/>. [Aufruf 11.08.2014].
- [19] MAURYA, A.: *Running Lean: Iterate from Plan A to a Plan That Works*. O'Reilly Media, Inc., 2012.
- [20] MCCLURE, D.: *Master of 500 Hats: A blog about Geeks, Entrepreneurs, & Startups in Silicon Valley, by Dave McClure. The Internet Revolution, Act III.*. Online verfügbar unter <http://500hats.typepad.com/>. [Aufruf 20.11.2014].
- [21] ODDOY, M.: *Softwareentwicklung mit Natürlicher Sprache („Lean Modelling“)*. Unveröffentlichter Großer Beleg, Technische Universität Dresden, 2013.
- [22] OSTERWALDER, A.: *Business Model Alchemist*. Online verfügbar unter <http://businessmodelalchemist.com/>. [Aufruf 11.08.2014].
- [23] OSTERWALDER, A. und Y. PIGNEUR: *Business Model Generation: A Handbook For Visionaries, Game Changers, And Challengers*. Wiley, 2010.
- [24] POHL, K., G. BÖCKLE und F. VAN DER LINDEN: *Software Product Line Engineering: Foundations, principles, and techniques*. Springer, New York, 1. Aufl., 2005.
- [25] PORST, R.: *Praxis der Umfrageforschung*. Studienskripten zur Soziologie. Teubner, 2000.

- [26] PORST, R.: *Fragebogen: Ein Arbeitsbuch*. Springer, 3. Aufl., 2011.
- [27] RIES, E.: *Startup Lessons Learned*. Online verfügbar unter <http://www.startuplessonslearned.com/>. [Aufruf 11.08.2014].
- [28] RUPP, C. und D. SCHÜPFERLING: *Warum Sie in Interviews mit Kunden nie die ganze Wahrheit erfahren*. Online verfügbar unter <http://jaxenter.de/artikel/Warum-Sie-in-Interviews-mit-Kunden-nie-ganze-Wahrheit-erfahren>, 2013. [Aufruf 22.07.2014].
- [29] RUPP, C. und D. SCHÜPFERLING: *Warum Sie in Interviews nie die ganze Wahrheit erfahren: Fragen und Antworten: Teil 2*. Online verfügbar unter <http://jaxenter.de/artikel/Warum-Sie-in-Interviews-nie-ganze-Wahrheit-erfahren-Fragen-Antworten>, 2013. [Aufruf 22.07.2014].
- [30] RUPP, C. und D. SCHÜPFERLING: *Warum Sie in Interviews nie die ganze Wahrheit erfahren: Fragen und Antworten: Teil 3*. Online verfügbar unter <http://jaxenter.de/artikel/Warum-Sie-in-Interviews-nie-ganze-Wahrheit-erfahren-Fragen-Antworten-1>, 2013. [Aufruf 22.07.2014].
- [31] RUPP, C. und D. SCHÜPFERLING: *Warum Sie in Interviews nie die ganze Wahrheit erfahren: Fragen und Antworten: Teil 4*. Online verfügbar unter <http://jaxenter.de/artikel/Warum-Sie-in-Interviews-nie-ganze-Wahrheit-erfahren-Fragen-Antworten-0>, 2013. [Aufruf 22.07.2014].
- [32] STARKE, G. und P. HRUSCHKA: *Das arc42 Template*. Online verfügbar unter <http://www.arc42.de/template/>, 2012. [Aufruf 29.07.2014].



# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Belegarbeit mit dem Titel „Organisierte Software-Startups mit kollaborativen Canvases“ selbstständig verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt. Die wörtlichen oder sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht.

Dresden, den 4. Januar 2015





# Customer Interview Canvas

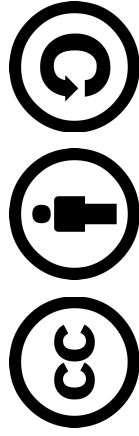
<p><b>Goal/Paradox Inverse Goal</b>  <i>What is the goal of the project?          What has to be avoided?</i></p>	<p><b>System Users</b>  <i>Who has access to the system?          For what type of user does the system offer a certain functionality?</i></p>	<p><b>Project Status Within Schedule</b>  <i>What is the schedule for the project (important dates)?          What is the current state of the project?</i></p>	<p><b>Domain Model</b>  <i>What are the objects identified within the project domain?          How do the objects relate to each other?</i></p>	<p><b>Success Indicators/Criteria</b>  <i>What measures are suitable to verify the success of the project?          What criteria does the project have to meet?</i></p>
<p><b>Resources</b>  <i>What are the resources for this project, e.g. developer team, experience &amp; expertise, money, time</i></p>	<p><b>Risks</b>  <i>What are the main risks to be addressed?          How are they weighted?</i></p>	<p><b>Questions</b>  <i>What topics need to be discussed in this meeting?          What aspects need further clarification?</i></p> <p><b>Answers</b>  <i>What are the answers as the interviewer has understood them?          Have they been understood correctly from the point of the customer?</i></p>		



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Customer Interview Canvas

Project Status within Schedule		Goal/Inverse Goal	Success Indicators
Use Cases	Scenarios		Risks
Questions		Answers	



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Iteration Planning Canvas

Release Plan	Personal Estimates	Special Qualifications
Previous Achievements #	Tasks	Sought Experience
Personal Subscriptions		



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

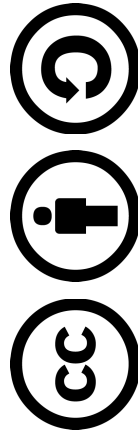
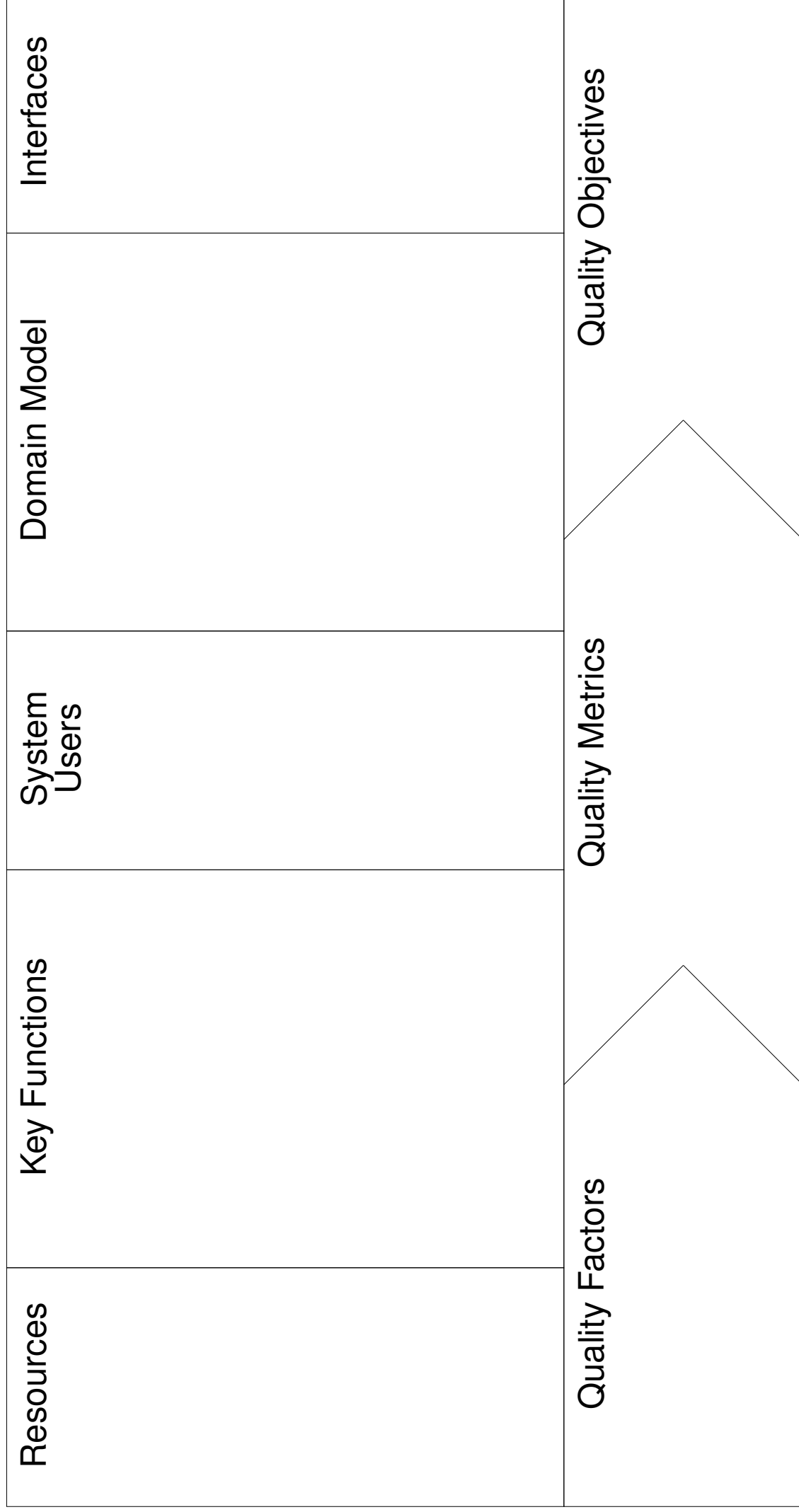
# Iteration Planning Canvas

Release Plan	Personal Estimates	Special Qualifications
Previous Achievements #	Tasks	Sought Experience
Personal Subscriptions		



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Product Vision Canvas



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Iteration Planning Canvas

<p><b>Acceptance Tests</b>  <i>What criteria does a story implementation have to meet to get accepted?          What effect/output is expected for a specific action/input?</i></p>	<p><b>Previous Achievements #</b>  <i>How many story points were done in the last release/iteration by the whole team?          How many estimated hours of work did you complete in the last iteration?</i></p> <p><b>Stories</b>  <i>What type of user can execute what kind of action for what reason?          (a single story should not exceed the workload for two persons for the whole iteration)</i></p>	<p><b>Release Plan</b>  <i>What features can/should be implemented?          How many story points are scheduled (based on previous releases)?</i></p>	<p><b>Special Qualifications</b>  <i>In which field is who the expert/has who currently a lot of practice?</i></p> <p><b>Sought Experience</b>  <i>In which field does who seek to gain more experience?</i></p>	<p><b>Personal Subscriptions</b>  <i>What tasks have you subscribed for?</i></p>
<p><b>Tasks</b>  <i>How are the stories subdivided into "simple units of work", e.g. db schema, html page servlet</i></p>			<p><b>Personal Estimates</b>  <i>What is your personal estimate for this task?</i></p>	



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.