Friedrich-Alexander-Universität Erlangen-Nürnberg Technische Fakultät, Department Informatik

BERND GRILLENBERGER
MASTER THESIS

A QUALITY MODEL FOR INNER SOURCE

Eingereicht am 3. August 2017

Betreuer: Prof. Dr. Dirk Riehle, M.B.A., M.Sc. Maximilian Capraro

Professur für Open-Source-Software

Department Informatik, Technische Fakultät

Friedrich-Alexander-Universität Erlangen-Nürnberg

Versicherung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, 3. August 2017

License

This work is licensed under the Creative Commons Attribution 4.0 International license (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/

Erlangen, 3. August 2017

Abstract

Inner source (IS) is the utilization of open source (OS) development practices within an organization. Some organizations run IS projects or even IS programs. So far, there are published OS quality models, but no quality model is specifically known for IS.

This paper presents a quality model for IS programs and projects. We conduct five interviews with IS experts and analyze them by means of thematic analysis. Based on the resulting insights, we develop a hierarchical quality model for IS programs and IS projects, which we subsequently delimitate in relation to researched OS quality models.

Zusammenfassung

Inner Source (IS) ist die Verwendung von Open Source (OS) Entwicklungspraktiken innerhalb einer Organisation. Einige Organisationen führen IS Projekte oder sogar IS Programme durch. Bisher gibt es zwar veröffentlichte OS Qualitätsmodelle, allerdings ist kein Qualitätsmodell speziell für IS bekannt.

Dieses Papier präsentiert ein Qualitätsmodell für IS Programme und Projekte. Wir führen fünf Interviews mit IS Experten durch und analysieren diese mittels thematischer Analyse. Anhand der daraus entstandenen Einblicke entwickeln wir ein hierarchisches Qualitätsmodell für IS Programme und IS Projekte, die wir anschließend zu recherchierten OS Qualitätsmodellen abgrenzen.

Inhaltsverzeichnis

1	Vor	wort	1												
	1.1	Ursprüngliche Thesis Ziele	1												
	1.2	Änderungen der Thesis Ziele	1												
2	Fors	schung	2												
	2.1	Einleitung	2												
	2.2	Verwandte Arbeiten	3												
		2.2.1 Inner Source	3												
		2.2.2 Open Source	3												
	2.3	Forschungsfrage	6												
	2.4	Forschungsmethode	6												
		2.4.1 Experteninterviews	7												
		2.4.2 Literaturrecherche	9												
		2.4.3 Thematische Analyse	10												
	2.5		12												
	2.6		14												
			16												
		2.6.2 Hierarchisches Qualitätsmodell Projekt	18												
			20												
	2.7														
			23												
			24												
	2.8		25												
3	Näh	ere Ausarbeitung	26												
	3.1	Experteninterviews	26												
		-	26												
			27												
	3.2	•	27												
4	Dan	ksagung	29												

Anhänge	
Anhang A	Interview-Guideline
Anhang B	Vorabinformation Interviewpartner
Anhang C	Transkriptionsregeln
Anhang D	Open Source Qualitätsmodelle
Anhang E	Analyse - Suchen nach Themen
Anhang F	Inhalte der CD

Abbildungsverzeichnis

2.1	Zuordnung OS Qualitätscharakteristiken - Veröffentlichungen	
	(Adewumi, Misra, Omoregbe, Crawford, & Soto, 2016)	5
2.2	MAXQDA: Code Matrix Erfolgsindikatoren	14
2.3	Qualitätsmodell Programm	16
2.4	Qualitätsmodell Projekt	18
2.5	MAXQDA: Code Matrix Metriken	20
4.1	OSMM Qualitätsmodell	
	(Miguel, Mauricio, & Rodríguez, 2014)	35
4.2	OpenBRR Qualitätsmodell	
	(Miguel, Mauricio, & Rodríguez, 2014)	36
4.3	SQO-OSS Qualitätsmodell	
	(Miguel, Mauricio, & Rodríguez, 2014)	37
4.4	QualOSS Qualitätsmodell	
	(Miguel, Mauricio, & Rodríguez, 2014)	38
4.5	Programm Analyse: Suche nach Themen	
4.6	Projekt Analyse: Suche nach Themen	40

Tabellenverzeichnis

2.1	Rolle und Phase der Interviewpartner	13
4.1	Interview-Guideline	31

1 Vorwort

1.1 Ursprüngliche Thesis Ziele

Das übergeordnete Ziel der Thesis ist es, Erfolgsindikatoren für IS herauszuarbeiten. Dies soll sowohl mittels der Durchführung von fünf Interviews mit IS Experten als auch Literaturrecherche von OS Qualitätsmodellen und IS Literatur erfolgen. Dabei sollen im Laufe der Arbeit drei Ergebnisse entstehen, die in dieser Reihenfolge aufeinander aufbauen:

- 1. Interview-Guideline für Experteninterviews
- 2. Transkription der fünf Experteninterviews (fremde Hilfe möglich) und Code-Systeme aufgrund der Analyse der Interviews
- 3. Hierarchisches Qualitätsmodell (hierarchische Erfolgsindikatoren) von IS Programmen und Projekten

1.2 Änderungen der Thesis Ziele

Die Ziele der Thesis wurden nicht geändert. Die Anfertigung der Transkripte wurde ohne fremde Hilfe vorgenommen und zusätzlich eine ausführliche Recherche zu Transkriptionsmethoden durchgeführt.

2 Forschung

2.1 Einleitung

Inner Source (IS) ist die Verwendung von Open Source (OS) Entwicklungspraktiken innerhalb einer Organisation (Capraro & Riehle, 2016). Bei IS unterscheidet man zwischen einem IS Programm und einem IS Projekt. Capraro und Riehle (2016) definieren das IS Programm als einen koordinierten Aufwand einer Organisation, ein oder mehrere IS Projekte zu ermöglichen und zu führen. Das IS Projekt definieren sie als ein Software Projekt mit dem Ziel IS Software zu entwickeln und zu warten. Es ähnelt einem OS Projekt und hat keine feste Dauer oder definiertes Enddatum.

In der bestehenden Literatur gibt es mehrere veröffentlichte OS Qualitätsmodelle, die teilweise auf unterschiedliche Aspekte von Qualität eingehen.

Allerdings gibt es bis dato keine veröffentlichten Qualitätsmodelle speziell für IS, die Unternehmen dabei unterstützen, ihre IS Programme und IS Projekte zu bewerten und im Gesamtkontext entscheiden zu können, ob ein IS Programm oder IS Projekt in ihrem Unternehmen gut läuft bzw. erfolgreich ist.

Wie bereits erwähnt gibt es zwei Perspektiven, die sich voneinander unterscheiden. Die projektspezifische und programmweite Perspektive. Der Blick auf ein IS Programm als Ganzes konzentriert sich auf die komplette IS Landschaft einer Organisation. Im Gegensatz dazu ist der Fokus bei einer projektspezifischen Sicht auf die Umgebung eines bestimmten Projekts gerichtet, die beteiligten Parteien und ihre Interessen.

Um Unternehmen eine Hilfestellung zu geben, entwickeln wir in dieser Arbeit ein IS Qualitätsmodell für IS Programme und IS Projekte. Wir verwenden dabei einen qualitativen Ansatz, indem wir fünf Interviews mit IS Experten durchführen und per Audio aufzeichnen. Anschließend werden diese transkribiert und mit dem Tool MAXQDA¹ nach der thematischen Analyse nach Braun und Clarke (2006) analysiert. Die bestehende OS Literatur wird im Hinblick auf OS Qualitätsindikatoren näher betrachtet und zu unseren entworfenen Modellen in Abgrenzung

¹http://www.maxqda.de/

gesetzt. Wir liefern somit Erfolgsindikatoren für IS Programme und IS Projekte und ergänzend listen wir Metriken zum Vermessen einzelner Aspekte auf, die in den Interviews genannt wurden.

Diese Arbeit ist wie folgt strukturiert: Zunächst werden verwandte Arbeiten näher betrachtet. In Abschnitt 2.3 wird genauer auf die Forschungsfrage eingegangen. Im folgenden Abschnitt werden die benutzten Forschungsmethoden dargestellt. Abschnitt 2.5 erläutert die verwendeten Datenquellen. Abschnitt 2.6 zeigt die herausgearbeiteten Qualitätsmodelle und Metriken. Die letzten beiden Abschnitte Diskussion der Ergebnisse und zukünftige Arbeit bilden den Abschluss.

2.2 Verwandte Arbeiten

2.2.1 Inner Source

Bisherige Veröffentlichungen zu IS beschäftigten sich unter anderem mit Themen wie Definitionen, Herausforderungen und Vorteile (Capraro & Riehle, 2016) oder Schlüsselfaktoren bei der Einführung von IS (Stol, Avgeriou, Babar, Lucas, & Fitzgerald, 2014). Keine der bekannten IS Arbeiten geht aber konkreter auf Erfolgsindikatoren von IS Programmen und Projekten ein.

2.2.2 Open Source

Das erste OS Qualitätsmodell veröffentlichten Duijnhouwer und Widdows (2003) mit dem Namen Open Source Maturity Model (OSMM). Dieses Modell gibt eine Hilfestellung bei der Auswahl, welches Open Source Produkt zur Verwendung für die eigenen Bedürfnisse geeignet ist. Das Modell sieht eine Proportion der Qualität eines OS Projekts zum Reifegrad des Projekts. Es unterteilt in die zwei Gruppen Produkt und Application Indicators und definiert dazu einige Indikatoren und Subindikatoren. Unter den Subindikatoren für ein Produkt befinden sich unter anderem auch Aspekte wie die Developer Community oder die User Community. Die Source Code Qualität betrachtet dieses Modell nicht.

Das Qualification and Selection of Open Source Software (QSOS) ist ein weiteres Modell zur Auswahl von OS Software, das im Jahr 2004 veröffentlicht wurde. Es besteht aus den vier iterativen Schritten Define, Evaluate, Qualify und Select und bietet auch Tool Support an².

Das Open Business Readiness Rating (OpenBRR) entstand 2005 und enthält Charakteristiken hohen Levels wie Functionality, Operational Software, Support,

²http://www.qsos.org/Method.html

Documentation, Software Technology, Communication and Adoption und Development Process zu denen einige Subcharakteristiken genannt werden. Es bezieht somit im Gegensatz zu OSMM den Development Prozess mit ein (Samoladas, Gousios, Spinellis, & Stamelos, 2008). Die ursprüngliche Webseite und Quelle des OpenBRR³ ist allerdings nicht mehr erreichbar.

Samoladas u. a. (2008) präsentieren ein weiteres hierarchisches Modell, das Source Code und Community Prozesse auf Basis von automatischer Berechnung von Messwerten evaluiert. Software Quality Observatory for Open Source Software (SQO-OSS) unterscheidet sich von den vorher aufgeführten Modellen dahingehend, dass der Fokus auf die Automatisierung, auf das kontinuierliche Überwachungssystem und auf den Source Code gerichtet ist. Die zwei Qualitätscharakteristiken sind Product Code Quality und die Community Quality, welche jeweils Subcharakteristiken enthalten. In der Community Quality werden jedoch lediglich Aspekte betrachtet, die automatisch gemessen werden können. Sie enthält Mailing List Quality, Documentation Quality und Developer Base Quality.

Ein weiteres Modell namens Quality of Open Source Software (QualOSS) veröffentlichten Soto und Ciolkowski (2009). Es soll sowohl die Robustness des Produkts als auch die Evolvability abdecken. Es wird auf Produktcharakteristiken eingegangen aber auch genauer auf die Prozesse und die Community Mitglieder, die für die Evolvability eine entscheidende Rolle spielen.

Aberdour (2007) analysiert empirische Daten, um Schlussfolgerungen daraus zu ziehen, wie OS Software Qualität erreicht werden kann und welche Faktoren dabei eine entscheidende Rolle spielen. Als Aspekte dafür werden eine große und nachhaltige Community, Code Modularität, gute Dokumentation, Tutorials, Entwicklungswerkzeuge, Belohnungs- und Anerkennungskultur, kurze Release Zyklen, Code Review durch Personen außerhalb des Projekts und Tool Support für die Zusammenarbeit erwähnt.

Ähnlich dazu erwähnen Crowston, Wei, Howison, und Wiggins (2012) unterschiedliche Forschungen, die verschieden Erfolgsaspekte bei Free/Libre Open Source Software (FLOSS) Projekte aufzeigen. Dabei werden Aspekte wie System- und Informationsqualität, Benutzerzufriedenheit, Benutzung, Individuelle und organisatorische Einflüsse, Projekt Output, Prozess und Erfolg für Projektmitglieder, Software Qualität und Community Qualität genannt.

Miguel, Mauricio, und Rodríguez (2014) fassen bestehende Qualitätsmodelle zur Evaluation von Software Produkten zusammen. Darunter befinden sich auch die OS Qualitätsmodelle OSMM, OpenBRR, SQO-OSS und QualOSS. Diese Arbeit stellt Grafiken der einzelnen OS Qualitätsmodellen bereit, die sich im Anhang D befinden.

³http://www.openbrr.org

Adewumi, Misra, und Omoregbe (2015) evaluieren die bestehenden OS Qualitätsmodelle hinsichtlich der Norm ISO 25010. Das ist zwar nicht unser Augenmerk, allerdings wird in dieser Veröffentlichung genauer auf die einzelnen Aspekte der Modelle eingegangen. Die acht genannten Modelle sind OSMM, QSOS, Open-BRR, Sung et al., QualOSS, OMM, SQO-OSS und EFFORT.

Adewumi, Misra, Omoregbe, Crawford, und Soto (2016) geben eine gute Übersicht der bestehenden Modelle oder Veröffentlichungen, die es zur Bewertung von OS Qualität gibt und deren Erscheinungsjahr. Außerdem verdeutlichen sie, auf welche unterschiedlichen Qualitätscharakteristiken in welcher Veröffentlichung bzw. in welchem Modell eingegangen wird, siehe Abbildung 2.1.

Quality character- istics	OSMM	QSOS	Open BRR	Sung et al.	Qua- IOSS	ОММ	SQO- OSS	EFFORT	Raf- foul et al.	Alfonzo et al.	Wray and Mathleu	Mul- ler	Chirila et al.	OS- UMM	Adewumi et al.	Sudhaman and Thangavel	Sohn et al.	Kuwata et al.	Sarrab and Rehman
ISO 25010																			
Product qua	lity																		
Func- tional Suit- ability	X		X	х	х	X		х	X	х		X							х
Reliability	X				X	X	X	X	X				X		X				x
Perfor- mance effi- clency	x		X		X	X		х									х		х
Operabil- Ity	X		X	X	X				X	X									
Security	X		X	X	X	X	x		X				X						X
Compat- Ibility	X			Х	X	X			X	X									
Maintain- ability	X	X	X	X	X		Х	X		X			X		X				х
Tranfer- ability	X			X	X			X					X				Х		
Quality in us	e																		
Effective- ness							Х		X	X									
Efficiency											X	X				X			
Satisfac- tion																			
Safety																			
	Х		X	X				X	Х	X		X		х			х		X
Community		lity charac	teristics																
Mainte- nance Capac- Ity	X		X		х	X	X	Х							х		X	Х	
Sustain- ability	X		X		X	X	X	X				X						х	
Process Matu- rity	X		Х		х		X	x							х			x	

Abbildung 2.1: Zuordnung OS Qualitätscharakteristiken - Veröffentlichungen (Adewumi, Misra, Omoregbe, Crawford, & Soto, 2016)

Wir wollen in dieser Arbeit kein Modell zur Auswahl eines Inner Source Produkts entwickeln, wie es das QSOS Modell macht. Außerdem gehen die Modelle OSMM und OpenBRR nur undetailliert auf die Community Aspekte ein, auf die wir besonders unseren Fokus legen. Das SQO-OSS Modell fokussiert sich nur auf die Aspekte, die sich automatisiert vermessen lassen, was für uns erst einmal

zweitrangig ist. Wir liefern zwar auch Metriken, allerdings sind diese für uns eher Mittel zum Zweck. Beim Entwickeln von QualOSS wurden zusätzlich zu früheren Modellen und generellen Software Standards wie ISO 9126 ebenfalls Experteninterviews als Quelle verwendet und auch besonderes Augenmerk auf Prozesse und die Community gelegt. Des Weiteren gehen wir nicht wie die meisten Modelle auf die Code Qualität an sich ein, da es zum einen nicht Teil dieser Arbeit ist und zum anderen Soto und Ciolkowski (2009) betont, dass sich die Produkt Aspekte von OS Software mit den gleichen verfügbaren Techniken anderer Software Typen vermessen lassen. Ähnlich zu den bestehenden Modellen sollen unsere Qualitätsmodelle hierarchisch entworfen werden.

2.3 Forschungsfrage

In dieser Arbeit soll die Frage beantwortet werden, welche Erfolgsindikatoren es innerhalb von IS gibt. Dazu arbeiten wir grundsätzlich zwei Fragen heraus:

- Was sind Erfolgsindikatoren für hohe Qualität von IS Programmen?
- Was sind Erfolgsindikatoren für hohe Qualität von IS Projekten?

Im Vornherein grenzen wir zunächst den weitläufigen Begriff Qualität für IS genauer ab, da es verschiedene Dimensionen für Qualität gibt.

In unserer Arbeit definieren wir die Qualität über die IS Praktiken und besonders über die IS Community und nicht wie bereits erwähnt über die Code Qualität. Außerdem verstehen wir unter einem Erfolgsindikator einen bestimmten Aspekt, der eine Bewertung der Qualität eines IS Programms oder IS Projekts ermöglicht. Dieser kann aus weiteren Suberfolgsindikatoren bestehen.

2.4 Forschungsmethode

Zum einen führen wir fünf Interviews mit IS Experten durch, die wir per Audio aufzeichnen und transkribieren, um qualitative Daten zu IS Programmen und IS Projekten zu erhalten. Sie stellen unsere Basis der entstehenden Modelle dar. Anschließend verwenden wir das Programm MAXQDA für die qualitative Datenanalyse, bei der wir die Vorgehensweise der thematischen Analyse nach Braun und Clarke (2006) einsetzen.

Zum anderen recherchieren wir die bestehende Literatur zu OS im Hinblick auf Qualitätsmodelle und Erfolgsindikatoren und diskutieren diese in Bezug auf unsere herausgearbeiteten Modelle.

2.4.1 Experteninterviews

Zu Beginn qualitativer Forschung muss die Methode für das Sammeln der relevanten Daten gewählt werden. Oft werden hierzu Interviews geführt (Kaufmann & Riehle, 2015).

Das Experteninterview grenzt durch die Befragung einer besonderen Zielgruppe von anderen qualitativen Interviews ab. Prinzipiell gibt es zwei forschungspraktische Gründe, warum Experteninterviews sehr beliebt sind. Zum einen erspart man sich lange Forschungswege die aufgrund von beispielsweise Beobachtungen über einen längeren Zeitraum erfolgen. Zum anderen stellen die Experten Kristallisationspunkte praktischen Insiderwissens dar und dienen stellvertretend für eine Vielzahl der sonst zu befragenden Akteuren. Experteninterviews haben somit den Zweck besonders exklusives, detailliertes und umfassende Wissen über Wissensbestände und Praktiken offen zu legen (Bogner, Littig, & Menz, 2005).

In der qualitativen Forschung kommen gerne die nichtstandardisierten Interviews zur Anwendung, da diese sowohl dem Interviewer als auch dem Interviewpartner mehr Freiraum lassen. Leitfadeninterviews stellen eine bestimmte Form von nichtstandisierten Interviews dar. Bei diesen Interviews verwendet der Interviewer keinen konkreten Fragebogen, sondern eine Aufzeichnung mit Themenblöcken und jeweiligen Fragen dazu, einer sogenannten Interview-Guideline. Dabei sind die Reihenfolge als auch die exakte Verwendung der Frageformulierung nicht verbindlich. Ein natürlicher Gesprächsverlauf steht im Vordergrund, wobei alle Themenblöcke und deren Fragen in einer freien Art und Weise beantwortet werden sollen.

Interview-Guideline

Nach den theoretischen Vorüberlegungen und der Literaturrecherche entwickelten wir die Interview-Guideline. Diese Guideline ist gegliedert in Hauptkategorien mit dazugehörigen Leitfragen, die auf das Untersuchungsfeld gerichtet sind. Sie charakterisieren das benötigte Wissen im Hinblick zur Beantwortung der Forschungsfrage. Bei der Formulierung der Leitfragen wurde darauf geachtet, dass der Interviewpartner nach seinem Wissen und Interesse frei nach dem sogenannten Prinzip der Offenheit antworten kann (Gläser & Laudel, 2009).

Zur Entwicklung von Interviewleitfäden hat Helfferich (2004) ein methodisches Verfahren gestaltet. Das nun von uns ausgewählte SPSS-Verfahren hat sich in der Forschungspraxis bewährt und besteht im Wesentlichen aus vier Schritten, die zirkulär miteinander verbunden sind (Kruse, 2015).

• S: Sammeln sehr vieler Fragen in einem offenen Brainstorming

- P: Prüfen der Eignung sämtlicher Fragen, alle ungeeigneten Fragen werden gestrichen
- S: Sortieren der Fragen sowohl inhaltlich als auch nach Fragetyp (Erzählaufforderung, Aufrechterhaltungsfragen, konkrete Nachfragen)
- S: Subsumieren der Fragen in den Leitfaden

Nach mehrmaliger Anwendung des SPSS-Verfahren und Überprüfung der Guideline durch erfahrene wissenschaftliche Mitarbeiter entstanden folgende Hauptkategorien:

- A: Hintergrund des Interviewten
- B: IS Erfolgsindikatoren
 - B1: IS Erfolgsindikatoren Programm
 - B2: IS Erfolgsindikatoren Projekt
- C: IS Erfolgsindikatoren aus verwendeten Metriken
 - C1: Existierende IS Metriken
 - C2: Erwünschte IS Metriken

Vor einem Interview erfolgt anfangs eine kurze Vorstellung. Ergänzend wird sich zu Beginn für die genommene Zeit des Interviewpartners und für die Zustimmung zur Audioaufzeichnung bedankt. Bevor die Fragen gestellt werden, wird noch einmal klar das Ziel der Forschung erklärt. Nach dem Stellen der Leitfragen wird sich erneut für das Interview bedankt und dem Befragten die Möglichkeit gegeben, etwas von sich aus zu erzählen. Abschließend wird die Möglichkeit geäußert, sich bei Fragen oder Anregungen gerne mit uns in Verbindung zu setzen.

Die vollständige Interview-Guideline inklusive aller Leitfragen und auch den Anmerkungen vor und nach dem Interview befindet sich im Anhang A.

Vorabinformation der Interviewpartner

Im Sinne der ethischen Aspekte einer Expertenbefragung, ist eine Vorabinformation des Interviewpartners obligatorisch. Folgend die beschriebenen Punkte nach Kaiser (2014).

- Kurze Beschreibung des Forschungsthemas
- Organisatorischen Hintergrund (studentisches Forschungsprojekt)

- Beabsichtigte Verwertung erhobener Forschungsdaten (Master Thesis mit anschließender Veröffentlichung)
- Technische Durchführung des Interviews (Audioaufzeichnung)
- Zeitraum bzw. Dauer des Interviews
- Vertrauliche Behandlung der Interviewdaten

Unsere vollständige Information, die jedem Interviewpartner im Voraus zugeschickt wurde, befindet sich im Anhang B.

Interview Transkription

Da die Art des Transkriptionssystems die Auswertung der Daten beeinflusst, müssen zuvor die verschiedenen Systeme betrachtet und in Bezug zu unserem Forschungsvorhaben gesetzt werden. Unterschieden werden diese prinzipiell sowohl im Umfang des transkribierten Textes und Informationsgehalt als auch nach sprachlicher Genauigkeit (von Kardorff, Steinke, & Flick, 2008).

Für unser Forschungsvorhaben verwendeten wir die wörtliche Transkription, weil wir die Transkription vom ersten Auswertungsschritt getrennt halten möchten. Weiterhin ist angedacht, die Interviews für weitere Forschungszwecke zu verwenden. Für die Interpretation wurden wichtige Auffälligkeiten wie das Lachen des Interviewpartners mit vermerkt. Für die wörtliche Transkription verwendeten wir die Literarische Umschrift, die einen Dialekt bzw. Umgangssprache in gebräuchliches Alphabet bereinigt.

Für eine bessere Basis bei der späteren Analyse der Transkripte spielt die Einheitlichkeit eine wichtige Rolle. Daher bedarf es klar vorgegebene Regeln, die das Transkriptionssystem beschreiben. Für unsere Experteninterviews verwendeten wir die Regeln nach Dresing und Pehl (2013), die den Fokus auf den Inhalt setzen. Die Auflistung der Regeln mit minimaler Anpassung an die eigenen Bedürfnisse befinden sich im Anhang C.

2.4.2 Literaturrecherche

Van der Linden, Lundell, und Marttiin (2009) und Capraro und Riehle (2016) betonen die Ähnlichkeit der IS Entwicklung zur OS Entwicklung mit dem Unterschied der Begrenzung auf die eigene Organisation statt weltweiter Offenheit. Daher betrachteten wir die Literatur zu OS Qualitätsmodellen und Erfolgsindikatoren, um diese später zu unseren beiden aus den Interviews entstandenen Modellen in Kapitel 2.7 abzugrenzen.

2.4.3 Thematische Analyse

In der qualitative Datenanalyse werden beispielsweise Interviews oder wissenschaftliche Veröffentlichungen mit sogenannten Codes versehen, um diese zu strukturieren und in ein hierarchisches Codesystem zu übertragen. Die Theoriebildung erfolgt dann auf Basis der Ergebnisse der Codierung.

Ein wichtiger Aspekt in der Forschung ist, dass die Schritte der Analyse nachvollziehbar sein müssen. Daher wird empfohlen ein bestehendes und bewährtes Vorgehen bzw. Verfahren zu verwenden und die einzelnen Schritte der Analyse zu dokumentieren.

Wir wählten die thematische Analyse nach Braun und Clarke (2006), da sich bei diesem Verfahren die Hauptmerkmale einer großen Datensammlung sinnvoll zusammenfassen lassen und die Gemeinsamkeiten als auch Unterschiede in der Datensammlung hervorgehoben werden können. Außerdem ist dieses Verfahren schnell zu erlernen und anzuwenden.

Für die computergestützte qualitative Datenanalyse wählten wir das Tool MAXQDA (Version 11).

1 Vertrautmachen mit den Daten

Gemäß des ersten Schrittes der thematischen Analyse wurde sich mit den Daten vertraut gemacht. Die Interviews transkribiert, Korrektur gehört, wiederholt gelesen und die ersten Ideen als Notizen festgehalten, sofern diese nicht schon während des Interviews notiert wurden.

2 Erstellen der Anfangscodes

Die Transkripte der Interviews wurden in MAXQDA importiert.

Im nächsten Schritt haben wir die Erfolgsindikatoren der Interviewdaten in einer systematischen Weise über den ganzen Datensatz codiert und in den von den Interviewpartnern genannten Kontext in Programm oder Projekt einsortiert.

Da unser Fokus in erster Linie auf den Erfolgsindikatoren liegt, codierten wir vorerst die genannten Metriken mit einem Platzhalter. Anschließend wurden diese nach näherer Betrachtung ebenfalls codiert, um möglicherweise Erfolgsindikatoren ableiten zu können.

3 Suchen nach Themen

Anschließend wurden die Codes von Programmen und Projekten getrennt zu potentiellen Themen sortiert und in einem ersten Entwurf hierarchisch geordnet.

4 Überprüfung der Themen

Folgend überprüften wir die Themen, indem die gesamte Datenbasis erneut gelesen wurde und die Themen im Kontext dazu gesetzt wurden. Es wurden Themen überarbeitet, neue Themen erstellt und Attribute umgeordnet oder entfernt.

5 Definition und Benennung der Themen

Daraufhin passten wir teilweise die zu Beginn vergebenen Bezeichnungen nochmal an und bestimmten klare Definitionen für jedes Attribut.

Die Attribute wurden auch in MAXQDA umbenannt und hierarchisch gruppiert.

6 Ergebnisbericht

Im letzten Schritt extrahierten wir klare überzeugende Beispiele auch im Hinblick an die Forschungsfrage. Neben der Codierung in MAXQDA erstellten wir parallel ein Codebook, das diese Informationen übersichtlich bereitstellt.

Während der thematischen Analyse beachteten wir auch die 15-Punkte Checkliste von Braun und Clarke (2006).

Die beiden entstandenen hierarchischen Modelle werden in Kapitel 2.6 näher erläutert.

2.5 Verwendete Datenquellen

Für uns waren insbesondere bei der Auswahl der Experten zwei Dimensionen maßgeblich. Zum einen die Rolle, die der Interviewpartner innerhalb von IS einnimmt und zum anderen der Reifegrad des IS Programms.

Die unterschiedlichen möglichen Rollen werden folgend aufgelistet und kurz erläutert.

- Contributor (Entwickler, Architekt)
 - Einzelpersonen, die als Beitragender in IS Projekten tätig sind
- Committer (Entwickler, Architekt)
 - Einzelpersonen, die als Committer in IS Projekten tätig sind
- Product Manager
 - Einzelpersonen mit Verantwortlichkeit der Marktanalyse und Priorisierung von Features
- Engineering Manager (Projektleiter, Vize Präsident Engineering)
 - Einzelperson, die eine Reihe von Ingenieuren betreut
 - Oft interessiert an lokalem Nutzen (nicht primär globale Unternehmensperspektive)
- Top-Level Manager (CTOs, CEOs)
 - Einzelperson, die eine Vielzahl von Ingenieuren und andere Manager betreut
 - Typischerweise interessiert an einem Nutzen für die globale Unternehmensperspektive

Die zweite Dimension ergibt sich aus den verschiedenen IS Hintergründen der ausgewählten Experten. Dieser Hintergrund ist abhängig vom Reifegrad des IS Programms, in dem sie tätig sind. Folgend unterscheiden wir zwischen zwei Reifegraden von IS Programmen.

- Einführungsphase
 - Programme, die aktuell in der Erstellung sind bzw. Unternehmen, die gerade IS eingeführt haben
- Etablierte Phase
 - Programme, die bereits nachweislich etabliert wurden
 - Indikator: Programme, die seit mehr als 18 Monaten existieren

Mit Nutzung der Kontakte des Lehrstuhls wurden potentielle Kandidaten für ein Interview kontaktiert.

Die Tabelle 2.1 zeigt die Rollen und von welcher Phase unsere Interviewpartner kommen.

Rolle	Einführungsphase	Etablierte Phase
Contributor	Interview 2, Interview 4	
Committer	Interview 1	
Product Manager		
Engineering Manager	Interview 5	
Top-level Manager	Interview 3	

Tabelle 2.1: Rolle und Phase der Interviewpartner

Aufgrund des Umfangs der Thesis und der vorher begrenzten Anzahl an Interviewpartner konnten nur fünf Interviews mit Experten aus der Einführungsphase durchgeführt werden. Welche Einschränkungen das für die Modelle bedeutet, wird in Kapitel 2.7.2 näher beschrieben. Es wurde bei der Auswahl der Experten darauf geachtet, die Erfolgsindikatoren aus verschiedenen Perspektiven zu betrachten und dabei die unterschiedlichen Rollen, die die Personen im IS Umfeld inne haben, in das Qualitätsmodell mit einzubeziehen, um ein von der Rolle der Experten unabhängiges Qualitätsmodell erstellen zu können.

2.6 Forschungsergebnisse

Die Abbildung 2.2 zeigt die Anfangscodierung in MAXQDA und in wie vielen unterschiedlichen Interviews diese zur Aussprache kamen. Dadurch lässt sich eine erste Tendenz erkennen, welche Erfolgsindikatoren auch für mehrere Interviewpartner wichtig sind.

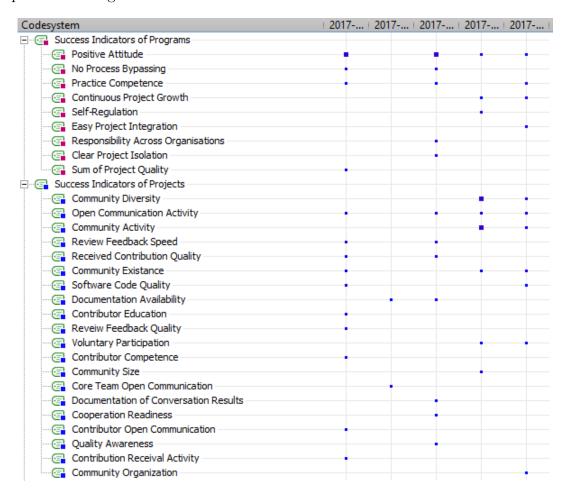


Abbildung 2.2: MAXQDA: Code Matrix Erfolgsindikatoren

Folgend beschreiben wir zunächst ein paar Terminologien, die wir verwendeten. Ein Core Team ist eine Gruppe von Entwicklern, die Verantwortung für die Entwicklung, Wartung und Support eines Inner Source Projekts hat (Höst, Stol, & Oručević-Alagić, 2014).

Unter Community verstehen wir eine Gruppe von Entwicklern, die sich rund um ein Projekt oder auch Programm gebildet hat. Dies beinhaltet sowohl die beitragenden Entwickler (Contributors) als auch das Core Team.

Dementsprechend beinhaltet die Contributor Community nur die beitragenden Entwickler ohne Core Team.

Die Zeichnungen des Zwischenschritts zum Suchen der Themen befinden sich im Anhang E.

Warum manche Erfolgsindikatoren bei der Überprüfung der Themen entfernt wurden ist folgend beschrieben.

Programm:

- Responsibility Across Organization: Für diesen Erfolgsindikator ist der Begriff Core Team Diversity besser geeignet und dieser ist aber projektspezifisch. Daher wurde dieser Erfolgsindikator zu den Erfolgsindikatoren der Projekte verschoben.
- Self-Regulation: Dieser Erfolgsindikator wurde zuerst zu No Explizit Responsibility geändert. Allerdings sehen wir diesen Punkt sehr kritisch und nicht geeignet für einen Erfolgsindikator.
- Continoous Project Growth: Diesen Erfolgsindikator haben wir entfernt, da es nicht zwingend heißt, dass wenn die Projekte nicht mehr wachsen, das Programm deswegen schlecht ist, da eine Stagnation des kontinuierlichen Wachstums eines Projektes nicht zwingend die Wertigkeit des Programms definiert.

Projekt:

- Quality Awareness: Dieser Erfolgsindikator steckt bereits in Contribution Quality. Daher wurde dieser Erfolgsindikator entfernt.
- Cooperation Readiness: Dieser Erfolgsindikator steckt bereits in Contribution Activity, da die Kooperationsbereitschaft eine Voraussetzung für Beitragsaktivitäten sind. Daher wurde dieser Erfolgsindikator entfernt.
- Voluntary Participation: Diesen Erfolgsindikator decken wir teilweise durch Cooperation Readiness ab. Außerdem ist es schwer zu definieren, was aufgezwungen ist und was freiwillig. Daher wurde dieser Erfolgsindikator entfernt.
- Community Organization: Dies ist ein sehr allgemeiner Punkt und zielt im Prinzip auf die Kommunikationskanäle ab. Diesen Aspekt haben wir schon in Open Communication enthalten. Daher wurde dieser Erfolgsindikator entfernt.
- Software Code Quality: Wir legen in dieser Arbeit kein Augenmerk auf die interne Code Qualität. Daher wurde dieser Erfolgsindikator entfernt.

Außerdem wurde die Community Diversity ebenfalls auf die Programm Ebene übertragen, da dies auch ein wichtiger Erfolgsindikator für ein IS Programm ist. Die Aspekte Organization Reach und Project Generality sind im Laufe der thematischen Analyse zum Qualitätsmodell des Programms hinzugekommen.

2.6.1 Hierarchisches Qualitätsmodell Programm

Die Abbildung 2.3 zeigt das hierarchische Qualitätsmodell für ein IS Programm, das aus der thematischen Analyse der fünf Experteninterviews entstanden ist.

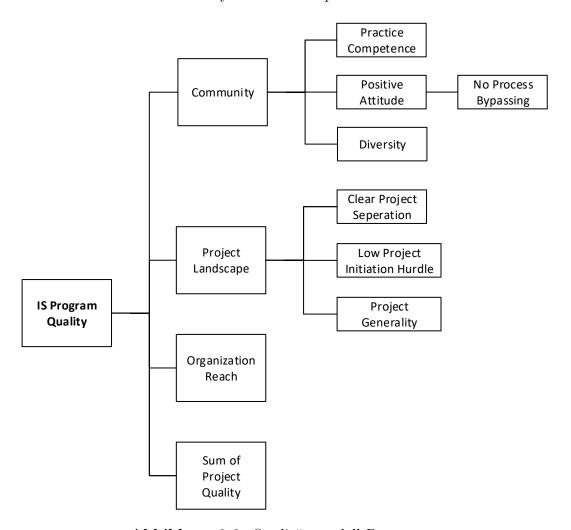


Abbildung 2.3: Qualitätsmodell Programm

Community: Die Existenz und vor allem Gegenwärtigkeit einer Community

- Practice Competence: Das Verständnis über IS Praktiken und der IS Arbeitsweise der Mitarbeiter einer Organisation
- Positive Attitude: Die Akzeptanz von IS und positive Einstellung zu IS Praktiken der Mitarbeiter einer Organisation
 - No Process Bypassing: Kein Umgehen der bestehenden IS Praktiken, Prozesse und etablierten Verfahren

• Diversity: Die verschiedenen Hintergründe von denen die Community Mitglieder kommen (z.B. Business Units), was eine breitere Expertise innerhalb eines Programms bringt

Project Landscape: Das Vorhandensein einer strukturierten Projektlandschaft

- Clear Project Seperation: Die klare Definition und Abgrenzung der unterschiedlichen Projekte
- Low Project Initiation Hurdle: Das Vorhandensein einer geringen Projektstart Hürde durch Bereitstellen einer einfachen Entwicklungsumgebung
- **Project Generality:** Die Generalität für die Eignung für eine Vielzahl an Community Mitgliedern

Organization Reach: Die Reichweite über verschiedene Business Units und Divisions hinweg

Sum of Project Quality: Die Summe der Qualität der einzelnen Projekte innerhalb eines Programms

2.6.2 Hierarchisches Qualitätsmodell Projekt

Die Abbildung 2.4 zeigt das hierarchische Qualitätsmodell für ein IS Projekt, das aus der thematischen Analyse der fünf Experteninterviews entstanden ist.

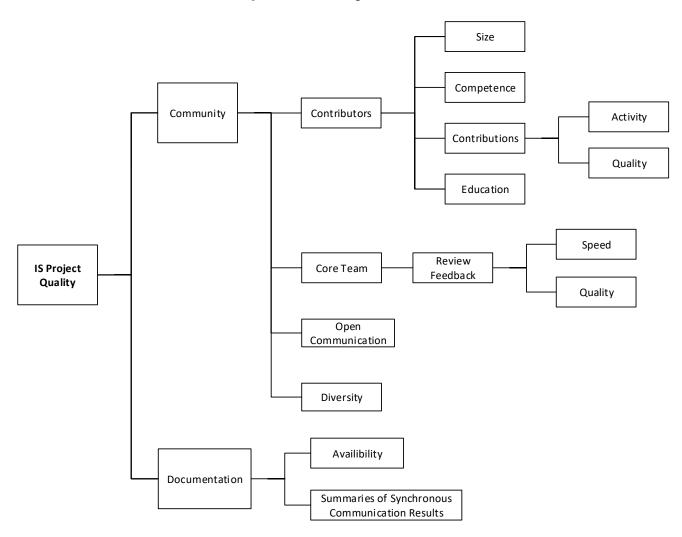


Abbildung 2.4: Qualitätsmodell Projekt

Community: Die Existenz und Gegenwärtigkeit einer Community

- Contributor Community: Die Existenz und Gegenwärtigkeit einer Contributor Community
 - Size: Die Anzahl an Contributors innerhalb einer Community
 - Competence: Die gewisse fachliche Kompetenz, die die Contributors besitzen

- Contributions: Die Beiträge der Contributors
 - * Activity: Die Regelmäßigkeit, in der die Contributors zu einem Projekt beitragen
 - * Quality: Die hohe Qualität und das Bewusstsein über die hohe Qualität der Beiträge, die die Contributors liefern
- Education: Die Ausbildung, die die Contributors durch das Verteilen und Bereitstellen von Wissen der Core Team Mitglieder erhalten
- Core Team: Die Existenz und Gegenwärtigkeit eines Core Teams
 - Review Feedback: Das Feedback, das die Committer an die Contributors zu ihren Beiträgen geben
 - * Quality: Das gute, konstruktive, hilfreiche und faire Feedback, das die Committer an die Contributors zu ihren Beiträgen geben
 - * **Speed:** Das angemessen schnelle Feedback, das die Committer an die Contributors zu ihren Beiträgen geben
- Open Communication: Die Existenz und die Verwendung von offenen Kommunikationsplattformen sowohl durch das Core Team als auch der Contributors
- **Diversity:** Die verschiedenen Hintergründe von denen die Community Mitglieder kommen (z.B. Business Units), was eine breitere Expertise innerhalb eines Projekts bringt

Documentation: Die Dokumentation von wichtigen Informationen

- Availability: Die Verfügbarkeit der Dokumentation und Informationen über ein Projekt, sodass diese eingesehen und das Projekt und deren Software verstanden werden können
- Summaries of Synchronous Communication Results: Die Dokumentation und Bereitstellung der Ergebnisse von synchron geführten Gesprächen

2.6.3 Metriken

Metriken erlauben einen betrachteten Aspekt zu vermessen und somit auch zu bewerten und zu vergleichen. Das Augenmerk in dieser Arbeit liegt zwar auf den Erfolgsindikatoren von IS, allerdings wollen wir trotzdem die Metriken auflisten, die in den Interviews aufgekommen sind. Die Abbildung 2.5 zeigt wie oft diese Metriken in den unterschiedlichen Interviews genannt wurden.

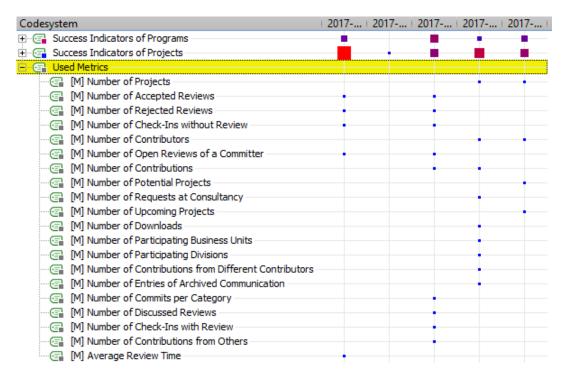


Abbildung 2.5: MAXQDA: Code Matrix Metriken

Zu den Metriken geben wir die Zuordnung zu einem oder mehreren Erfolgsindikator, eine kurze Definition und eine positive Tendenz an. Wir unterscheiden in welchem Kontext die einzelnen Metriken im Interview genannt wurden zwischen Projekt und Programm, wobei vereinzelt Metriken auch beiden zugeordnet werden können. Außerdem lassen sich Metriken eines Programms teilweise herunter brechen auf die Ebene eines Projekts. Die Einteilung ist also nicht als absolut anzusehen.

Zuerst erläutern wir die Metriken, die im Kontext eines IS Programms genannt wurden. Die Auflistung ist sortiert nach der Häufigkeit der Nennung in den Interviews.

Number of Projects (Low Project Initiation Hurdle, Positive Attitude): Die Anzahl der Projekte, die existieren. Eine große Anzahl ist als positiv anzusehen.

Number of Contributors (Positive Attitude, Contributions Activity, Size, Project Generality): Die Anzahl aller Contributors. Eine große Anzahl ist als positiv anzusehen.

Number of Check-Ins without Review (Positive Attitude, No Process Bypassing): Die Anzahl von Check-Ins, bei denen kein Code Review Objekt angehängt ist. Eine kleine Anzahl ist als positiv anzusehen.

Number of Check-Ins with Review (Positive Attitude, No Process Bypassing): Die Anzahl von Check-Ins, bei denen ein Code Review Objekt angehängt ist. Eine große Anzahl ist als positiv anzusehen.

Number of Participating Business Units (Positive Attitude, Diversity, Organization Reach, Contributions Activity, Project Generality): Die Anzahl der verschiedenen Business Units, die zu anderen Projekten beitragen. Eine große Anzahl ist als positiv anzusehen.

Number of Participating Divisions (Positive Attitude, Diversity, Organization Reach, Contributions Activity, Project Generality): Die Anzahl der verschiedenen Divisionen, die zu anderen Projekten beitragen. Eine große Anzahl ist als positiv anzusehen.

Number of Upcoming Projects (Positive Attitude, Low Project Initiation Hurdle): Die Anzahl der Projekte, die kurz davor sind in das Programm einzusteigen (eingerichtet aber noch nicht public). Eine große Anzahl ist als positiv anzusehen.

Number of Potential Projects (Positive Attitude): Die Anzahl an Projekten, die prinzipiell für das IS Programm geeignet wären. Eine große Anzahl ist als positiv anzusehen.

Number of Requests at Consultancy (Positive Attitude): Die Anzahl an interessierten Mitarbeiter, die Anfragen zum Thema IS an die Beratung stellen.

Folgend listen wir die Metriken auf, die im Projekt Kontext genannt wurden.

Number of Accepted Reviews (Competence, Contributions Quality): Die Anzahl der Review Objekte, die über einen bestimmten Zeitpunkt akzeptiert wurden. Eine große Anzahl ist als positiv anzusehen.

Number of Rejected Reviews (Competence, Contributions Quality): Die Anzahl der Review Objekte, die über einen bestimmten Zeitpunkt abgelehnt wurden. Eine kleine Anzahl ist als positiv anzusehen.

Number of Open Reviews of a Committer (Review Feedback Speed): Die Anzahl offener und unbearbeiteter Reviews eines Committers zu einem bestimmten Zeitpunkt. Eine kleine Anzahl ist als positiv anzusehen.

Average Review Time (Review Feedback Speed): Die durchschnittliche Zeit vom Stellen des Requests bis zum Erhalt des Feedbacks. Eine kurze Zeit ist als positiv anzusehen.

Number of Contributions (Contributions Activity): Die Anzahl der Beiträge, die von allen Beitragenden kommen. Eine große Anzahl ist als positiv anzusehen.

Numbers of Contributions from Others (Contributions Activity, Diversity): Die Anzahl der Beiträge, die von Personen anderer Projekte kommen. Eine große Anzahl ist als positiv anzusehen.

Number of Contributions from Different Contributors (Diversity, Project Generality): Die Anzahl der Beiträge, die von verschiedenen Contributors kommen. Eine große Anzahl ist als positiv anzusehen.

Number of Discussed Reviews (Competence, Contributions Quality): Die Anzahl der Reviews, die z.B. in einem Diskussion Board diskutiert wurden. Eine kleine Anzahl ist als positiv anzusehen.

Number of Commits per Category: Die Anzahl der jeweiligen Kategorie zugehörigen Commits (Fehlerbehebung, Change Request, neues Feature, Produktänderung). Es ist als positiv anzusehen, wenn die Commits auch aus unterschiedlichen Kategorien kommen.

Number of Entries of Archived Communication (Open Communication, Contributions Activity): Die Anzahl der Einträge, die in der archivierten Kommunikation existieren. Eine große Anzahl ist als positiv anzusehen.

Number of Downloads: Die Anzahl der Downloads eines IS Produkts. Eine große Anzahl ist als positiv anzusehen.

Ergänzend wurden die Metriken genauer betrachtet, um daraus eventuell noch nicht identifizierte Erfolgsindikatoren abzuleiten. Allerdings konnten keine neuen Indikatoren daraus abgeleitet werden, sondern lediglich den Bestehenden zugeordnet werden.

2.7 Diskussion der Ergebnisse

2.7.1 Abgrenzung zu Open Source Qualitätsmodellen

Bei der Literaturrecherche wurden einige OS Veröffentlichungen identifiziert, allerdings beruhten sich viele ihrer Aspekte auf die OS Software Qualität und nicht auf die Bewertung eines OS Projekts mit derer Community und Praktiken an sich. Außerdem unterscheiden die identifizierten Qualitätsmodelle nicht zwischen einem Projekt und einem Programm in der Gesamtheit, weshalb das Qualitätsmodell für IS Programme nicht abgegrenzt werden kann.

Folgend werden die einzelnen bereits in Kapitel 2.2 erwähnten Veröffentlichungen zu OS Qualität näher betrachtet.

Da wir den Fokus auf die Praktiken und auf die Community setzen, sind QSOS, und Sung, Kim, und Rhew (2007) nicht für die Abgrenzung unserer Modelle geeignet, da sie ihr Augenmerk auf die Auswahl von OS Software setzen.

Im OSMM Modell wird in der Developer Community die Aktivität der Entwickler und die dokumentierten Prozeduren, ein Community Mitglied zu werden, aufgeführt. In der User Community spielt die Größe und die Anzahl der Untergruppen eine Rolle. Weitere genannte Indikatoren sind Ease of Deployment (Documentation HOWTO's), Support, Reporting und Training. In unseren Modellen decken diese Aspekte Contributions Activity, Documentation, Contributors Size als auch Core Team und Education ab. Der Aspekt der verschiedenen Untergruppen innerhalb einer Community kam in unseren Interviews nicht auf.

OpenBRR führt Documentation, Community und Support als Attribut auf, wobei Support noch die Subattribute Training, Service, Consulting Service enthält. Auch hier beinhalten Documentation, Community, Core Team, Education in einer Art und Weise die genannten Aspekte. Lediglich Consulting Service befindet sich nicht direkt in unserem Modell.

Das SQO-OSS unterteilt die Community lediglich in Mailing List Quality, Documentation Quality und Developer Base Quality. Entsprechend in unserem Modell sind das Open Communication, Documentation, Contributors Competence, und Contributions Quality.

Das QualOSS Modell listet Documentation mit Subattribut Availability auf. Außerdem das Attribut Community Members mit den Subattributen Size and Regeneration Adequacy, Interactivity and Workload Adequacy und Composition Adequacy. In Kontrast zu unserem Modell sind es Documentation Availability, Contributors Size, Contributors Activity, Community Diversity.

Des Weiteren Software Process mit den Subattributen Capability of Requirements and Change Management, Capability of Release Management, Capability of Support and Community Management. Diese beinhalten Core Team, Review

Feedback und auch den Erfolgsindikator eines Programms wie Low Project Initiation Hurdle, das mit der Bereitstellung einer einfachen Entwicklungsumgebung einhergeht.

EFFORT führt unter Community Trustworthiness die Subattribute Community Activity, Support Tools, Support Services und Documentation auf. Auch hier sind die Aspekte wie bereits oben erwähnt durch unser Modell abgedeckt.

Die hier genannten OS Modelle bekräftigen die in unserer Arbeit teilweise in einem detaillierterem Grad herausgearbeiteten Erfolgsindikatoren.

Die Community Aspekte haben wir in unserem Modell spezialisierter in Contributor und Core Team aufgespalten, wohingegen die bisherigen Modelle mit Begriff Community die Contributor Community in unserem Sinne darstellen. In den Interviews kam deutlich hervor, dass das Core Team auch sehr wichtig ist im Hinblick auf das Review Feedback, weshalb ein eigenständiger Erfolgsindikator entstand.

Insgesamt tauchten in der OS Literatur kaum erkenntnisreiche neue Aspekte auf, die wir in unseren Modellen nicht in einer Art und Weise enthalten haben, wenn man die interne Software Qualität wie Functionality, Maintainability, Security, Usability, Reliability etc. außen vor lässt. Dies spricht durchaus für die Qualität der Interviews.

Die spezielle Kategorisierung der Modelle wurde zunächst kritisch geprüft, da sie ein Artefakt (Project), einen Akteur (Community) und Praktiken (Project Landscape) beinhalten. Allerdings ist dies bei QualOSS ähnlich, das ebenfalls ein Artefakt (Product), einen Akteur (Community) und Praktiken (Software Process) enthält. Außerdem konnte keine Normalisierung gefunden werden, die alle in den Interviews genannten Erfolgsindikatoren abdeckt.

2.7.2 Einschränkungen

Da aufgrund des definierten Umfangs der Thesis keine Interviews mit Experten aus der etablierten Phase durchgeführt wurden, zeigt das Qualitätsmodell erste Erfolgsindikatoren auf, die für unsere Interviewpartner vor allem zu Beginn von IS wichtig sind. Es ist durchaus wahrscheinlich, dass in der etablierten Phase der Programme andere Erfolgsindikatoren zum Ausdruck kommen oder dass der ein oder andere Erfolgsindikator nur in der Einführungsphase relevant ist und deshalb das Qualitätsmodell an neue Erkenntnisse angepasst werden muss. Beispielsweise fiel uns sowohl in der Codierung als auch in der Auflistung der Metriken auf, dass für unsere Interviewpartner besonders die grundsätzliche positive Einstellung der Mitarbeiter zu IS in der Startphase eine hohe Priorität hat. Bei diesem Erfolgsindikator könnten wir uns vorstellen, dass dieser in der etablierten Phase immer

mehr als selbstverständlich angesehen wird.

Des Weiteren waren aufgrund des definierten Umfangs der Arbeit von vornherein nur fünf Interviews geplant. Diese Anzahl ist überschaubar und möglicherweise bringen weitere Interviews noch nicht entdeckte Erfolgsindikatoren zum Vorschein. Es ist daher nicht auszuschließen, dass weitere Erfolgsindikatoren in das Modell einzuarbeiten wären.

2.8 Zukünftige Arbeit

In dieser Arbeit haben wir zwei Qualitätsmodelle anhand der Analyse von fünf Experteninterviews entwickelt, die Unternehmen dabei unterstützen, die Qualität ihrer IS Programme und IS Projekte zu bewerten.

Allerdings stammten alle Interviewpartner aus der Einführungsphase. Für zukünftige Forschungen ist es daher besonders interessant, Interviewpartner mit verschiedenen Rollen aus der etablierten Phase zu interviewen. Dies bietet die Möglichkeit weitere Erfolgsindikatoren oder Metriken herauszukristallisieren, die im Laufe der IS Programmen oder IS Projekten entstanden sind, und diese im Modell zu ergänzen. Oder es hat sich mit der Zeit herausgestellt, dass Erfolgsindikatoren oder Metriken, die zu Beginn festgelegt wurden, sich als nicht wirklich aussagekräftig erwiesen haben und somit das Modell angepasst werden muss. Außerdem gilt es auch festzusetzen, wie sich die Gewichtung bzw. Priorisierung der Erfolgsindikatoren möglicherweise in der etablierten Phase verändert. Das hierarchische Modell ist daher nicht als vollständig und abgeschlossen anzusehen, sondern bietet einen ersten Ansatz für Unternehmen, die sich in der Einführungsphase von IS befinden, und für darauf aufbauende Forschungen.

Neben den Erfolgsindikatoren sind auch persönliche Aspekte der Teilnehmer von IS interessant. Dieses Thema kam bei einem Interview zur Sprache. Zu untersuchen wäre in diesem Fall, was sich die einzelne Person selbst darunter vorstellt, ein Mitglied in einer IS Community zu sein und im weiteren Verlauf ergänzend was sie sich erhofft hat und wie sie es tatsächlich empfunden hat. Eine Gegenüberstellung der IS Erwartungen und IS Erfahrungen wäre ein weiteres zu untersuchendes Forschungsthema.

3 Nähere Ausarbeitung

3.1 Experteninterviews

3.1.1 Varianten

Gläser und Laudel (2009) unterscheiden Interviews anhand ihres Standardisierungsgrades. Die strengste Form sind standardisierte Interviews. Bei diesen sind die Fragen des Interviewers und Antwortmöglichkeiten immer identisch. Es werden ausformulierte geschlossen Fragen in einer festen Reihenfolge verwendet, die dem Interviewpartner in der Beantwortung keine Freiheit lässt. Diese Form findet ihre Anwendung besonders in der quantitativen Forschung. Halbstandardisierte bzw. teilstandardisierte Interviews haben einen standardisierten Fragebogen, überlassen jedoch dem Interviewpartner alle Freiheiten bei der Beantwortung. Bei den nichtstandardisierten Interviews sind weder die Fragen noch die Antworten standardisiert.

Außerdem geht Kaiser (2014) auf drei Varianten qualitativer Experteninterviews ein. Das explorative Experteninterview beschreibt er als allgemeine Informationssammlung in einem bisher kaum erforschten Themengebiet mit dem Ziel der Hypothesenbildung oder der "Felderöffnung", um relevante Experten zu identifizieren

Die zweite Variante ist das Leitfaden-gestützte Experteninterview. Dieses ist eine stärker strukturierte Form der Befragung mit dem Ziel Fakten zu bekommen, die sich aus anderen Quellen wie z.B. Literatur nur bedingt ermitteln lassen. Das Leitfaden-gestützte Experteninterview dient dabei der Beantwortung einer bereits präzisen Forschungsfrage durch Abfrage spezifischem Wissens.

Das Plausibilisierungsgespräch wird als dritte Variante genannt. Durch sie gelangt man an Handlungsempfehlungen oder Hinweise zur Präsentation der Forschungsergebnisse. Sie werden nach Abschluss des Forschungsprogramms geführt. (Kaiser, 2014). Für unsere bereits formulierte Forschungsfrage benötigen wir spezifisches Wissen, das uns erlaubt, die Forschungsfrage zu beantworten. Daher werden die Experteninterviews als Leitfaden-gestütztes Experteninterview geführt.

3.1.2 Ethische Aspekte

Des Weiteren geht Kaiser (2014) auf die ethischen Aspekte ein, die es bei der Expertenbefragung zu beachten gilt. Als erstes dürfen keine personenbezogenen Daten in der Forschung erhoben und verarbeitet werden, sofern diese nicht für den Forschungszweck notwendig sind.

Ergänzend erfolgt eine Information an den Interviewpartner über den Zweck und Inhalt der Forschung, zu der er seine explizite Zustimmung zu seiner Teilnahme gibt und diese auch später noch widerrufen kann.

Außerdem muss das gesammelte Material wie z.B. Audioaufzeichnung und Transkripte vertraulich behandelt werden und darf nur dem Forschungsteam zur Verfügung gestellt werden.

Zuletzt gilt es die Grundsätze der Integrität und Objektivität als ethische Leitlinien zu befolgen.

3.2 Transkriptionsarten

Höld (2009) beschreibt vier Arten von Transkriptionssystemen:

- Wörtlich: Vollständige Textfassung aus dem verbalen Datenmaterial
- Kommentiert: Festhalten sprachlicher Auffälligkeiten wie z.B. langer Pausen oder Betonungen, Kommentare zu Gesprächsverlauf
- Zusammenfassend: Verschriften von Zusammenfassungen vom Tonband
- Selektiv: Verschriften bestimmter Teile des Datenmaterials mittels vorher festgelegten Kategorienschema

Ein zusammenfassendes oder selektives Transkriptionssystem verbindet die Transkription mit den ersten inhaltsanalytischen Auswertungsschritten. Da wir aber eine saubere Trennung der Transkription und Auswertung der Daten mittels Thematischer Analyse nach Braun und Clarke (2006) und keinerlei möglichen kommunikativen Informationsverlust bei der Transkription bevorzugen, kommen diese beiden Systeme nicht in Frage. Außerdem wollen wir die inhaltlichen Aspekte und nicht die sprachliche Form analysieren, was für die Verwendung der wörtlichen Transkription spricht. Hierbei bieten sich drei unterschiedliche Vorgehensweisen an:

- Internationale Phonetische Alphabet (IPA): Darstellung der Aussprache von Wörtern bzw. Silben mittels speziellen Zeichen (Sprachwissenschaft)
- Literarische Umschrift: Wiedergabe des Dialekts bzw. Umgangssprache mit gebräuchlichem Alphabet

• Normales Schriftdeutsch: Bereinigung des Dialekts und Ausbesserung von Grammatikfehlern

Daher sollte man sich im Voraus zwei Fragen stellen. Zum einen, was unbedingt transkribiert werden muss und zum anderen, ob man die so festgehaltenen Eigenschaften überhaupt später interpretieren möchte.

Bei unserer qualitativen Forschung ist der Inhalt der Interviews wichtig, weswegen die Vorgehensweise nach IPA für unser Vorhaben keinen Sinn macht. Da wir möglichst nah das geführte Interview mit dem Transkript widerspiegeln wollen, verwenden wir die Vorgehensweise der Literarischen Umschrift, da wir Grammatikfehler bzw. Satzbaufehler nicht ausbessern, jedoch eine leserliche Form ohne Dialekt oder Wortverschleifungen bevorzugen.

4 Danksagung

Wir möchten uns natürlich bei allen teilnehmenden Interviewpartnern, die uns ihr Expertenwissen zur Verfügung gestellt und uns somit die Daten für die qualitative Analyse ermöglicht haben, herzlich bedanken.

Anhang A Interview-Guideline

Thema	Id	Frage
		Vorstellung: - Namen und Tätigkeit (Max Forscher, Bernd Masterand) - Affiliation: FAU + OSR Prof. Dirk Riehle Danksagung: - Für Ihre Zeit - Für Zustimmung zur Audioaufzeichnung Ziel erklären: - Verstehen was gutes/gesundes Inner Source auszeichnet - Indikatoren für Qualität → auch gebräuchlich: Erfolgsind. → Ableitungen von Metriken, die helfen IS zu managen
A Hintergrund des Interviewpartners	1	In welcher Rolle sind Sie tätig?
A Hintergrund des Interviewpartners	2	In welchem Zusammenhang steht Ihre Arbeit mit Inner Source?
A Hintergrund des Interviewpartners	3	(Definition + Beispiel Programm) Können Sie die Inner Source Programme/ Vorhaben in denen Sie tätig waren kurz beschreiben?
		Je nach Erfahrungsschatz: Definition Inner Source – Code lesend und für Beiträge öffnen – Offen zugängliche archivierte Kommunikation, Kommunikation ermöglichen
B1 Erfolgsindikatoren Projekte	1	(Definition + Beispiel Projekt) Was zeichnet ein gutes Inner Source Projekt aus?
B1 Erfolgsindikatoren Projekte	2	Mature: Denken Sie an ein erfolgreiches Projekt. Was zeichnete dieses Programm aus? Adoption: Stellen sie sich den Idealfall/Erfolgsfall vor. Was macht das Projekt gut?
B1 Erfolgsindikatoren Projekte	3	Was zeichnet ein schlechtes Inner Source Projekt aus?
B1 Erfolgsindikatoren Projekte	4	Mature: Denken Sie an ein gescheitertes Inner Source Projekt. Was lief schlecht? Adoption: Stellen sie sich den Worst-Case vor. Was läuft schlecht?

B1 Erfolgsindikatoren Projekte	5	"Jedes Inner Source Projekt ist gleich gut/schlecht". Was denken Sie über diese These?
B2 Erfolgsindikatoren Programme	1	Was zeichnet ein gutes Inner Source Programm/guten Inner Source aus?
B2 Erfolgsindikatoren Programme	2	Was zeichnet ein schlechtes Inner Source Programm/schlechtes Inner Source aus?
B2 Erfolgsindikatoren Programme	3	Mature: Denken Sie zurück an die Programme mit denen Sie gearbeitet haben. Waren das gute/schlechte Programme? Warum?
B2 Erfolgsindikatoren Programme	4	Adoption: Denken Sie an den Idealfall. Wie sieht Ihr Inner Source Programm aus?
B2 Erfolgsindikatoren Programme	5	Adoption: Denken Sie an den Worst-Case. Wie sieht Ihr Inner Source Programm aus?
B2 Erfolgsindikatoren Programme	6	"Jedes Inner Source Programm ist gleich gut/schlecht". Was denken Sie über diese These?
C1 Existierende	1	Welche Inner Source Metriken erheben Sie
Metriken		momentan? Warum?
		(Ggf. getrennt Programm, Projekt)
		(Ggf. zeitnah geplant)
C1 Existierende	2	Wie nutzen Sie diese Metriken? Was für eine
Metriken		Stellung haben diese für Ihre Arbeit?
C1 Existierende	3	Wie nutzen Kollegen diese Metriken?
Metriken		Bedeutung für tägl. Arbeit?
C1 Existierende	4	Was lässt sich nicht messen, ist aber trotzdem
Metriken		wichtig?
C1 Existierende	5	Was messen Sie bewusst nicht? Welche Metriken
Metriken		nutzen Sie bewusst nicht?
C2 Erwünschte	1	Wenn Sie träumen dürften: Welche Metrik hätten
Metriken		Sie gern? Warum?
D Bonus	1	Wie haben Sie die Metriken gewonnen?
		 Nochmals für Zeit und Beiträge bedanken Wir transkribieren und werten aus Falls Ihnen noch was einfällt oder Fragen → Gern melden

Tabelle 4.1: Interview-Guideline

Anhang B Vorabinformation Interviewpartner

Nach einer ersten Anfrage, ob die potentiellen Interviewpartner prinzipiell Interesse/Zeit für das Interview zum Thema Inner Source Qualität haben, erhielten die Interviewpartner vorab eine E-Mail mit Informationen zum bevorstehenden Interview.

Sehr geehrte Frau/Herr [XXX],

mein Name ist [XXX]. Ich bin wissenschaftlicher Mitarbeiter in der Open Source Research Group von Prof. [XXX], M.B.A. und forsche an Inner Source (der Verwendung von Open-Source-Software-Entwicklungsmethoden innerhalb von Unternehmen). Gemeinsam mit meinem Kollegen [XXX] und unserem Master-Studenten [XXX] (im cc) arbeiten wir momentan an messbaren Qualitätsattributen von Inner Source. Vielen Dank, dass Sie uns zu diesem Thema als Interviewpartner bereitstehen möchten!

Im Interview werden wir darüber sprechen, was aus Ihrer Sicht gute und schlechte Inner Source Projekte und Programme ausmacht, und wie man die Qualität von Inner Source vermessen und bewerten kann. Dabei werden offene Fragen gestellt, d.h. es gibt hier kein falsch und richtig. Persönliche Meinungen und Erfahrungengern auch provokative Ideen - sind ebenfalls sehr willkommen. Je nach Gesprächsverlauf wird ein Interview 30 bis 60 Minuten Zeit in Anspruch nehmen. Für die wissenschaftliche Auswertung möchten wir gern eine Audioaufzeichnung des Interviews anfertigen und bitten Sie dafür um Ihre Zustimmung. Selbstverständlich werden die Audioaufzeichnung und Gesprächsnotizen streng vertraulich behandelt und nur von [XXX], [XXX], und Mitgliedern der Open Source Research Group eingesehen und auf einem dedizierten, besonders gesicherten Server gespeichert. Bei Interesse machen wir Ihnen die Ergebnisse zu gegebenem Zeitpunkt gern zugänglich. Wir bedanken uns nochmals für Ihre Zeit und ihr Interesse. Gern stehen wir Ihnen für Rückfragen vor und nach dem Interview zur Verfügung.

Mit freundlichen Grüßen

[XXX], [XXX], [XXX]

Anhang C Transkriptionsregeln

Nachfolgend sind die Transkriptionsregeln von Dresing und Pehl (2013) aufgeführt.

- 1. Es wird wörtlich transkribiert, also nicht lautsprachlich oder zusammenfassend. Vorhandene Dialekte werden möglichst wortgenau ins Hochdeutsche übersetzt. Wenn keine eindeutige Übersetzung möglich ist, wird der Dialekt beibehalten, zum Beispiel: Ich gehe heuer auf das Oktoberfest.
- 2. Wortverschleifungen werden nicht transkribiert, sondern an das Schriftdeutsch angenähert. Beispielsweise "Er hatte noch so'n Buch genannt" wird zu "Er hatte noch so ein Buch genannt" und "hamma" wird zu "haben wir". Die Satzform wird beibehalten, auch wenn sie syntaktische Fehler beinhaltet, beispielsweise: "bin ich nach Kaufhaus gegangen".
- 3. Wort- und Satzabbrüche sowie Stottern werden geglättet bzw. Ausgelassen, Wortdoppelungen nur erfasst, wenn sie als Stilmittel zur Betonung genutzt werden: "Das ist mir sehr, sehr wichtig.". "Ganze" Halbsätze, denen nur die Vollendung fehlt, werden jedoch erfasst und mit dem Abbruchzeichen / gekennzeichnet.
- 4. Interpunktion wird zu Gunsten der Lesbarkeit geglättet, das heißt bei kurzem Senken der Stimme oder uneindeutiger Betonung wird eher ein Punkt als ein Komma gesetzt. Dabei sollen Sinneinheiten beibehalten werden.
- 5. Pausen werden durch drei Auslassungspunkte in Klammern (...) markiert.
- 6. Verständnissignale des gerade nicht Sprechenden wie "mhm, aha, ja, genau, ähm" etc. werden nicht transkribiert. AUSNAHME: Eine Antwort besteht NUR aus "mhm" ohne jegliche weitere Ausführung. Dies wird als "mhm (bejahend)", oder "mhm (verneinend)" erfasst, je nach Interpretation.
- 7. Besonders betonte Wörter oder Äußerungen werden durch GROSSSCHREI-BUNG gekennzeichnet.
- 8. Jeder Sprecherbeitrag erhält eigene Absätze. Zwischen den Sprechern gibt es eine freie, leere Zeile. Auch kurze Einwürfe werden in einem separaten Absatz transkribiert. Mindestens am Ende eines Absatzes werden Zeitmarken eingefügt.
- 9. Emotionale nonverbale Äußerungen der befragten Person und des Interviewers, die die Aussage unterstützen oder verdeutlichen (etwa wie lachen oder seufzen), werden beim Einsatz in Klammern notiert.
- 10. Unverständliche Wörter werden mit (unv.) gekennzeichnet. Längere unverständliche Passagen sollen möglichst mit der Ursache versehen werden

(unv., Handystörgeräusch) oder (unv., Mikrofon rauscht). Vermutet man einen Wortlaut, ist sich aber nicht sicher, wird das Wort bzw. der Satzteil mit einem Fragezeichen in Klammern gesetzt. Zum Beispiel: (Xylomethanolin?). Generell werden alle unverständlichen Stellen mit einer Zeitmarke versehen, wenn innerhalb von einer Minute keine Zeitmarke gesetzt ist.

- 11. Die interviewende Person wird durch ein "I:", die befragte Person durch ein "B:" gekennzeichnet. Bei mehreren Interviewpartnern (z.B. Gruppendiskussion) wird dem Kürzel "B" eine entsprechende Kennnummer oder ein Name zugeordnet (z.B. "B1:", "Peter:").
- 12. Das Transkript wird als Rich Text Format (.rtf-Datei) gespeichert. Benennung der Datei entsprechend des Audiodateinamens (ohne Endung wav, mp3). Beispielsweise: Interview_04022011.rtf oder Interview_schmitt.rtf

Lediglich bei Regel 8 wurde das Einfügen der Zeitmarken für jeden Absatz weggelassen, da die Audioaufzeichnungen nach der vollständigen Transkription gelöscht werden sollen. Bei Regel 12 wurde die Transkription als Textdatei gespeichert statt als Rich Text Format, um die Transkripte neutral von Formatierung zu speichern. Die Benennung der Transkripte erfolgte im Format YYYY-MM-DD – IS Quality – Interview [Nr.] – [NACHNAME].

Als weitere eigene Regel werden im Interview genannte Namen durch ihre Rolle ersetzt wie z.B. "Das habe ich von Hr. XYZ bekommen." zu "Das habe ich von [meinem Chef] bekommen."

Anhang D Open Source Qualitätsmodelle

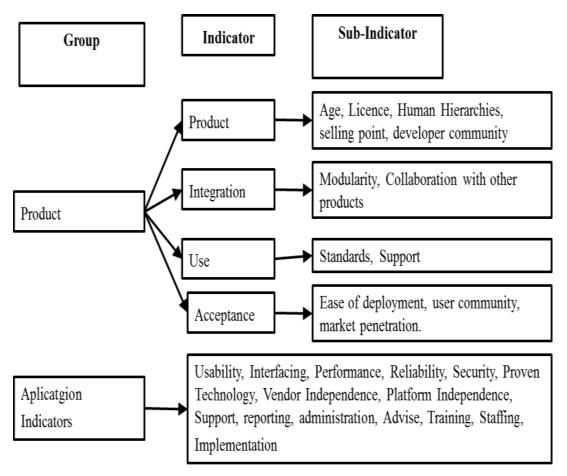


Abbildung 4.1: OSMM Qualitätsmodell (Miguel, Mauricio, & Rodríguez, 2014)

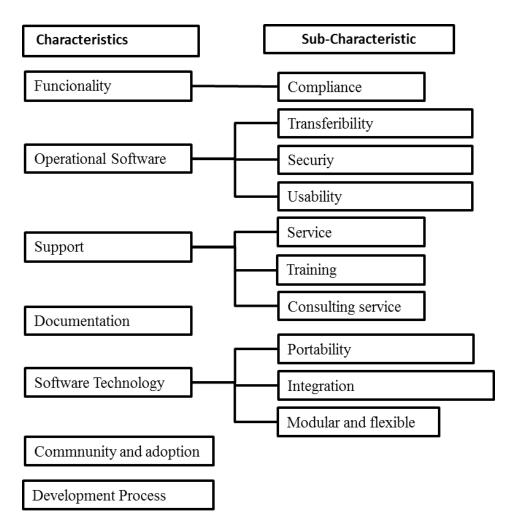


Abbildung 4.2: OpenBRR Qualitätsmodell (Miguel, Mauricio, & Rodríguez, 2014)

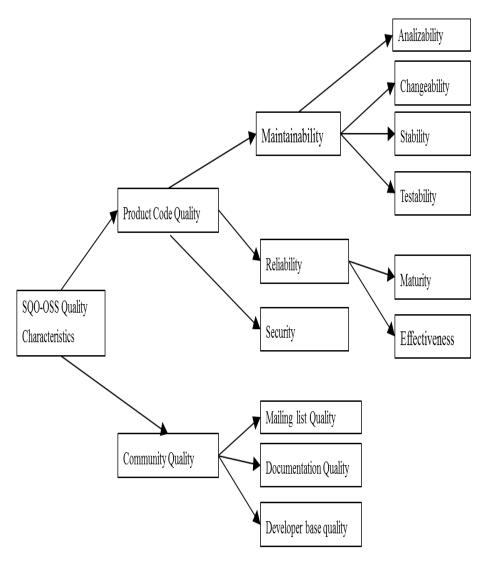


Abbildung 4.3: SQO-OSS Qualitätsmodell (Miguel, Mauricio, & Rodríguez, 2014)

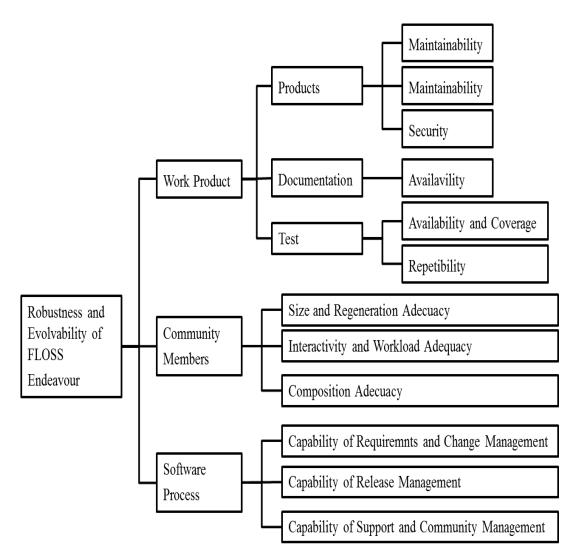


Abbildung 4.4: QualOSS Qualitätsmodell (Miguel, Mauricio, & Rodríguez, 2014)

Anhang E Analyse - Suchen nach Themen

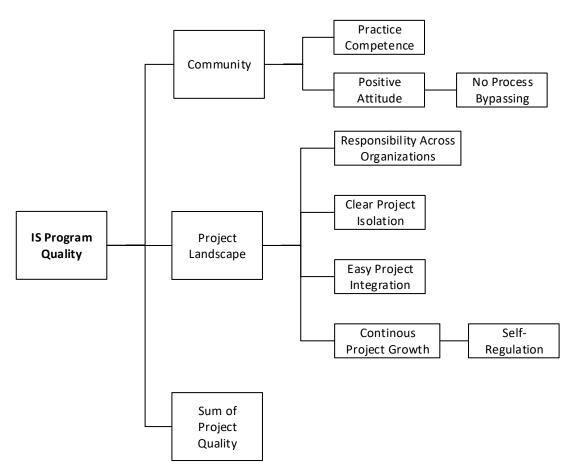


Abbildung 4.5: Programm Analyse: Suche nach Themen

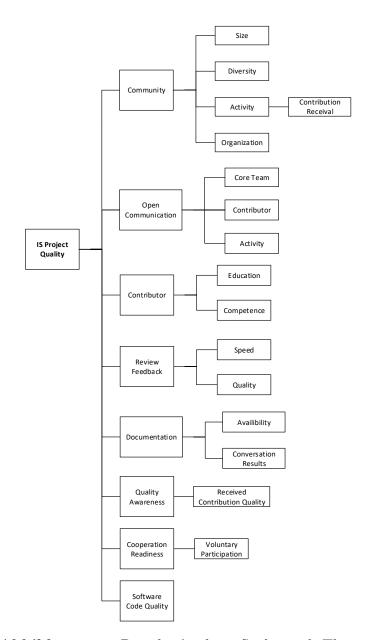


Abbildung 4.6: Projekt Analyse: Suche nach Themen

Anhang F Inhalte der CD

Auf der beigefügten CD befinden sich die Thesis als PDF-Datei und zwei Ordner IS_Quality_Interviews und Interview_Analyse. In IS_Quality_Interviews liegen die Transkripte der einzelnen durchgeführten Interviews als Text-Datei, falls diese für andere Forschungszwecke noch einmal verwendet werden und die Interview-Guideline in einer Druckversion als PDF-Datei für mögliche weitere Interviews.

Interview_Analyse beinhaltet die MAXQDA-Datei als mx5-Datei (erstellt mit MAXQDA 11) und das Codebook als xlsx-Datei in englisch. Das Codebook befindet sich auch im gemeinsamen Google Drive im Ordner Codeschema (Codebook v8). Es wird empfohlen dieses dort zu öffnen, da beim Export die Formatierung gelitten hat.

Literaturverzeichnis

- Aberdour, M. (2007). Achieving quality in open-source software. *IEEE software*, 24(1).
- Adewumi, A., Misra, S., & Omoregbe, N. (2015). Evaluating Open Source Software Quality Models against ISO 25010. In Computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing (cit/iucc/dasc/picom), 2015 ieee international conference on (S. 872–877). IEEE.
- Adewumi, A., Misra, S., Omoregbe, N., Crawford, B., & Soto, R. (2016). A systematic literature review of open source software quality assessment models. SpringerPlus, 5(1), 1936.
- Bogner, A., Littig, B., & Menz, W. (2005). Das Experteninterview: Theorien, Methoden, Anwendungen.
- Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 77–101.
- Capraro, M. & Riehle, D. (2016). Inner Source Definition, Benefits, and Challenges. ACM Computing Surveys (CSUR), 49(4), 67.
- Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, 44(2), 7.
- Dresing, T. & Pehl, T. (2013). Praxisbuch Interview, Transkription & Analyse. Anleitungen und Regelsysteme für qualitativ Forschende, 5. Auflage, Marburg. Letzter Zugriff 08.03. 2014.
- Duijnhouwer, F. & Widdows, C. (2003). Open source maturity model. *Cappemini Expert Letter*.
- Gläser, J. & Laudel, G. (2009). Experteninterviews und qualitative Inhaltsanalyse als Instrumente rekonstruierender Untersuchungen. 3., überarbeitete Auflage. VS-Verlag für Sozialwissenschaften, Wiesbaden.
- Helfferich, C. (2004). Die Qualität qualitativer Daten. Springer.
- Höld, R. (2009). Zur Transkription von Audiodaten. Qualitative Marktforschung, 655–668.

- Höst, M., Stol, K.-J., & Oručević-Alagić, A. (2014). Inner source project management. In *Software project management in a changing world* (S. 343–369). Springer.
- Kaiser, R. (2014). Qualitative Experteninterviews: Konzeptionelle Grundlagen und praktische Durchführung. Springer-Verlag.
- Kaufmann, A. & Riehle, D. (2015). Improving Traceability of Requirements Through Qualitative Data Analysis. In *Software engineering & management* (S. 165–170).
- Kruse, J. (2015). Qualitative Interviewforschung.
- Miguel, J. P., Mauricio, D., & Rodríguez, G. (2014). A review of software quality models for the evaluation of software products. arXiv preprint ar-Xiv:1412.2977.
- Samoladas, I., Gousios, G., Spinellis, D., & Stamelos, I. (2008). The sqo-oss quality model: measurement based open source software evaluation. *Open source development, communities and quality*, 237–248.
- Soto, M. & Ciolkowski, M. (2009). The qualoss open source assessment model measuring the performance of open source communities. In *Empirical* software engineering and measurement, 2009. esem 2009. 3rd international symposium on (S. 498–501). IEEE.
- Stol, K.-J., Avgeriou, P., Babar, M. A., Lucas, Y., & Fitzgerald, B. (2014). Key factors for adopting inner source. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 23(2), 18.
- Sung, W. J., Kim, J. H., & Rhew, S. Y. (2007). A quality model for open source software selection. In Advanced language processing and web information technology, 2007. alpit 2007. sixth international conference on (S. 515–519). IEEE.
- Van der Linden, F., Lundell, B., & Marttiin, P. (2009). Commodification of industrial software: A case for open source. *IEEE software*, 26(4).
- von Kardorff, E., Steinke, I., & Flick, U. (2008). Qualitative Forschung: Ein Handbuch. Rowohlt Taschenbuch-Verlag.