

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

Statistische Analyse des Zusammenhangs von Softwarekritikalität und Entwicklungskontext

Bachelorarbeit

im Studiengang Technische Informatik

von

Christian Münkel

Prüfer: Prof. Dr. Kurt Schneider

Zweitprüfer: Dr. Jil Klünder

Betreuer: Dr. Jil Klünder

Hannover, 05.12.2019

Zusammenfassung

Für die Organisation von Softwareprojekten existieren viele Methoden und Praktiken. Dabei verwenden Unternehmen heute meist auf ihre Bedürfnisse angepasste hybride Entwicklungsmethoden. Da bei kritischen Projekten, bei denen ein Ausfall zum Beispiel Menschenleben gefährden kann, das fehlerfreie Funktionieren der Software besonders wichtig ist, stellt sich die Frage, ob die Kritikalität einen Einfluss auf den jeweiligen Entwicklungskontext hat.

Um dies zu untersuchen, wird in dieser Arbeit ein Datensatz, der aus einer internationalen Umfrage zum Thema Softwareentwicklung entstanden ist, analysiert. Dabei werden mit Hilfe von statistischen Methoden die Auswirkungen von insgesamt neun Kritikalitätskategorien auf den Entwicklungskontext von Softwareprojekten untersucht. Die Ergebnisse zeigen, dass bei der Entwicklung von kritischen Projekten ein größerer Aufwand getrieben wird, indem häufiger Praktiken wie End-to-End Testing und Design Reviews genutzt und externe Standards implementiert werden. Außerdem sind Aktivitäten wie das Projekt- und Risikomanagement häufiger traditionell organisiert. Auch bei den verwendeten Methoden setzen mehr Unternehmen auf traditionelle Ansätze. Dabei unterscheidet sich vor allem der Anteil an Projekten, die das V-Modell verwenden. Während dieses nur von 5% der Unternehmen für alle Projekte verwendet wird, sind es bei den Unternehmen, die gesundheits- oder lebensgefährdende Software entwickeln, 21,3%.

Allerdings ist die Nutzung von agilen Verfahren auch bei der Entwicklung von kritischer Software weit verbreitet.

Abstract

There are many methods and practices for organizing software projects. Today companies usually use hybrid development Methods that are specifically tailored to their needs. A flawless operation is especially important for critical software, which includes for example software that threatens human life or health. This begs the question: How does criticality affect the context of software development?

This work will analyze the results of a large international survey, which included questions about many aspects of software development, in order to answer this question. Effects of a total of nine criticality categories on the context of development are examined using statistical methods.

The results show: Critical software projects are larger, meaning they take longer to complete, partially because methods like end-to-end testing and design reviews are used more frequently. Additionally some activities during development, like risk- and project management, are organized in a more traditional manner. Companies also tend to use more traditional methods, especially the V-Model. While this is only used in 5% of all projects, 21,3% of companies who are developing software which threatens human life or health are using it for all of their projects.

Despite of these differences, the findings also show that agile methods and practices are widely used, even for critical projects.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung	1
1.3. Struktur der Arbeit	2
2. Grundlagen	3
2.1. Kritikalität	3
2.2. Statistik	4
2.2.1. Signifikanz	4
2.2.2. Statistische Unabhängigkeit	4
2.2.3. Korrelationskoeffizienten	5
2.2.4. Der Chi-Quadrat Test	5
2.2.5. Fishers Exakter Test	7
3. Verwandte Arbeiten	9
3.1. Arbeiten zur Kritikalität	9
3.2. Arbeiten zur HELENA-Studie	10
4. Durchführung	13
4.1. Forschungsfragen	13
4.1.1. Beantwortung der Fragen	14
4.2. Aufbau der HELENA-Studie	14
4.2.1. Frage zur Kritikalität	15
4.3. R Umgebung	16
4.4. Vorverarbeitung der Daten	16
4.4.1. Entfernen unvollständiger Datenpunkte	17
4.4.2. Freitextfelder	17
4.4.3. Entfernen nicht relevanter Fragen	17
4.5. Analyse	19
5. Auswertung	23
5.1. Lebens- oder Gesundheitsgefährdend	24
5.2. Kompletter Systemverlust	27
5.3. Systemdegradierung	29
5.4. Finanzielle Kosten	29
5.5. Zivilrechtliche Folgen	31
5.6. Strafrechtliche Folgen	31
5.7. Umweltgefährdend	32
5.8. Reputationsschädigend und Geschäftsschädigend	35

Inhaltsverzeichnis

6. Diskussion	37
6.1. Beantwortung der Forschungsfragen	37
6.2. Empfehlungen	38
6.3. Einschränkung der Gültigkeit	39
7. Fazit und Ausblick	41
7.1. Fazit	41
7.2. Ausblick	41
A. Korrelationskoeffizienten	43
B. Ergebnisse des Chi-Quadrat Tests	45
C. Inhalt der beiliegenden CD	49
Literaturverzeichnis	51

1. Einleitung

1.1. Motivation

Von der Kasse am Supermarkt bis zur Steuerung von autonomen Fahrzeugen - Software findet heute fast überall Verwendung. Gerade in besonders kritischen Bereichen, wie zum Beispiel der Luft- und Raumfahrt, ist es dabei wichtig, dass sie möglichst fehlerfrei funktioniert. Dies wirft die Frage auf, wie Software mit hoher Kritikalität entwickelt wird. Welche Methoden und Praktiken werden verwendet?

In der modernen Softwareentwicklung verwenden Unternehmen häufig genau auf ihre Bedürfnisse angepasste hybride Entwicklungsprozesse, die durch Erfahrungen aus vorangegangenen Projekten entstanden sind [15, 18]. Allerdings lassen sich agile Methoden nicht immer mit den Zielen von kritischen Projekten, wie beispielsweise einer Zertifizierung, vereinbaren [13]. Werden deshalb in solchen Fällen mehr traditionelle Methoden verwendet als üblich? Wie werden diese mit agilen Prozessen verbunden? Um dies zu beantworten wird diese Arbeit eine globale Umfrage mit 1467 Datenpunkten [14] auswerten und dabei den Aufbau von Projekten in neun Kritikalitätsklassen untersuchen.

1.2. Zielsetzung

Auf Basis der Hybrid dEveLopmENT Approaches in software systems development (HELENA) Umfrage [14] soll in dieser Arbeit der Aufbau von kritischen Softwareprojekten im Hinblick auf verwendete Methoden und Praktiken, sowie Agilität des Entwicklungsprozesses, untersucht werden. Die HELENA-Studie ist eine globale Umfrage, an der 1467 Mitarbeiter von Softwareprojekten teilgenommen und Fragen zu den verwendeten Methoden und Praktiken, Projektgröße, Kritikalität und anderen Eigenschaften des Entwicklungskontextes, beantwortet haben. Dieser umfangreiche Datensatz eignet sich gut, um die Struktur von Projekten mit unterschiedlicher Kritikalität zu beschreiben. Diese Arbeit soll dabei mit Hilfe der Statistik insbesondere untersuchen, ob sich die Entwicklung von kritischen und nicht kritischen Projekten unterscheidet. Außerdem soll die Hypothese untersucht werden, dass der Grad an Agilität bei hoher Kritikalität geringer ist. Um dies zu erreichen, wird, für jede in der HELENA Studie verwendete Kritikalitätsklasse, eine Beschreibung über den Aufbau der darin enthaltenen Projekte gegeben. Weiterhin soll jeweils der Anteil an verwendeten agilen und traditionellen Methoden im Entwicklungsprozess verglichen werden. Abschließend wird auf dieser Grundlage eine Empfehlung für den Aufbau von kritischen Projekten gegeben.

1.3. Struktur der Arbeit

Um die oben genannten Ziele zu erreichen, ist diese Arbeit in sechs Kapitel unterteilt: In den Grundlagen wird zunächst ein einheitliches Verständnis der verwendeten Begriffe und Werkzeuge geschaffen. Zu diesem Zweck werden relevante Standards und Definitionen von Kritikalität und anschließend die verwendeten statistischen Verfahren beschrieben. Obwohl hier einige wichtige Definitionen in der Statistik genannt werden, wird für diese Arbeit ein grundlegendes Verständnis der Statistik vorausgesetzt.

Im zweiten Kapitel werden Arbeiten vorgestellt, die sich mit ähnlichen Themen befassen. Dabei liegt der Fokus auf Veröffentlichungen zu dem Thema Kritikalität in der Softwareentwicklung sowie auf Arbeiten, die den HELENA Datensatz auswerten. Dies dient der besseren Einordnung dieser Arbeit in die vorhandene Literatur.

Um den Aufbau der Analyse nachvollziehbar zu gestalten und eine Begründung für die verwendeten Methoden zu geben, widmet sich das dritte Kapitel der Struktur der nachfolgenden Auswertung der Daten. Weiterhin wird hier die Vorverarbeitung und Sortierung der Rohdaten erläutert.

Das vierte Kapitel stellt die Ergebnisse der vorher beschriebenen statistischen Analyse vor. Zunächst wird zu diesem Zweck eine Übersicht aller in den gesammelten Daten vorhandener Variablen und deren Relevanz für kritische Projekte im allgemeinen gegeben. Mit dem Ziel eine detaillierte Beschreibung für alle in den Umfrage erfassten Kategorien von kritischen Projekten zu erstellen, wird anschließend auf jede dieser Kategorien einzeln eingegangen. Dabei werden jeweils die am häufigsten verwendeten Methoden und Praktiken sowie andere zum Entwicklungskontext zugehörige Variablen, wie zum Beispiel Größe von Projekt und Unternehmen, vorgestellt.

Mit Implikationen und Gültigkeit der vorher beschriebenen Ergebnisse befasst sich das fünfte Kapitel. Hier werden Probleme, die zum Beispiel durch eine ungenügende Menge an Daten entstehen können, diskutiert. Außerdem wird auf Grundlage der Analyse eine Empfehlung für den Aufbau kritischer Projekte gegeben.

Das letzte Kapitel enthält eine Zusammenfassung der Arbeit und gibt einen Ausblick auf Forschungsfragen für zukünftige Arbeiten.

2. Grundlagen

Dieses Kapitel stellt wichtige Definitionen und Verfahren, die in dieser Arbeit Verwendung finden, vor. Dabei wird ein grundlegendes Verständnis von Statistik vorausgesetzt.

2.1. Kritikalität

Kritikalität im Rahmen von Software Projekten beschreibt die möglichen Folgen im Fall von Systemversagen. Kritische Systeme können nach M. Hinchey und L. Coyle [5] in vier Kategorien eingeteilt werden:

Safety-Critical Ein System, dass bei Versagen zu Verlust menschlichen Lebens, Verletzungen oder Umweltschäden führt. Ein Beispiel hierfür ist der Autopilot von Flugzeugen.

Mission-Critical Fehler der Software führen entweder zum kompletten Verlust des Systems oder verhindern das Erreichen des Projektziels. Software zur Steuerung einer autonomen Mondsonde fällt in diese Kategorie.

Business-Critical Ausfall kann finanzielle Kosten oder Schaden an der Reputation verursachen. Dies ist zum Beispiel bei der Website eines Online Versandhandels der Fall.

Security-Critical Fehler können zum Verlust sensibler Daten führen. Unter anderem gehören Verschlüsselungssoftware und Systeme zum Aufbewahren von Zugangsdaten dieser Kategorie an.

Oft erfüllen Software Projekte mehrere dieser Bedingungen, so kann beispielsweise der Ausfall von Software in autonomen Fahrzeugen sowohl zum Verlust von Menschenleben als auch zu finanziellem Schaden führen. Da Kritikalität für viele Projekte eine Rolle spielt, gibt es mehrere internationale Standards die sich mit diesem Thema befassen. Im Folgenden werden einige dieser Standards kurz vorgestellt.

IEC 61508 [1] Dieser Standard der International Electrotechnical Commission ist für Projekte relevant, die einer Gefährdung für menschliche Gesundheit und die Umwelt darstellen. Er enthält unter anderem Vorgaben für die Projektarchitektur, Qualitätsprüfungen und Dokumentation des Projekts.

ISO 26262 [2] Eine Anpassung des IEC 61508 an die Bedürfnisse der Automobilindustrie, er befasst sich also ebenfalls mit Safety-Critical Projekten.

ISO 27001 [3] Diese Norm gibt Vorgaben für Projekte, die in den Bereich Security-

2. Grundlagen

Critical fallen. Sie beschreibt Vorgehensweisen um Sicherheitsrisiken innerhalb eines Unternehmens zu erkennen und zu minimieren.

ISO 9001 [4] Ein sehr allgemeiner Standard, der Anforderungen an die Qualitätssicherung eines Unternehmens stellt. Er ist sehr weit verbreitet und es ist möglich ein Zertifikat für die Einhaltung dieses Standards zu erhalten.

2.2. Statistik

Um die in der HELENA Studie erhobenen Daten zu analysieren, werden in dieser Arbeit verschiedene statistische Verfahren verwendet, die in diesem Kapitel erläutert werden.

2.2.1. Signifikanz

Signifikanz ist ein Maß für die Aussagekraft von Ergebnissen. Ein Resultat gilt dann als statistisch signifikant, wenn es, unter Annahme der Nullhypothese, mit einer Wahrscheinlichkeit von unter 0,05 zufällig auftritt [7]. Zum Beispiel könnte eine Münze mit der Vermutung, dass Kopf und Zahl beim Werfen mit gleicher relativer Häufigkeit auftreten, untersucht werden. Wenn bei zehnfachem Werfen jedes mal Kopf oben erscheint, kann die Nullhypothese widerlegt werden, da die Wahrscheinlichkeit für dieses Ergebnis nur bei $\frac{1}{2^{10}} \approx 0,001$ liegt, wenn Kopf und Zahl, wie anfangs vermutet, gleich wahrscheinlich sind.

2.2.2. Statistische Unabhängigkeit

Die Statistische Unabhängigkeit beschreibt das Verhältnis zwischen zwei oder mehreren Zufallsvariablen und ist wie folgt definiert [6, S. 10]:

Zwei Zufallsvariablen A und B sind statistisch unabhängig, wenn gilt

$$P(A \cap B) = P(A)P(B) .$$

Daraus folgt, dass nach dem Messen einer Zufallsvariablen A nur dann eine Vorhersage über den Wert einer zweiten Variablen B getroffen werden kann, wenn A und B nicht statistisch unabhängig sind. Das Überprüfen auf statistische Unabhängigkeit ist also nützlich, um Zusammenhänge zwischen Variablen zu erkennen. Dabei ist zu beachten, dass auf diese Art kein kausaler Zusammenhang zwischen den getesteten Variablen nachgewiesen werden kann.

2.2.3. Korrelationskoeffizienten

Die Korrelation ist ein Spezialfall der statistischen Abhängigkeit [7], korrelierte Variablen sind also immer auch statistisch abhängig, während statistisch abhängige Variablen nicht korreliert sein müssen. Genauer gesagt beschreibt Korrelation den linearen Zusammenhang zwischen Zufallsvariablen. Dieser kann mit dem Korrelationskoeffizienten ρ quantifiziert werden, der für zwei Variablen x und y wie folgt berechnet wird:

$$\rho = \frac{\sum_{i=1}^n (x_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^n (x_i - m_x)^2 \sum_{i=1}^n (y_i - m_y)^2}}$$

Der Wertebereich von ρ ist $[-1, 1]$. Je näher der Korrelationskoeffizient an den Grenzen des Wertebereichs liegt, desto stärker der lineare Zusammenhang zwischen x und y , während ein Wert von 0 bedeutet, dass die beiden Variablen nicht linear voneinander abhängig sind. Üblicherweise wird für den Fall von $|\rho| < 0,3$ von vernachlässigbar geringer und bei $|\rho| > 0,9$ von sehr hoher Korrelation gesprochen [10]. Das Vorzeichen von ρ gibt an, ob es sich um einen positiven oder negativen linearen Zusammenhang handelt.

2.2.4. Der Chi-Quadrat Test

Um Variablen auf statistische Abhängigkeit zu überprüfen, können verschiedene Tests verwendet werden. Häufig liegen dabei Daten vor, die nur mit Hilfe einer Nominalskala dargestellt werden können. Zu diesen Variablen gehören zum Beispiel die in einem Softwareprojekt verwendeten Methoden und Praktiken, da ihnen kein Zahlenwert zugewiesen werden kann und sie auch keiner Rangfolge unterliegen (zum Beispiel ist das V-Model nicht größer oder höher als Extreme Programming). Deshalb werden diese Merkmale auch kategoriale Merkmale genannt. Eine weit verbreitete Methode zum Testen von kategorialen Merkmalen ist Pearsons Chi-Quadrat Test für Unabhängigkeit [8], nachfolgend als Chi-Quadrat Test bezeichnet. Dabei wird von der Nullhypothese ausgegangen, dass die zu testenden Variablen statistisch unabhängig sind und anschließend die Wahrscheinlichkeit berechnet, dass die beobachteten Werte unter dieser Annahme auftreten. Die Funktionsweise wird nachfolgend an einem Beispiel verdeutlicht:

Beobachtet	<i>nicht ausgefallen</i>	<i>ausgefallen</i>	Σ
<i>getestet</i>	40	11	51
<i>nicht getestet</i>	10	25	35
Σ	50	36	86

Die oben stehende Tabelle zeigt, wie Entwickler ihren Code getestet haben und ob die von ihnen entwickelte Software ausgefallen ist. Um mit dem Chi-Quadrat Test zu

2. Grundlagen

überprüfen, ob ein Zusammenhang zwischen den beiden Variablen besteht, wird zunächst von der Nullhypothese ausgegangen, dass das Testen der Software keinen Einfluss auf die Ausfallrate hat. Im ersten Schritt wird dann für jede Kombination von Kategorien die erwartete Häufigkeit

$$E_i = \frac{\text{SummeZeile} * \text{SummeSpalte}}{\text{SummeGesamt}}$$

berechnet. Für die erste Zelle der Tabelle also

$$\frac{51 * 50}{86} \approx 29,65$$

, die restlichen Ergebnisse können in der nachfolgenden Tabelle abgelesen werden. Diese Werte bilden auch eine wichtige Einschränkung für den Chi-Quadrat Test, da dieser bei ungenügender Anzahl an Datenpunkten nicht verwendet werden kann. Üblicherweise wird diese Grenze so definiert, dass keines der E_i kleiner als 5 sein darf. Sollte dies der Fall sein, muss ein anderer Test, wie zum Beispiel Fishers Exakter Test, verwendet werden.

Erwartet	<i>nicht ausgefallen</i>	<i>ausgefallen</i>
<i>getestet</i>	29,65	21,34
<i>nicht getestet</i>	20,34	14,65

Anschließend wird der Chi-Quadrat Wert χ^2 für die Daten folgendermaßen berechnet:

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$

Dabei ist n die Anzahl an Zellen, O_i der beobachtete Wert in der Zelle i und E_i der jeweilige erwartete Wert. Für das Beispiel ist

$$\chi^2 = \frac{(40 - 29,65)^2}{29,65} + \dots + \frac{(25 - 14,65)^2}{14,65} = 21,19.$$

Um die Wahrscheinlichkeit für das Auftreten der gemessenen Verteilung zu bestimmen, muss nun noch der Freiheitsgrad

$$df = (\text{AnzahlZeilen} - 1)(\text{AnzahlSpalten} - 1)$$

, in diesem Beispiel also

$$df = (2 - 1)(2 - 1) = 1$$

, berechnet werden. Im letzten Schritt wird der Wert der Chi-Quadrat Verteilung $F(df, \chi^2)$ entweder berechnet oder aus einer Tabelle abgelesen. Die Wahrscheinlichkeit für das Zutreffen der Nullhypothese in dem Beispiel liegt somit bei $1,1 * 10^{-5}$, was geringer als 0,05, also der Schwelle für Signifikanz, ist. Dieses Ergebnis bedeutet, dass ein Zusammenhang zwischen dem Testen der Software und deren Ausfallrate existiert. Da der Chi-Quadrat Test nur auf statistische Unabhängigkeit prüft, kann so noch kein kausaler Zusammenhang zwischen den beiden Variablen nachgewiesen werden.

2.2.5. Fishers Exakter Test

Wie oben schon erwähnt, wird Fishers Exakter Test ebenfalls zum Überprüfen von kategorialen Variablen auf statistische Unabhängigkeit verwendet [9]. Dabei wird von der selben Nullhypothese ausgegangen und die am Ende berechneten Wahrscheinlichkeiten können auf die gleiche Art interpretiert werden. Im Gegensatz zu dem Chi-Quadrat Test muss hier allerdings keine minimale Anzahl an Datenpunkten gegeben sein. Funktionsweise und Eigenschaften des exakten Fisher Tests werden im Folgenden an einem Beispiel verdeutlicht:

Beobachtet	<i>nicht ausgefallen</i>	<i>ausgefallen</i>	Σ
<i>getestet</i>	1(a)	8(b)	9(a+b)
<i>nicht getestet</i>	6(c)	5(d)	11(a+d)
Σ	7(a+c)	13(b+d)	20(n)

Da dieser Datensatz sehr klein ist und es deshalb Zellen gibt, für die ein Wert kleiner als 5 erwartet wird, kann der Chi-Quadrat Test hier nicht angewendet werden. Dies kann durch Berechnen des erwarteten Wertes für die oberen linke Zelle gezeigt werden

$$E_1 = \frac{9 \cdot 7}{20} = 3,15.$$

Da der exakte Fisher Test dieser Beschränkung nicht unterliegt, kann er hier angewendet werden. Bei diesem Test wird die Wahrscheinlichkeit für das Auftreten der beobachteten Werte mit Hilfe einer hypergeometrischen Verteilung berechnet:

$$p(a = 1) = \frac{\binom{a+b}{a} \binom{c+d}{c}}{\binom{n}{a+c}} = \frac{\binom{1+8}{1} \binom{6+5}{6}}{\binom{20}{1+6}} = 0,05$$

Im nächsten Schritt werden die Wahrscheinlichkeiten für alle anderen möglichen Verteilungen, die mindestens so extrem sind wie die Messdaten, mit gleichen Randhäufigkeiten berechnet und summiert. Hier führt dies zu einer Gesamtwahrscheinlichkeit von 0,07, die Nullhypothese wurde also nicht widerlegt.

Da die zur Berechnung verwendete hypergeometrische Verteilung Binomialkoeffizienten enthält, müssen für den exakten Fisher Test mehrere Fakultätsfunktionen berechnet werden. Bei sehr großen Zahlen ist dies sehr aufwändig, weswegen sich der exakte Fisher Test besonders gut für kleine Datensätze eignet.

2. Grundlagen

3. Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt, die sich mit für diese Arbeit relevanten Themenbereichen beschäftigen. Dabei liegt der Fokus auf Veröffentlichungen zu dem Thema Kritikalität in der Softwareentwicklung und auf Arbeiten, die auf der HELENA-Studie basieren.

3.1. Arbeiten zur Kritikalität

In einer Arbeit [5] aus dem Jahr 2010 haben M. Hinchey und L. Coyle vorgeschlagen, neue Methoden zur Weiterentwicklung bereits vorhandener Software zu erforschen. Es wurden vor allem vier Szenarien, die besonders kritisch sein könnten, hervorgehoben. Das erste Szenario ist das Portieren von Software, die nur einen Thread verwendet, auf neue Hardware mit mehreren parallel laufenden Prozessorkernen. Laut den Autoren gehören Kommunikation zwischen den Threads, Speicherzugriffe und externe Interrupts zu Problemen, die zu diesem Zweck Lösungen erfordern. Die Weiterentwicklung von kritischen Netzwerk Systemen, die bei Ausfall einen starken Einfluss auf das gesamte Netzwerk haben, ist ein weiteres Szenario. Hier wurde empfohlen Methoden zu entwickeln, mit denen Software verändert werden kann, ohne das System von dem Netzwerk zu trennen. Außerdem werden autonome Agenten zur Erforschung der Umgebung, wie zum Beispiel Weltraumsonden hervorgehoben. In diesem Bereich nennen die Autoren das selbstständige Weiterentwickeln von Software in unvorhergesehenen Situationen als möglichen Forschungsbereich. Letztlich wird für Software im Bereich der Automobilindustrie empfohlen, Möglichkeiten für das Installieren von Upgrades, während des Betriebs und ohne Gefährdung der Sicherheit, zu suchen. Außerdem wird darauf hingewiesen, dass Probleme bei der Kommunikation mit Software von anderen Herstellern entstehen könnten. Die Autoren stellen ein Modell vor, dass die zu entwickelnden Methoden in drei Stufen unterteilt: Planung des Upgrades, Installation und Bewertung des Ergebnisses.

Um die Limitierungen von agilen Entwicklungsansätzen zu untersuchen, haben D. Turk, R. France und B. Rumpe [13] Veröffentlichungen zum Thema agile Prozesse analysiert. Dabei sind sie zu dem Ergebnis gekommen, dass unter anderem kritische Software nicht vollständig agil entwickelt werden kann. Dies wird durch unzureichende Maßnahmen zur Qualitätskontrolle bei agilen Verfahren begründet. Um dennoch bei der Entwicklung nicht vollständig auf diese Methoden verzichten zu müssen, empfehlen die Autoren zusätzlich traditionelle Verfahren zum Testen der Software einzusetzen und diese in den agilen Entwicklungsprozess einzubinden.

3. Verwandte Arbeiten

Auf der Agile Conference im Jahr 2010 wurde von X. Ge et al. [11] ein Verfahren zur Anpassung von agilen Methoden an die Bedürfnisse von Softwareprojekten mit hoher Kritikalität vorgestellt. Da am Ende eines traditionellen Verfahrens oft eine Zertifizierung steht, bei der das gesamte System strengen Tests unterzogen wird, muss eine Möglichkeit gefunden werden, dieses Verfahren auf agile Prozesse anzupassen. Die Autoren schlagen vor, die einzelnen Module eines Projektes iterativ zu entwickeln und simultan bei jedem Schritt Tests an dem aktuellen Modul durchzuführen. Bei Fertigstellung eines Moduls wird die Interaktion mit den anderen bereits abgeschlossenen Teilen eines Projektes getestet. Auf diese Weise kann schrittweise ein intensiv überprüfbares Produkt mit einem möglichst agilen Verfahren entwickelt werden.

Für Projekte, die in die Kategorie Safety-Critical fallen, ist das Testen ein besonders wichtiger Teil des Entwicklungsprozesses. Wie R. Baker und I. Habli [12] gezeigt haben, können die üblicherweise in solchen Fällen verwendeten Testverfahren nicht alle vorhandenen Fehler erkennen. Deshalb untersuchten sie die Anwendung von Mutation-Testing auf kritische Projekte. Die Forscher kamen zu dem Ergebnis, dass das von ihnen vorgestellte Verfahren Fehler sehr verlässlich erkennen kann und so eine Ergänzung zu den bereits verwendeten Testmethoden sein kann. Jedoch standen für die Studie nicht genügend Daten zu Verfügung, um zu entscheiden, ob Mutation-Testing die anderen Verfahren komplett ersetzen kann.

3.2. Arbeiten zur HELENA-Studie

In der Arbeit von J. Klünder et al. [15] wurde untersucht, welche Methoden und Praktiken bei der modernen Softwareentwicklung zum Einsatz kommen. Dabei haben die Autoren besonders die Verwendung hybrider Entwicklungsmethoden erforscht und dabei beantwortet, welche Methoden verwendet werden, wie diese kombiniert wurden und welche Vorteile die unterschiedlichen Ansätze haben. Das Ergebnis zeigt, dass 76,8% der teilnehmenden Unternehmen hybride Methoden implementieren. Zu den am häufigsten verwendeten Methoden gehören Scrum (84,7%), Iterative Programming (80,9%) und Kanban (66,7%). Die Entwicklung von hybriden Methoden wird in den meisten Fällen (78,5%) durch Erweiterung von bestehenden Praktiken mit Hilfe von Erfahrungen aus früheren Projekten durchgeführt. Außerdem zeigen die Autoren, dass eine Kombination aus geplanten Verbesserungen mit Erfahrungen aus früheren Projekten ein guter Ansatz ist, um hybride Methoden für ein Unternehmen oder Projekt zu entwickeln.

M. Marinho et al. [17] haben die Nutzung von agilen Entwicklungsmethoden in globalen Softwareprojekten untersucht. Sie zeigen, dass 25% aller großen Projekte vorwiegend agile und nur 5% hauptsächlich traditionelle Methoden einsetzen. Jedoch ist mit 72% die am weitesten verbreitete Organisationsform eine Kombination von beiden Ansätzen, die auf Bedürfnisse und Erfahrungen des jeweiligen Unternehmens angepasst

ist. Am häufigsten wird dabei Scrum (86%) mit einem traditionellen Verfahren, in den meisten Fällen mit dem klassischen Wasserfall, verbunden.

Um ein besseres Verständnis für den Entwicklungsprozess von hybriden Prozessen zu erlangen, haben P. Tell et al. [18] untersucht, welche Methoden und Praktiken in Softwareprojekten zum Einsatz kommen. Die Autoren stellten fest, dass die Häufigkeit der Verwendung von hybriden Methoden nicht von Unternehmensgröße oder Industriesektor abhängt. Außerdem wurde gezeigt, dass 17 Kombinationen von zwei oder mehr Methoden einen Großteil der verwendeten Prozesse ausmachen. Dabei werden Scrum, Iterative Development und Kanban am häufigsten verwendet. Weiterhin wurde gezeigt, dass sich der Aufbau von verschiedenen hybriden Prozessen, obwohl er oft auf die individuellen Anforderung von Projekten angepasst ist, nur geringfügig voneinander unterscheidet.

3. Verwandte Arbeiten

4. Durchführung

Dieses Kapitel stellt die zentralen Forschungsfragen dieser Arbeit und die Methoden, die zu deren Beantwortung genutzt wurden, vor.

4.1. Forschungsfragen

Um die Ergebnisse dieser Arbeit übersichtlicher zu gestalten, werden im Folgenden drei Forschungsfragen bestimmt.

Wie wird der Grad der Agilität von Projekten durch deren Kritikalität beeinflusst?

Wie D. Turk, R. France und B. Rumpe [13] festgestellt haben, sind agile Prozesse nicht immer mit den Zielen von kritischen Projekten vereinbar. Da dies in besonderem Maße für Projekte mit hoher Kritikalität, also bei Software die im Fall eines Fehlers Menschenleben gefährdet, gilt, soll untersucht werden, ob der Grad der Agilität bei diesen Projekten tatsächlich geringer ist. Um diese Fragen zu beantworten, werden in dieser Arbeit Projekte von unterschiedlicher Kritikalität im Hinblick auf die jeweils verwendeten Methoden und Praktiken untersucht. Anschließend wird verglichen, wie oft agile Methoden, wie zum Beispiel Extreme Programming und traditionelle Methoden, wie zum Beispiel der klassische Wasserfall, verwendet werden. Außerdem wird der Grad der Agilität von Aktivitäten während des Entwicklungsprozesses, wie zum Beispiel Projekt- und Qualitätsmanagement, untersucht.

Welche Methoden und Praktiken wurden für kritische Projekte besonders oft verwendet?

Viele Unternehmen verwenden hybride Entwicklungsmethoden, die durch Erfahrungen aus vorherigen Projekten durch eine Kombination von agilen und traditionellen Verfahren entstanden sind [15]. Da kritische Projekte teilweise andere Anforderungen, wie zum Beispiel eine Zertifizierung, haben, wirft dies die Frage auf, ob durch abweichende Erfahrungen auch andere hybride Methoden entstehen. Zur Beantwortung wird in dieser Arbeit untersucht, welche Entwicklungsmethoden und Praktiken in kritischen Projekten häufiger als bei nicht kritischen Projekten verwendet werden.

4. Durchführung

Welche Unterschiede gibt es zwischen dem Entwicklungskontext von kritischen und nicht kritischen Projekten?

Zu dem Entwicklungskontext zählen unter anderem Unternehmens- und Projektgröße, Verwendung von externen Standards und Zertifizierung. Die Beantwortung dieser Frage soll einen Überblick über den Aufwand der nötig ist um kritische Projekte abzuschließen geben. Da das Einhalten externer Standards und die Zertifizierung sehr aufwändig sind, legt dies die Vermutung nahe, dass kritische Projekte von größeren Teams entwickelt werden. Durch die dadurch entstehenden finanziellen Kosten, ist zu erwarten, dass Unternehmen, die kritische Software entwickeln, besonders groß sind.

4.1.1. Beantwortung der Fragen

Um die Forschungsfragen zu beantworten, werden Daten, die Variablen zu Kritikalität und den anderen, bereits genannten, Faktoren des Entwicklungskontextes von Softwareprojekten enthalten, benötigt. Ein Datensatz, der diese Kriterien erfüllt, ist im Rahmen der HELENA-Studie entstanden.

4.2. Aufbau der HELENA-Studie

Die HELENA-Studie wurde in drei Phasen durchgeführt [14]:

Phase 0 Entwicklung des Fragebogens und Test an 15 Teilnehmern

Phase 1 Überarbeitung des Fragebogens mit Erfahrung aus Phase 0 und Test der überarbeiteten Version in Europa

Phase 2 Finale Verbesserungen des Fragebogens und Durchführung der Umfrage in 55 Ländern weltweit

Da der Fragebogen mit jeder Phase verbessert wurde und die Anzahl an Teilnehmern hier am größten war, werden in dieser Arbeit nur die Ergebnisse aus Phase 2 der HELENA-Studie betrachtet. Der in der Umfrage entstandene Datensatz enthält insgesamt 1467 Datenpunkte, die aus den Antworten auf 38 Fragen aus den folgenden 5 Kategorien zusammengesetzt sind:

Demographics Fragen zur Demographie, zum Beispiel Unternehmensgröße oder Kritikalität

Processes Used Fragen zu den genutzten Prozessen und der Art der Implementierung

Processes Used and Standards Fragen zur Agilität und externen Standards

Experience Fragen zu Erfahrungen mit Vor- und Nachteilen des gewählten Entwicklungsprozesses

Closing Fragen zum Umfeld der Studie, zum Beispiel über die Teilname an zukünftigen Studien

In dem Datensatz wird jeder Frage sowie jeder Antwortmöglichkeit zur einfacheren Identifizierung eine Bezeichnung, die aus den Anfangsbuchstaben der Kategorie, gefolgt von einer fortlaufenden Nummer für die Frage, gefolgt von einem Unterstrich und einer Nummer für die Antwortmöglichkeit besteht, zugeordnet. So hat zum Beispiel die erste Antwortmöglichkeit der sechsten Frage in der Kategorie Demographic die Bezeichnung D006_01. Die Fragen konnten mehrheitlich durch Ankreuzen von vorgegebenen Antworten und teilweise auch durch freie Texte beantwortet werden.

4.2.1. Frage zur Kritikalität

Die für diese Arbeit wichtigste Frage befasst sich mit der Kritikalität des aktuellen Projektes der Teilnehmer. Zum Beantworten standen neun ankreuzbare Antwortmöglichkeiten und ein Freitextfeld zur Verfügung. Abbildung 4.1 stellt die verfügbaren Antwortmöglichkeiten und die interne Bezeichnung dafür dar:

In dem Projekt oder Produkt, auf das Sie sich beziehen, kann ein Softwarefehler:	
<i>Bezeichnung</i>	<i>Antwortmöglichkeit</i>
D006_01	...menschliches Leben oder Gesundheit bedrohen
D006_02	...zu komplettem Systemverlust führen
D006_03	...zu System (Service) Degradierung führen
D006_04	...zu finanziellem Verlust führen
D006_05	...rechtliche Konsequenzen haben (Zivilgesetz)
D006_06	...die Umwelt schädigen
D006_07	...die Reputation des Unternehmens beeinträchtigen
D006_08	...das Geschäft des Unternehmens beeinträchtigen
D006_09	<i>Anderes (Freitextfeld)</i>
D006_10	...rechtliche Konsequenzen haben (Strafgesetz)

Abbildung 4.1.: Tabelle zu den Antwortmöglichkeiten auf die Kritikalitätsfrage und den zugehörigen internen Bezeichnung.

Von M. Hinchey und L. Coyle [5] wurden vier Kategorien von kritischen Projekten vorgestellt. Diese können wie folgt den hier vorgegebenen Antwortmöglichkeiten zugeordnet werden:

Safety-Critical

- ...menschliches Leben oder Gesundheit bedrohen
- ...die Umwelt schädigen

Mission-Critical

- ...zu komplettem Systemverlust führen

4. Durchführung

- ...zu System (Service) Degradierung führen

Business-Critical

- ...zu finanziellem Verlust führen
- ...rechtliche Konsequenzen haben (Zivilgesetz)
- ...die Reputation des Unternehmens beeinträchtigen
- ...das Geschäft des Unternehmens beeinträchtigen
- ...rechtliche Konsequenzen haben (Strafgesetz)

Security-Critical

- keine der Antwortmöglichkeiten passt in diese Kategorie

Von den insgesamt 1467 Teilnehmern der Studie, haben 935 den Fragebogen mindestens bis zu dieser Frage beantwortet.

4.3. R Umgebung

Alle Arbeiten an dem Datensatz wurden mit Hilfe von R, einer Interpretersprache für Statistik, durchgeführt. R wurde gewählt, weil es ein weit verbreitetes Tool zur Analyse von Daten ist und die Standardbibliothek eine große Anzahl an für die Statistik nützlichen Datenstrukturen und Algorithmen enthält. Auf der CD im Anhang befindet sich der in dieser Arbeit entstandene R Workspace, die folgende Tabelle enthält die für diese Arbeit wichtigsten Variablen und Funktionen:

data_raw Data Frame mit den Rohdaten der HELENA-Studie

data Data Frame, enthält in der Analyse verwendeten Datensatz

cor() Funktion, die Korrelationskoeffizienten von übergebenen Variablen berechnet

chisq.test() Führt Chi-Quadrat Test für zwei Variablen durch

fisher.test() Führt Fisher-Test für zwei Variablen durch

ChiFisher() Nicht Teil der R Funktionen, wurde für diese Arbeit entwickelt. Berechnet entweder den Chi-Quadrat Test oder, wenn die Bedingungen für diesen nicht erfüllt sind, den Fisher-Test für eine beliebige Anzahl an Variablen

4.4. Vorverarbeitung der Daten

Vor der eigentlichen Analyse wurden an dem Rohdatensatz der HELENA-Studie einige Änderungen vorgenommen. Dabei war das Ziel, die Auswertung der Daten einfacher zu gestalten, ohne dabei die Aussagekraft der Ergebnisse zu beeinträchtigen. In diesem Abschnitt wird erläutert, in welcher Weise der Datensatz verändert wurde und

warum dadurch das Ergebnis nicht beeinflusst wurde.

Zunächst wurde der gesamte Rohdatensatz in die R Umgebung importiert:

```
data_raw <- read.csv(file = "HELENA-Data.csv", sep = ",")
```

4.4.1. Entfernen unvollständiger Datenpunkte

Wie bereits erwähnt, haben 935 von 1467 Teilnehmern den Fragebogen bis zu der Kritikalitätsfrage beantwortet. Da die Einstufung der Kritikalität eines Projektes für diese Arbeit essenziell ist, wurden nur diese 935 Datenpunkte betrachtet. Das Ergebnis wird durch die Reduktion des Datensatzes nicht beeinträchtigt, da über die 532 übrigen Projekte keine Angabe hinsichtlich ihrer Kritikalität gemacht werden kann. Da das Feld D006 des Datensatzes die Anzahl an angekreuzten Antworten zur Kritikalitätsfrage enthält, kann es genutzt werden, um die unvollständigen Datenpunkte folgendermaßen zu entfernen:

```
data <- data_raw[!is.na.data.frame(data_raw$D006)]
```

4.4.2. Freitextfelder

Der Datensatz enthält insgesamt 21 Freitextfelder, in die von den Teilnehmer beliebige Texte eintragen werden konnten. Dabei handelt es sich zum größten Teil um Felder mit denen nicht vorhergesehene Antworten, wie zum Beispiel ein nicht vorgegebener Industriesektor, abgedeckt werden konnten. Außerdem wurden Freitextfelder für Fragen benutzt, für die keine Antwortmöglichkeiten vorgegeben werden können, zum Beispiel: *Bitte beschreiben kurz (d.h., weniger als 100 Wörter) Sie das Projekt oder das Produkt, auf das sich Ihre Antwort bezieht, oder geben Sie ein Akronym an.*

In den 935 für diese Arbeit betrachteten Fragebögen finden sich also insgesamt bis zu $935 \times 21 = 19635$ ausgefüllte Freitextfelder. Da diese nicht automatisiert ausgewertet werden können und eine manuelle Auswertung den Umfang dieser Arbeit übersteigen würde, wurden alle Freitextfelder bei der Analyse der Daten ignoriert. Im Gegensatz zu den anderen Änderungen an dem Datensatz wird hierdurch die Aussagekraft der Ergebnisse beeinträchtigt. Kapitel 6.3 geht hierauf näher ein. Da keine automatisierte Auswertungsmöglichkeit für diese Felder zur Verfügung steht, kann dies allerdings nicht vermieden werden. Zur Entfernung wurde manuell eine Liste mit den Indizes der Freitextfelder erstellt, anschließend wurde der Datensatz verkleinert:

```
data <- data[index_no_text]
```

4.4.3. Entfernen nicht relevanter Fragen

Der Fragebogen für die HELENA-Studie beinhaltet auch Fragen, die für diese Arbeit nicht relevant sind, zum Beispiel: *Wie haben Sie von dieser Studie erfahren, bzw. wie*

4. Durchführung

wurden Sie kontaktiert? Um die Analyse einfacher zu gestalten, wurden diese Fragen ebenfalls aus dem Datensatz entfernt. Im Folgenden werden Begründungen für das als nicht relevant Einstufen einzelner Fragen und Kategorien gegeben.

Experience

Diese Kategorie enthält zwei Fragen zu den persönlichen Erfahrungen des Umfrageteilnehmers mit den genutzten Methoden und Praktiken. Da dies nicht Teil des Entwicklungskontextes des Projektes ist, wurden sie entfernt.

Closing

In dem letzten Abschnitt der Umfrage wurden ausschließlich Fragen zum Umfeld der Studie oder der Teilnehmer gestellt. Ein Beispiel für die insgesamt acht Fragen in diesem Abschnitt ist: *Haben Sie bereits in der 1. Stufe von HELENA teilgenommen?* Weil dieser Abschnitt keine Verbindung zu dem Entwicklungskontext der behandelten Projekte hat, wurde er für diese Arbeit komplett aus dem Datensatz entfernt.

In welchem Land arbeiten Sie persönlich?

Viele Unternehmen agieren heute global, auch Softwareprojekte können Länderübergreifend entwickelt werden. Deshalb und weil sich diese Frage nur auf den Mitarbeiter bezieht, der den Fragebogen beantwortet hat, wurde sie ebenfalls als für diese Arbeit nicht relevant eingestuft.

Wird das Projekt bzw. Produkt, auf das sich Ihre Antwort bezieht, (global) verteilt entwickelt?

Um ein Produkt global zu entwickeln muss das entsprechende Unternehmen entweder Standorte in mehreren Ländern haben oder mit anderen kooperieren. Da nicht zu erwarten ist, dass die Kritikalität eines Projektes ausschlaggebend für Kooperation mit anderen Unternehmen oder die Verteilung auf mehrere Standorte ist, wurde diese Frage entfernt.

Welche Rolle haben Sie im Projekt bzw. im Produkt hauptsächlich inne?

Die Kritikalität hat keinen Einfluss auf die Rolle des an der Umfrage teilnehmenden Mitarbeiters, weswegen sie für die Analyse nicht betrachtet wurde.

Wie viele Jahre Erfahrung haben Sie in der Software- und Systementwicklung?

Auch hier wird nur Bezug auf den teilnehmenden Mitarbeiter genommen.

Wie wurden die Kombinationen von Frameworks, Methoden und Praktiken in Ihrem Unternehmen entwickelt?

Wie wird der projektspezifische Entwicklungsprozess festgelegt/definiert?

Weichen Sie bewusst von definierten Regeln ab?

Welches sind die grundsätzlichen Ziele, welche Sie mit Ihrer Auswahl und Kombination von Entwicklungsansätzen verfolgen?

Mit diesen Themen befassen sich bereits andere Arbeiten, zum Beispiel J. Klünder et al. [15]. Außerdem soll diese Arbeit nur einen Überblick über den Entwicklungskontext von kritischen Projekten und nicht dessen Entstehung geben.

Warum haben Sie die zuvor genannten Standards implementiert?

Wie wird die Compliance des Entwicklungsprozesses überprüft?

Diese Fragen wurden ebenfalls entfernt, da die Antworten von dem jeweils gewählten Standard abhängen und dieser nur in einem Freitextfeld erwähnt werden konnte.

Indizes für alle nicht relevanten Fragen wurden in ein Array geschrieben, welches zur Entfernung benutzt wurde:

```
data <- data[index_relevant]
```

4.5. Analyse

Nach der Vorverarbeitung enthält der Datensatz die Antworten von 935 Umfrageteilnehmern auf Fragen mit insgesamt 117 Antwortmöglichkeiten. Im ersten Schritt wurde eine Möglichkeit zur Identifizierung von Verbindungen aus zwei Variablen gesucht, um Zusammenhänge in den Daten zu erkennen. Wie in Kapitel 2.2.2 erläutert, ist das Testen auf statistische Abhängigkeit dafür gut geeignet. Da außerdem die Vermutung aufgestellt wurde, dass es lineare Abhängigkeiten geben könnte, zum Beispiel zwischen Projektkritikalität und Unternehmensgröße, wurden zunächst, für alle Kombinationen von Variablen, Korrelationskoeffizienten berechnet und die Ergebnisse in einer Matrix gespeichert. Anschließend wurde diese so verkleinert, dass darin nur noch Kombinationen enthalten waren, in denen mindestens eine Variable Teil der Kritikalitätsfrage war, da in dieser Arbeit nur die Zusammenhänge zwischen Kritikalität und den anderen Variablen untersucht werden. Außerdem wurden die Ergebnisse auf zwei Nachkommastellen gerundet. Dies ist ausreichend, um den Grad der Korrelation zu bestimmen. Zu diesem Zweck können die folgenden Befehle in R verwendet werden:

```
cor_mat <- round(data.matrix(cor(data)))
cor_mat <- cor_mat[33:43, ]
```

Abbildung 4.2 zeigt den Ausschnitt der Korrelationsmatrix mit den höchsten Korrelationskoeffizienten. Nach der in Kapitel 2.2.3 vorgestellten Kategorisierung von Korrelationskoeffizienten ρ , müsste folgendes gelten, damit überhaupt von einem linearen Zusammenhang zwischen zwei Variablen gesprochen werden kann:

$$|\rho| > 0,3$$

Allerdings wurde, wie in Abbildung 4.2 zu sehen, diese Bedingung für keine der Variablen erfüllt, also gibt es keine linearen Zusammenhänge zwischen den hier untersuchten Variablen. Vermutlich ist dies dadurch begründet, dass der Datensatz ausschließlich kategoriale Variablen enthält. Da die Berechnung der Korrelationskoeffizienten nicht zu dem Ziel geführt hat, Zusammenhänge in den Daten zu identifizieren, wurden sie bei der weiteren Analyse nicht verwendet.

4. Durchführung

	Lebensgefährdend	Systemverlust	System Schädigung	Finanzieller Schaden	Zivilrecht	Strafrecht	Umweltgefährdend	Reputationschädigend	Geschäftsschädigend
Projektgröße	0,23	0,11	0,1	0,09	0,11	0,13	0,11	0,05	0,17
Stage-Gate	0,14	0,22	0,16	0,03	0,03	0,07	0,12	0,12	0,02
V-Modell	0,24	0,32	0,26	0,12	-0,02	0,13	0,16	0,21	0,07
Design Reviews	0,21	0,13	0,14	0,13	0,12	0,07	0,09	0,05	0,15
End-to-End Testing	0,23	0,17	0,14	0,13	0,1	0,13	0,14	0,06	0,1
Externe Standards	0,27	0,25	0,15	0,14	0,08	0,16	0,2	0,17	0,12

Abbildung 4.2.: Ausschnitt aus der Korrelationsmatrix. Es werden nur die Ergebnisse mit dem höchsten Betrag gezeigt.

Der nächste Schritt zur Erreichung dieses Ziels war eine allgemeine Prüfung auf statistische Abhängigkeit in dem Datensatz. Da es sich, wie bereits erwähnt, um kategoriale Daten handelt, ist der Chi-Quadrat Test hierfür geeignet. Für dessen Verwendung muss die in Kapitel 2.2.4 erläuterte Bedingung für den erwarteten Wert einer Zelle gelten. Sollte dies nicht der Fall sein, kann auf Fishers exakten Test zurückgegriffen werden. Da die von R für den Chi-Quadrat Test zur Verfügung stehende Funktion `chisq.test()` zwar eine Warnung ausgibt, wenn die für den Test notwendige Bedingung nicht erfüllt ist, aber keine Möglichkeit enthält in diesem Fall den Fisher Test zu verwenden, wurde hierzu eine neue Funktion `chiFisher` geschrieben. Der Quellcode befindet sich auf der CD im Anhang. Die Funktion akzeptiert zwei Data Frames mit Messergebnissen für alle zu testenden Variablen und berechnet daraus, mit Hilfe der in R zur Verfügung stehenden Funktionen, die Wahrscheinlichkeiten für statistische Unabhängigkeit von allen Kombinationen der Variablen aus dem ersten Data Frame mit denen aus dem zweiten. Dabei wird nur dann der Fisher Test an Stelle des Chi-Quadrat Tests verwendet, wenn die notwendige Bedingung für diesen nicht erfüllt ist. Das Ergebnis der Berechnung wird in einem neuen Data Frame gespeichert. Folgender Befehl wendet die Funktion auf die Kritikalitätsfrage und die anderen Variablen an:

```
hybrid_p_vals <- chiFisher(data, subset(data,
  select=D006_01:D006_09))
```

In Abbildung 4.3 ist der Teil der so entstandenen Ergebnisse zu sehen, der die Wahrscheinlichkeiten für statistische Unabhängigkeit zwischen dem Industriesektor eines Projektes und dessen Kritikalität zeigt. Die vollständigen Ergebnisse können dem Anhang entnommen werden.

Wie zu sehen ist, besteht zum Beispiel ein statistisch signifikanter Zusammenhang zwischen Projekten in der Automobilindustrie und Software, die bei einem Fehler Leben gefährdet. In diesem Beispiel lässt sich leicht erahnen, dass in der Automobilindustrie eine vergleichbar höhere Anzahl an Safety-Critical Projekten durchgeführt werden, als in anderen Industriesektoren. Wie bereits erwähnt kann allerdings nur durch das Testen auf statistische Unabhängigkeit im Allgemeinen die Art des Zusammenhangs noch nicht bestimmt werden.

	Lebensgefährdend	Systemverlust	System Schädigung	Finanzieller Schaden	Zivilrecht	Strafrecht	Umweltgefährdend	Reputationschädigend	Geschäftsschädigend
Automobil	8,71E-08	0,124	0,571	0,995	0,044	0,00149	7,25E-08	0,36	0,043
Luffahrt	0,0177	0,0443	0,972	1	0,38	0,528	0,0502	0,117	0,385
Cloud	0,118	0,395	0,297	0,853	0,594	0,721	0,614	0,0586	0,00246
Verteidigung	0,187	0,000702	0,457	1	0,771	1	1	0,557	0,128
Energie	0,0134	0,505	0,968	1	0,611	0,998	0,00041	0,466	0,57
Finanzsektor	1,24E-06	0,431	0,184	1,29E-10	1,12E-08	0,00147	0,0568	0,291	0,000278
Spiele	0,341	0,755	0,545	0,707	0,245	0,758	1	1	0,292
Smart Home	0,411	0,268	0,0546	0,349	0,2	0,788	0,241	0,565	0,702
Logistik	0,507	0,158	0,0658	0,37	0,38	0,151	0,18	0,0164	0,175
Entertainment	0,211	0,894	0,431	0,275	0,655	0,141	1	0,485	0,821
Medizin	2,23E-19	0,669	0,145	0,0191	0,58	0,272	0,629	0,14	0,782
Mobil Sektor	0,0178	0,886	0,716	0,474	0,133	0,286	0,952	0,00192	0,0155
Embedded	1	0,00211	0,633	0,939	0,0057	0,0519	1	0,935	0,568
Information	0,0177	0,941	0,213	0,23	0,913	0,318	0,529	0,194	0,923
Öffentlicher Sektor	0,636	0,541	0,78	0,0316	0,0135	0,223	1	0,924	0,514
Robotik	0,0127	0,00364	0,155	0,0996	0,311	0,342	1	0,958	0,0854
Raumfahrt	0,353	0,000268	0,797	1	0,106	0,468	0,507	0,642	0,00839
Telekommunikation	0,204	0,069	0,333	0,718	1	0,72	0,571	0,953	1
Web Anwendung	2,77E-05	0,179	0,473	0,507	0,91	0,142	0,336	0,000176	0,0122

Abbildung 4.3.: Ausschnitt aus den Ergebnissen des Chi-Quadrat Tests. Werte unter 0,05 sind grün und Werte unter 0,01 blau hinterlegt.

Deshalb wurden im letzten Schritt der Analyse die Kombinationen von Variablen, die voneinander statistisch abhängig sind, genauer untersucht. Eine Möglichkeit dafür ist, die Antworten auf eine Frage für den Fall, dass das Projekt einer bestimmten Kritikalitätsklasse angehört mit denen für den Fall, dass es dieser nicht angehört, zu vergleichen. Zum Beispiel könnten die Industriesektoren von Projekten, die umweltgefährdend sind mit den Industriesektoren von Projekten verglichen werden, die dieser Kategorie nicht angehören. In dem HELENA-Datensatz enthält eine Zelle die Zahl 2, wenn eine Antwortmöglichkeit von dem entsprechenden Umfrageteilnehmer angekreuzt wurde und die Zahl 1, wenn dies nicht der Fall war. Also können für das oben genannte Beispiel in R mit folgendem Befehl die Anzahl der Projekte im Automobilsektor(D005_01), in denen menschliches Leben gefährdet wird(D006_01), ausgegeben werden:

```
table(unlist(data$D005_01[data$D006_01 == 2]))
```

Dies führt zu der Ausgabe:

```
1 2
96 32
```

Umgekehrt wird die Zugehörigkeit zum Automobilsektor von nicht lebensgefährdenden

4. Durchführung

Projekten mit diesem Befehl ausgewertet:

```
table(unlist(data$D005_01[data$D006_01 == 1]))
```

Mit folgender Ausgabe:

```
 1  2  
737 70
```

Aus den Ergebnissen folgt, dass 31,3% aller Softwareprojekte im Automobilbereich und nur 13,7% aller Projekte in anderen Industriesektoren Menschenleben gefährden. Die oben aufgestellte Vermutung wurde also bestätigt.

Mit den bis hier gezeigten Methoden konnten nun die Forschungsfragen beantwortet werden. Beispielsweise bezieht sich Forschungsfrage 1 auf die für die Entwicklung genutzten Methoden und Praktiken. Zur Beantwortung wurde also für jede Kombination von Fragen zu diesen Themen und den Kritikalitätsklassen die voneinander statistisch abhängig sind, die genauen Verteilungen, wie in dem Beispiel beschrieben, näher untersucht. Analog können auch die anderen Forschungsfragen mit den für sie relevanten Variablen beantwortet werden.

5. Auswertung

In diesem Kapitel werden Unterschiede zwischen den Kritikalitätskategorien vorgestellt, indem auf jede Antwortmöglichkeit für die Kritikalitätsfrage in der HELENA-Studie einzeln eingegangen wird.

Da viele Kritikalitätskategorien voneinander abhängen, zum Beispiel sind Projekte, die strafrechtliche Folgen haben können auch potentiell reputationsschädigend, wurde zuerst untersucht, welche Antwortmöglichkeiten von den Teilnehmern zusammen angekreuzt wurden. Wie in 5.1 zu erkennen ist, wurden die Kategorien Reputationsschädi-

	Lebensgefährdend	Systemverlust	System Schädigung	Finanzieller Schaden	Zivilrecht	Strafrecht	Umweltgefährdend	Reputationsschädigend	Geschäftsschädigend
Lebensgefährdend	1,00	0,34	0,59	0,59	0,53	0,35	0,25	0,80	0,65
System Verlust	0,33	1,00	0,78	0,77	0,57	0,30	0,20	0,88	0,80
System Schädigung	0,16	0,21	1,00	0,64	0,46	0,21	0,08	0,80	0,73
Finanzieller Schaden	0,13	0,18	0,55	1,00	0,51	0,21	0,08	0,82	0,80
Zivilrecht	0,19	0,21	0,62	0,80	1,00	0,34	0,10	0,85	0,82
Strafrecht	0,31	0,28	0,70	0,81	0,86	1,00	0,17	0,88	0,85
Umweltgefährdend	0,56	0,47	0,68	0,75	0,63	0,44	1,00	0,84	0,74
Reputationsschädigend	0,15	0,17	0,57	0,69	0,45	0,19	0,07	1,00	0,82
Geschäftsschädigend	0,13	0,16	0,54	0,69	0,45	0,19	0,06	0,84	1,00

Abbildung 5.1.: Relativer Anteil zusammen gewählter Antwortmöglichkeiten von Kritikalitätsfrage. Spalten zeigen, wovon einzelne Kategorien beeinflusst wurden. Beispiel: Zeile 1, Spalte 4 bedeutet, dass 77% aller Projekte, die zu Systemverlust führen können, auch zu finanziellem Schaden führen können.

gend, Geschäftsschädigend, Zivilrecht, Finanzieller Schaden und System Schädigung oft mit anderen Kategorien zusammen gewählt. Dies deutet darauf hin, dass es sich bei ihnen um weniger kritische Kategorien handelt, da die Einordnung meist als Folge von negativen Auswirkungen anderer Kritikalitätskategorien entsteht. Beispielsweise führt der komplette Systemverlust zu finanziellen Kosten, wenn das defekte System ausgetauscht werden muss. In den folgenden Abschnitten wird auf die einzelnen Antworten auf die Frage nach der Kritikalität eines Projektes eingegangen. Dabei werden nur Variablen betrachtet, die nach dem Chi-Quadrat beziehungsweise Fisher Test statistisch von der jeweiligen Kritikalitätsklasse abhängig sind.

Abbildung 5.2 zeigt Variablen, die eine besonders hohe Wahrscheinlichkeit auf statistische Abhängigkeit von den Kritikalitätsklassen haben. Die Analyse in den folgenden Unterkapiteln wird zeigen, wie diese und andere Variablen von der Kritikalität eines

5. Auswertung

	Lebensgefährdend	Systemverlust	System Schädigung	Finanzieller Schaden	Zivilrecht	Strafrecht	Umweltgefährdend	Reputationschädigend	Geschäftsschädigend
Unternehmensgröße	0,0561	0,0107	0,00152	0,201	0,719	0,0486	0,234	0,0538	0,239
Projektgröße	0,00011	0,0206	0,013	5,36E-06	0,000881	0,000299	0,919	5,17E-12	1,15E-12
Stage-Gate	1,17E-08	0,000574	0,502	0,489	0,513	0,0379	0,00278	0,335	0,0602
V-Modell	4,72E-21	6,10E-11	0,00627	0,89	0,00826	1,96E-05	0,000214	0,319	0,0632
Design Reviews	0,000157	0,00829	0,000534	0,0429	0,736	0,459	0,34	0,000306	0,0043
End-to-End Testing	0,000144	0,00546	0,0045	0,00384	0,00748	0,0122	0,395	0,0275	0,227
Externe Standards	3,81E-11	0,000101	0,000215	0,0319	3,28E-05	1,49E-07	1,74E-05	0,00245	0,76

Abbildung 5.2.: Ausschnitt aus den Ergebnissen des Chi-Quadrat Tests. Einzelne Werte zeigen Wahrscheinlichkeit auf statistische Unabhängigkeit.

Projektes beeinflusst werden. Die Tabelle mit allen Ergebnissen des Chi-Quadrat beziehungsweise Fisher Tests befindet sich im Anhang.

5.1. Lebens- oder Gesundheitsgefährdend

Alle Projekte, bei denen die Frage nach möglichen Folgen im Falle eines Softwarefehlers mit "...*menschliches Leben oder Gesundheit bedrohen*" beantwortet wurde, sind Teil dieser Kategorie. Sie sind Teil der Klasse Safety-Critical [5] und haben damit die höchste Kritikalität aller in der HELENA-Umfrage behandelten Projekte. Von den insgesamt 935 für diese Arbeit betrachteten Projekte, gehören 128 (13,7%) dieser Kategorie an. Im Vergleich zu anderen Projekten sind sie besonders groß, das heißt es werden mehr Personenjahre benötigt, um sie abzuschließen. Abbildung 5.3 zeigt die relativen Anteile an Softwareprojekten in den von der Studie vorgegeben Größenintervallen. Es ist zu erkennen, dass die überwiegende Mehrheit (75%) aller lebensgefährdenden Projekte sehr groß sind, allerdings sind auch 58% der gesamten Menge an Projekten Teil dieser Kategorie. Dies liegt vermutlich daran, dass das Intervall für die Aufwand sehr großer Projekte schon bei einem Personenjahr beginnt und viele große globale Softwareprojekte diesen Aufwand überschreiten. Dennoch ist der Unterschied zu weniger kritischen Projekten noch deutlich erkennbar. Der höhere Aufwand könnte unter anderem durch die weiter verbreitete Nutzung externer Standards entstehen. Diese werden in 67,7% der lebensgefährdenden und nur in 36,9% aller Softwareprojekte verwendet. Zusätzlich unterliegen sie häufig einer Zertifizierung. Im Gegensatz zu allgemeinen Projekten (42,7%) ist das hier bei 64% aller Projekte der Fall. Die Zertifizierung könnte bei hoch kritischen Projekten eine Möglichkeit für den Auftraggeber sein, abzusichern, dass das Produkt geltende Standards einhält. Zu den für die Projekte der HELENA-Studie verwendeten Standards gehören unter anderem ISO 9001 und IEC 27001, für beide existiert zusätzlich ein Zertifizierungsverfahren.

Wie aus Abbildung 5.4 entnommen werden kann, sind die Anwendungsgebiete, in denen am häufigsten lebensgefährdende Software eingesetzt wird die Automobilindus-

5.1. Lebens- oder Gesundheitsgefährdend

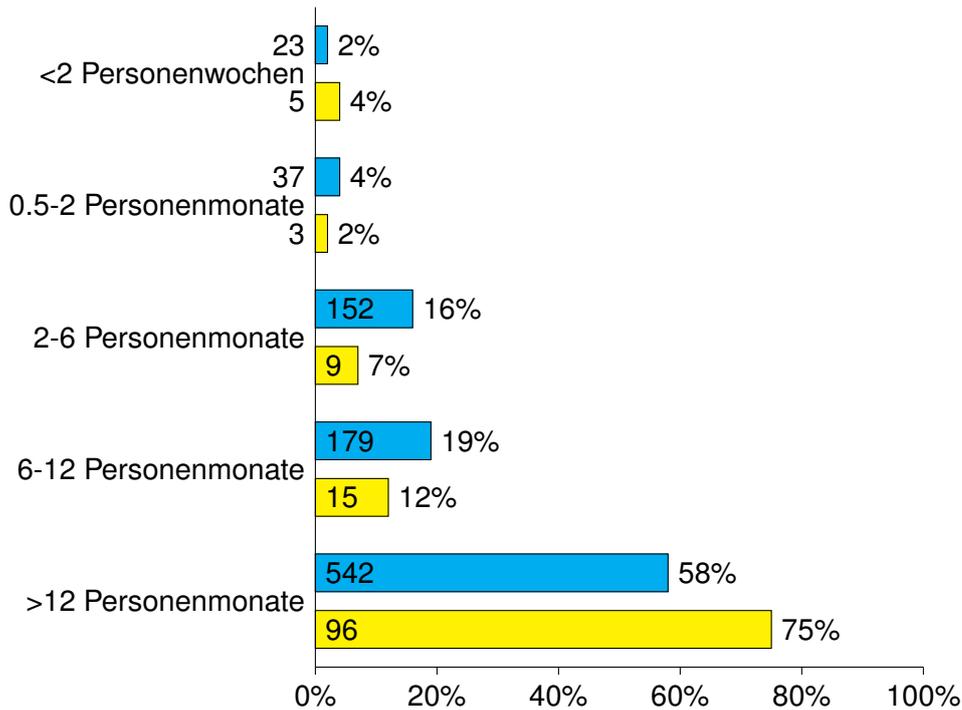


Abbildung 5.3.: Größe von Softwareprojekten. Blaue Balken zeigen die Größen aller Projekte, gelbe Balken zeigen die Größen von lebensgefährdenden Projekten. Graph zeigt relative Häufigkeiten der einzelnen Größen. Absolute Häufigkeiten stehen auf- oder links neben den Balken.

trie (31,4%), die Luftfahrt (29,4%) und der medizinische Bereich (45%). Dabei handelt es sich vor allem in der Luftfahrt (60%) und der Automobilindustrie (50%) und Projekte, bei denen die Software zusammen mit Hardware entwickelt wird. Dies lässt darauf schließen, dass Software in diesen Industriesektoren häufig Hardware steuert, die bei Fehlverhalten Menschen gefährden kann, wie zum Beispiel der Autopilot eines Flugzeugs. Im Gegensatz dazu wird nur 17% der Software im medizinischen Bereich Teil zusammen mit Hardware entwickelt. Die hohe Kritikalität wird hier also vermutlich hauptsächlich durch medizinische Diagnosesoftware verursacht.

	Lebensgefährdend		Hardware Projekt
	JA	Nein	
Automobil	32 (31,4%)	70 (68,6%)	16 (50%)
Luftfahrt	10 (29,4%)	24 (70,6%)	6 (60%)
Medizinisch	41 (45%)	50 (55%)	7 (17%)
Energie	13(26,5%)	36(73,5%)	3(23%)

Abbildung 5.4.: Industriesektoren mit höchstem Anteil an lebensgefährdenden Projekten und Anteil an Hardwareprojekten die Safety-Critical sind.

5. Auswertung

Das Betrachten der für Projekte mit hoher Kritikalität verwendeten Methoden und Praktiken zeigt weitere Gründe für den erhöhten Aufwand. Der größte Unterschied ist dabei die weit verbreitete Nutzung des V-Modells, siehe Abbildung 5.5. Insgesamt wird dieses in 35% der lebensgefährdenden Projekte und nur in 11% aller Projekte oft oder immer verwendet. Außerdem ist der Anteil an Entwicklern von hoch kritischen Projekten, denen das Modell unbekannt ist, mit 21% nur etwa halb so hoch wie im Durchschnitt. Auch das Stage-Gate Modell wird besonders häufig genutzt. Während

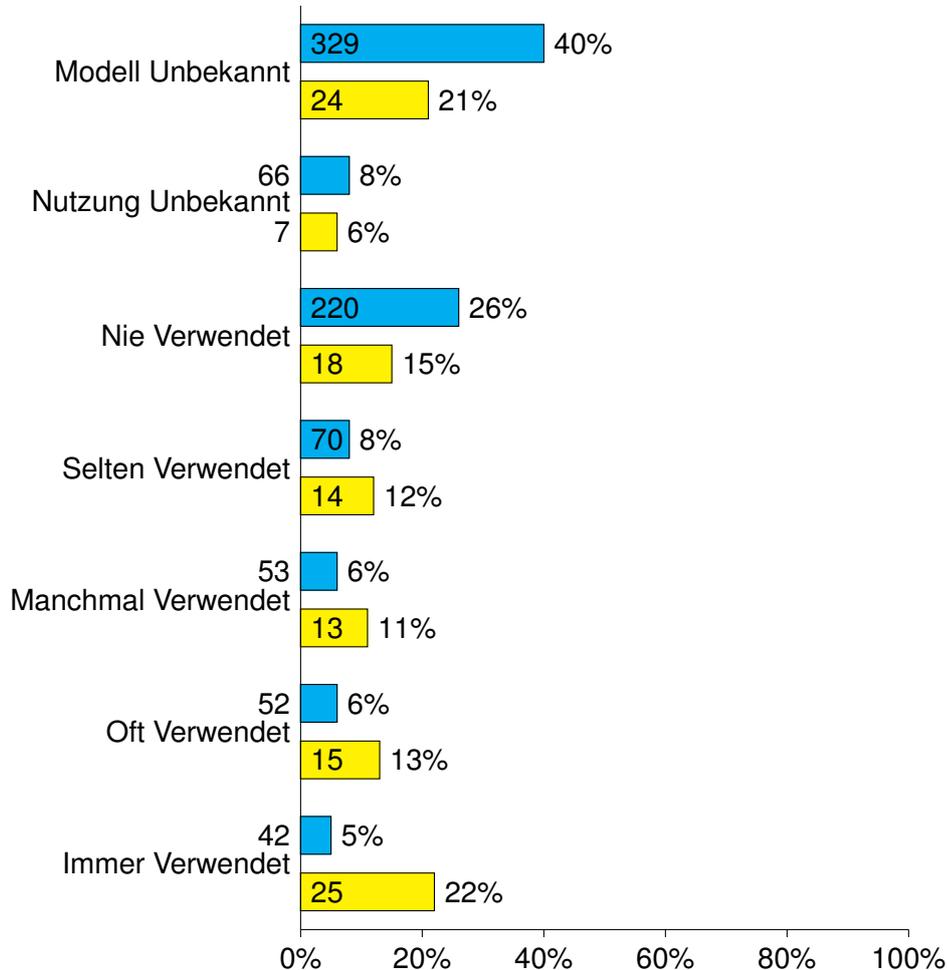


Abbildung 5.5.: Verwendung des V-Modells bei allen(blau) und bei lebensgefährdenden(gelb) Projekten

nur 8,4% aller Teilnehmer der Umfrage angegeben haben, dass sie es oft oder immer benutzen, waren es bei den Teilnehmern, die an Produkten gearbeitet haben durch die Menschenleben gefährdet werden können, etwa 27%. Ein weiterer Unterschied zu nicht kritischen Projekten ist die Verwendung von aufwändigen oder mit eventuell hohen finanziellen Kosten verbundener Praktiken. Diese Praktiken sind Design-Reviews, Destructive Testing, detaillierte Design Spezifikationen und Ent-to-End Testing. Dabei werden Design Reviews für 33,7% der lebensgefährdenden und 17,5% aller Projekte

immer verwendet. End-to-End Testing wird sogar von 71% der Befragten, die an sehr kritischen Projekten arbeiten, oft oder immer durchgeführt, detaillierte Design Spezifikationen werden von 44% der Befragten oft oder immer erstellt.

Letztlich finden sich auch bei dem Grad der Agilität von bestimmten Teilen des Entwicklungsprozesses Unterschiede zu durchschnittlichen Projekten. Projekt-, Qualitäts- und Risikomanagement werden bei lebensgefährdenden Softwareprojekten weniger agil betrieben. In Abbildung 5.6 sind der Anteil an traditionellen und agilen Teilen dieser Bereiche für kritische und allgemeine Projekte dargestellt. Es ist zu erkennen, dass bei kritischen Projekten zwar mehr traditionelle Methoden verwendet werden, der Anteil an agilen Methoden in diesen Bereichen aber dennoch überall mehr als 15% beträgt.

	Projektmgmt.		Qualitätsmgmt.		Risikomgmt.	
	tredit.	agil	tredit.	agil	tredit.	agil
Lebensgefährdend	42,1%	32,1%	53%	24,1%	55,5%	15,6%
Allgemein	27,9%	43,1%	34,3%	35,7%	38,5%	23,3%

Abbildung 5.6.: Anteil an Projekten bei denen Projekt-, Qualitäts- und Risikomanagement größtenteils oder vollständig traditionell beziehungsweise agil betrieben werden.

5.2. Kompletter Systemverlust

Die Antwortmöglichkeit "...zu komplettem Systemverlust führen" wurde von 132 (15,8%) der insgesamt 935 Teilnehmer ausgewählt. Projekte, die dieses Kriterium erfüllen sind Teil der Kategorie Mission-Critical [5], was nach der in Kapitel 2.1 vorgestellten Einteilung der zweithöchsten Kritikalität entspricht. Sie sind, ähnlich der lebensgefährdenden Projekte, meist sehr groß (siehe Abbildung 5.5). Auch hier sind die meisten Projekte (70%) in dem selben Intervall für die Größe eines Softwareprojektes. Deshalb ist der Unterschied zu weniger kritischen Projekten nicht genau bestimmbar, da die Verteilung innerhalb dieses Intervalls unbekannt ist. Die Verwendung externer Standards ist auch bei der Entwicklung von Software, die bei einem Fehler zu komplettem Systemverlust führen können, mit 54,3%, deutlich weiter verbreitet, als bei dem Durchschnitt aller Projekte (36,9%). Eine weitere Ähnlichkeit zu den lebensgefährdenden Projekten ist außerdem das häufige Teilnehmen an Zertifizierungsverfahren für 50,9% der Projekte. Damit liegen die missionskritischen Projekte in ihrer Größe, der Verwendung von externen Standards und der Teilnahme an Zertifizierungsverfahren zwischen dem Durchschnitt aller Projekte und den lebensgefährdenden Projekten. Die Anwendungsgebiete, in denen am häufigsten missionskritische Software zum Einsatz kommt sind Robotik (40%), Verteidigungssysteme (37,5%) und die Raumfahrt (35%).

Verwendete Methoden und Praktiken ähneln denen von lebensgefährdenden Projekten, zum Beispiel haben (15,8%) der Befragten angegeben, das V-Model oft oder immer zu benutzen. Abbildung 5.6 zeigt die am häufigsten verwendeten Methoden und Praktiken für missionskritische Projekte. Im Vergleich mit der zuvor behandelten Kritikalitätsklasse, werden bei missionskritischen Projekten automatisierte Verfahren,

5. Auswertung

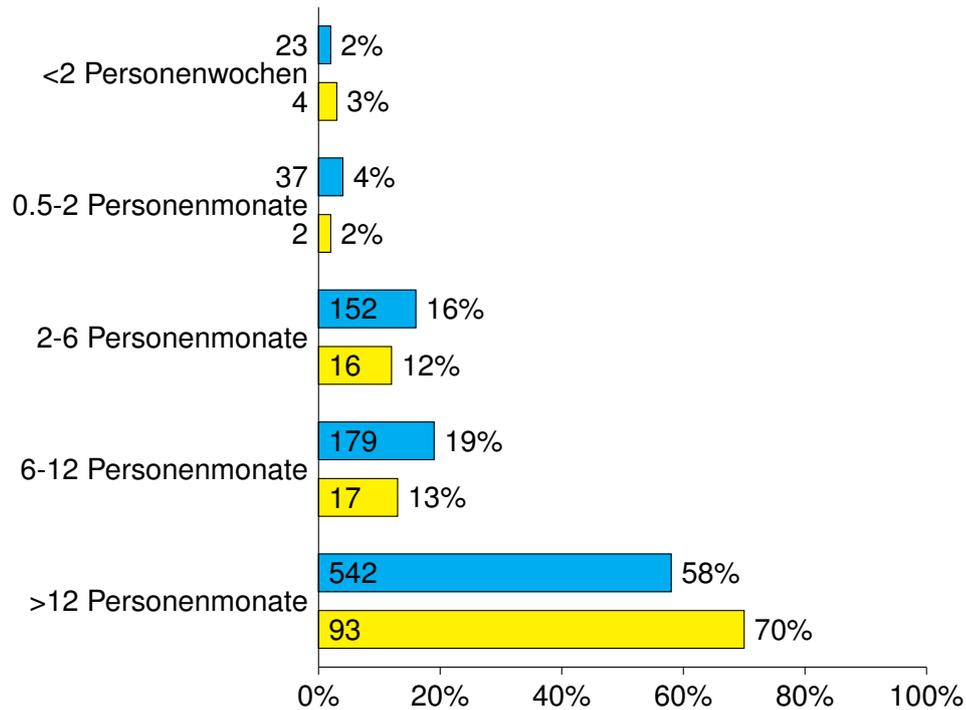


Abbildung 5.7.: Größe von Softwareprojekten. Blaue Balken zeigen die Größen aller Projekte, gelbe Balken zeigen die Größen von Projekten, die bei Ausfall zu komplettem Systemverlust führen können.

	Missionskritisch	Allgemein
Wasserfall	28(23,3%)	164(19%)
Stage-Gate	21(17,4%)	70(8,2%)
V-Model	42(26,7%)	94(11,2%)
Design Reviews	59(55,2%)	299(39%)
End-to-End Test	72(63,7%)	390(50,8%)
Automatisierte Codegenerierung	30(26,5%)	141(18,4%)
Automatisierte Unit Tests	67(59,4%)	386(50,3%)

Abbildung 5.8.: Verwendeten Methoden und Praktiken für missionskritische Software im Vergleich mit Software allgemein.

wie zum Beispiel Code Generation, häufiger verwendet. Ähnlichkeiten finden sich bei der Verwendung von traditionellen Modellen, wie dem V-Model und dem Stage-Gate Model. Design Reviews und Destructive Testing wird zwar auch bei missionkritischer Software häufiger benutzt, als im Durchschnitt, allerdings nicht so oft, wie bei lebensgefährdenden Projekten. Von den Befragten gaben für Destructive Testing 11,5% und für Design Reviews 55,2% an, die Verfahren oft oder immer zu verwenden.

Bei den Angaben zum Grad der Agilität einzelner Aktivitäten innerhalb des Projekts gibt es nur bei der Codeentwicklung signifikante Unterschiede zu dem durchschnittlichen Softwareprojekt. Während dies insgesamt bei 62,6% aller Projekte hauptsäch-

lich oder ausschließlich agil durchgeführt wird, setzen missionskritische Projekte zum großen Teil (32,6%) auf einen zwischen agil und traditionell ausgeglichenen Ansatz. Allerdings wird Code auch hier zu 47,8% größtenteils oder vollständig agil geschrieben.

5.3. Systemdegradierung

Dieser Abschnitt befasst sich mit allen Projekte, bei denen die Befragten auf die Kritikalitätsfrage der HELENA-Studie mit "...zu *System(Service) Degradierung führen*" geantwortet haben. Systemdegradierung zählt nach M. Hinchey [5] ebenfalls zu den missionskritischen Projekten. Im Gegensatz zu den Projekten, die im vorherigen Abschnitt behandelt wurden, ist die Kritikalität hier allerdings geringer, da Softwarefehler nicht zu kompletten Systemverlust führen. Dies wird besonders durch die Anzahl an Fragen mit signifikanten Unterschieden zu den Antworten bei durchschnittlichen Projekten deutlich: Die Anzahl an statistisch abhängigen Variablen ist nur etwa halb so hoch, wie bei den bereits behandelten Kritikalitätskategorien. Dies ist vermutlich dadurch zu erklären, dass mehr als die Hälfte (51,8%) der 935 Befragten angaben, dass die von ihnen entwickelte Software bei Fehlern Systemdegradierung verursachen kann. Auch bei Variablen, die nach dem Chi-Quadrat Test mit einer Wahrscheinlichkeit von über 99% statistisch von dieser Kritikalitätsklasse abhängig sind, ist meist keine große Abweichung von dem Durchschnitt aller Datenpunkte vorhanden. Ein Beispiel dafür ist in Abbildung 5.6, ein Vergleich der Anzahl an Mitarbeitern aller Firmen, zu sehen. Wie zu erkennen, sind Unternehmen, die potentiell systemdegradierende Software entwickeln, etwas größer als der Durchschnitt. Insgesamt hat die Unternehmensgröße allerdings kaum einen Einfluss auf die Kritikalität der von ihnen entwickelten Software. Dieses Ergebnis wird dadurch bestätigt, dass zwischen dieser Variable und den anderen Kritikalitätsklassen nach dem Chi-Quadrat Test keine Abhängigkeit besteht.

Unterschiede zwischen dem Durchschnitt aller Projekte und solchen, die zu Systemdegradierung führen können, finden sich bei den verwendeten Methoden und Praktiken und bei der Implementierung externer Standards. Dabei werden hauptsächlich die gleichen Methoden vermehrt verwendet, wie bei den Projekten, die zu komplettem Systemverlust führen können. Auch hier werden von vielen Unternehmen Design Reviews (43,9%), End-to-End Testing (56,5%) und automatisierte Code Generation (20,8%) für die meisten oder alle Projekte verwendet. Im Gegensatz zu Projekten, die zu komplettem Systemverlust führen können, werden traditionelle Modelle, wie Stage-Gate oder das V-Model, nicht überdurchschnittlich häufig verwendet.

5.4. Finanzielle Kosten

Projekte, bei denen die Kritikalitätsfrage mit "...zu *finanziellem Verlust führen*" beantwortet wurde, sind nach M. Hinchey und L. Coyle [5] Teil der Kategorie Business-Critical. Insgesamt haben 564 (60,3%) der 935 Teilnehmer diese Antwortmöglichkeit

5. Auswertung

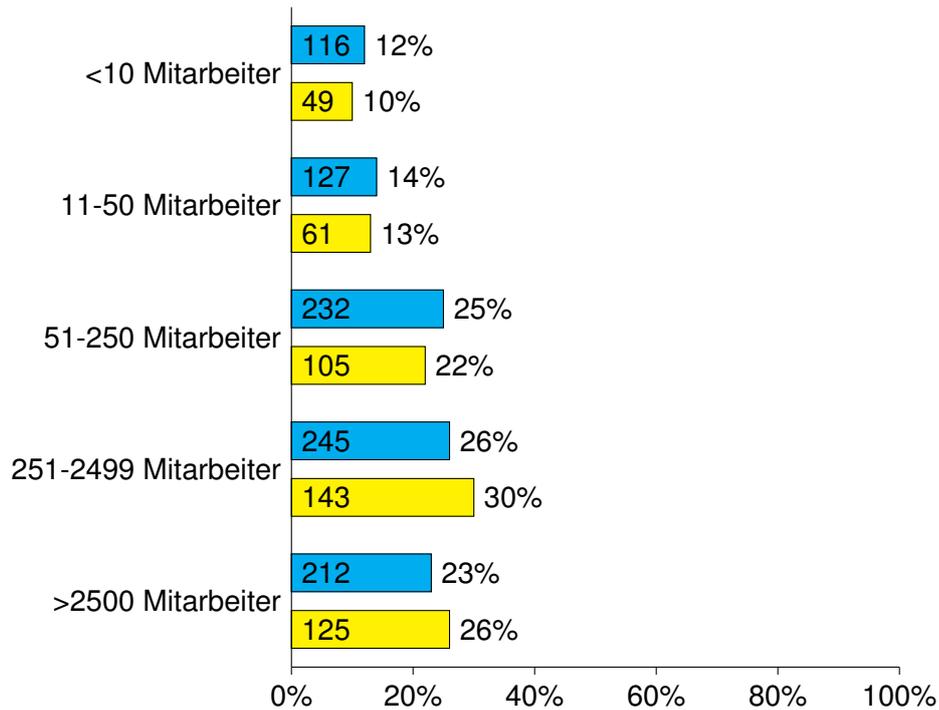


Abbildung 5.9.: Anzahl an Mitarbeitern von Softwareunternehmen, blaue Balken zeigen Alle Projekte, gelbe zeigen nur solche, die bei Fehler zu Systemdegradierung führen können.

gewählt. Die hohe Zahl entsteht vor allem durch Überschneidung mit anderen Kritikalitätsklassen, wie zum Beispiel der Systemdegradierung, da das Eintreten der anderen negativen Folgen häufig auch in finanziellen Kosten zu deren Beseitigung resultiert. Abbildung 5.1 zeigt Überschneidungen von Antworten auf die Kritikalitätsfrage der HELENA-Studie. Wie zu erkennen ist, entstehen bei sehr vielen Projekten in anderen Kritikalitätskategorien auch finanzielle Schäden, besonders bei strafrechtlichen (81%) und zivilrechtlichen (80%) Folgen. Anders als die vorher behandelten Kategorien, unterscheidet sich die Größe von Projekten, die zu finanziellen Schäden führen können, nicht stark von der durchschnittlichen Größe aller Projekte. Beispielsweise benötigen 63,8% aller Projekte in dieser Kategorie mehr als ein Personenjahr bis zum Abschluss, während im Durchschnitt 58% aller Projekte mindestens so lange dauern.

Das Anwendungsgebiet, bei dem am häufigsten Software dieser Kategorie zum Einsatz kommt ist der Finanzsektor, hier können 79% der Softwareprojekte bei Fehlern zu finanziellem Schaden führen. Verwendete Methoden und Praktiken bei der Entwicklung von Softwareprojekten in dieser Kategorie unterscheiden sich nicht von den allgemein verwendeten, wie sie zum Beispiel von J. Klünder et al. [15] und M. Kuhrmann et al. [16] vorgestellt wurden. Auch die Einhaltung von externen Standards oder die Teilnahme an Zertifizierungsverfahren ist nicht weiter verbreitet, als üblich. Auch der Grad der Agilität von Aktivitäten während der Entwicklung, wie zum Beispiel das Qualitätsmanagement, ist durchschnittlich.

5.5. Zivilrechtliche Folgen

Die fünfte Antwortmöglichkeit auf die Kritikalitätsfrage ist *"...rechtliche Konsequenzen haben (Zivilgesetz)"*; sie gehört ebenfalls der Kategorie Business-Critical [5] an. Der Anteil an Befragten, die an einem Projekt dieser Kritikalitätsklasse arbeiten, ist 38,4%. Wie Abbildung 5.1 zeigt, treten zivilrechtliche Folgen oft zusammen mit anderen Konsequenzen von kritischen Projekten auf, ins Besondere, wenn es sie dabei um strafrechtliche Konsequenzen handelt. Damit sind Projekte dieser Kategorie, ähnlich der im vorherigen Abschnitt behandelten Projekte, weniger kritisch als zum Beispiel solche, die zu komplettem Systemverlust führen. Im Hinblick auf verwendete Methoden und Praktiken, Projektgröße und auftraggebende Industriesektoren, ähneln sie stark den Projekten, die zu finanziellem Schaden führen, das heißt sie unterscheiden sich nicht oder kaum von durchschnittlichen Projekten. Die einzige Variable, die sich anders verhält, ist die Einhaltung externer Standards. Während diese nur von durchschnittlich 36,9% aller Softwareprojekte implementiert werden, sind es bei Projekten, die zu zivilrechtlichen Verfahren führen können, 46,5%. Im Gegensatz zu Projekten mit höherer Kritikalität, wie totaler Systemverlust, wird hier allerdings nur durchschnittlich oft an Zertifizierungsverfahren teilgenommen.

5.6. Strafrechtliche Folgen

Wie Projekte mit eventuellen zivilrechtlichen Folgen, gehören auch Projekte bei denen die Kritikalitätsfrage mit *"...rechtliche Konsequenzen haben (Strafgesetz)"* beantwortet wurde der Kategorie Business-Critical [5] an. Da strafrechtliche Verfahren höhere Kosten für ein Unternehmen nach sich ziehen können als zivilrechtliche Verfahren, haben Projekte dieser Kategorie eine höhere Kritikalität. Die Anzahl an Projekten (143) mit möglichen strafrechtlichen Folgen ist vergleichbar mit der Anzahl an missionskritischen (132) oder lebensgefährdenden (128) Projekten. Abbildung 5.10 zeigt die Größe von eventuell strafrechtlich relevanten Softwareprojekten. Im Vergleich mit den Abbildungen 5.3 und 5.7 wird deutlich, dass Projekte dieser drei Kategorien etwa gleich aufwändig sind. Eine weitere Ähnlichkeit zu den anderen Kritikalitätskategorien ist die häufige Verwendung externer Standards (59,5%) und die Teilnahme an Zertifizierungsverfahren (52,2%) im Vergleich zum Durchschnitt aller Projekte (37% und 42,7%). Außerdem werden ebenfalls vermehrt traditionelle Methoden und Praktiken eingesetzt. Das V-Modell wird von 21,9%, detaillierte Design Reviews von 42,1% und das Stage-Gate Modell von 15% aller Befragten, die an Projekten mit möglichen strafrechtlichen Folgen arbeiten, oft oder immer verwendet. Auch der Anteil an traditionell durchgeführten Aktivitäten während der Entwicklung ist vergleichbar mit den Kategorien Lebensgefährdend und Missionskritisch (siehe Abbildung 5.13). Beispielsweise wird das Risikomanagement in 49,5%, das Konfigurationsmanagement in 44,5% und das Änderungsmanagement in 35,6% der Unternehmen, die an Projekten, die zu strafrechtlichen Prozessen führen können, hauptsächlich oder vollständig traditionell durchgeführt.

5. Auswertung

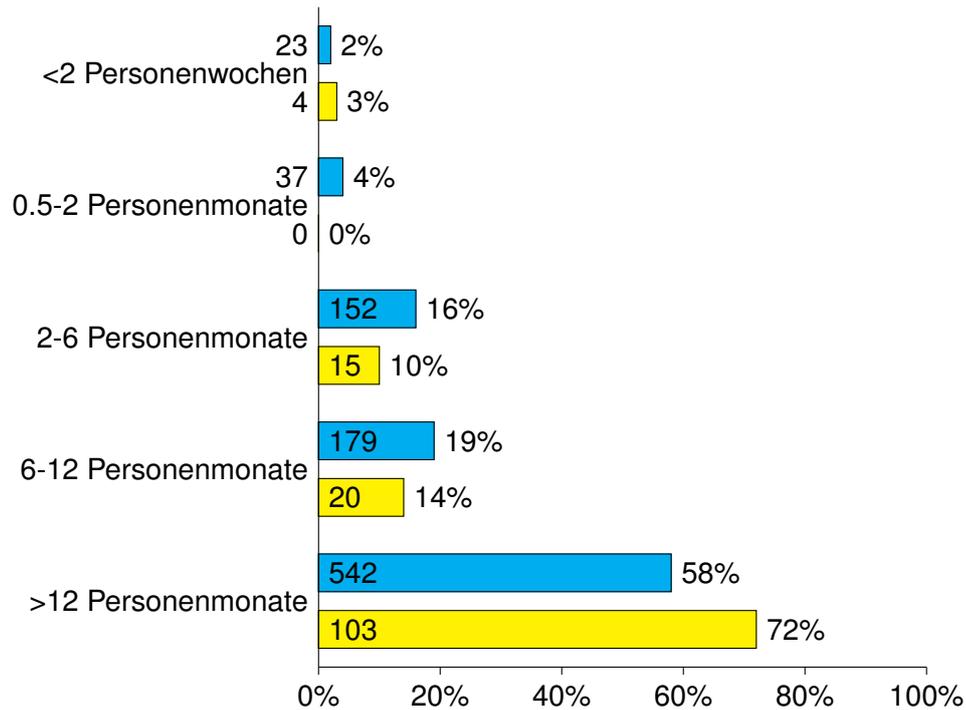


Abbildung 5.10.: Größe von Softwareprojekten. Blaue Balken zeigen die Größen aller Projekte, gelbe Balken zeigen die Größen von Projekten, die bei Fehlern zu strafrechtlichen Folgen führen können.

5.7. Umweltgefährdend

Neben lebensgefährdenden gehören auch umweltgefährdende Projekte, bei denen die Befragten mit *"...die Umwelt schädigen"* geantwortet haben, in die Kategorie Safety-Critical [5], sie haben also eine vergleichsweise hohe Kritikalität. Mehr als die Hälfte der 57 Projekte mit Risiko auf Umweltschäden werden in der Automobilindustrie (33,3%) und dem Energiesektor (17,5%) entwickelt. Da es viele Ähnlichkeiten mit den bereits vorgestellten Ergebnissen für lebensgefährdende, missionskritische und strafrechtlich relevante Projekte gibt, enthält dieser Abschnitt einen Vergleich der vier Kategorien. Dabei wird auf Gemeinsamkeiten und Unterschiede bei der Implementierung externer Standards, den verwendeten Methoden und Praktiken und dem Grad der Agilität einzelner Aktivitäten während des Projektes, zum Beispiel Risiko- und Projektmanagement, eingegangen.

Wie in Abbildung 5.11 zu erkennen, setzen Unternehmen bei der Entwicklung umweltgefährdender Software häufig auf Zertifizierungsverfahren und die Implementierung externer Standards. Zu diesen gehören unter anderem ISO 26262 [2] für die Automobilindustrie und ISO 9001 [4], für beide Standards existieren Zertifizierungsverfahren. Abbildung 5.12 zeigt, dass bei den Kategorien mit hoher Kritikalität vermehrt traditionelle Methoden, also Stage-Gate und das V-Model, und aufwändige Praktiken, wie Design Reviews und End-to-End Testing genutzt werden. Zum Beispiel wird das Stage-

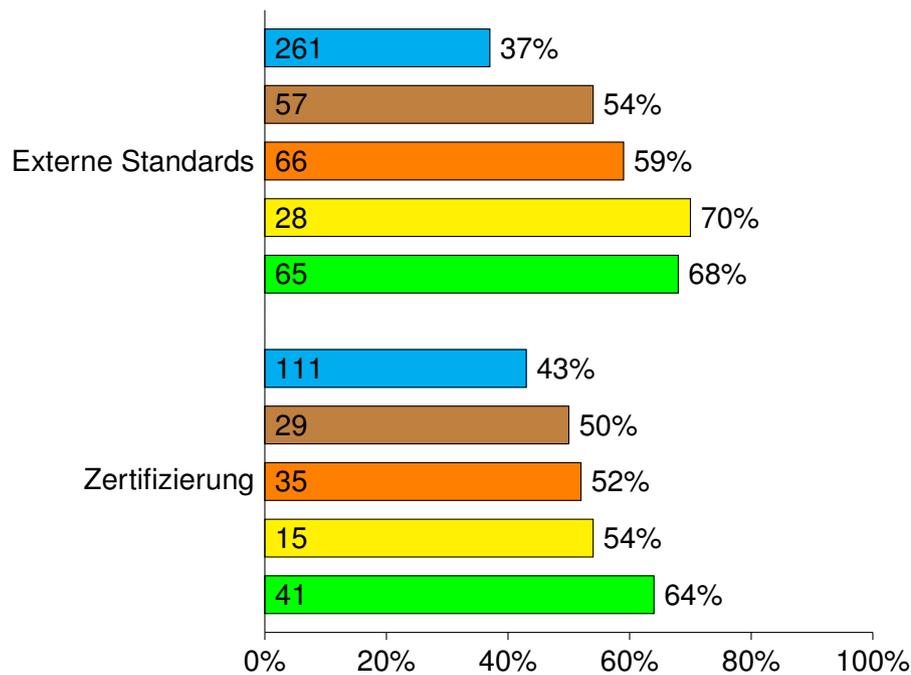


Abbildung 5.11.: Anteil an Projekten, die externen Standards implementieren beziehungsweise and Zertifizierungsverfahren teilnehmen. Blaue Balken: Durchschnitt aller Projekte, gelbe Balken: Umweltgefährdende Projekte, grüne Balken: lebensgefährdende Projekte, braune Balken: kompletter Systemverlust

Gate Modell von 24% und das V-Model von 28% alle Befragten, die an umweltgefährdenden Projekten arbeiten, für alle oder fast alle Projekte genutzt, während es durchschnittlich nur 11% für das V-Model und 8% für das Stage-Gate Modell sind. Dies führt vermutlich zu dem größeren Aufwand (siehe Abbildungen 5.3, 5.7 und 5.10), der zum Abschluss von kritischen Projekten getrieben werden muss. Dieser ist auch bei umweltgefährdenden Projekten, von denen 64,9% mehr als 1 Personenjahr bis zur Fertigstellung benötigen, etwas höher als bei dem Durchschnitt aller Projekte (58%). Dieser Unterschied ist allerdings nicht so groß, wie bei lebensgefährdenden oder missionskritischen Softwareprojekten. Da jedoch keine Informationen über die genaue Größe von Projekten, die mehr als 1 Personenjahr bis zum Abschluss benötigen, vorliegen, könnte dieser Unterschied auch noch größer sein. Beispielsweise könnte die durchschnittliche Größe eines umweltgefährdenden Projektes in der Kategorie *sehr groß* das doppelte der durchschnittlichen Größe aller Projekte in dieser Kategorie betragen, ohne das dies aus dem HELENA-Datensatz ersichtlich wäre. Auch bei dem Grad der Agilität verschiedener Aktivitäten während der Entwicklung gibt es Ähnlichkeiten. In Projekten der vier Kritikalitätskategorien werden diese Aktivitäten öfter traditionell durchgeführt (Abbildung 5.13). Dabei werden für Projekte, die bei Fehlern Umweltschäden verursachen können, besonders das Projekt- und Risikomanagement weniger agil betrieben. Verglichen mit dem Durchschnitt aller Projekte, bei denen das Risikomanagement zu 38,6% größtenteils oder vollständig traditionell organisiert ist, ist dies bei umweltge-

5. Auswertung

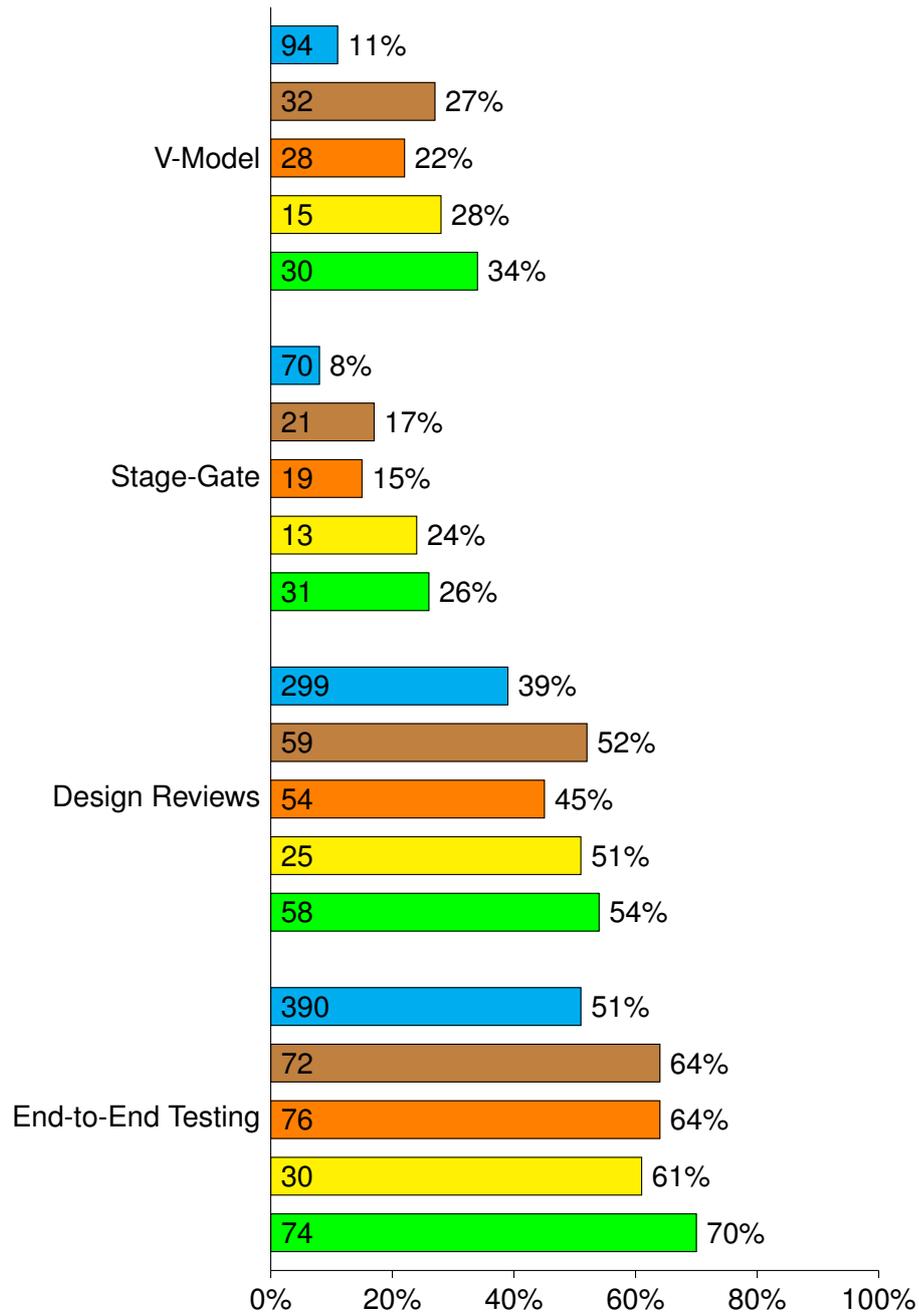


Abbildung 5.12.: Vergleich des Anteils an Projekten, die verschiedene Methoden und Praktiken häufig oder immer nutzen. Blaue Balken: Durchschnitt aller Projekte, gelbe Balken: Umweltgefährdende Projekte, grüne Balken: lebensgefährdende Projekte, braune Balken: kompletter Systemverlust

fährdenden Projekten zu 57,8% der Fall.

Trotz dessen ist Agilität auch bei kritischen Projekten weit verbreitet, beispielsweise

5.8. Reputationsschädigend und Geschäftsschädigend

	Projektmanagement		Risikomanagement	
	Traditionell	Agil	Traditionell	Agil
Allgemein	158 (28%)	244 (43,1%)	218 (38,6%)	132 (23,3%)
Systemverlust	35 (36,9%)	30 (31,5%)	48 (50,1%)	18 (18,8%)
Strafrechtliche Folgen	37 (36,6%)	37 (36,5%)	50 (49,5%)	22 (21,7%)
Umweltgefährdend	19 (52,8%)	12 (33,2%)	21 (57,8%)	5 (13,9%)
Lebensgefährdend	35 (42,1%)	26 (31,3%)	46 (54,4%)	13 (15,6%)

Abbildung 5.13.: Anzahl an Projekten bei denen Projekt- und Risikomanagement hauptsächlich oder vollständig agil beziehungsweise traditionell organisiert sind.

ist das Projektmanagement von über 30% aller kritischen Projekte überwiegend oder vollständig agil organisiert. Außerdem werden einige agile Methoden nahezu unabhängig von der Kritikalität des Projektes betrieben. Zum Beispiel wird Scrum für die Entwicklung von 54,7% der lebensgefährdenden und 49,1% der umweltgefährdenden Software verwendet.

5.8. Reputationsschädigend und Geschäftsschädigend

Möglicher Schaden an dem Geschäft eines Unternehmens ist die häufigste Folge bei Ausfall oder Fehler der Software. Von den 935 Teilnehmern der HELENA Studie haben 673 (72%) die Kritikalitätsfrage mit "*...das Geschäft des Unternehmens beeinträchtigen*" beantwortet. Das liegt vor allem daran, dass die meisten anderen Folgen von Softwarefehlern zusätzlich auch zu Geschäftsschäden führen (siehe Abbildung 5.1). Dies kann zum Beispiel geschehen, wenn eine Anklage in einem strafrechtlichen Verfahren öffentlich wird. Gleiches gilt für Projekte, die Reputationsverlust verursachen können. Insgesamt haben 654 (70%) Befragte die Antwortmöglichkeit "*...die Reputation des Unternehmens beeinträchtigen*" gewählt. Da, wie in Abbildung 5.1 zu sehen, über 80% aller Projekte, die zu Reputationsschäden führen können, auch Schäden am Geschäft eines Unternehmens verursachen können, werden diese beiden Kritikalitätskategorien gemeinsam behandelt. Beide gehören der Kategorie Business-Critical [5] an und haben damit eine geringere Kritikalität. Dies und die Tatsache, dass ein sehr großer Teil der Projekte in eine oder beide dieser Kategorien fällt, führt dazu, dass es kaum signifikante Unterschiede zwischen dem Durchschnitt aller Projekte und denen in den Kategorien Reputationsschädigend beziehungsweise Geschäftsschädigend in den Daten gibt. Zwar liefert der Chi-Quadrat Test für einige der Variablen eine Wahrscheinlichkeit auf statistische Abhängigkeit von über 99%, allerdings fällt bei genauer Betrachtung dieser auf, dass sie von den beiden Kritikalitätskategorien nicht oder kaum beeinflusst werden. Abbildung 5.14 zeigt dies an dem Beispiel des Anteils der Projekte, die Standards für die Code Entwicklung nutzen. Wie zu erkennen ist, unterscheiden sich die Anteile an Antworten in den einzelnen Kategorien maximal um 3 Prozentpunkte, obwohl die in dem Beispiel gezeigte Variable nach dem Chi-Quadrat Test die größte Wahrscheinlichkeit auf statistische Abhängigkeit von beiden Kritikalitätskategorien hat.

5. Auswertung

tätsklassen hat. Die Abweichung vom Durchschnitt für die anderen Variablen fällt noch geringer aus.

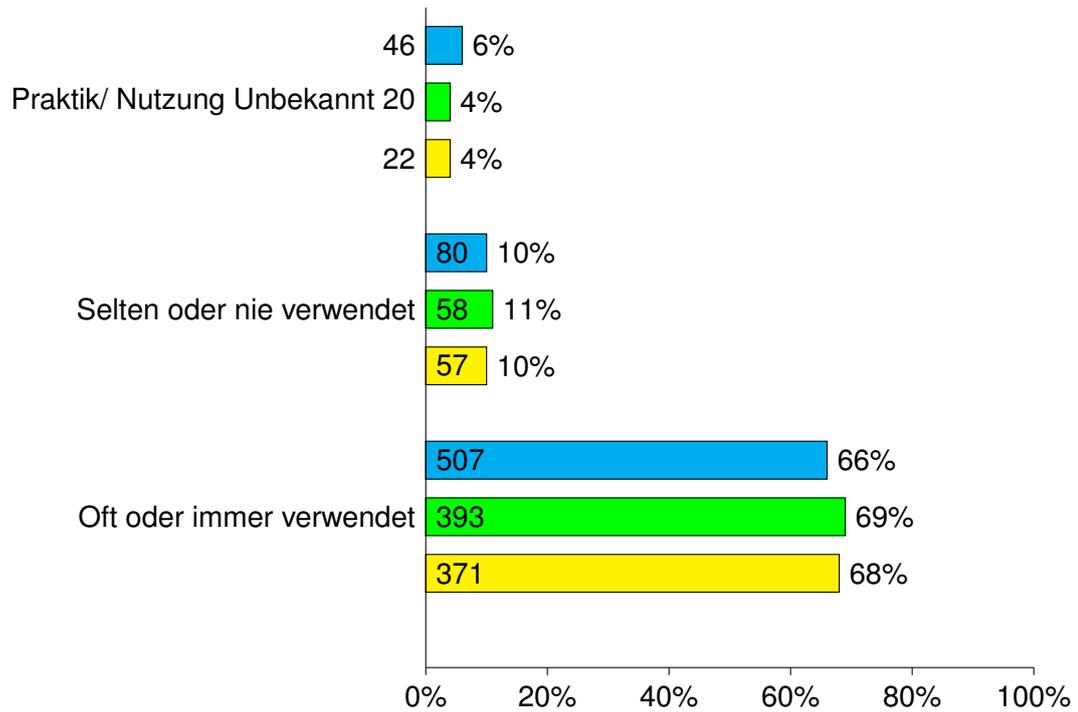


Abbildung 5.14.: Anteil an Projekten, die Standards für die Codeentwicklung verwenden. Blaue Balken: Durchschnitt aller Projekte, gelbe Balken: reputationsschädigende Projekte, grüne Balken: geschäftsschädigende Projekte

6. Diskussion

In diesem Kapitel werden die drei Forschungsfragen mit Hilfe der Ergebnisse aus dem vorherigen Kapitel beantwortet. Dies wird anschließend genutzt um eine Empfehlung für den Aufbau kritischer Projekte zu geben. Außerdem werden Einschränkungen für die Gültigkeit der Ergebnisse dieser Arbeit erläutert.

6.1. Beantwortung der Forschungsfragen

Wie wird der Grad der Agilität von Projekten durch deren Kritikalität beeinflusst?

Im letzten Kapitel wurde gezeigt, dass Unternehmen für Projekte, die bei Ausfall entweder lebens- oder umweltgefährdend sind oder zu strafrechtlichen Konsequenzen oder komplettem Systemverlust führen, bestimmte Methoden und Praktiken häufiger verwenden. Zu diesen zählen besonders das V-Modell und das Stage-Gate Modell. Wie in Abbildung 5.12 zu sehen ist, werden diese von Unternehmen für kritische Projekte wesentlich häufiger verwendet als durchschnittlich. Insbesondere wird das V-Modell bei 34% und das Stage-Gate Modell bei 26% der lebensgefährdenden Projekte verwendet, während dies nur bei 11% beziehungsweise 8% aller Projekte der Fall ist. Weiterhin werden Projekt-, Qualitäts- und Risikomanagement bei Projekten dieser vier Kategorien öfter traditionell und seltener agil betrieben (siehe Abbildung 5.13). Daraus folgt, dass Projekte mit hoher Kritikalität einen geringeren Grad an Agilität haben, als weniger kritische Projekte. Wie bereits erwähnt bedeutet das allerdings nicht, dass sie komplett traditionell entwickelt werden. Vor allem Scrum ist auch bei kritischen Projekten weit verbreitet.

Welche Methoden und Praktiken wurden für kritische Projekte besonders oft verwendet?

Neben traditionellen Methoden wie dem V-Modell und dem Stage-Gate Modell wurden auch einiger aufwändige Praktiken vermehrt bei Projekten mit hoher Kritikalität eingesetzt. Dazu gehören für alle Kritikalitätskategorien Design-Reviews und End-to-End Testing. Bei missionskritischen Projekten werden, wie in Abbildung 5.7 zu sehen, vermehrt automatisierte Verfahren zur Code Generierung und für Unit Tests eingesetzt. Für lebensgefährdende oder missionskritische Projekte werden zusätzlich häufig detaillierte Design Spezifikationen erstellt. Außerdem verwenden Unternehmen für Projekte, die menschliches Leben bedrohen können, destruktive Testverfahren.

Welche Unterschiede gibt es zwischen dem Entwicklungskontext von kritischen und nicht kritischen Projekten?

Wie bereits erwähnt, werden für kritische Projekte vermehrt traditionelle Methoden und aufwendige Praktiken verwendet. Außerdem werden einige Aktivitäten, wie das Projektmanagement eher traditionell organisiert. Weitere Unterschiede finden sich bei der Implementierung externer Standards und der Teilnahme an Zertifizierungsverfahren, beides wird bei kritischen Projekten häufiger als bei dem Durchschnitt aller Projekte angewendet (Abbildung 5.11). Da dies zu einem erhöhten Aufwand für die Entwicklung der Software führt, sind kritische Projekte größer als nicht kritische Projekte. Aufgrund der hohen Anzahl an Softwareprojekten in der Kategorie *sehr groß* kann allerdings keine genaue Angabe über den Größenunterschied gemacht werden. Anders als anfangs vermutet, haben Unternehmen, die kritische Software entwickeln, im Durchschnitt nicht mehr Mitarbeiter, als andere Unternehmen.

6.2. Empfehlungen

Bisher wurde in dieser Arbeit nur der aktuelle Stand in der Softwareindustrie dargestellt und keine Angaben über die Erfolgsquote für die unterschiedlichen Entwicklungsansätze gemacht, da der Datensatz keine Variable zum Messen von Erfolg enthält. In diesem Abschnitt werden nun einige Empfehlungen für den Aufbau zukünftiger Projekte gegeben, wobei diese soweit wie möglich anhand der Daten begründet werden. Diese Arbeit hat gezeigt, dass Unternehmen bei der Entwicklung von kritischen Projekten häufiger auf traditionelle Methoden, vor allem V-Modell und Stage-Gate, setzen. Allerdings wird die Mehrheit dieser Projekte heute ebenfalls agil entwickelt. Davon ausgehend, dass die Erfolgswahrscheinlichkeit bei beiden Ansätzen etwa gleich hoch ist, da sich agile Entwicklungsmethoden sonst bei kritischen Projekten nicht durchgesetzt hätten, wirft dies die Frage auf, wie der Unterschied zwischen der Verwendung von Methoden und Praktiken bei kritischen und nicht kritischen Projekten entsteht. Da die agile Softwareentwicklung erst nach der traditionellen entstanden ist, mussten Unternehmen entscheiden, ob sie die alten bereits bewährten Methoden durch neue ersetzen. Die bei Ausfall von kritischer Software entstehenden hohen Kosten könnten dazu geführt haben, dass die Unternehmen, die häufig beispielsweise lebensgefährdende Software entwickeln, deshalb öfter die Entscheidung getroffen haben das Risiko einer Umstellung auf neue Prozesse nicht einzugehen. Deshalb und aufgrund der geringen Nutzung außerhalb von kritischen Projekten, wird empfohlen, V-Modell und Stage-Gate Modell auch nicht für die Entwicklung kritischer Projekte zu nutzen, und an deren Stelle auf agile Prozesse zu setzen. Um den besonderen Ansprüchen im Bezug auf die Ausfallwahrscheinlichkeit gerecht zu werden, wird stattdessen empfohlen vermehrt Verfahren wie das End-to-End Testing oder Destructive Testing zu verwenden. Wie die Ergebnisse dieser Arbeit zeigen, ist dies bei den meisten kritischen Projekten bereits der Fall.

6.3. Einschränkung der Gültigkeit

Bei der Analyse des HELENA-Datensatzes wurde alle Freitextfelder ignoriert, daraus entstehen einige Einschränkungen für die Gültigkeit der Ergebnisse. Unter Anderem war es den Befragten auch möglich auf die Kritikalitätsfrage mit einem beliebigen Text zu Antworten, um so eventuell in den Antwortmöglichkeiten fehlende Folgen von Softwarefehlern abzudecken. Deshalb fehlen in dieser Arbeit zum Beispiel Aussagen zu Software, die bei Ausfall zum Verlust von sensiblen Daten führt. Außerdem standen Freitextfelder zur Auflistung von Methoden und Praktiken, die nicht in den vorgegebenen Antwortmöglichkeiten enthalten waren, zur Verfügung. Über einen Einfluss von Kritikalität auf diese Methoden und Praktiken kann in dieser Arbeit keine Aussage getroffen werden.

Weitere Einschränkungen entstehen durch die, für bestimmte Variablenkombinationen, geringe Anzahl an Datenpunkten. Vor allem die Kategorie für umweltschädliche Projekte ist davon betroffen, da für diese insgesamt nur 57 Antworten gegeben wurden. Zum Beispiel haben 49 Befragte, die an umweltgefährdenden Projekten arbeiten, auf Fragen nach den verwendeten Praktiken geantwortet. Weil für diese Fragen jeweils sieben, sich gegenseitig ausschließende, Antwortmöglichkeiten zur Verfügung standen, haben durchschnittlich nur sieben Befragte eine der Antworten gegeben. Aus diesem Grund sind solche Variablen besonders anfällig für zufällige Schwankungen. Da die Ergebnisse dieser Arbeit auf einem Datensatz beruhen, der durch eine Umfrage entstanden ist, können einzelne Datenpunkte durch Fehler während des Ausfüllens verfälscht werden. So kann zum Beispiel eine Frage von den Teilnehmern nicht richtig interpretiert oder eine Antwortmöglichkeit unbeabsichtigt angekreuzt worden sein.

6. *Diskussion*

7. Fazit und Ausblick

7.1. Fazit

Die Auswertung hat gezeigt, dass für Projekte, die einer von vier der neun Kritikalitätsklassen angehören, bestimmte Variablen des Entwicklungskontextes von denen bei Projekten in den anderen Kategorien abweichen. Davon sind besonders Safety-Critical, also lebens- und umweltgefährdende, Projekte betroffen. Auch bei Projekten, die bei Ausfall zu komplettem Systemverlust oder strafrechtlichen Konsequenzen führen können, wurde diese Abweichung festgestellt. Für die anderen Kritikalitätskategorien konnten kaum Abweichungen von dem Durchschnitt aller Projekte festgestellt werden. Eine Erklärung dafür ist, dass Software von hoher Kritikalität auch andere Kritikalitätskriterien erfüllt. Zum Beispiel kann Software, die bei Ausfall zu komplettem Systemverlust führt, zusätzlich zum Beispiel finanzielle Kosten durch das Ersetzen des Systems verursachen. Wie gezeigt wurde, hängen die fünf Kategorien mit niedriger Kritikalität oft von denen mit hoher Kritikalität ab.

Der größte dieser Unterschiede ist die häufige Implementierung von externen Standards. Zum Beispiel werden diese bei 68% der umweltgefährdenden Projekte verwendet, während dies nur 37% aller Softwareprojekte der Fall ist. Für viele der verwendeten Standards existieren Zertifizierungsverfahren, die Teilnahme daran ist bei kritischen Projekten ebenfalls höher. Weiterhin unterscheidet sich der Anteil an Projekten, die bestimmte Methoden und Praktiken verwenden. Dabei werden das V-Modell und das Stage-Gate Modell besonders häufig bei der Entwicklung kritischer Projekte verwendet. Außerdem sind die Praktiken End-to-End Testing und Design Reviews weiter verbreitet. Die Ergebnisse zeigen allerdings auch, dass agile Methoden auch von vielen Unternehmen bei der Entwicklung kritischer Software eingesetzt wird.

Letztlich sind kritische Projekte aufwändiger als durchschnittliche Projekte. Der genaue Größenunterschied konnte aus den Daten allerdings nicht entnommen werden.

7.2. Ausblick

Die Ergebnisse dieser Arbeit lassen vor allem eine Frage offen: *Welche Methoden und Praktiken sind bei der Entwicklung kritischer Projekte erfolgreich und welche nicht?*

Da der verwendete Datensatz keine Variablen zum Messen des Erfolgs von Softwareprojekten enthält, konnte diese Frage nicht beantwortet werden. Es bietet sich also an, in einer weiteren Arbeit auf diesen Aspekt einzugehen. Mit Hilfe einer Umfrage kann ein weiterer Datensatz erstellt werden, der ein Maß für den Erfolg von einzelnen Softwareprojekten enthält. Außerdem sollten, ähnlich der HELENA-Daten, auch

7. Fazit und Ausblick

Informationen über die genutzten Methoden und Praktiken, den Grad der Agilität von Aktivitäten während der Entwicklung und die Kritikalität des Projektes enthalten sein. Um eine detailliertere Beschreibung von den Auswirkungen der Kritikalität auf den Entwicklungskontext zu erhalten, ist es sinnvoll neben der Art der Folgen bei Fehlern der Software auch noch deren Schwere zu bestimmen. Beispielsweise könnten zwei Projekte von dem selben Unternehmen unterschiedlich entwickelt werden, wenn eines zu einem vielfach höheren finanziellen Schaden führen kann, als das andere. Auf diese Art, kann auch der Unterschied bei der Entwicklung von kritischen Projekten in der selben Kategorie untersucht werden.

A. Korrelationskoeffizienten

	D006	D006_01	D006_02	D006_03	D006_04	D006_05	D006_10	D006_06	D006_07	D006_08	D006_09
D001	0,07	0,09	0,06	0,11	-0,03	0,03	0,07	0,07	-0,04	-0,01	0,03
D002	0,05	-0,03	0,05	0,06	0,04	0,04	-0,05	0	0,04	0,04	0,01
D002_01	-0,04	-0,09	-0,06	-0,08	0,05	0,01	-0,15	-0,05	0,06	0,05	0,02
D002_02	0,01	-0,1	0,02	0,03	0	0,02	-0,03	-0,02	0,02	0,08	-0,02
D002_03	0,17	0,21	0,25	0,12	0,01	0	0,06	0,12	0,08	0,04	-0,02
D002_04	0,01	-0,02	0	-0,01	0,06	0,01	-0,03	-0,04	-0,02	0,03	0,04
D002_05	0,07	0,02	0,02	0,09	0,06	0,1	0	0	0,01	-0,01	-0,06
D002_06	0	-0,06	-0,01	0,01	0,01	0,02	-0,03	-0,02	0,04	0,01	0,01
D002_07	-0,1	0,05	0,03	-0,03	-0,12	-0,07	-0,08	-0,02	-0,07	-0,1	0,06
D002_08	0,07	-0,06	-0,03	0,1	0,06	0,04	0,12	0,03	-0,01	0,02	0,01
D005	0,1	-0,03	0,09	0,08	0,02	0,03	0,03	0,05	0,1	0,07	-0,02
D005_01	0,14	0,2	0,11	0,03	0,01	0,1	0,16	0,26	-0,01	-0,04	-0,04
D005_02	0,08	0,13	0,1	-0,01	0,01	0,03	0,05	0,11	0,07	-0,06	0
D005_03	0,01	-0,08	-0,01	0,02	-0,02	-0,02	-0,03	0,01	0,06	0,1	-0,02
D005_04	0,07	0,07	0,18	0	0,02	0,01	0,04	0,03	0,08	-0,04	-0,03
D005_20	0,07	0,09	0,04	0,02	-0,01	0,06	0	0,12	0,07	0	-0,04
D005_05	0,1	-0,16	-0,01	0,03	0,2	0,18	0,09	-0,08	-0,01	0,09	-0,03
D005_06	-0,03	-0,04	-0,04	0	0,01	-0,03	-0,04	-0,04	0,03	-0,01	-0,03
D005_07	0,04	-0,02	0,05	0,07	-0,03	-0,02	-0,02	0,04	0,06	0,06	-0,03
D005_08	0,11	-0,01	0,07	0,06	0,04	0,06	0,05	0,08	0,11	0,08	-0,05
D005_19	0,02	-0,04	0	0,04	0,02	0,02	-0,05	-0,02	0,03	0,03	0,04
D005_09	0,01	0,32	-0,04	-0,04	-0,09	-0,03	0,03	-0,02	0,02	-0,04	-0,01
D005_10	0	-0,08	-0,01	0,02	0,02	-0,07	-0,05	-0,03	0,09	0,07	-0,04
D005_11	0	0,02	0,11	0,01	0	-0,09	-0,06	0,03	0,01	0,02	0,01
D005_12	-0,03	-0,06	0,03	0,05	0,02	-0,04	-0,04	-0,01	-0,06	0,01	-0,05
D005_13	0,03	-0,03	-0,01	0,03	-0,07	0,13	0,06	-0,01	0	-0,01	0,03
D005_14	-0,01	0,07	0,06	0,06	-0,07	-0,04	-0,03	-0,03	0,01	-0,06	-0,02
D005_15	-0,02	-0,05	0,16	0,05	0	-0,06	-0,02	-0,02	-0,03	-0,11	0
D005_16	0,02	-0,03	0,05	0,03	-0,02	0,01	0,05	-0,03	-0,02	-0,02	0,09
D005_17	0	-0,16	-0,06	0,03	0,01	0	-0,06	-0,04	0,09	0,09	-0,04
D005_18	-0,04	-0,04	-0,04	-0,03	-0,04	-0,1	-0,02	0,02	0	0,01	0,15
D006	1	0,37	0,52	0,48	0,53	0,64	0,56	0,42	0,55	0,48	0,07
D006_01	0,37	1	0,24	0,05	-0,02	0,15	0,25	0,33	0,07	-0,06	0,02
D006_02	0,52	0,24	1	0,23	0,18	0,17	0,18	0,26	0,16	0,1	0,04
D006_03	0,48	0,05	0,23	1	0,07	0,18	0,18	0,12	0,17	0,05	0
D006_04	0,53	-0,02	0,18	0,07	1	0,31	0,15	0,09	0,22	0,24	-0,04
D006_05	0,64	0,15	0,17	0,18	0,31	1	0,44	0,14	0,22	0,21	-0,03
D006_10	0,56	0,25	0,18	0,18	0,15	0,44	1	0,25	0,13	0,12	0,03
D006_06	0,42	0,33	0,26	0,12	0,09	0,14	0,25	1	0,11	0,08	0,02
D006_07	0,55	0,07	0,16	0,17	0,22	0,22	0,13	0,11	1	0,37	-0,02
D006_08	0,48	-0,06	0,1	0,05	0,24	0,21	0,12	0,08	0,37	1	-0,08
D006_09	0,07	0,02	0,04	0	-0,04	-0,03	0,03	0,02	-0,02	-0,08	1
D009	0,23	0,11	0,1	0,09	0,11	0,13	0,11	0,05	0,17	0,16	-0,02
PU09_01	0,19	0,13	0,13	0,15	0,09	0,11	0,11	0,04	0,07	0,02	0,05
PU09_02	0,03	-0,05	0,06	-0,04	0,08	0,04	0,06	0	-0,01	-0,01	-0,02
PU09_03	0,1	-0,07	-0,01	0,04	0,07	0,06	0,04	-0,01	0,11	0,13	0,04
PU09_04	0,1	0,06	-0,01	0,03	0,08	0,1	0,05	0,06	0,05	0,03	-0,01
PU09_05	0	-0,05	0,03	-0,03	0	0,04	0,08	-0,02	-0,01	-0,03	-0,02
PU09_06	0,04	-0,04	-0,03	-0,03	0,05	0,03	0	-0,01	0,09	0,09	-0,02
PU09_07	0,07	0,1	0,02	0	-0,03	0,01	0,02	0,09	0,15	0,04	-0,06
PU09_08	0,14	0,09	0,04	0,03	0,06	0,08	0,02	0,02	0,17	0,13	-0,03
PU09_09	0,03	-0,05	-0,05	-0,02	0,11	0,06	-0,01	0,01	0	0,06	-0,03
PU09_10	0,05	0,01	0,02	0	0,05	0,06	0,02	0,04	0,02	0,02	-0,04
PU09_11	0	0,05	0	-0,04	0,03	-0,01	-0,06	0	0,02	0,03	-0,05
PU09_12	0,1	0,14	0,13	0,01	-0,02	0,08	0,05	0,12	0,06	0	0

A. Korrelationskoeffizienten

	D006	D006_01	D006_02	D006_03	D006_04	D006_05	D006_10	D006_06	D006_07	D006_08	D006_09
PU09_13	-0,06	-0,02	0	-0,04	-0,08	0,03	0,03	0	-0,1	-0,07	-0,07
PU09_14	0	0,1	0,07	-0,04	-0,01	0,01	0,06	-0,01	-0,07	-0,04	-0,02
PU09_15	0,14	0,22	0,16	0,03	0,03	0,07	0,12	0,12	0,02	-0,04	0,02
PU09_16	0,08	0,01	0,02	0,05	0,04	0,1	0,07	-0,04	0,08	0	-0,04
PU09_17	0,04	-0,03	-0,02	0,02	0,06	0,06	0,02	0,03	0	0,03	0,02
PU09_18	0,08	0,01	0,04	-0,06	0,08	0,07	0,1	0,08	0,04	0,05	-0,02
PU09_19	0,08	-0,03	-0,03	0,01	0,08	0,1	0	-0,04	0,1	0,13	-0,06
PU09_20	0,05	-0,03	0,04	-0,02	0,07	0,11	0,02	0	0,02	0	-0,03
PU09_21	0,07	0,06	0,07	-0,02	0,01	0,05	0,06	0,02	0,03	0,05	0
PU09_22	0,03	0,07	0,08	0	0	0,04	0,08	0,01	-0,03	-0,04	-0,03
PU09_23	0,02	0,06	0,07	-0,08	-0,01	0,06	0,04	0,02	-0,05	0,02	0,01
PU09_24	0,24	0,32	0,26	0,12	-0,02	0,13	0,16	0,21	0,07	0,02	-0,04
PU10_01	0,14	0,07	0,09	0,06	0,07	0,06	0,03	0,04	0,14	0,08	0,02
PU10_02	0,12	0,06	0,1	0,03	0,06	0,1	0,03	0,09	0,08	0,03	0,03
PU10_03	0,05	0,02	0,08	-0,05	0,03	0,02	0,02	0,02	-0,01	0,1	0,04
PU10_04	0,16	0,1	0,09	0,04	0,09	0,1	-0,01	0,04	0,17	0,07	0,03
PU10_05	0,06	-0,02	-0,02	0,02	0,09	0,07	-0,05	-0,04	0,06	0,08	-0,01
PU10_06	0,12	0,09	0,01	-0,01	0,12	0,06	0,04	0,05	0,11	0,09	-0,04
PU10_07	0,2	0,06	0,1	0,09	0,1	0,11	0,02	0,02	0,17	0,18	0,01
PU10_08	0,18	0,04	0,08	0,08	0,12	0,12	0,05	0,02	0,16	0,12	-0,04
PU10_09	0,08	-0,03	0,05	0,01	0,11	0,04	-0,03	-0,01	0,09	0,05	0,08
PU10_10	-0,01	-0,05	0	-0,05	0	0,03	-0,03	-0,06	0,04	0,04	-0,02
PU10_11	0,07	0,03	0,01	0,04	0,05	0,05	-0,03	0	0,09	0,03	0,04
PU10_12	0,11	0,02	0	0	0,15	0,1	0,01	0	0,13	0,07	-0,02
PU10_13	0,11	0,03	0	0,02	0,1	0,07	0,02	0	0,13	0,08	-0,02
PU10_14	0,21	0,13	0,14	0,13	0,12	0,07	0,09	0,05	0,15	0,09	0,01
PU10_15	0,18	0,1	0,12	0,04	0,1	0,13	0,14	0,04	0,06	0,07	-0,01
PU10_16	0,14	0,14	0,08	-0,02	0,08	0,03	0,12	0,03	0,11	0,11	0,02
PU10_18	0,23	0,17	0,14	0,13	0,1	0,13	0,14	0,06	0,1	0,07	-0,01
PU10_19	0,1	-0,01	-0,02	0	0,11	0,09	0,01	-0,01	0,13	0,12	-0,06
PU10_20	0,03	-0,03	-0,01	-0,07	0,06	0,05	0,04	0,03	-0,02	0,08	0,01
PU10_21	0,12	0,04	0,08	0	0,05	0,1	0,07	0,06	0,08	0,09	0
PU10_22	0,12	0,02	0,02	-0,03	0,12	0,09	0,01	0	0,11	0,18	-0,04
PU10_23	0,1	-0,04	-0,02	0	0,12	0,1	0	-0,03	0,13	0,17	-0,06
PU10_17	0,06	-0,04	-0,02	-0,04	0,14	0,1	-0,01	0,01	0,01	0,13	-0,06
PU10_24	0,16	0,09	0,12	0,06	0,08	0,1	0,1	0,1	0,07	0,03	-0,01
PU10_25	0,18	0,03	0,11	0,01	0,16	0,1	0,08	0,07	0,14	0,12	0,02
PU10_26	0,09	-0,06	-0,02	0	0,1	0,09	0,01	-0,02	0,09	0,16	-0,01
PU10_27	0,15	0,05	0,05	0,07	0,1	0,08	0,06	0,06	0,09	0,09	0,01
PU10_28	0,08	0	-0,02	-0,03	0,08	0,07	-0,04	0,01	0,14	0,1	0
PU10_29	0,18	0,06	0,01	0,08	0,18	0,14	0,07	0,07	0,1	0,1	-0,01
PU10_30	0,11	0,04	-0,02	-0,02	0,11	0,11	0,02	0,01	0,1	0,13	-0,04
PU10_31	0,06	0,04	0,02	-0,02	0,06	0,06	0,04	-0,03	0,01	0,07	-0,03
PU10_32	0,18	0,08	0,08	0	0,1	0,16	0,11	0,05	0,14	0,1	-0,02
PU10_33	0,07	-0,04	-0,01	0,01	0,09	0,05	-0,01	0,04	0,09	0,08	-0,02
PU10_36	0,11	0,05	0,06	0,01	0,08	0,09	0,06	0,05	0,03	0,06	0
PU10_34	0,09	-0,03	-0,01	-0,01	0,13	0,06	0,03	-0,07	0,15	0,09	-0,04
PU10_35	0,08	0	-0,02	-0,05	0,09	0,08	0,02	-0,01	0,12	0,11	-0,04
PS01	0,27	0,25	0,15	0,14	0,08	0,16	0,2	0,17	0,12	0,01	0,02

B. Ergebnisse des Chi-Quadrat Tests

	D006_01	D006_02	D006_03	D006_04	D006_05	D006_10	D006_06	D006_07	D006_08	D006_09
D001	0,0561	0,0107	0,00152	0,201	0,719	0,0486	0,234	0,0538	0,239	0,534
D002	0,925	0,405	0,203	0,559	0,0433	0,218	0,266	0,00475	0,057	0,861
D002_01	0,00486	0,0392	0,0347	0,0795	0,732	0,0454	0,19	0,073	0,0943	0,739
D002_02	0,00467	0,837	0,99	1	0,998	0,627	0,655	0,383	0,0846	0,884
D002_03	7,64E-07	9,83E-13	4,90E-05	0,647	0,841	0,307	0,0043	0,113	0,727	0,758
D002_04	0,911	0,96	0,748	0,126	0,277	0,686	0,682	0,509	0,454	0,898
D002_05	1	1	0,0135	0,164	0,00056	0,318	1	0,308	0,709	0,106
D002_06	0,189	0,874	0,562	0,701	0,393	0,79	0,883	0,142	0,324	1
D002_07	0,381	0,228	0,83	0,0158	0,401	0,138	0,682	0,511	0,031	0,114
D002_08	0,454	0,245	0,0122	0,258	0,352	0,0305	0,896	0,357	0,0133	0,755
D005	0,0576	0,17	0,124	0,154	0,108	0,188	0,125	0,00233	0,0504	0,419
D005_01	8,71E-08	0,124	0,571	0,995	0,044	0,00149	7,25E-08	0,36	0,043	0,76
D005_02	0,0177	0,0443	0,972	1	0,38	0,528	0,0502	0,117	0,385	1
D005_03	0,118	0,395	0,297	0,853	0,594	0,721	0,614	0,0586	0,00246	0,603
D005_04	0,187	0,000702	0,457	1	0,771	1	1	0,557	0,128	0,248
D005_20	0,0134	0,505	0,968	1	0,611	0,998	0,00041	0,466	0,57	1
D005_05	1,24E-06	0,431	0,184	1,29E-10	1,12E-08	0,00147	0,0568	0,291	0,000278	0,623
D005_06	0,341	0,755	0,545	0,707	0,245	0,758	1	1	0,292	1
D005_07	0,411	0,268	0,0546	0,349	0,2	0,788	0,241	0,565	0,702	1
D005_08	0,507	0,158	0,0658	0,37	0,38	0,151	0,18	0,0164	0,175	0,715
D005_19	0,211	0,894	0,431	0,275	0,655	0,141	1	0,485	0,821	0,305
D005_09	2,23E-19	0,669	0,145	0,0191	0,58	0,272	0,629	0,14	0,782	0,748
D005_10	0,0178	0,886	0,716	0,474	0,133	0,286	0,952	0,00192	0,0155	0,294
D005_11	1	0,00211	0,633	0,939	0,0057	0,0519	1	0,935	0,568	1
D005_12	0,0177	0,941	0,213	0,23	0,913	0,318	0,529	0,194	0,923	0,163
D005_13	0,636	0,541	0,78	0,0316	0,0135	0,223	1	0,924	0,514	0,384
D005_14	0,0127	0,00364	0,155	0,0996	0,311	0,342	1	0,958	0,0854	1
D005_15	0,353	0,000268	0,797	1	0,106	0,468	0,507	0,642	0,00839	0,339
D005_16	0,204	0,069	0,333	0,718	1	0,72	0,571	0,953	1	0,0219
D005_17	2,77E-05	0,179	0,473	0,507	0,91	0,142	0,336	0,000176	0,0122	0,242
D005_18	0,215	0,174	0,553	0,204	0,00786	0,414	0,396	0,729	0,939	2,46E-06
D006_01	1,66E-203	3,70E-12	0,0785	0,739	0,00033	4,46E-11	4,39E-21	0,0472	0,211	0,786
D006_02	3,70E-12	1,49E-203	1,34E-10	6,13E-05	4,23E-06	4,67E-07	4,37E-13	1,83E-05	0,00698	0,579
D006_03	0,0785	1,34E-10	1,79E-204	0,00892	8,06E-07	3,61E-06	0,0139	1,41E-07	0,0327	0,444
D006_04	0,739	6,13E-05	0,00892	1,95E-204	1,82E-22	5,61E-08	0,0233	1,37E-16	1,07E-15	0,0851
D006_05	0,00033	4,23E-06	8,06E-07	1,82E-22	2,00E-204	1,48E-36	0,00013	1,77E-12	7,48E-11	0,2
D006_10	4,46E-11	4,67E-07	3,61E-06	5,61E-08	1,48E-36	1,14E-203	2,05E-09	4,96E-06	4,95E-05	0,603
D006_06	4,39E-21	4,37E-13	0,0139	0,0233	0,00013	2,05E-09	1,07E-92	0,0489	0,627	0,686
D006_07	0,0472	1,83E-05	1,41E-07	1,37E-16	1,77E-12	4,96E-06	0,0489	2,88E-204	1,16E-37	0,118
D006_08	0,211	0,00698	0,0327	1,07E-15	7,48E-11	4,95E-05	0,627	1,16E-37	2,61E-204	0,00303
D006_09	0,786	0,579	0,444	0,0851	0,2	0,603	0,686	0,118	0,00303	3,01E-54
D009	0,00011	0,0206	0,013	5,36E-06	0,000881	0,000299	0,919	5,17E-12	1,15E-12	0,661
PU09_01	0,107	0,000119	0,0149	0,000262	0,018	0,0171	0,011	4,38E-06	0,000308	0,38
PU09_02	0,579	0,456	0,149	0,394	0,779	0,379	0,274	0,129	0,0649	0,907
PU09_03	0,248	0,991	0,54	0,111	0,0941	0,065	0,391	0,0119	0,00272	0,386
PU09_04	0,0511	0,386	0,453	0,304	0,0272	0,175	0,194	0,205	0,179	0,745
PU09_05	0,00716	0,387	0,132	0,622	0,99	0,125	0,0272	0,0316	0,021	0,848
PU09_06	0,692	0,784	0,714	0,368	0,792	0,111	0,953	9,46E-06	0,0157	0,193
PU09_07	0,0246	0,823	0,472	0,0393	0,488	0,633	0,39	0,0129	0,741	0,934
PU09_08	0,141	0,0647	0,245	0,0236	0,0501	0,00252	0,389	2,90E-07	0,000875	0,209
PU09_09	0,619	0,0243	0,031	0,0573	0,291	0,00132	0,957	0,561	6,00E-04	0,534
PU09_10	0,0909	0,0859	0,634	0,186	0,242	0,0237	0,0171	0,866	0,594	0,904
PU09_11	0,304	0,605	0,756	0,302	0,738	0,684	0,714	0,0613	0,273	0,895
PU09_12	0,00908	0,0057	0,856	0,925	0,581	0,209	0,0245	0,647	0,442	0,778

B. Ergebnisse des Chi-Quadrat Tests

	D006_01	D006_02	D006_03	D006_04	D006_05	D006_10	D006_06	D006_07	D006_08	D006_09
PU09_13	0,446	0,0263	0,371	0,0255	0,616	0,403	0,887	0,0209	0,00459	0,52
PU09_14	0,0489	0,0209	0,461	0,367	0,453	0,366	0,432	0,066	0,431	0,568
PU09_15	1,17E-08	0,000574	0,502	0,489	0,513	0,0379	0,00278	0,335	0,0602	0,724
PU09_16	0,401	0,45	0,78	0,959	0,00734	0,164	0,498	0,65	0,0242	0,984
PU09_17	0,245	0,575	0,0889	0,604	0,437	0,448	0,791	0,158	0,281	0,91
PU09_18	0,282	0,312	0,174	0,45	0,442	0,276	0,068	0,583	0,629	0,832
PU09_19	0,441	0,312	0,51	0,161	0,176	0,315	0,529	2,82E-05	0,0173	0,154
PU09_20	0,928	0,00974	0,243	0,586	0,627	0,927	0,615	0,754	0,787	0,93
PU09_21	0,38	0,0379	0,49	0,335	0,144	0,175	0,728	0,297	0,285	0,43
PU09_22	0,0254	0,107	0,122	0,383	0,461	0,34	0,989	0,301	0,152	0,618
PU09_23	0,419	0,0275	0,0753	0,249	0,391	0,133	0,0641	0,168	0,696	0,212
PU09_24	4,72E-21	6,10E-11	0,00627	0,89	0,00826	1,96E-05	0,000214	0,319	0,0632	0,526
PU10_01	0,12	0,0192	0,0738	0,126	0,612	0,285	0,323	0,00381	0,0753	0,542
PU10_02	0,578	0,00657	0,00897	0,576	0,0337	0,814	0,151	0,0414	0,0237	0,904
PU10_03	0,148	0,0101	0,24	0,415	0,882	0,731	0,175	0,31	0,0141	0,249
PU10_04	0,198	0,00246	0,011	0,278	0,135	0,141	0,244	5,29E-05	0,0612	0,797
PU10_05	0,314	0,864	0,388	0,104	0,32	0,592	0,572	0,0237	0,0424	0,577
PU10_06	0,274	0,375	0,38	0,0357	0,317	0,437	0,143	0,0228	0,371	0,449
PU10_07	0,0599	0,0517	0,0133	0,0147	0,00815	0,365	0,255	3,40E-05	7,74E-05	0,78
PU10_08	0,494	0,27	0,0213	0,00264	0,0396	0,455	0,971	2,16E-06	0,000119	0,37
PU10_09	0,272	0,199	0,615	0,0771	0,569	0,781	0,727	0,0582	0,165	0,29
PU10_10	0,512	0,592	0,516	0,595	0,175	0,599	0,507	0,0992	0,727	0,854
PU10_11	0,594	0,237	0,151	0,664	0,0276	0,382	0,317	0,0125	0,0156	0,633
PU10_12	0,384	0,0448	0,53	0,0012	0,0133	0,00325	0,533	0,0103	0,238	0,861
PU10_13	0,822	0,079	0,14	0,148	0,621	0,164	0,911	0,0181	0,201	0,702
PU10_14	0,000157	0,00829	0,000534	0,0429	0,736	0,459	0,34	0,000306	0,0043	0,509
PU10_15	0,0101	0,00102	0,0672	0,0225	0,0375	0,0196	0,208	0,494	0,363	0,0042
PU10_16	0,00782	0,144	0,192	0,365	0,424	0,0014	0,377	0,00609	0,0146	0,721
PU10_18	0,000144	0,00546	0,0045	0,00384	0,00748	0,0122	0,395	0,0275	0,227	0,343
PU10_19	0,591	0,98	0,246	0,141	0,477	0,445	0,71	0,0159	0,0206	0,605
PU10_20	0,957	0,383	0,058	0,736	0,37	0,414	0,691	0,609	0,57	0,371
PU10_21	0,535	0,107	0,751	0,505	0,0718	0,176	0,259	0,683	0,198	0,451
PU10_22	0,142	0,0307	0,363	0,0463	0,138	0,628	0,747	0,00892	0,000135	0,397
PU10_23	0,53	0,514	0,267	0,0303	0,127	0,673	0,689	0,000849	0,000238	0,304
PU10_17	0,661	0,194	0,176	0,00609	0,0286	0,871	0,286	0,753	0,0154	0,264
PU10_24	0,167	0,00115	0,142	0,532	0,131	0,113	0,0231	0,121	0,655	0,282
PU10_25	0,134	0,013	0,345	0,000568	0,174	0,096	0,389	0,00299	0,0701	0,64
PU10_26	0,468	0,502	0,592	0,0817	0,0183	0,936	0,359	0,00364	8,63E-05	0,755
PU10_27	0,294	0,16	0,537	0,00263	0,0435	0,525	0,614	0,0288	0,00113	0,979
PU10_28	0,933	0,306	0,252	0,0704	0,134	0,942	0,584	1,24E-05	0,00326	0,954
PU10_29	0,428	0,83	0,17	2,05E-06	0,00352	0,363	0,224	0,00701	0,00131	0,852
PU10_30	0,743	0,617	0,493	0,111	0,0125	0,274	0,463	0,238	0,0193	0,235
PU10_31	0,428	0,385	0,853	0,577	0,124	0,555	0,836	0,556	0,447	0,795
PU10_32	0,128	0,0527	0,276	0,241	0,00233	0,22	0,562	0,00584	0,0285	0,86
PU10_33	0,82	0,208	0,727	0,621	0,341	0,804	0,825	0,00115	0,302	0,608
PU10_36	0,0981	0,0665	0,392	0,455	0,122	0,176	0,561	0,0559	0,348	0,576
PU10_34	0,76	0,782	0,602	0,00356	0,6	0,964	0,257	0,00406	0,0218	0,393
PU10_35	0,402	0,318	0,0492	0,643	0,0155	0,631	0,368	0,0228	0,115	0,176
PS01	3,81E-11	0,000101	0,000215	0,0319	3,28E-05	1,49E-07	1,74E-05	0,00245	0,76	0,658
PU01	0,783	0,72	0,44	0,0595	0,252	0,511	0,0295	0,365	0,763	0,533
PU04	0,177	0,0401	0,0157	0,159	0,359	0,0193	0,619	0,617	0,914	0,0545
PU05_01	0,00601	0,124	0,156	0,398	0,466	0,349	0,0022	0,0738	0,148	0,804
PU05_02	0,00499	0,127	0,742	0,726	0,792	0,109	0,297	0,0588	0,0713	0,907
PU05_03	0,00529	0,0512	0,204	0,765	0,328	0,00478	0,00704	0,493	0,866	0,491

	D006_01	D006_02	D006_03	D006_04	D006_05	D006_10	D006_06	D006_07	D006_08	D006_09
PU05_04	0,0894	0,315	0,478	0,854	0,122	0,0154	0,361	0,553	0,233	0,211
PU05_05	0,46	0,0488	0,326	0,987	0,306	0,021	0,426	0,927	0,802	0,794
PU05_06	0,00356	0,167	0,68	0,253	0,69	0,421	0,175	0,183	0,00134	0,0961
PU05_07	0,0133	0,0232	0,173	0,0774	0,129	0,166	0,0435	0,181	0,00919	0,13
PU05_08	0,219	0,00513	0,468	0,962	0,408	0,51	0,416	0,154	0,307	0,464
PU05_09	0,329	0,057	0,689	0,741	0,211	0,0522	0,0769	0,643	0,355	0,0505
PU05_10	0,01	0,128	0,417	0,735	0,797	0,0486	0,00329	0,584	0,557	0,304
PU05_11	0,172	0,0906	0,108	0,963	0,455	0,364	0,559	0,405	0,56	0,388

B. Ergebnisse des Chi-Quadrat Tests

C. Inhalt der beiliegenden CD

Die dieser Arbeit beiliegende CD enthält folgende Dateien und Ordner:

LaTeX Ordner, der das Latex Projekt für diese Arbeit enthält

R Ordner, der den R Workspace enthält

HELENA Ordner, der die Rohdaten und die Beschreibungen der Variablen der HELENA-Studie enthält.

Bachelorarbeit.pdf PDF Version dieses Dokumentes

C. Inhalt der beiliegenden CD

Literaturverzeichnis

- [1] Website des IEC zu dem Standard IEC 61508,
www.iec.ch/functionalsafety/
(abgerufen am 16.11.2019)
- [2] Website der ISO zu dem Standard ISO 26262,
www.iso.org/standard/68383.html
(abgerufen am 16.11.2019)
- [3] Website der ISO zu dem Standard ISO 27001,
<https://www.iso.org/isoiec-27001-information-security.html>
(abgerufen am 16.11.2019)
- [4] Website der ISO zu der ISO 9000 Familie,
<https://www.iso.org/iso-9001-quality-management.html>
(abgerufen am 16.11.2019)
- [5] Hinchey, Mike/Coyle, Lorcan,
Evolving Critical Systems: a Research Agenda for Computer-Based Systems,
17th IEEE International Conference and Workshops on Engineering of Computer
Based Systems, S.430-435, 2010
- [6] Gallager, Robert G.
Stochastic Processes Theory for Applications,
Cambridge University Press, 2010
- [7] Dalgaard, Peter,
Introductory Statistics with R,
Springer Science+Business Media, 2010
- [8] Pearson, Karl,
*On the criterion that a given system of deviations from the probable in the case
of a correlated system of variables is such that it can be reasonably supposed to
have arisen from random sampling*,
The London, Edinburgh, and Dublin Philosophical Magazine and Journal of
Science, Volume 50, S.157-175, 1900
- [9] Fisher, Ronald A.,
On the Interpretation of χ^2 from Contingency Tables, and the Calculation of P,
Journal of the Royal Statistical Society, Volume 85, S. 87-94, 1922
- [10] Mukaka, M. M.,
A guide to appropriate use of correlation coefficient in medical research,
Malawi Medical Journal, Volume 24, S. 69-71, 2012

LITERATURVERZEICHNIS

- [11] Ge, Xiaocheng/Paige, Richard F./McDermid, John A.,
An Iterative Approach for Development of Safety-Critical Software and Safety Arguments,
2010 Agile Conference
- [12] Baker, Richard und Habli, Ibrahim,
An Empirical Evaluation of Mutation Testing for Improving the Test Quality of Safety-Critical Software,
IEEE Transactions on Software Engineering, Volume 39, 2013
- [13] Turk, Dan/ France, Robert/ Rumpe, Bernhard,
Limitations of Agile Software Processes,
Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, S.43-46, 2002
- [14] Kuhrmann, Marco et al.,
HELENA Stage 2 Results,
www.researchgate.net/publication/329246439_HELENA_Stage_2_Results
(abgerufen am 15.11.2019)
- [15] Klünder, Jil et al.,
Catching up with Method and Process Practice: An Industry-Informed Baseline for Researchers,
- [16] Kuhrmann, Marco et al.,
Walking through the Method Zoo: Does Higher Education really meet Software Industry Demands?,
- [17] Marinho, Marcelo et al.,
Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development,
- [18] Tell, Paolo et al.,
What are Hybrid Development Methods Made Of? An Evidence-based Characterization,
- [19] *RDocumentation*,
www.rdocumentation.org/
(abgerufen am 19.11.2019)

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 05.12.2019

Christian Münkel