Institut für Softwaretechnologie

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Experimentelle Untersuchung des Placeboeffekts beim Verstehen von Quellcode

Andreas Dominik Preikschat

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. Stefan Wagner
Betreuer:	Marvin Wyrich M.Sc.
Beginn am:	20. November 2019
Beendet am:	26. Juni 2020

Kurzfassung

Hintergrund: Softwaremetriken zur Messung von Komplexität sind fester Bestandteil der Softwaretechnik. Metriken werden zum Beispiel verwendet, um die Komplexität von Quellcode zu quantifizieren, viele dieser Metriken sind jedoch nicht ausreichend validiert. In der Softwaretechnik sind Placeboeffekte – nach unserem Wissen – bisher nicht untersucht. Außerhalb der Softwaretechnik sind sie in einer Vielzahl von Kontexten bekannt und können sich zum Beispiel auf Kognition auswirken. Es ist ungeklärt, ob nicht ausreichend validierte Metriken durch Placeboeffekte einen Einfluss auf Kognition von Softwareentwicklern haben.

Ziel: In dieser Arbeit untersuchen wir den Einfluss von nicht validierten Softwaremetriken im Kontext von Placeboeffekten beim Verstehen von Quellcode in Hinblick auf das Codeverständnis von Softwareentwicklern.

Methode: Wir führen ein doppelt-blindes Experiment mit 45 Teilnehmern, einer unabhängigen Variable und zwei Treatment-Gruppen durch. Teilnehmer müssen Java-Methoden verstehen und Rückgabewerte berechnen. Als Treatment wird jeder Gruppe eine andere Bewertung der Verständlichkeit, in Form einer manipulierten Metrik, präsentiert. Wir untersuchen, welchen Einfluss eine manipulierte Metrik auf die subjektive Wahrnehmung hat (RQ1) und, ob die Manipulation einen Einfluss auf das Codeverständnis hat (RQ2). Weiter untersuchen wir explorativ, welche individuellen Charakteristiken dabei eine Rolle spielen (RQ3).

Ergebnisse: Die beiden Gruppen haben identische Java-Methoden signifikant unterschiedlich bewertet (RQ1). Die Gruppen waren bezüglich ihres Codeverständnisses nicht signifikant unterschiedlich (RQ2). In der explorativen Analyse wurde eine Korrelation mit dem Grad, mit dem Teilnehmer sich an den vorgegebenen Metrikwert gehalten haben, und dem individuellen Charakteristikum „Ängstlichkeit“ gefunden (RQ3).

Limitationen: Um einen möglichen starken Placeboeffekt zu erreichen, haben wir die Metrik sehr prominent platziert und beworben, in Entwicklungsumgebungen ist dies meistens nicht der Fall.

Schlussfolgerung: Diese Arbeit zeigt, dass Verankerung durch Softwaremetriken möglich ist und bei Experimenten, die Teilnehmer nach einer subjektiven Bewertung fragen, darauf geachtet werden sollte, dass Teilnehmer nicht durch angezeigte Metriken in ihrer Bewertung verankert werden. In der Praxis könnte sich dies beispielsweise bei Aufwandsschätzungen unter Zuhilfenahme von Metriken zeigen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Listings

1. Einleitung

In diesem Kapitel legen wir die Motivation dieser Arbeit dar. Wir erläutern, welche Erkenntnisse es zu Placebos gibt. Außerdem legen wir dar, dass Softwaremetriken oftmals nicht validiert sind. Wir kombinieren diese beiden Aspekte, leiten daraus eine Problemstellung ab und verfeinern diese zu einem Forschungsziel. Abschließend erläutern wir die Struktur der Arbeit.

1.1. Motivation

In der Medizin gibt es viele Untersuchungen zu Placeboeffekten. Durch Placebos können zum Beispiel Schmerzen gelindert werden. Dies wird eindrücklich in einem Experiment von Levine et al. [LGF78] gezeigt, in dem Probanden, die aufgrund eines Placebos berichteten, sie hätten weniger Schmerzen, ein Mittel verabreicht wird, das schmerzlindernde Stoffe blockiert. Daraufhin nahm bei diesen Patienten der Schmerz zu. Bei den Teilnehmern, bei denen das Placebo im Vorfeld keine Verbesserung zeigte, führte das Mittel zu keiner Verschlechterung der Schmerzen [LGF78]. Ferner können Placeboeffekte bei Depressionen, Angst und dem Fatigue-Syndrom[1] beobachtet werden [WA15]. Durch Scheinoperationen – also nur vorgespielte Operationen – hat sich der Zustand von Parkinson-Patienten im auf die Scheinoperation folgenden Jahr ähnlich wie bei der Operationsgruppe entwickelt [MCY+04].

Außerhalb der Medizin wurden Placeboeffekte ebenfalls beobachtet: Draganich und Erdal haben einen „Placeboschlaf“ beobachtet. Probanden, denen gesagt wird, sie hätten in der vergangenen Nacht viele REM-Phasen gehabt, zeigen eine bessere Performance in Tests als die Vergleichsgruppe, der gesagt wird, sie hätten wenige REM-Phasen gehabt [DE14]. Eine scheinbar vorgenommene Elektrostimulation des Gehirns hat in einem Experiment von Turi et al. [TBG+18] abhängig davon, ob den Probanden erzählt wird, dass die Elektrostimulation die Leistung steigere bzw. senke, die kognitive Leistung der Probanden verbessert bzw. verschlechtert [TBG+18]. In einem weiteren Experiment wird ein „kreativitätsfördernder Duft“ – das Placebo – in einem Geruchslabor „getestet“. Die Teilnehmer, denen gesagt wird, dass der Duft Hemmungen senke und kreativitätsfördernd sei, zeigten im Vergleich zur Kontrollgruppe ausgeprägtere Kreativität [RMI+17]. Durch die Manipulation von Uhren sind Sportler im Stande, höhere Leistungen zu erbringen [Mor09]. In einer Metaanalyse wird geschlussfolgert, „dass bei Sportlern die leistungssteigernde Wirkung verschiedener Placebos bedeutend ist“ [BKSB11]. In der Softwaretechnik sind uns ähnliche Effekte, die zu einer Steigerung der Performance führen, nicht bekannt. Wir stellen uns die Frage, ob ähnliche Ergebnisse in der Softwaretechnik zu finden sind.

[1] Das Fatigue-Syndrom ist ein Krankheitsbild, bei dem der Betroffene unter chronischer Erschöpfung leidet [NE98].

In der Softwaretechnik sind Metriken ein fester Bestandteil. Die zyklomatische Komplexität wurde 1976 vorgestellt, um Softwareentwickler bei der Aufrechterhaltung von Wartbarkeit und Testbarkeit zu unterstützen [McC76]. Im Jahr 1994 haben Shepperd und Ince [SI94] in ihrer Arbeit verschiedene Metriken kritisiert, darunter die zyklomatische Komplexität. Sie kritisieren, dass die Modelle dieser Metriken unzureichend seien und damit die Validierung schwer sei [SI94]. In den Worten von Shepperd und Ince:

> „The importance of validation cannot be overstressed: metrics based on flawed models are worse than valueless: they are potentially misleading.“ [SI94]

Weiter verdeutlichen Scalabrino et al. [SBV+19] in ihrer Arbeit, dass viele Metriken nicht ausreichend validiert sind. Sie führen ein Experiment durch, in dem Teilnehmer Quellcode verstehen müssen. Die Autoren entwickeln Proxies, um zu quantifizieren, wie hoch das Codeverständnis der Teilnehmer ist. Das Codeverständnis der Teilnehmer korrelieren sie anschließend mit den Werten der ausgewählten Metriken. Laut der Autoren korreliere keine der über 70 untersuchten Metriken ausreichend mit Codeverständlichkeit [SBV+19]. Nilson et al. [NAG19] untersuchen Werkzeuge wie SonarQube und ob die darin eingesetzten Metriken validiert sind. Sie kommen zu dem Schluss, dass einige Metriken „irgendwie“ validiert sind, die „erdrückende“ Mehrheit aber nicht validiert ist [NAG19]. Wir fragen uns, wie sich nicht validierte Metriken auf Softwareentwickler auswirken.

1.2. Zielsetzung

Aufgrund der Beobachtungen in anderen Disziplinen, dass durch Placebos die Performance von Personen beeinflusst werden kann und aufgrund der angesprochenen Mängel vieler Softwaremetriken stellen wir uns die Frage, ob Metriken einen negativen oder positiven Einfluss auf die Performance von Softwareentwicklern beim Verstehen von Quellcode haben können. Da Softwareentwickler bei der Wartung von Software zwischen 30 %-50 % der Zeit damit verbringen, Quellcode zu verstehen [MML15], erachten wir diese Fragestellung für relevant. Weiter stellen wir uns die Frage, ob durch geschickte Manipulation von Metriken bewusst die Performance gesteuert werden kann. Als Forschungsziel formulieren wir deshalb:

Forschungsziel: *Diese Arbeit analysiert den Einfluss von manipulierten Softwaremetriken auf das Verstehen von Quellcode, mit dem Zweck zu verstehen, welchen Einfluss manipulierte Metriken auf die Performance von Softwareentwicklern haben. Die Arbeit wird im Kontext von Placeboeffekten in der Softwaretechnik an der Universität Stuttgart durchgeführt und strebt an, Codeverstehen realistisch abzubilden.*

1.3. Gliederung

Die Arbeit ist in folgender Weise strukturiert: In *Kapitel 2* wird Literatur mit Hintergrundinformationen zur Validierung von Metriken sowie zu Placebos und Placeboeffekten angeführt. Weiter wird in *Kapitel 3* Literatur vorgestellt, welche für die Planung, Durchführung und Einordnung dieser Arbeit relevant ist. In *Kapitel 4* wird die Planung und Methodik des Experiments beschrieben und diskutiert. Dazu gehört, dass Forschungsfragen aufgestellt und Hypothesen formuliert werden.

Weiter werden Maßnahmen beschrieben, die im Vorfeld ergriffen werden, um die Gültigkeit der Ergebnisse zu wahren. In *Kapitel 5* werden die Ergebnisse zu den Forschungsfragen vorgestellt und die Forschungsfragen beantwortet und in *Kapitel 6* wird besprochen, wie die Ergebnisse zu interpretieren sind und welche Limitationen beim Interpretieren berücksichtigt werden müssen. Abschließend wird in *Kapitel 7* die Arbeit zusammengefasst und ein Ausblick gegeben.

2. Hintergrund

In diesem Kapitel wird auf relevante Hintergründe für diese Arbeit eingegangen. Es wird erläutert, welche Rolle Modelle für Metriken spielen und welche Konsequenzen unvollständige Modelle haben können. Weiter wird grundlegend auf Placebos und Placeboeffekte eingegangen: Es werden die Begriffe *Placebo* und *Placeboeffekt* vorgestellt und beschrieben, welche möglichen Faktoren bei der Wirkung von Placebos einen Einfluss haben. Darüber hinaus wird anhand einer Studie aus den 1970ern die Wirkung von Placebos verdeutlicht. Es werden Beispiele für Placeboeffekte gegeben, die auf eine körperliche Leistungssteigerung eingehen. Außerdem wird Literatur über die Rolle der Geisteshaltung bei der Wirkung von Placebos vorgestellt. Abschließend werden kognitive Verzerrungen und der Ankereffekt erklärt und eingeführt.

2.1. Validierung von Metriken

In der Arbeit von Shepperd und Ince [SI94] werden drei Metriken einem Review unterzogen. Zu den untersuchten Metriken gehört zum Beispiel die zyklomatische Komplexität, die 1976 von McCabe vorgestellt wurde [McC76]. Shepperd und Ince bezeichnen die Vorstellungen der drei untersuchen Metriken zwar als Meilensteine in der Softwaretechnik, sie zeigen durch Verweise auf Literatur aber auch Unzulänglichkeiten dieser Metriken auf. Zum Beispiel sei die zyklomatische Komplexität von McCabe nicht für beliebige Programmiersprachen, sondern für Fortran ausgelegt gewesen. Weiter haben Studien Belege gefunden, dass strukturelle Verbesserungen an Quellcode nicht zwangsläufig in der zyklomatischen Komplexität widergespiegelt werden, sondern sogar zu einer Verschlechterung selbiger führen können. Die Autoren schreiben, dass diese Unzulänglichkeiten darauf beruhen, dass die zu Grunde liegenden Modelle unvollständigen seien. Shepperd und Ince erläutern, dass Modelle wichtig seien, da sie den Zweck einer Metrik beschreiben. Weiter beschreibe ein Modell den Zusammenhang von realer Welt und den Ein- und Ausgabewerten. Schlussendlich diene ein Modell dem Zweck einer Validierung. Sie fassen zusammen, dass die Unzulänglichkeiten der begutachteten Metriken darauf basieren, dass die Modelle dieser Metriken nicht vollständig seien und als Folge nur schlecht validierbar seien. Sie schreiben, dass nicht validierte Metriken potenziell mehr schaden würden, als sie nützen.

Scalabrino et al. [SBV+19] evaluieren in ihrer Arbeit über 70 verschiedene Metriken[1] im Rahmen eines Experiments. Eins der Forschungsziele ihrer Studie ist es, diese Metriken auf eine Korrelation mit Codeverständnis von Codeschnipsel hin zu untersuchen. Einleitend legen sie dar, dass es kein empirisch belegtes Modell gebe, um objektiv Codeverstehen zu messen. Weiter erläutern sie, dass bei der Evaluation vieler Verständlichkeitsmetriken nur die wahrgenommene Verständlichkeit – die subjektive Einschätzung der Probanden – untersucht werde. Folglich unterscheiden sie in ihrer

[1] Ursprünglich wurden 121 Metriken untersucht, durch Ausschluss redundanter Metriken wurde die Zahl der untersuchten Metriken auf 73 reduziert.

Arbeit zwischen *wahrgenommenem* Verstehen und *tatsächlichem* Verstehen von Codeschnipseln. Weiter führen sie Proxies ein, um Codeverständnis subjektiv und objektiv zu beschreiben. Zum Beispiel messen sie die tatsächliche Verständlichkeit (engl. *actual understandability (AU)*), indem sie den Probanden Verständnisfragen zu den Codeschnipseln stellen. Von 73 Metriken kann bei 51 Metriken keine Korrelation festgestellt werden und bei keiner der untersuchten Metriken wird eine mittlere oder starke Korrelation festgestellt.

Die Autoren Nilson et al. [NAG19] untersuchen in ihrer Arbeit, welche validierten Metriken existieren, um Softwarequalität zu quantifizieren und welche statischen Quellcodeanalysewerkzeuge diese Metriken einsetzen. Dazu sammeln die Autoren eine Liste von Metriken und exkludieren nicht validierte Metriken: Sie wählen nur Metriken zur genaueren Analyse aus, die in einer englischsprachigen Publikation wissenschaftlich validiert wurden. Werden gegensätzliche Ergebnisse zu einer Metrik in unterschiedlichen Publikationen gefunden, wird diese Metrik ausgeschlossen. Das Hauptkriterium der Autoren, damit eine Metrik berücksichtigt wird, ist, dass die Metrik mit einem Attribut von externer Softwarequalität korreliert, also nicht nur interne Softwarequalität quantifiziert, sondern auch externe Softwarequalität. Für die weitere Betrachtung haben Nilson et al. zwölf Metriken identifiziert. Weiter recherchieren sie 130 Analysewerkzeuge, die interne Softwarequalität abbilden können. Anhand von Kriterien wählen sie sechs Werkzeuge für eine genauere Betrachtung aus. Zu den ausgewählten Werkzeugen gehören beispielsweise *SonarQube* und das *Eclipse Metrics Plugin*. Sie resümieren, dass „beliebte“ Quellcodeanalysewerkzeuge diese validierten Metriken teilweise verwenden würden. Gleichzeitig würde eine Vielzahl von nicht validierten Metriken in diesen Werkzeugen verwendet werden. Laut Autoren könne dies zu Verwirrung bei Entwicklern führen.

2.2. Placebos und Placeboeffekte

Shapiro [Sha68] beschreibt in seiner Arbeit historische und aktuelle Definitionen von Placebos. Er stellt unterschiedliche Definitionen vor, dabei widmet er sich der Etymologie des Wortes und nennt Definitionen von Placebos in Lexika aus dem Jahr 1785. Weiter vergleicht er in seiner Arbeit, welche Definitionen in wissenschaftlichen Publikationen verwendet werden. Er beschreibt, dass in diesen Publikationen nicht nur Medikamente als Placebo bezeichnet werden, sondern auch der Einsatz von zum Beispiel Wärmebehandlungen, Inhalationsmitteln und Operationen. Laut Shapiro kann es dem Patienten bekannt oder unbekannt sein, dass ein Placebo eingesetzt wird. Jede Art der medizinischen Behandlung sei möglich. Dabei sei unerheblich, in welcher Form Medikamente verabreicht werden. Es seien therapeutische, chirurgische und andere mechanische Formen einer Therapie möglich. Der Autor definiert den Placeboeffekt als Ergebnis eines Placebos. Shapiro schlägt als Definition für *Placebo* folgendes vor:

> „Ein Placebo ist jede Therapie, die wohlüberlegt und wissentlich wegen ihrer unspezifischen oder psycho-physiologischen Wirkung benutzt wird, oder die unbewußt wegen ihrer vermuteten oder geglaubten spezifischen Wirkung auf eine Patientin/einen Patienten[,] ein Symptom oder eine Krankheit eingesetzt wird, aber die, ohne dass Patient(in) oder Therapeut(in) es wissen, ohne spezifische Wirkung auf die behandelte Problematik ist.“ [Win07]

Wager und Atlas [WA15] schreiben in ihrer Arbeit über den Fortschritt der Erforschung des Placeboeffektes, indem sie Erkenntnisse unterschiedlicher wissenschaftlicher Arbeiten zusammenfassen. Sie leiten ihre Arbeit ein, indem sie konstatieren, dass viele Krankheiten gut behandelt werden können, wenn die Vorgänge und Zusammenhänge im Körper verstanden werden. Weiter schreiben sie, dass Schmerzen, Depressionen und andere Krankheitsbilder, deren Ursachen überwiegend im Gehirn verortet sind, oft schwerer zu behandeln seien. Als Grund führen die Autoren an, dass die entsprechenden Hirnfunktionen sehr komplex und von vielen Einflussfaktoren abhängig seien. Zu den Einflussfaktoren zählen gemäß den Autoren der Zustand sowie die Funktionsweise des Gehirns. Weiter nennen sie als wichtige Faktoren die Wahrnehmung der Umgebung und den sozialen Kontext einer Behandlung. Die Gesamtheit dieser Faktoren bezeichnen sie als Kontext. Wager und Atlas erläutern, dass Placebostudien auf diesem Gebiet zu neuen Erkenntnissen geführt hätten, und zwar indem ebendieser Kontext manipuliert wird. Die Autoren beschreiben Placebos und den Placeboeffekt folgendermaßen:

> „Ein Placebo ist eine Behandlung, ein Medikament oder ein Gerät, das physisch und pharmakologisch inert[2] ist. Ein Placebo hat – per Definition – keine direkte therapeutische Auswirkung auf den Körper. Jedoch wird eine Behandlung in einem Kontext durchgeführt, zum Kontext gehören soziale und physische Hinweisreize, verbale Äußerungen und der Behandlungsverlauf. Dieser Kontext wird vom Gehirn aktiv interpretiert und kann Erwartungen, Erinnerungen und Gefühle wecken, die ihrerseits eine Veränderung der Gesundheit in Körper und Gehirn beeinflussen können. Folglich sind Placeboeffekte eine Gehirn-Körper-Reaktion auf Kontextinformationen, die Gesundheit und Wohlbefinden fördern.“ (Eigene Übersetzung nach Wager und Atlas [WA15])

Im weiteren Verlauf ihrer Arbeit stellen sie ein Framework vor, um den Kontext und die Einflussfaktoren auf den Placeboeffekt zu beschreiben. Als Erstes gliedern sie die Einflussfaktoren in zwei Gruppen: präkognitiven Assoziationen und gedankliche Prozesse. Unter *präkognitive Assoziationen* verstehen die Autoren „links between events and/or dates that exist outside conscious awareness“ [WA15]. Solche Assoziationen werden laut Wager und Atlas überwiegend durch Konditionierung geschaffen. Ein Beispiel für eine solche Konditionierung ist die Verknüpfung des Ortes ”Arztpraxis” mit einer anschließenden Genesung. Eine solche Assoziation wird als Ortskontext beschrieben. Andere Assoziation können Sinneshinweisreize (engl. *sensory cues*) sein, die unbewusst mit positiven und negativen Folgen verbunden werden. Präkognitive Assoziationen können unbewusst Veränderungen von Affektzustand, Gefühlen und Motivation bewirken. Unter *gedanklichen Prozessen* hingegen wird verstanden, dass aktuelle Informationen interpretiert werden und zusammen mit vergangenen Erfahrungen zu einer Beeinflussung von Erwartungen und einer Veränderung von Bewertungen und Erinnerungen führen können. Beispiele dafür sind soziale und verbale Hinweise, wie zum Beispiel, dass ein Arzt *Experte* auf einem Gebiet ist – ein Hinweis, der die Erwartung stärkt, dass es dem Patienten nach einer Behandlung besser gehen werde. Ferner beschreiben sie, dass Placeboeffekte durch drei unterschiedlichen Arten sichtbar werden können: Sie können zu einer Veränderung der Wahrnehmung führen, zum Beispiel einer subjektiven Verbesserung von Symptomen (engl. *reported experiences (symptoms)*). Ebenso kann ein Placebo zu einer Veränderung des Verhaltens führen (engl. *Behaviour*), wie beispielsweise Veränderungen der Essgewohnheiten, die sich wiederum langfristig positiv auf die Gesundheit auswirken können. Drittens kann es zu einer Zustandsverbesserung kommen (engl. *pathophysiology*

[2] Der Begriff *inert* kommt aus der Chemie und bedeutet, dass ein Stoff keine Reaktion mit anderen Stoffen zeigt [Mul94].

(signs)). Diese Zusammenhänge sind in Abbildung 2.1 illustriert. Wager und Atlas zeigen in ihrer Arbeit die Vielfalt von Placebos und Placeboeffekten auf. Placebos könnten dazu führen, dass Menschen sich besser fühlen. Sie könnten Schmerz – nicht nur subjektiv – lindern und können bei Herz-Kreislauf-Erkrankungen zu einer geringeren Sterberate führen, so die Autoren. Ferner könnten Placeboeffekt bei Depressionen und der Parkinsonkrankheit beobachtet werden. Placebos können sich auf den Hormonhaushalt auswirken und in Kombination mit Konditionierung das Immunsystem beeinflussen. Die Erkenntnisse bezüglich Faktoren, welche die Wirksamkeit von Placebos beeinflussen, seien für wirksame Medikamente wichtig: Sie nennen mehrere Studien die zeigen, dass Medikamente die unwissend verabreicht werden infolgedessen oftmals weniger gut wirken. Die Autoren zitieren eine Studie, die Indizien gefunden habe, dass Menschen, die gut auf Medikamente reagieren, auch gut auf Placebos reagieren. Sie resümieren, dass Placeboeffekte eine Reaktion auf den Kontext einer Behandlung seien. Weiter halten sie fest, dass bei der Untersuchung von Placeboeffekten stets eine Reduktion der Hirnaktivität in Gehirnarealen für negative Gefühle und Schmerz beobachtet werde. Der Tenor ihrer Arbeit ist, dass die Auslöser für Placeboeffekte vielfältig seien und unterschiedlich in Erscheinungen treten würden.

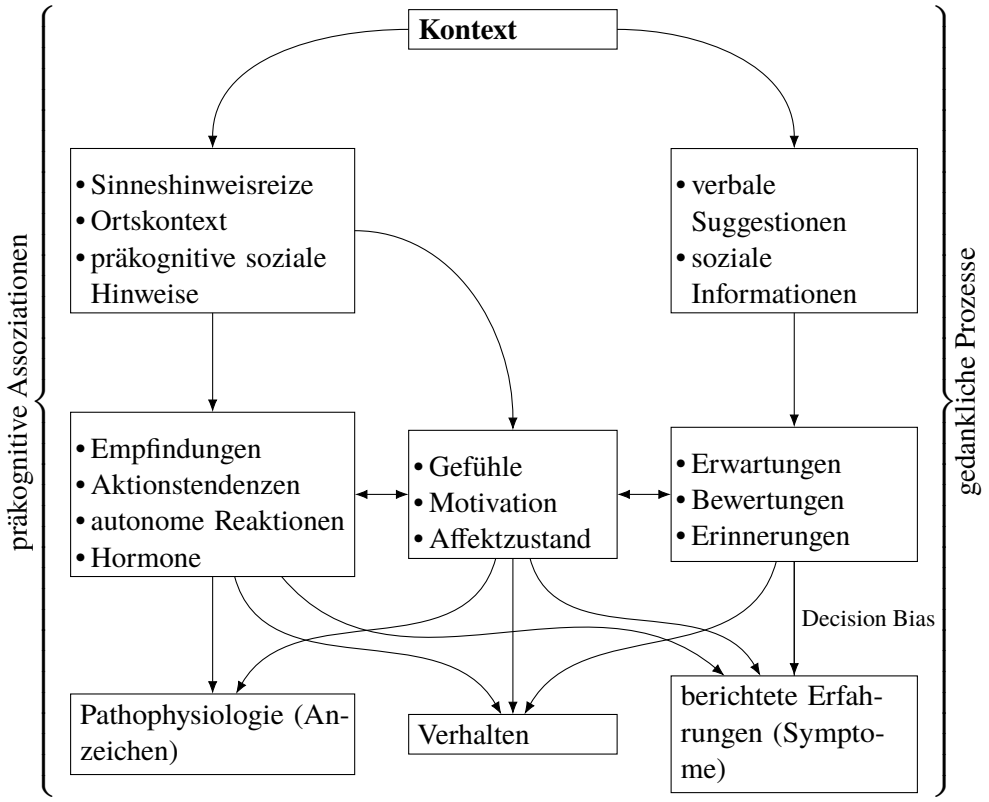

Abbildung 2.1.: Zusammenhang des Kontexts und der Placeboeffekten nach Wager und Atlas [WA15].

2.3. Wirksamkeit von Placebos am Beispiel Schmerz

Levine et al. [LGF78] haben in den 1970ern eine Placebostudie durchgeführt. Die Probanden ihrer Studie haben postoperative Schmerzen nach einer Zahnoperation. Deshalb erhalten die Patienten Schmerzplacebos: Bei einigen Patienten führt dies zu einer Linderung der Schmerzen, bei anderen Patienten verbessert sich durch die Gabe des Placebos das Schmerzempfinden nicht. Anschließend wird den Patienten Naloxon[3] verabreicht. Dies führt bei den Patienten, bei denen das Placebo Schmerzen gelindert hat dazu, dass die Schmerzen größer werden. Bei den Patienten, bei denen das Placebo keinen Effekt gezeigt hat, verändern sich die Schmerzen durch die Gabe von Naloxon nicht. Daraus leiten sie ab, dass die Gabe des Placebos zu einer Ausschüttung körpereigener Endorphine führe, die eine Schmerzlinderung bewirken würden.

2.4. Placeboeffekte im Sport

Bérdi et al. [BKSB11] führten 2011 eine Metaanalyse durch, um die Untersuchungen zu Placeboeffekten im Sport zu konsolidieren. Ihre Metaanalyse umfasst nach Ausschluss einiger Arbeiten 14 wissenschaftliche Arbeiten. Die ausgewählten Arbeiten untersuchen Placebos bei unterschiedlichen Sportarten und Disziplinen, wie beispielsweise Bankdrücken, Fahrrad fahren und Laufen. Die Sportarten bzw. die Daten der Studien wurden in Ausdauer- und Kraftsport unterteilt und separat betrachtet. Die aggregierten Effektstärken sind klein bzw. moderat gemäß Cohens d [Coh13]. Nach der Analyse werden von Bérdi et al. die Ergebnisse und Limitationen diskutiert. Zwar gebe es ein paar Limitationen, gemäß Autoren könne allgemein gesagt werden, dass Placeboeffekte im Sport existieren würden und Sportler durch Placebos leistungsfähiger seien.

2.5. Placebos und die Geisteshaltung von Menschen

Crum und Langer [CL07] haben eine Studie durchgeführt, um den Einfluss der Geisteshaltung im Kontext von Placebos zu untersuchen. Die Studie wird in sieben Hotels mit 84 Reinigungsfachkräften durchgeführt. Die Hotels werden zufällig entweder der Kontrollgruppe oder der Treatmentgruppe zugewiesen. Den Reinigungsfachkräften in der Treatmentgruppe wird berichtet, dass die Bewegung die sie durch ihre Arbeit haben, ein gutes Training sei. Es wird erklärt, dass ein guter Lebensstil daraus bestehe, sich zu bewegen und durch diese Bewegung im Schnitt 200 Kalorien am Tag zu verbrennen; schweres und anstrengendes Training sei nicht erforderlich. Den Reinigungsfachkräften wird erklärt, wie viele Kalorien bei den einzelnen Tätigkeiten ihrer Arbeit verbraucht werden. Weiter wird gesagt, dass sie mit ihrer Tätigkeit die Empfehlungen für einen gesunden Lebensstil überträfen.[4] Den Reinigungsfachkräften der Kontrollgruppe werden diese Informationen nicht gegeben. Zu Beginn der Studie werden Körpergewicht, Anteil von Körperfett, der Body-Mass-Index, Blutdruck und weitere Faktoren gemessen. Zum Beispiel wird abgefragt, wie viel Sport in der eigenen Wahrnehmung gemacht wird. Nach vier Wochen wird die Studie beendet: Die gleichen Faktoren werden erneut gemessen und Teilnehmer aufgeklärt. Die Auswertung der Daten von Crum und Langer ergibt,

[3] Naloxon ist ein Opiatantagonist. Das heißt, Naloxon kann eine schmerzlindernde Wirkung aufheben [Sch07].

[4] Die Informationen, die den Teilnehmern präsentiert werden, entsprechen den *Surgeon General's recommendations*.

dass obwohl Teilnehmer nicht mehr Sport machen, die Wahrnehmung der Treatmentgruppe sich dahingehend verändert hat, dass ihre Tätigkeiten mehr als Sport wahrgenommen werden. Darüber hinaus hat die Treatmentgruppe in den vier Wochen im Schnitt umgerechnet 800 g Gewicht verloren. Der Gewichtsverlust sei verglichen mit der Kontrollgruppe signifikant. Weiter habe sich der obere Blutdruck und das Körperfett signifikant reduziert. Crum und Langer diskutieren medizinische und psychologische Hintergründe, wie es zu solch einer Veränderung kommen kann. Abschließend schlussfolgern sie, dass die Geisteshaltung bei der Wirkung von Placebo wichtig sei.

2.6. Ankereffekt und kognitive Verzerrungen

Furnham und Boo [FB11] fassen in ihrem Literaturreview die Erkenntnisse über den Ankereffekt zusammen. Der Ankereffekt ist eine kognitive Verzerrung, bei dem Menschen beispielsweise bei einer Schätzung durch eine vorher genannte Zahl – dem Anker – beeinflusst werden. Beispielsweise werden Teilnehmer einer Studie gefragt, wie viele afrikanische Länder in Prozent ein Mitglied der Vereinten Nationen sind. Bevor die Teilnehmer ihre Schätzung abgeben, wird ein Glücksrad mit den Zahlen von 0 bis 100 gedreht und sie sollten beurteilen, ob der wahre Wert unter oder über der Zahl auf dem Glücksrad ist. Anschließend soll die eigene Schätzung abgegeben werden [FB11]. Personen, die den Wert von 10 auf dem Glücksrad drehten, haben im Mittel 25 % geschätzt, Personen die 45 auf dem Glücksrad drehten schätzen im Mittel 45 % [TK74]. Furnham und Boo [FB11] erklären, dass es keine Rolle spiele, ob der Anker relevant für die Schätzung ist oder nicht. Weiter tragen sie Literatur zusammen, die Zusammenhänge zwischen dem Ankereffekt und Einflussfaktoren wie Gemütslage, Erfahrung und Motivation untersucht. Sie führen unterschiedliche individuelle Charakteristiken an, für die in der Literatur ein Zusammenhang gesehen werde. Sie führen beispielsweise eine Publikation an, welche Menschen mit den individuellen Charakteristiken von Gewissenhaftigkeit und Verträglichkeit sowie geringem Anteil an Extraversion eine höhere Anfälligkeit für den Ankereffekt attestiere. Weiter wird eine Studie angeführt, laut der Menschen mit dem individuellen Charakteristikum „Offenheit“ eher von Ankereffekten beeinflusst seien. Außerdem diskutieren sie die Frage, ob Experten und Nicht-Experten gleichermaßen für Ankereffekte anfällig seien. Trotz der unterschiedlichen Ergebnisse in der Literatur kommen sie zu dem Schluss, dass das Wissen einer Person keinen Einfluss auf den Ankereffekt habe. Sie nennen als Beispiel eine Studie, in der Automobilkaufmänner, welche alle notwendigen Informationen zur Verfügung haben, um den Wert eines Autos zu schätzen, sich trotzdem von einem Anker haben beeinflussen lassen. Furnham und Boo tragen unterschiedliche Erklärungsansätze für diesen Effekt zusammen. Der aktuelle Ansatz erkläre den Ankereffekt damit, dass je nach Anker jene Informationen berücksichtigt werden, die konsistent mit dem Anker sind (engl. *confirmatory hypothesis testing*).

Mohanani et al. [MST+18] führen ein systematisches Literaturreview über kognitive Verzerrungen (engl. *cognitive biases*) in der Softwaretechnik durch. Sie beschrieben kognitive Verzerrungen als „systematische Abweichungen von optimalen Schlussfolgerungen“ [MST+18]. In ihrem Review identifizieren sie insgesamt 37 Verzerrungen, wie beispielsweise den Bestätigungsfehler (engl. *confirmation bias*). Der Bestätigungsfehler sei eine Verzerrung, die dazu führe, dass Informationen die eine *Erwartung* bestätigen, betont und zu unseren Erwartungen konträre Informationen verworfen oder ignoriert werden. Zum Beispiel führe der *positive Test Bias* dazu, dass Tests in einer Form geschrieben werden, die die korrekte Funktionsweise einer Software bestätigen, anstatt dass Tests Fehler aufdecken. In ihrem Review untersuchen sie neben den Verzerrungen auch die Taxonomie von Verzerrungen. Sie ordnen die Verzerrungen Kategorien zu. Gemäß Mohanani et al. gehört der

Bestätigungsfehler zur Kategorie *Interessenverzerrungen* (engl. *interest biases*) und Ankereffekte zu der Kategorie *Stabilitätsverzerrungen* (engl. *stability biases*). Als weitere Kategorie nennen sie Entscheidungsverzerrungen (engl. *decision biases*). Eine ihrer Schlussfolgerungen der Arbeit ist, dass kognitive Verzerrungen an vielen Stellen der Softwaretechnik zu finden seien und die psychologischen und soziologischen Hintergründe nicht ausreichend untersucht seien. Ebenso kritisieren sie, dass es bei Verzerrungen und der Taxonomie ein „allgemein verbreitetes Durcheinander“ [MST+18] gebe. Verzerrungen würden in Arbeiten teilweise unterschiedlich definiert oder haben unterschiedliche Namen. Ferner gebe es Verzerrungen, die sich sehr ähnlich sind oder sogar – so vermuten die Autoren – auf die gleichen kognitiven Vorgänge zurückzuführen seien.

2.7. Zusammenfassung

In diesem Kapitel wurde anhand einer Arbeit aus dem Jahr 1994 von Shepperd und Ince [SI94] erläutert, dass das Modell einer Metrik eine wichtige Rolle einnimmt. Ein Modell beschreibt den Zweck einer Metrik und ermöglicht eine Validierung selbiger. Gemäß der Autoren seien solche Modelle oft unzureichend und schlussendlich schade eine nicht validierte Metrik mehr, als sie nütze [SI94]. Dass das Problem von nicht validierten Metriken nach über 25 Jahren immer noch existiert, wurde anhand von Literatur aus dem Jahr 2019 gezeigt: Scalabrino et al. [SBV+19] verdeutlichen, dass viele Metriken nicht validiert seien, keine der untersuchten Verständlichkeitsmetriken habe eine starke Korrelation mit Verständlichkeit. Und Nilson et al. [NAG19] legen dar, dass in häufig verwendeten Analysewerkzeugen kaum validierte Metriken zum Einsatz kämen. Weiter wurden Placebos und Placeboeffekte definiert und vorgestellt. Laut Shapiro [Sha68] können Medikamente, Operationen, Inhalationsmittel und Wärmebehandlungen als Placebos dienen. Neben der Vielfältigkeit, die [Sha68] beschreibt, erklären Wager und Atlas [WA15], welche unterschiedlichen Einflussfaktoren es auf die Wirkung von Placebos gibt. Zum Beispiel können verbale Suggestionen die Erwartungen, Bewertungen und Erinnerungen beeinflussen und sich auf Verhalten und Gefühle auswirken. Am Beispiel von Schmerzen wurde von Levine et al. [LGF78] verdeutlicht, dass Placebos nicht reine Einbildung sind, sondern Reaktionen im Körper auslösen. Von Crum und Langer [CL07] wurde präsentiert, welche Rolle die Geisteshaltung bei der Wirkung von Placebos spielt und wie sich eine veränderte Geisteshaltung auf den Körper auswirken kann, selbst wenn sich das Verhalten nicht ändert. Dass es durch Placebos Leistungssteigerungen im Sport gibt, wurde in der Metaanalyse von Bérdi et al. [BKSB11] herausgearbeitet. Abschließend wurden kognitive Verzerrungen eingeführt [MST+18] und der Ankereffekt als eine solche Verzerrung beschrieben: Der Ankereffet führt dazu, dass Menschen sich bei beispielsweise Schätzungen beeinflussen lassen [FB11].

Dass viele Metriken nicht validiert sind [SBV+19] und dass nicht validierte Metriken möglicherweise Entwickler verwirren [NAG19] und eventuell mehr schaden als nützen [SI94], ist für diese Arbeit ein interessanter Aspekt, da dadurch die Frage aufkommt, welche – möglicherweise negativen – Effekte manipulierte oder nicht validierte Metriken auf Softwareentwickler haben könnten. In unserer Arbeit möchten wir untersuchen, ob es Placeboeffekte in der Softwaretechnik gibt. Da Placeboeffekte zu Veränderungen im und am Körper führen [CL07; LGF78] und weil Placeboeffekte durch unterschiedliche Einflussfaktoren, wie beispielsweise verbale Suggestionen, beeinflusst werden können [WA15], sehen wir die Möglichkeit, eine Metrik als Placebo zu verwenden. Ob eine Manipulation von Softwareentwicklern durch eine Metrik möglich ist, wollen wir anhand des Ankereffekts untersuchen.

3. Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt, die für das Design unseres Experiments oder für eine Einordnung unserer Ergebnisse wichtig sind. Es wird auf individuelle Charakteristiken, wie beispielsweise dispositionellen Optimismus, als Einflussfaktoren auf die Wirksamkeit von Placebos eingegangen. Weiter werden Beispiele für Placeboeffekt bei kognitiver Leistung und Kreativität angeführt.

3.1. Optimismus als Einflussfaktor auf die Wirksamkeit von Placebos

Morton et al. [MWEJ09] untersuchen im Kontext von Schmerzbehandlung, welche individuellen Charakteristiken bei Placeboeffekten eine Rolle spielen. Vor dem Experiment müssen Probanden Fragebögen ausfüllen: beispielsweise den LOT-R-Fragebogen, um *dispositionellen Optimismus*[1] zu messen. In ihrem Experiment führen sie Probanden an den Unterarmen schmerzhafte Wärmeimpulse mit einem Laser zu. Probanden müssen die Schmerzen auf einer Skala von 0 bis 10 bewerten. Davor wird Teilnehmern auf beide Unterarme eine Creme gegeben. Der Kontrollgruppe wird mitgeteilt, dass auf beiden Unterarmen eine inaktive Creme verwendet wird. Ebenso wird die Kontrollgruppe darüber aufgeklärt, dass die schmerzhaften Wärmeimpulse an einen Arm niedriger sein werden, als beim anderen. Der Placebogruppe[2] wird gesagt, dass sie an einem Unterarm eine inaktive Creme erhalten werden und an dem anderen entweder eine wirksame oder inaktive Creme erhalten werden – in Realität sind aber beide Cremen inaktiv. Jeder Proband hat mit einem Abstand von wenigstens zwei Wochen an einer zweiten Wiederholungssitzung teilgenommen. Morton et al. haben die Bewertungen der Schmerzen der zwei Gruppen und den Unterarmen zwischen den zwei Sitzungen verglichen. In ihrer Regressionsanalyse kommen sie zu dem Ergebnis, dass wenn Teilnehmer der Placebogruppe das individuelle Charakteristikum *dispositionellen Optimismus* besitzen und in der ersten Sitzung auf das Placebo angesprochen haben, die Erwartung geweckt wird, dass es in der zweiten Sitzung erneut zu einer Schmerzlinderung kommt. Diese Erwartung führe bei jenen Probanden in der zweiten Sitzung zu einer geringeren Angst.

Geers et al. untersuchen in ihrer Arbeit [GWF+10] die Wirkung von Placebos im Kontext von Schmerz bei Menschen. Dazu haben sie ein Experiment mit zwei Gruppen und einer wirkungslosen Creme durchgeführt. Der Placebogruppe wird erzählt, dass die Creme mit dem englischen Namen *Trivaricane* eine zeitgemäße Creme zur örtlichen Betäubung sei und dass dieser Creme in Vorstudien anderer Forscher eine schmerzlindernde Wirkung attestiert wurde. Weiter wird berichtet, dass die Wirkung nach 30 Sekunden eintreten werde. Der Kontrollgruppe wird berichtet, dass diese Creme üblicherweise in ähnlichen Experimenten zur Reinigung der Hände verwendet werde. Nach

[1] Dispositioneller Optimismus ist nach [SC85] ein individuelles Charakteristikum von Menschen. Menschen mit diesem Charakteristikum sehen generell positiv in die Zukunft und erwarten, dass das Gute und nicht das Schlechte eintrifft.
[2] Hier ist die Placebogruppe die Gruppe, die das „Treatment“ erhält.

Auftragen der Creme haben Teilnehmer anschließend ihre Hand für zwei Minuten in 4 °C kaltes Wasser mit zerstoßenem Eis gehalten. Teilnehmern wird gesagt, dass sie die Hand früher aus dem Wasser nehmen dürfen, wenn es zu unangenehm sein sollte. Nach Ablauf der zwei Minuten haben Teilnehmer über einen Fragebogen (SF-MPQ) die Schmerzen bewertet. Anhand dieser Daten wird eine Regressionsanalyse durchgeführt. Neben den Schmerzwerten werden die Ergebnisse des LOT-R-Fragebogens, der dispositionellen Optimismus und Pessimismus misst, verwendet. Sie kommen zu dem Ergebnis, dass in der Placebogruppe bei Personen mit hohen Werten für Optimismus geringere Schmerzwerte berichten werden, als Personen mit geringen Optimismuswerten. In der Kontrollgruppe wird dieser Unterschied nicht gefunden.

3.2. Persönlichkeit als Einflussfaktor auf die Wirkung von Placebos

In der Arbeit von Darragh et al. [DBC14] wird untersucht, welche individuellen Charakteristiken bei der Wirkung von Placebos im Kontext von Stress eine Rolle spielen. Vor der Studie werden online einige Fragebögen ausgefüllt. Dazu gehört ein Fragebogen zur Messung von Neurotizismus und Extraversion (EPQ-R), der LOT-R-Fragebogen, um Optimismus zu quantifizieren und weitere Fragebögen, wie beispielsweise ein Fragebogen, um Empathie zu messen. Die Studie ist in zwei Phasen aufgeteilt. In der ersten Phase müssen alle Teilnehmer eine Subtraktion berechnen, um Stress zu erzeugen. Anschließend werden Herzfrequenz, Herzfrequenzvariabilität und wahrgenommener Stress gemessen. Wahrgenommener Stress wird durch Fragebögen ermittelt und aus Herzfrequenz und Herzfrequenzvariabilität wird physiologischer Stress abgeleitet. Vor der zweiten Phase – dem zweiten Stresstest – wird der Kontrollgruppe ein Video vorgespielt, in dem erklärt wird, das es wichtig sei, Störfaktoren in Experimenten zu kontrollieren. Der Placebogruppe wird in dem Video erklärt, dass ihnen vor und nach dem zweiten Stresstest ein Nasenspray verabreicht wird, dass die Regeneration vom Stress verbessere. Das Nasenspray ist in Wahrheit eine wirkungslose 5-prozentige Salzlösung. Darragh et al. stellen fest, dass die Placebogruppe sich durch das Placebo schneller vom wahrgenommenen und physiologischen Stress erholt habe. Niedrige Werte bezüglich Optimismus stünden in Beziehung zur schnelleren wahrgenommenen Regeneration vom Stress. Anlog gelte dies für Empathie. Gemäß der Regressionsanalyse der Autoren konnten Zusammenhänge zwischen der Herzfrequenz und Herzfrequenzvariabilität und individuellen Charakteristiken gefunden werden. Ein solcher Regressor sind hohe Neurotizismuswerte.

Peciña et al. [PAL+13] untersuchen in ihrer Arbeit, welche Rolle individuelle Charakteristiken bei der Wirkung von Placebos bei Schmerzen haben und in welchem Zusammenhang die individuellen Charakteristiken mit Resilienz stehen. Dazu führen sie ein Experiment mit einem within-subjects Design durch. Weiter werden verschiedene Fragebögen zu individuellen Charakteristiken ausgefüllt [PAL+13]. Darunter der PANAS-Fragebogen, um Gefühle zu erfassen [BB16] und der LOT-R, um dispositionellen Optimismus zu quantifizieren. Weiter verwenden Peciña et al. [PAL+13] den Revised NEO Personality Inventory-Fragebogen, um – wie der Big Five-Fragebogen – individuelle Charakteristiken zu bestimmen. In ihrer Regressionsanalyse finden sie, dass Verträglichkeit ein positiver und Neurotizismus ein negativer Regressor für die Wirksamkeit von Placebos seien. Eine Korrelation mit Optimismus haben die Autoren nicht beobachtet.

3.3. Kognitive Leistung, Kreativität und Placebos

Draganich und Erdal [DE14] haben in zwei Experimenten[3] den „Placeboschlaf“ untersucht. In den Experimenten haben die Teilnehmer zu Beginn auf einer Skala von 1 bis 10 bewertet, wie qualitativ der eigene Schlaf in der vergangenen Nacht empfunden wurde. Die Studienleiter haben anschließend unter dem Vorwand, mehr Hintergrundinformationen zur Teilnahme zu geben, den Teilnehmern grundlegende Informationen über Schlafqualität und kognitive Funktionen vermittelt. Den Probanden wird zum Beispiel berichtet, dass Menschen durchschnittlich 20-25 % der Nacht in REM-Phasen sind. Weiter wird berichtet, dass Personen die unter 20 % dieser REM-Phasen haben, durchschnittlich schlechtere Leistung und jene die über 25 % REM-Phasen haben, bessere Leistungen bringen würden. Es wird von einem angeblich neuen Verfahren berichtet, das anhand von Herzfrequenz, Blutdruck und der Frequenz von Hirnströmen bestimmen könne, wie viel REM-Schlaf eine Person in der vorherigen Nacht gehabt habe. Den Teilnehmern wird gesagt, dass führende Schlafspezialisten die Korrelation des Verfahrens und Quantität des REM-Schlafs bestätigt hätten. Das Ergebnis dieses Verfahrens wird von den Autoren manipuliert, sodass den Probanden entweder 28,7 % oder 16,2 % REM-Schlafphasen attestiert werden. Anschließend führen die Probanden den *Paced Auditory Serial Addition Test* (PASAT) durch. Der PASAT ermittelt beispielsweise, wie schnell Informationen verarbeitet werden. Das Ergebnis dieser Experimente ist, dass die subjektive Bewertung der Schlafqualität der Teilnehmer nicht mit den Ergebnissen des PASATs korreliere. Zwischen der durch das gefälschte Verfahren attestierten Schlafqualität und der gemessenen Performance im PASAT bestehe dagegen ein Zusammenhang. Sie schlussfolgern, dass die Geisteshaltung die kognitive Performance beeinflussen kann.

In der Arbeit von Rozenkrantz et al. [RMI+17] wird untersucht, ob Placebos einen Einfluss auf Kognition von Probanden haben. Dazu führen Rozenkrantz et al. ein Experiment in einem Geruchslabor durch. Die Teilnehmer dieses Experiments werden gebeten, an einem Duftstoff zu riechen und diesen anhand unterschiedlicher Eigenschaften zu bewerten. Beispielsweise soll bewertet werden, wie „angenehm“ dieser ist. Es gibt zwei Gruppen, eine Placebo- und eine Kontrollgruppe. Der Placebogruppe wird zusätzlich gesagt, dass dieser Duftstoff einzigartig sei, Kreativität fördere und Hemmungen verringere. Anschließend absolvieren die Teilnehmer verschiedene Tests. Zu den Tests gehört das *creative foraging game* (CFG), ein Spiel, um Kreativität zu quantifizieren. Nach einer zehnminütigen Spielzeit werden die Probanden gebeten, noch einmal an dem Duftstoff zu riechen und diesen zu bewerten. Der Kontrollgruppe wird gesagt, dass man Bewertungen zu unterschiedlichen Zeiten brauche, der Placebogruppe wird hingegen gesagt, dass so der kreativitätsfördernde Effekt des Duftstoffs aufrecht erhalten werden solle. Anschließend wird der *alternate use test* (AUT) sowie der *Torrance test of creative thinking* durchgeführt – beide quantifizieren bestimmte Aspekte von Kreativität. Das Ergebnis des Experiments ist, dass die Placebogruppe eine signifikant höhere Originalität und eine höhere, aber nicht signifikant höhere, Fähigkeit habe „out of the box“ zu denken.

[3] Das zweite Experiment ist eine revidierte Replikation des ersten Experiments.

3.4. Zusammenfassung

In diesem Kapitel wurde auf die Rolle von dispositionellem Optimismus bei der Wirkung von Placebos eingegangen: Es wurden Arbeiten aus der Schmerzbehandlung angeführt [GWF+10; MWEJ09] und eine Arbeit im Bereich von Stress [DBC14]. Weiter widmen sich Darragh et al. [DBC14] und Peciña et al. [PAL+13] der Untersuchung von weiteren individuellen Charakteristiken. Es wurde ein Experiment über den „Placeboschlaf“ von Draganich und Erdal [DE14] vorgestellt, das illustriert, dass kognitive Performance durch Placebos manipuliert werden kann. Mit [RMI+17] wurde eine Arbeit vorgestellt, die den Einfluss von Placebos auf die Kreativität verdeutlicht.

Aus der Literatur geht hervor, dass Placebos in vielen anderen Bereichen vorzufinden sind (vgl. Kapitel 2) und dass beispielsweise Optimismus ein wahrscheinlicher Einflussfaktor auf die Wirkung von Placebos ist [DBC14; GWF+10; MWEJ09]. Weiter haben Peciña et al. [PAL+13] individuelle Charakteristiken identifiziert. Die Experimente von Draganich und Erdal [DE14] und Rozenkrantz et al. [RMI+17] bewegen sich außerhalb des Felds der Schmerzbehandlung. Beispielsweise ist die Arbeit von [DE14] in einem Psychologie-Journal veröffentlicht. Die Ideen der Experimente könnten beim Design eines eigenen Experiments hilfreich sein. Wir wissen nicht, ob Placeboeffekte in der Softwaretechnik zu finden sind und, wenn ja, ob die gleichen Einflussfaktoren relevant sind. Unsere Arbeit soll diese Lücke schließen, indem einerseits ein Placeboexperiment im Kontext der Softwaretechnik durchgeführt wird und anderseits die Ergebnisse zu individuellen Charakteristiken repliziert werden.

4. Methodik

In diesem Kapitel wird beschrieben, welche Forschungsfragen mit dem Experiment beantwortet werden. Es wird beschrieben, wie Teilnehmer für das Experiment gewonnen werden, wie die Aufgabenstellung lautet und welche Materialien benötigt werden. Es werden Treatments, Variablen und Hypothesen und das Design erläutert und begründet. Weiter wird beschrieben, wie das Experiment durchgeführt wird. Abschließend wird dargelegt, wie die Daten analysiert werden und welche Aspekte dabei zu berücksichtigen sind.

4.1. Forschungsfragen

Mit diesem Experiment sollen folgende Forschungsfragen beantwortet werden:

RQ1 Hat eine Manipulation der Verständlichkeitsmetrik einen Einfluss auf die subjektive Bewertung der Codeverständlichkeit?

Wager und Atlas [WA15] erklären in ihrer Arbeit die unterschiedlichen Einflussfaktoren und Komponenten der Wirkungsweise von Placebos. Die Veränderung des Kontextes – beispielsweise durch eine Manipulation der Verständlichkeitsmetrik – könnte einen Einfluss auf die Erwartung der Probanden haben. Diese veränderte Erwartung könnte sich, wenn wir die Erkenntnisse von Wager und Atlas übertragen, zum einen auf das Verhalten der Probanden auswirken und zum anderen sich durch einen Decision Bias [WA15] bzw. Verankerung (vgl. [FB11]) manifestieren. Ließe sich dies beobachten, würde dies Anlass geben, die Wirkung von Metriken auf den Menschen ausführlicher zu untersuchen.

RQ2 Hat eine Manipulation der Verständlichkeitsmetrik einen Einfluss auf das tatsächliche Codeverständnis?

Draganich und Erdal [DE14] schreiben in ihrer Arbeit über den Placeboschlaf. Dort werden Teilnehmer mit einem technischen Hilfsmittel dazu gebracht zu glauben, dass sie viele REM-Phasen gehabt hätten. Durch diese Manipulation sind die Probanden in einem Test besser, als die Probanden, denen gesagt wird, sie hätten wenige REM-Phasen gehabt. Autoren wie Rozenkrantz et al. [RMI+17] und Turi et al. [TBG+18] zeigen, dass Placebos sich auf Kreativität [RMI+17] und Lernperformance [TBG+18] auswirken können. Darüber hinaus haben Bérdi et al. [BKSB11] und Morton [Mor09] körperliche Leistungssteigerungen durch Placebos im Sport beobachtet. In der Arbeit von Morton [Mor09] werden die Uhren von Sportlern manipuliert, mit dem Ergebnis, dass männliche[1] Sportler signifikant länger bis zur Erschöpfung auf einem Fahrradergometer trainieren konnten. Diese Arbeiten zeigen,

[1] Bei Frauen wurde der Effekt auch beobachtet, war aber nicht signifikant.

dass Placeboeffekte vermutlich nicht nur in der Medizin verortet werden können. Ob es Leistungsverbesserungen durch Manipulation von Metriken gibt, ist uns nicht bekannt. Die Untersuchungen zu dieser Forschungsfrage soll erste Erkenntnisse dazu liefern.

RQ3 Welchen Einfluss haben individuelle Charakteristiken auf den Grad der Abweichung vom angezeigten und manipulierten Wert einer Verständlichkeitsmetrik?

In der Medizin werden eine Vielzahl von Einflussfaktoren auf die Wirkung von Placeboeffekten beobachtet. Zu den Einflussfaktoren zählen beispielsweise Gefühle [WA15]. Die Bedeutung von individuellen Charakteristiken auf die Wirksamkeit von Schmerzplacebos wird von Peciña et al. [PAL+13] gezeigt. Geers et al. [GWF+10] und Morton et al. [MWEJ09] haben dispositionellen Optimismus als ein individuelles Charakteristikum, das sich positiv auf die Wirksamkeit von Placeboeffekten auswirkt, identifiziert. Die Ergebnisse von Darragh et al. [DBC14] stehen dem zwar entgegen, da hier der Einfluss von dispositionellem Optimismus geringer war. Trotzdem schreiben sie, dass die Wirksamkeit von Placebos neben dem Kontext abhängig von individuellen Charakteristiken sei. Die Untersuchung von Einflussfaktoren auf die Wirksamkeit von Manipulationen von Metriken im Kontext der Softwaretechnik ist damit ein interessanter Aspekt. Aufgrund der Vielzahl von möglichen Einflussfaktoren wird diese Forschungsfrage explorativ untersucht.

4.2. Teilnehmer

Um die Forschungsfragen zu überprüfen, wird ein kontrolliertes Experiment durchgeführt. Es ist geplant, dass an dem Experiment 30 bis 50 Teilnehmer teilnehmen. Die Teilnehmer werden größtenteils im Masterstudium der Studiengänge Softwaretechnik und Informatik sein. Die Teilnehmer werden über E-Mail und über die Lernplattform ILIAS eingeladen. Der Großteil der Personen wird über die Vorlesung *Forschungsmethoden der Softwaretechnik* akquiriert; die Studierenden dieser Veranstaltung haben die Scheinbedingung, an einem Experiment der Abteilung teilzunehmen. Neben dem Erwerb dieses Scheins dürfen sich Teilnehmer am Ende des Experiments eine Süßigkeit nehmen. Es ist vorgesehen, bei Bedarf weitere Teilnehmer über persönliche Kontakte zu gewinnen. Darüber hinaus wird nicht öffentlich zum Experiment eingeladen. Teilnahmebedingung für eine Teilnahme an dem Experiment sind gute Java-Kenntnisse sowie – da das Experiment auf Deutsch durchgeführt wird – gute Deutschkenntnisse.

Die Teilnehmer werden nicht anhand bestimmter Merkmale einem Treatment zugeordnet. Die Zuteilung soll implizit dadurch geschehen, indem sich Teilnehmer für ein Zeitfenster anmelden. Folglich werden Teilnehmer dem Treatment nicht anhand bestimmter Merkmale zugeordnet. Eine explizite und doppelt-blinde Zuteilung von *Zeitfenstern zu Treatments* wird von uns vorgenommen, dies geschieht aber unabhängig von den entsprechenden Teilnehmern in einem Zeitfenster (vgl. Abschnitt 4.6).

Nachdem den Teilnehmern das Ziel und die Aufgabe des Experiments erklärt wurden, wird ihnen eine Einverständniserklärung ausgehändigt. Es wird zusätzlich mündlich erklärt, in welcher Form Daten anonymisiert erfasst werden, welche da sind Fragebögen und Bearbeitungszeiten. Weiter wird erklärt, dass die Teilnahme an dem Experiment zu jeder Zeit beendet werden kann. Die Anonymität der Datensätze wird gewährleistet, indem Namen und Experimentdaten nicht verknüpft werden. Eine nachträgliche Zuordnung ist ebenfalls nicht möglich, da immer zwei Teilnehmer zusammen an

dem Experiment teilnehmen werden und nicht aufgezeichnet wird, welcher Teilnehmer an welchem Laptop arbeitet.[2] Die Teilnehmer werden circa zwei Wochen nach dem Experiment per E-Mail und in der Vorlesung *Forschungsmethoden der Softwaretechnik* über den wahren Grund des Experiments aufgeklärt. Wir erwarten nicht, dass Teilnehmer – unabhängig vom Treatment – einen Schaden durch das Experiment nehmen werden.

4.3. Materialien

In dem Experiment werden die Teilnehmer mehrere Fragebögen ausfüllen. Zum einen werden demografische Daten abgefragt, zum anderen werden Fragebögen zu individuellen Charakteristiken ausgefüllt. Die demografischen Daten werden direkt nach dem Unterschreiben der Einverständniserklärung erfasst. Ebenso der Fragebogen zum Quantifizieren von negativen und positiven Gefühlen (SPANE) [RHS17]. Die restlichen Fragebögen werden, in gegebener Reihenfolge nach der Bearbeitung der Aufgaben, von den Teilnehmern ausgefüllt: Es wird der Big Five, zur Bestimmung von individuellen Charakteristiken, ausgefüllt [LLA01] und als letzter Fragebogen wird der Life-Orientation-Test (LOT-R) zur Messung von dispositionellem Optimismus und Pessimismus vorgelegt [GHKH08]. Für alle Fragebögen wird die validierte deutsche Version verwendet und alle Fragebögen werden in LibreOffice Calc ausgefüllt.

Den Teilnehmern werden im Laufe des Experiments mehrere Codeschnipsel gezeigt. Jedes Codeschnipsel besteht aus einer Klasse, die genau eine Methode enthält. Jede Methode verfügt über eine Dokumentation in Form eines Javadocs. Die Codeschnipsel sind im Anhang A abgebildet. Es gibt zwei Beispielaufgaben, diese Beispiele sind in Listings A.1 und A.2 zu sehen. Die Codeschnipsel A.3, A.4 und A.5 sind die Aufgaben, welche die Teilnehmer im Verlauf des Experiments bearbeiten werden. Die Codeschnipsel Listings A.2 bis A.5 – also alle Codeschnipsel bis auf das erste Beispiel A.1 – haben gemäß der Cognitive Complexity[3] [Cam18] die gleiche Komplexität; sie haben einen Komplexitätswert von 19. Die Codeschnipsel sind alle unter der Apache Lizenz in Version 2 lizenziert. Codeschnipsel A.1 wurde selbst programmiert. Hingegen sind A.2, A.3 und A.5 dem Projekt *Commons Lang*[4] entnommen. Codeschnipsel A.4 stammt aus dem Apache *Commons Collections*[5] Projekt. Die Codeschnipsel wurden teilweise modifiziert, insbesondere sind in allen Codeschnipseln[6] Bugs implementiert worden. Bei der Auswahl der Codeschnipsel für die Aufgaben wurde darauf verzichtet, generische Typen zu verwenden und darauf, dass Kenntnisse von Klassenhierarchien erforderlich sind. Es werden nur primitive Datentypen sowie objektorientierte Datentypen für die primitiven Datentypen, Strings, Arrays, Sets und Listen verwendet. Methodenaufrufe dürfen Seiteneffekte haben, diese sollen sich möglichst nur auf die übergebenen Parameter auswirken – also „statischen Charakter“ haben. Weiter werden Funktionen neuerer Java-Versionen, wie beispielsweise die Stream-API, die Verwendung von Lambdas und Methodenreferenzen, vermieden.

[2] Bei der Durchführung gab es ein Zeitfenster, in dem nur ein Teilnehmer anwesend war, hier wäre eine Zuordnung möglich gewesen.

[3] Um möglichst objektiv zu sein, verwenden wir diese Metrik, anstatt nach subjektivem Schwierigkeitsempfinden Codeschnipsel auszuwählen.

[4] https://commons.apache.org/proper/commons-lang/

[5] https://commons.apache.org/proper/commons-collections/

[6] Außer in A.2, denn dort ist es nicht erheblich für die Durchführung, dass ein Bug vorhanden wäre.

Die Teilnehmer werden an einem Laptop vom Typ Lenovo T520 arbeiten. Als Entwicklungsumgebung wird eine browserbasierte Lösung implementiert. Diese Entwicklungsumgebung basiert auf HTML-Dateien, die lokal im Browser geöffnet werden und die mit Bootstrap[7] und highlight.js[8] umgesetzt wird. Die Gründe für diesen Ansatz sind in Abschnitt 4.6 erläutert. In der Entwicklungsumgebung sind zwei Funktionen implementiert, die auch in Eclipse verfügbar sind:

1. Mit dem Mauszeiger kann über Methoden, Typen, Variablen, statische Variablen, Paketnamen usw. gefahren werden und die Entwicklungsumgebung zeigt ein Popup mit dem entsprechenden Javadoc an. Diese Funktion ist in Abbildung 4.1 zu sehen.
2. Wird eine Variable, Typ, Methode oder ähnliches markiert, werden alle anderen Vorkommen der Variable usw. hervorgehoben.

Abbildung 4.1.: Popup mit dem Javadoc einer statischen Methode in der Entwicklungsumgebung. Oben links wird die Verständlichkeitsmetrik angezeigt.

Bei der Bearbeitung der Aufgaben werden die Teilnehmer ihre Antworten auf dafür vorbereitete Antwortzettel eingetragen. Ein solcher Aufgabenzettel ist in Anhang B zu sehen.

4.4. Aufgabe

Die Aufgabe der Teilnehmer wird in der Einladung wie folgt beschrieben:

> „Du musst kurze in Java geschriebene Methoden verstehen und die Ergebnisse für gegebene Eingabewerte berechnen. Außerdem musst du bestimmen, welches Ergebnis du laut Javadoc erwartet würdest."

[7] https://getbootstrap.com/
[8] https://highlightjs.org/

Teilnehmern werden im Experiment drei Java-Methoden gezeigt. Jede Methode ist eine separate Aufgabe. Das bedeutet, dass Teilnehmer immer nur eine Methode zugleich bearbeiten. Die Aufgabe ist, dass sich Teilnehmer die Methode anschauen, sie verstehen und berechnen, was der Rückgabewert für gegebene Eingabeparameter ist. Zusätzlich dazu sollten die Teilnehmer für die Eingabeparameter bestimmen, welchen Rückgabewert sie laut der Dokumentation erwarten würden. Die Frage nach sowohl dem berechneten Rückgabewert, als auch dem erwarteten Rückgabewert gemäß Dokumentation ist wichtig, da in den Codeschnipseln Bugs implementiert wurden. Die Ungewissheit, wann ein Bug durch einen falschen Rückgabewert sichtbar wird, ist Teil der Aufgabe und soll Teilnehmer dazu bewegen, die Funktionsweise der Methode zu verstehen. Eine solche Aufgabe bewerten wir als realistisch.

Unsere Vorstellung von Codeverstehen ist, dass Codeverstehen auf Methodenebene geschieht, so wie zum Beispiel die Cognitive Complexity auch auf Methodenebene arbeitet. Wir gehen davon aus, dass die Dokumentation korrekt ist, aber im Quellcode Fehler vorhanden sind. Da wir davon ausgehen, dass Codeverstehen ein iterativer Vorgang ist, müssen Teilnehmer für mehrere Eingabeparameter die Ergebnisse bestimmen. Iterativ heißt in diesem Kontext, dass beim ersten Betrachten einer Methode Hypothesen, Annahmen oder Vermutungen über die Methode getroffen werden. Durch syntaktisches oder semantisches Verständnis von Teilen einer Methode werden neue, vielleicht unvollständige, Informationen gewonnen, die zur Bestätigung oder Ablehnung von Hypothesen, Annahmen oder Vermutungen führen. Eine neue Iteration beginnt; neue Hypothesen, Annahmen oder Vermutungen werden getroffen. Die Aufgabe, eine Methode zu verstehen und für Eingabewerte zu bestimmten, ob sich die Methode konform zur Dokumentation verhält, erachten wir als eine häufige Aufgabe eines Programmierers. Dementsprechend sind unsere Aufgabenstellung und die Aufgaben gestaltet: Teilnehmer haben für eine Methode mehrere Eingabewerte gegeben. Nachdem für alle Eingabeparameter der berechnete und erwartete Rückgabewert ermittelt wurde, müssen die Teilnehmer auf einer Likert-Skala von 0 bis 10 bewerten, wie verständlich die gezeigte Methode ist. Der Wert 0 steht für *sehr leicht* verständlich, der Wert 10 für *sehr schwer* verständlich.

4.5. Treatments, Variablen und Hypothesen

Beim Bearbeiten der Aufgabe wird den Teilnehmern eine manipulierte Metrik neben der Methode angezeigt. Den Teilnehmer wird erzählt, dass die angezeigte Metrik sehr zuverlässig arbeitet (vgl. Abschnitt 4.6). Der Metrikwert (*Mw*) ist der den Teilnehmern angezeigte und manipulierte Wert der Verständlichkeitsmetrik. Beim ersten Treatment wird ein Wert von 4 angezeigt, im Folgenden als Mw_4 bezeichnet, und beim zweiten Treatment wird der Wert 8 angezeigt, im Folgenden als Mw_8 bezeichnet. Die Menge aller Teilnehmer wird im Folgenden als $Mw_{4,8}$ bezeichnet. Dies sind unsere beiden Treatments.

Neben dem Metrikwert (*Mw*) gehören demografische Daten und individuelle Charakteristiken, die mit Fragebögen abgefragt werden, zu den unabhängigen Variablen des Experiments. Zu den demografischen Daten gehören Alter (*A*), Geschlecht (*G*), Studienfach (*Sf*), angestrebter akademischer Grad (*Grad*), Fachsemester (*Fs*) und die Anzahl an Jahren der Programmiererfahrung mit Java (*Pe*). Die individuellen Charakteristiken werden über den Big Five, den Life-Orientation-Test und den SPANE-Fragebogen quantifiziert. Diese Variablen der Fragebögen sind für RQ3 von Bedeutung. Zu den abhängigen Variablen gehört die Anzahl der korrekten Rückgabewerte laut Code (*kC*), die Anzahl der korrekten Rückgabewerte laut Dokumentation (*kD*) und die Bearbeitungszeit (*t*). Aus

diesen Werten wird das Codeverständnis (*TAU*) ermittelt. *TAU* ist der Arbeit von Scalabrino et al. [SBV+19] entnommen, da wir diese Berechnung als passende Grundlage sehen und auf Bestehendes zurückgreifen wollen. Da wir andere Variablennamen verwenden, werden diese in der Formel angepasst. Die Semantik wird unserer Einschätzung nach dabei nicht verändert. Berechnet wird das Codeverständnis folgendermaßen:

$$TAU = \frac{kC + kD}{30} \times (1 - \frac{t}{t_{\max}})$$

Der maximale Werte für die Variablen *kC* und *kD* beträgt jeweils 15. Der höchste Wert für die Summe von *kC* und *kD* ist dementsprechend 30. Der Wert für $t_{\max}$ entspricht der Zeit des Teilnehmers, der am längsten für die Bearbeitung benötigt. Dieser Teilnehmer erhält gemäß Formel den Wert $0{,}0$. Der theoretisch höchste Wert ist $1{,}0$. Die Variable *TAU* ist maßgeblich für die Beantwortung von RQ2. Eine weitere abhängige Variable ist die wahrgenommene Schwierigkeit (*wS*). Sie stellt eine Bewertung der Schwierigkeit der Aufgaben durch die Teilnehmer des Experiments dar. Der konkrete Werte für *wS* wird aus dem arithmetischen Mittel der abgegebenen Bewertungen für jede Aufgabe eines Teilnehmers ermittelt. Diese Variable ist für RQ1 entscheidend. Aus der wahrgenommenen Schwierigkeit und dem unabhängigen Metrikwert (*Mw*) wird die Metrikabweichung (*Ma*) abgeleitet, um zu beschreiben, in welchem Maß die Teilnehmer vom vorgegebenen Metrikwert abweichen:

$$Ma = |wS - Mw|$$

Die Metrikabweichung (*Ma*) wiederum ist für die Korrelationsbildung für RQ3 wichtig. In Tabelle C.1 in Anhang C sind alle unabhängigen und abhängigen Variablen des kontrollierten Experiments nach dem Schema aus Jedlitschka et al. [JCP08] aufgeführt. Nachfolgend sind die Hypothesen für die Forschungsfragen aufgeführt:

Hypothesen zu RQ1 Für Forschungsfrage RQ1 werden folgende Hypothesen aufgestellt, dafür wird die abhängige Variable *wS*, die wahrgenommene Schwierigkeit, erfasst:

H_{010}: Die wahrgenommene Schwierigkeit (*wS*) ist nicht signifikant unterschiedlich zwischen den beiden Treatments Mw_4 und Mw_8.

H_{110}: Die wahrgenommene Schwierigkeit (*wS*) ist signifikant unterschiedlich zwischen den beiden Treatments Mw_4 und Mw_8.

Hypothesen zu RQ2 Für diese Forschungsfrage wird die abhängige Variable *TAU* benötigt. Für Forschungsfrage RQ2 werden folgende Hypothesen aufgestellt:

H_{020}: Das Codeverständnis (*TAU*) ist nicht signifikant unterschiedlich zwischen den beiden Treatments Mw_4 und Mw_8.

H_{120}: Das Codeverständnis (*TAU*) ist signifikant unterschiedlich zwischen den beiden Treatments Mw_4 und Mw_8.

Forschungsfrage RQ3 Für Forschungsfrage RQ3 werden keine Hypothesen aufgestellt, da diese Forschungsfrage explorativ untersucht wird. Für diese Forschungsfrage werden der LOT-R-Fragebogen, der SPANE-Fragebogen und der Big Five-Fragebogen erfasst. Ebenso werden alle Variablen des demografischen Fragebogens mit nicht nominalen Skalen wie beispielsweise die Programmiererfahrung mit Java (*Pe*) in die Untersuchung miteinbezogen. Die unabhängigen Variablen können aus Tabelle C.1 entnommen werden. Der Life-Orientation-Test [GHKH08] kommt zum Einsatz, weil Geers et al. [GWF+10], Morton et al. [MWEJ09] und Draganich und Erdal [DE14] einen Zusammenhang mit der Wirksamkeit von Placebos und dispositionellem Optimismus, welcher vom LOT-R quantifiziert wird, sehen. Weiter wird der SPANE-Fragebogen [RHS17] verwendet, da laut Wager und Atlas [WA15] Gefühle eine Rolle bei der Wirksamkeit von Placebos spielen können. Der Big Five-Fragebogen [LLA01] wird verwendet, da Peciña et al. [PAL+13] einen großen Teil der Wirksamkeit von Placebos mit der Persönlichkeit verbinden.

4.6. Forschungsdesign

In dem Experiment gibt es zwei Gruppen, die jeweils ein Treatment erhalten. Das Experiment wird mit dem *between-subjects* Design durchgeführt. Es gibt keine Kontrollgruppe, dies hat mehrere Gründe: Wir rechnen damit, dass wir nicht genügend Teilnehmer akquirieren können, um eine Kontrollgruppe zu bilden. Ebenso wissen wir nicht, ob ein möglicher Placeboeffekt in beide Richtungen wirkt und zusätzlich wollen wir möglichst extreme Werte wählen, weswegen wir zwei Gruppen – eine mit einem niedrigen und eine mit einem hohen Metrikwert – haben wollen. Weiter ist es schwer, die „wahre“ Schwierigkeit der Aufgaben zu bestimmen – uns ist keine Verständlichkeitsmetrik bekannt, die ausreichend validiert ist, um die „wahre“ Schwierigkeit zu ermitteln [SBV+19]. Trotz alledem verwenden wir die Cognitive Complexity bei der Auswahl der Codeschnipsel, damit die Auswahl der Codeschnipsel nachvollziehbar ist und nicht nach subjektivem Schwierigkeitsempfinden ausgewählt wird.

Wie im vorherigen Abschnitt erläutert, gibt zwei Treatments. Bei Mw_4 wird als Metrikwert immer der Wert „4“ neben den Aufgaben angezeigt und bei Mw_8 wird immer der Wert „8“ angezeigt. McRae et al. [MCY+04] haben in einer klinischen Studie Parkinsonpatienten operiert, davon waren einige Operationen nur Scheinoperation – einige Operationen werden nur scheinbar durchgeführt. Die Studie kommt zu dem Ergebnis, dass es bei den Scheinoperationen einen starken Placeboeffekt gibt. Deshalb verwenden wir für die das Treatment Mw_8 einen möglichst hohen Wert – 8 von maximal 10. Davon erhoffen wir uns einen stärkeren Placeboeffekt und für das Treatment Mw_4 einen möglichst niedrigen Wert.[9]

Im Folgenden werden Maßnahmen erörtert, die wir unternehmen, um die Validität des Experiments wahren sollen.

[9] Dieser Gedankengang ist vom Blogeintrag von Jarrett [Jar19] inspiriert.

4.6.1. Maßnahmen zur Wahrung der Konstruktvalidität

Die Zuordnung der Treatments geschieht doppelt-blind. Das bedeutet, dass die Teilnehmer nicht wissen, welches Treatment sie erhalten bzw. nicht wissen, dass es Treatments gibt. Teilnehmern wird vor dem Experiment das eigentliche Ziel des Experiments – die Untersuchung des Placeboeffekts – nicht offenbart. In der Einladung zum Experiment wird geschrieben:

> „Ziel der Studie ist es, zu untersuchen, welche Faktoren das Verständnis von Quellcodes beeinflussen. Du musst kurze in Java geschriebene Methoden verstehen und die Ergebnisse für gegebene Eingabewerte berechnen. […] Darüber hinaus werden während des Experiments durch Fragebögen Daten über Programmiererfahrung etc. erhoben."

Vor Beginn des Experiments wird das Ziel den Teilnehmern ein weiteres Mal erzählt, um Fragen zum Ziel vorzubeugen. Nichtsdestotrotz entspricht das genannte Ziel dem Ziel des Experiments, wenngleich es nur einen Teil des Ziels widerspiegelt. Damit wird versucht zu unterbinden, dass Teilnehmer sich an das erwartete Ergebnis des Experiments anpassen (engl. *hypothesis guessing* [WRH+12]).

Es wird ein umfangreiches Skript zur Durchführung des Experiments erstellt, damit sich der Experimentleiter gegenüber allen Teilnehmern gleich verhält (engl. *experimenter expectancies* [WRH+12]). Außerdem wird das Experiment ausschließlich von einer Person betreut. Damit der Experimentleiter die Teilnehmer der zwei Gruppen nicht bewusst oder unbewusst unterschiedlich behandelt, werden die Treatments den Gruppen auf eine Art zugewiesen, sodass keine der involvierten Personen die Zuordnung kennt. Realisiert wird dies, indem der Experimentleiter vor der Durchführung für jeden Teilnehmer beide Treatments vorbereitet: Das eine Treatment, hier die HTML-Dateien, wird in einem Ordner *A* abgelegt und das andere Treatment in einem Ordner *AA* gespeichert. Der Betreuer dieser Arbeit ordnet jeden Teilnehmer eine Variante zu, indem der eine Ordner gelöscht wird und der Inhalt des anderen Ordner an eine entsprechende Stelle kopiert wird – ohne dabei zu wissen, welches Treatment welche Variante ist. Die Zuordnung, welche dem Experimentleiter erst nach Ende des Experiments offenbart wird, ist in Tabelle 4.1 zu sehen.

Tabelle 4.1.: Zuordnung von Treatments und Zeitfenstern.

	Tag 1	Tag 2	Tag 3	Tag 4	Tag 5	Tag 6
Zeitfenster 1, Teilnehmer A	AA	A	AA	AA	A	A
Zeitfenster 1, Teilnehmer B	AA	A	AA	AA	A	A
Zeitfenster 2, Teilnehmer A	AA	A	AA	A	A	AA
Zeitfenster 2, Teilnehmer B	AA	A	AA	A	A	AA
Zeitfenster 3, Teilnehmer A	A	A	A	AA	AA	AA
Zeitfenster 3, Teilnehmer B	A	A	A	AA	AA	AA
Zeitfenster 4, Teilnehmer A	AA	AA	A	A	AA	—
Zeitfenster 4, Teilnehmer B	AA	—	A	A	AA	—

Damit sich Teilnehmer nicht analysiert fühlen, was Wohlin et al. [WRH+12] als *evaluation apprehension* bezeichnen, wird darauf geachtet, dass der Experimentleiter während des Experiments erkennbar nicht in der Lage ist, die Antworten und Lösungen der Teilnehmer zu sehen. Die konkreten Maßnahmen, wie beispielsweise die Anordnung der Tische, ist in Abschnitt 4.7 erläutert. Ferner werden der Big Five-Fragebogen und der LOT-R-Fragebogen nach der Bearbeitung der Aufgaben erfasst, damit sich Teilnehmer aufgrund der persönlichen Fragen nicht analysiert fühlen.

4.6.2. Maßnahmen zur Wahrung der internen Validität

Dadurch, dass die zwei Teilnehmer eines Zeitfensters immer das gleiche Treatment erhalten, schließen wir aus, dass wenn ein Teilnehmer unaufgefordert den Metrikwert verrät, der andere Teilnehmer verwundert ist, dass bei ihm ein anderer Metrikwert angezeigt wird (engl. *diffusion or imitation of treatments* [WRH+12]).

Wager und Atlas [WA15] schreiben, dass Gefühle, Motivation und Affektzustände Einflussfaktoren von Placeboeffekten sind. Da der Verlauf des Experiments potenziell einen Einfluss auf die aktuellen Gefühle der Teilnehmer haben könnte, wird der SPANE-Fragebogen vor Bearbeitung der Aufgaben ausgefüllt. Um mögliche Gedanken oder Fragen über den Sinn und Zweck der Fragebögen zu unterbinden, werden der Big Five-Fragebogen und LOT-R Fragebogen aber erst nach Bearbeitung der Aufgaben erfasst. Wenn ein Teilnehmer für die Bearbeitung einer Aufgabe länger braucht, könnte sich dies auf die Gefühle der Teilnehmer auswirken. Deshalb wird Teilnehmern gesagt, dass sie beliebig lange für die Bearbeitung jeder Aufgabe haben. Damit soll vermieden werden, dass Teilnehmer Zeitdruck empfinden. In Wirklichkeit gibt es aber ein Zeitlimit für jede Aufgabe. Um die Zeitlimits zu bestimmten, haben einige Freiwillige die Aufgaben auf Zeit gerechnet; die verfügbare Zeit wird im Verhältnis der ermittelten Zeit auf die Aufgaben verteilt. Für die erste Aufgabe sind 15 Minuten vorgesehen, für Aufgabe 2 sind 50 Minuten und für die letzte Aufgabe sind 28 Minuten vorgesehen. Wenn ein Teilnehmer dieses Zeitlimit überschreitet, wird er gefragt: „Möchtest du mit der nächsten Aufgaben weitermachen?“. Diese Maßnahmen werden zur Wahrung der internen Validität (engl. *maturation* [WRH+12]) ergriffen.

Es wird darauf verzichtet, dass Teilnehmer vor Bearbeitung der Aufgaben Beispielaufgaben rechnen. Damit erschweren wir es Teilnehmern, Aufgaben miteinander zu vergleichen, denn für jede Beispielaufgabe hätten die Teilnehmer einen zusätzlichen Referenzpunkt zum Vergleichen der Schwierigkeit. Wir wollen nicht, dass Teilnehmer die Bewertung abhängig von anderen Beispielaufgaben bzw. Aufgaben machen. Um anderseits sicherzustellen, dass Teilnehmer die Aufgabenstellung richtig verstehen, werden den Teilnehmern Beispielaufgaben gezeigt (vgl. Abschnitt 4.7). Weiter sind die Codeschnipsel A.2, A.3, A.4 und A.5 gemäß Cognitive Complexity als gleich komplex bewertet (vgl. Abschnitt 4.3). Auf diesen Umstand haben wir Wert gelegt, damit es Teilnehmern möglichst erschwert wird, die Methoden miteinander zu vergleichen. So soll verhindert werden, dass Teilnehmer auf das Treatment mit der Zeit anders reagieren (engl. *maturation* [WRH+12]). Da wir Teilnehmer dahingehend manipulieren wollen, dass sie den angezeigten Metrikwerten glauben, könnte ein Vergleich der Aufgaben dazu führen, dass Teilnehmer an der Glaubwürdigkeit der Metrik zweifeln. Wager und Atlas [WA15] beschreiben unterschiedliche Einflussfaktoren auf die Wirksamkeit von Placebos, darunter *Erwartungen* und *Bewertungen*. Deshalb ist es wichtig, die Glaubwürdigkeit unserer Metrik sicherzustellen. Draganich und Erdal [DE14] haben in ihrem Experiment ebenso Maßnahmen ergriffen, um ihre Teilnehmer von der Glaubwürdigkeit ihres Placebos zu überzeugen.

Wir versuchen, in Teilnehmern die *Erwartung* zu wecken, dass die Metrik zuverlässig ist. Um die Teilnehmer von der Zuverlässigkeit der Metrik zu überzeugen, wird den Teilnehmern von einer wissenschaftlichen Publikation berichtet, die bestätigt, dass die Metrik *sehr gut* funktioniere:

> „Es gibt eine neue Metrik, die mit maschinellem Lernen arbeitet, die sehr gut misst, wie verständlich Quellcode ist. Eine gibt eine Publikation dazu, die nahelegt, dass das Codeverständnis der Entwickler und die Werte diese Metrik sehr gut übereinstimmen." (Siehe Anhang D)

Um die Glaubwürdigkeit der Metrik weiter zu untermauern, wird den Teilnehmer am Codeschnipsel A.2 gezeigt, dass die Metrik zuverlässig arbeitet. Dieses Beispiel wurde im Vorfeld ausgewählt, da es kurze Bezeichner verwendet und beispielsweise bitweise Verschiebungen verwendet. Weiter werden bei diesem Beispiel der Methodenname verändert und hilfreiche Kommentare entfernt. Weitere Details dazu in Abschnitt 4.7.3. Auf möglicherweise kritische Rückfragen zur Metrik werden Antworten ausgearbeitet, welche die Kritik möglichst neutralisieren. Beispielsweise soll auf die eventuelle Frage, warum die Metrik angezeigt wird, wenn die Codeschnipsel doch vom Teilnehmer bewertet werden sollen, geantwortet werden: „Die Metrik wird nur zur Information angezeigt."

Es soll verhindert werden, dass Teilnehmer sich Quellcode außerhalb der zu verstehenden Methode anschauen, da Metriken, wie zum Beispiel die Cognitive Complexity, auf Methodenebene arbeiten. Weiter soll die Erfahrung mit einer spezifischen Entwicklungsumgebung oder praktische Erfahrung beim Debuggen von Programmcode nicht unkontrolliert auf die abhängigen Variablen Einfluss nehmen (engl. *instrumentation* [WRH+12]), deshalb haben wir eine eigene minimale Entwicklungsumgebung, wie sie in Abschnitt 4.3 beschrieben ist, für unser Experiment implementiert.

4.6.3. Maßnahmen zur Wahrung der externen Validität

Wie im vorherigen Abschnitt dargelegt, wird auf eine vollständige Entwicklungsumgebung verzichtet, da wir die Entwicklungsumgebung nicht kontrollieren können. Alternativ könnte der Quellcode ausgedruckt werden. Darin sehen wir ein Problem für die Generalisierbarkeit, denn Quellcode wird in den seltensten Fällen während der Programmierung ausgedruckt. Dies legt erneut nahe, eine simple Entwicklungsumgebung zu entwickeln (engl. *interaction of setting and treatment* [WRH+12]). Weiter wird versucht, eine Aufgabenstellung zu formulieren, die sich außerhalb unseres Experiments in der Praxis wiederfinden lässt. Wir sehen davon ab, Codeschnipsel ohne sinnvolle Methodennamen, ohne Dokumentation und mit obfuskierten Variablenbezeichnern zu verwenden, da wir solche Aufgaben als nicht realistisch bewerten. Eine nähere Beschreibung unserer Aufgabenstellung ist in Abschnitt 4.4 zu finden. Ferner wird ein Szenario konstruiert und die Teilnehmer werden gebeten, sich in den Kontext des Szenarios „hinein zu fühlen" und es sich vorzustellen. Das Szenario wird formuliert, damit die Aufgabenstellung einen Bezug zur Realität und Alltag eines Softwareentwicklers hat. Das Szenario ist in Abschnitt 4.7.3 beschrieben. Mit diesen Maßnahmen wollen wir die Generalisierbarkeit stärken. Die Treatments werden über die Dauer der Durchführung gleichmäßig auf die Tageszeiten verteilt, so soll ein Einfluss der Tageszeit auf die Ergebnisse vermieden werden (engl. *interaction of history and treatment* [WRH+12]).

4.6.4. Maßnahmen zur Wahrung der Validität von Schlussfolgerungen

Um die Validität der Schlussfolgerungen zu wahren (engl. *random irrelevances in experimentation subjects* [WRH+12]), wird der Raum für den gesamten Zeitraum reserviert. Die Reservierung und die Bitte, nicht zu stören, werden auf einem A4-Hinweisschild ungefähr eine Woche vor Beginn des Experiments, ausgehängt. Das heißt, dass die Teilnehmer und der Experimentleiter während der Durchführung ungestört im Raum sind.

4.7. Vorgehensweise bei der Durchführung

Das Experiment wird in einem studentischen Arbeitsraum der Software Engineering Abteilung durchgeführt. Der Arbeitsraum wird für den Zeitraum der Durchführung reserviert. Im Arbeitsraum wird jedem Teilnehmer – immer zwei in einem Zeitfenster – jeweils ein Tisch zur Verfügung stehen. In Abbildung 4.2 ist eine Übersicht des Arbeitsraumes zu sehen.

Abbildung 4.2.: Übersicht über den studentischen Arbeitsraum während der Durchführung. Links sitzen die Teilnehmer, rechts sitzt der Experimentleiter.

4.7.1. Einleitung und Einweisung

Vor Beginn eines Zeitfensters werden die Teilnehmer im Wartebereich vor dem Arbeitsraum warten. Wenn beide Teilnehmer eingetroffen sind und die Vorbereitung abgeschlossen ist, werden die Teilnehmer in den Raum gebeten. Ihnen wird die Tür aufgehalten, sie werden hereingebeten und begrüßt. Die zwei freien Plätze vor den Laptops werden ihnen angeboten (vgl. Abbildung 4.3). Der Experimentleiter wird zwischen den zwei Teilnehmern Platz nehmen. Den Teilnehmern wird ein Überblick über das Ziel und die Aufgaben des Experiments gegeben. Anschließend wird den Teilnehmern eine Einverständniserklärung ausgehändigt, welche mündlich zusammengefasst wird. Außerdem wird erklärt, welche Daten erfasst werden. Die Teilnehmer werden mit dem Laptop und den Dateien auf dem Laptop vertraut gemacht. Im nächsten Schritt werden die Teilnehmer gebeten, zwei Fragebögen auszufüllen. Zum einen werden demografische Daten abgefragt, zum anderen wird SPANE abgefragt. Teilnehmer werden gebeten, die Fragebögen selbstständig zu speichern und den Fragebogen zu schließen, damit sich Teilnehmer beim Ausfüllen nicht kontrolliert oder überprüft fühlen. Der Experimentleiter wird währenddessen an dem Tisch gegenüber (vgl. Tisch im Hintergrund in Abbildung 4.3) verweilen. So können weder der Experimentleiter noch die Teilnehmer gegenseitig sehen, was in die Fragebögen eingetragen wird. Nachdem die Teilnehmer die ersten zwei Fragebögen ausgefüllt haben, wird den Teilnehmern ein Überblick über den weiteren Verlauf gegeben.

Abbildung 4.3.: Anordnung der Laptops zu Beginn des Experiments, diese ändert sich im Laufe der Durchführung.

4.7.2. Erklärung der Aufgabenstellung

Den Teilnehmern wird die Aufgabenstellung erklärt: Es werden Java-Methoden gezeigt, die Teilnehmer sollen sich die Methoden gründlich anschauen, verstehen und die Rückgabewerte für gegebene Eingabeparameter berechnen. Außerdem soll für jeden Eingabeparameter bestimmt werden, welchen Rückgabewert man laut Dokumentation erwarten würde. Es wird darauf hingewiesen, dass im Quellcode Fehler vorhanden sein können, sodass es zu einer Abweichung zwischen diesen zwei Werten kommen kann. Außerdem wird darauf hingewiesen, dass die Bearbeitungszeiten der Aufgaben variieren können. Den Teilnehmern wird ein vollständig ausgefüllter Aufgabenzettel für eine Beispielmethode (siehe Anhang B) gezeigt. Es wird die Struktur des Aufgabenzettels erklärt: Wo sind Eingabeparameter zu finden und in welche Zelle welche Antworten notiert werden sollen. Die Lösung wird beispielhaft für die ersten zwei Eingabeparameter berechnet. In eine Zelle ist „Weiß ich nicht“ eingetragen, um den Teilnehmern zu zeigen, dass sie nicht gezwungen sind, alles auszufüllen. Auf diese Möglichkeit wird aber nicht explizit hingewiesen.

4.7.3. Erklärung des Kontextes

Weiter wird den Teilnehmern der Kontext präsentiert. Die Teilnehmer werden gebeten, sich vorzustellen, dass sie als Softwareentwickler in einem Unternehmen arbeiten und ihre Haupttätigkeit daraus besteht, Quellcode zu verstehen und Fehler zu beheben. Es wird erzählt, dass es oft vorkommt, dass die Dokumentation einer Methode nicht dem entspricht, was eine Methode berechnet – das heißt, im Quellcode sind Bugs. Den Teilnehmern wird im Anschluss die Entwicklungsumgebung am Beispiel (siehe Listing A.1), zu dem sie bereits den Aufgabenzettel gesehen haben, gezeigt und erklärt: Sie konnten die Funktionen der Entwicklungsumgebung ausprobieren – so wird sichergestellt, dass alle Teilnehmer die Entwicklungsumgebung bedienen können. Weiter wird die Visualisierung der Metrik kurz erläutert (vgl. Anhang D). Anschließend wird an einem zweiten, schwerer bewerteten Beispiel (vgl. Listing A.2) die Metrik erklärt. Dies wird gemacht, um den Teilnehmern glaubhaft zu machen, dass die Metrik sehr gut funktioniert. Den Teilnehmer wird erklärt, warum die Metrik das zweite Beispiel mit dem Wert 9 bewertet:

> „Hier sieht man gut, dass dessen berechnete Werte sehr gut mit der Verständlichkeit der Methode passen. Wir haben hier ein eher schweres Beispiel [mit dem Wert 9]. Die Variablennamen sind eher kurz und die Kommentare sind kryptisch bzw. weniger hilfreich.“ (Vgl. Anhang D).

4.7.4. Wiederholung der Aufgabenstellung und letzte Instruktionen

Anschließend wird die Aufgabenstellung wiederholt und letzte Anweisungen gegeben: Es wird darauf hingewiesen, dass sie für die Bearbeitung der Aufgaben so viel Zeit haben, wie sie benötigen. Ihnen wird berichtet, dass ihr Arbeitgeber wollen würde, dass sie möglichst effizient sind, aber auch fehlerfrei arbeiten. Der Experimentleiter erklärt, dass manche Aufgaben schneller oder langsamer erledigt sind und dass während der Bearbeitung der Aufgaben keine Fragen beantwortet werden können bzw. Fragen jetzt gestellt werden sollen.

4.7.5. Bearbeitung der Aufgaben

Der Experimentleiter wird den Laptop des einen Teilnehmers auf den zweiten Tisch (in Abbildung 4.4 rechts) stellen, sodass sich die Teilnehmer gegenüber sitzen, wie in Abbildung 4.4 zu sehen ist. Zwischen den beiden Tischen ist ein Gang, den der Experimentleiter betreten kann, um den Teilnehmern die neue Aufgabe zu überreichen. So wird sichergestellt, dass der Experimentleiter nicht sieht, welches Treatment die Teilnehmer bekommen. An dem Tisch, in Abbildung 4.4 vorne,

Abbildung 4.4.: Zwei Teilnehmer bei der Bearbeitung einer Aufgabe.

sitzt der Experimentleiter an einem Laptop mit zweitem Bildschirm und erfasst die Bearbeitungszeiten. Laptop und zweiter Bildschirm sind auf Abbildung 4.4 beiseite geräumt worden. Direkter Sichtkontakt zu den Arbeitsplätzen der Teilnehmer wird soweit möglich vermieden.

Wenn die Teilnehmer bereit sind, startet der Experimentleiter die Zeiterfassung und die Teilnehmer beginnen die Bearbeitung der ersten Aufgabe. Wenn ein Teilnehmer mit einer Aufgabe fertig ist, wird die Zeit für den Teilnehmer gestoppt. Die Zeiten pro Teilnehmer und pro Aufgabe werden in LibreOffice Calc erfasst. Um bei der Zeiterfassung möglichst akkurat zu sein, wird der aktuelle Zeitstempel mit dem Tastenkürzel `Strg+Shift+;` eingefügt.

4.7.6. Big Five und LOT-R

Nachdem ein Teilnehmer alle drei Aufgaben bearbeitet hat, wird er gebeten, die letzten zwei Fragebögen auszufüllen, selbstständig zu speichern und LibreOffice Calc zu schließen.

Anschließend darf sich der Teilnehmer eine Süßigkeit nehmen und sich für den Schein der Vorlesung *Forschungsmethoden der Softwaretechnik* in eine Liste eintragen. Abschließend wird der Teilnehmer verabschiedet und verlässt den Raum. Ein vollständiges Skript für die Durchführung ist in Anhang D zu finden.

4.8. Vorgehen bei der Analyse

Die Antworten und Bewertungen der Teilnehmer werden von den Teilnehmern analog erfasst. Die Bearbeitungszeiten werden manuell vom Experimentleiter in LibreOffice Calc erfasst. Die Antworten und Bewertungen werden nach Ende des Experiments in diese Tabelle eingetragen. Diese Tabelle wird von einem Python-Programm in eine CSV-Datei überführt, welche nicht mehr zwischen den Aufgaben unterscheidet; dabei wird wahrgenommene Schwierigkeit (*wS*) aus den Bewertungen für alle Aufgaben durch das arithmetische Mittel bestimmt und das Codeverständnis (*TAU*) berechnet. Die Fragebögen werden digital erfasst. Die Antworten in den Fragebögen werden nicht händisch, sondern durch dokumentenübergreifende Verweise[10] übertragen; so sollen Übertragungsfehler vermieden werden. Schlussendlich wird die Metrikabweichung (*Ma*) ermittelt und der Datensatz um diesen Wert erweitert.

Für die Forschungsfragen RQ1, RQ2 und RQ3 wird jeweils überprüft, ob die Daten normalverteilt sind. Je nach Verteilung wird für RQ1 und RQ2 mit dem t-Test oder dem U-Test überprüft, ob sich die zwei Stichproben signifikant unterscheiden. Bei signifikanten Unterschieden wird mit Cohens d die Effektstärke berechnet. Beim Testen der Hypothesen wird auf ein Signifikanzniveau von $\alpha = 0{,}05$ getestet.

Für RQ3 werden keine Hypothesen aufgestellt: Die Daten für diese Forschungsfragen werden explorativ ausgewertet. Es werden Korrelationsmatrizen mit der Metrikabweichung (*Ma*) und den Daten der Fragebögen erstellt. Aus der Matrix werden solche Variablen mit nominalen Skalen (*G*, *Sf* und *Grad*) entfernt. Die Metrikabweichung (*Ma*) ist jene Variable, für welche nach Korrelationen gesucht wird. Falls der Datensatz eines Teilnehmers nicht vollständig sein sollte, werden die Daten des Teilnehmers nicht bei der Untersuchung für RQ3 berücksichtigt.[11] Es wird an entsprechenden Stellen auf die Größe der untersuchten Stichprobe hingewiesen. Wenn möglich, werden die Korrelationen nach Pearson berechnet, ansonsten nach Spearman. Bei der Untersuchung von RQ3 werden nicht nur die Daten aller Teilnehmer ($Mw_{4,8}$) untersucht, sondern es wird auf unterschiedliche Korrelationen bei den zwei Treatments Mw_4 und Mw_8 geprüft, um Erkenntnisse zu gewinnen, wie sich die Treatments unterscheiden. Bei der Analyse und Interpretation muss beachtet werden, dass wir über keine Kontrollgruppe verfügen (vgl. Abschnitt 4.6). Für die mit der Metrikabweichung (*Ma*) *korrelierenden Variablen* wird mit dem t-Test bzw. U-Test überprüft, ob sich die beiden Treatments Mw_4 und Mw_8 bezüglich der *korrelierenden Variablen* nicht signifikant unterscheiden. Bei dieser Forschungsfrage untersuchen wir explorativ viele Variablen. Wir vermeiden es, von Signifikanz zu sprechen. Wir verstehen Korrelationen mit einem p-Wert kleiner als dem Signifikanzniveau von $\alpha = 0{,}05$ als Indikator, dass eine Variable für spätere Untersuchungen interessant sein könnte.

[10] https://help.libreoffice.org/Calc/Referencing_a_Cell_in_Another_Document/de

[11] Während die Daten solcher Teilnehmer im Kapitel 5 nicht in den entsprechenden Berechnungen berücksichtigt werden, werden die Daten in den Korrelationsmatrizen im Anhang nicht vollständig entfernt. In den Matrizen wird die `rcorr`-Funktion des `Hmisc`-Pakets verwendet, das paarweise fehlende Werte, aber nicht den kompletten Datensatz, entfernt.

4.9. Zusammenfassung

In diesem Kapitel wurde die Methodik des Experiments beschrieben. An dem Experiment haben 45 Teilnehmer teilgenommen. Die Teilnehmer mussten Java-Methoden verstehen und für gegebene Eingabewerte die Rückgabewerte berechnen. Die Teilnehmer wurden in zwei Gruppen aufgeteilt und jeder Gruppe wurde ein anderer Metrikwert angezeigt. Der manipulierte Metrikwert war unser Treatment. Es wurden Forschungsfragen vorgestellt und begründet. Mit dem Experiment wurde untersucht, ob sich Teilnehmer in ihrer subjektiven Bewertung beeinflussen lassen (RQ1) und sich das Codeverständnis der Teilnehmer verändern lässt (RQ2). Weiter wurden explorativ individuelle Charakteristiken untersucht, die einen Einfluss auf diese Effekte haben könnten (RQ3). Im nachfolgenden Kapitel werden die Ergebnisse zu diesen Forschungsfragen präsentiert.

5. Ergebnisse

In diesem Kapitel werden die Ergebnisse des Experiments präsentiert. Als Erstes werden die Teilnehmer des Experiments beschrieben. Weiter werden zu berücksichtigende Aspekte bei der Auswertung diskutiert. Anschließend werden zuerst die Ergebnisse für RQ1, dann für RQ2 und abschließend für RQ3 präsentiert und eine Antwort auf die entsprechende Forschungsfrage gegeben.

An dem Experiment haben 45 Teilnehmer partizipiert. 22 Teilnehmer haben das Treatment Mw_4 erhalten. 23 Teilnehmer haben das Treatment Mw_8 erhalten. Von den 45 Teilnehmern studierten 43 Softwaretechnik und zwei Teilnehmer studierten Informatik. Insgesamt waren 35 Studierende im Master und zehn im Bachelor eingeschrieben. Das Durchschnittsalter beträgt im Median 24 Jahre. Die Standardabweichung beträgt $2{,}78$ Jahre. Von den Teilnehmern waren 43 männlich, zwei weiblich und niemand divers. Die Teilnehmer aus Mw_4 und Mw_8 haben im Median jeweils eine Programmiererfahrung mit Java von 6 Jahren. Das arithmetische Mittel der Programmiererfahrung mit Java beträgt für $Mw_{4,8}$: $5{,}7$ Jahre, für Mw_4: $5{,}1$ Jahre und für Mw_8: $6{,}3$ Jahre.

Anderes als geplant werden die Fachsemester (*Fs*) nicht ausgewertet, da die erfassten Daten nicht plausibel sind. Einige Masterstudierende haben nicht das Fachsemester angegeben, sondern ihr aktuelles Mastersemester. Zwei Teilnehmer mit dem Treatment Mw_4 haben jeweils einen Fragebogen nicht korrekt ausgefüllt. Ein Teilnehmer hat den SPANE-Fragebogen nicht korrekt ausgefüllt. Dieser Teilnehmer wird nicht bei Korrelationen mit den Variablen SPANE-P, SPANE-N und SPANE-B berücksichtigt. Ein anderer Teilnehmer hat den Big Five-Fragebogen nicht korrekt ausgefüllt. Dieser Teilnehmer wird bei Korrelationen mit Variablen aus dem Big Five nicht berücksichtigt.

5.1. Ergebnisse zu RQ1

Forschungsfrage RQ1: Hat eine Manipulation der Verständlichkeitsmetrik einen Einfluss auf die subjektive Bewertung der Codeverständlichkeit?

Für diese Forschungsfrage ist die wahrgenommene Schwierigkeit (*wS*) relevant. In Tabelle 5.1 sind die gemessenen Werte aller Teilnehmer ($Mw_{4,8}$) und der zwei Treatments zu sehen. Das mögliche

Tabelle 5.1.: Deskriptive Statistik für die wahrgenommene Schwierigkeit (*wS*).

	Minimum	1. Quantil	Median	Mean	3. Quantil	Maximum
$Mw_{4,8}$, $n = 45$	2,333	4,667	6,000	5,978	7,333	9,667
Mw_4, $n = 22$	2,333	4,083	4,667	4,970	5,583	7,667
Mw_8, $n = 23$	4,333	6,167	6,667	6,942	7,833	9,667

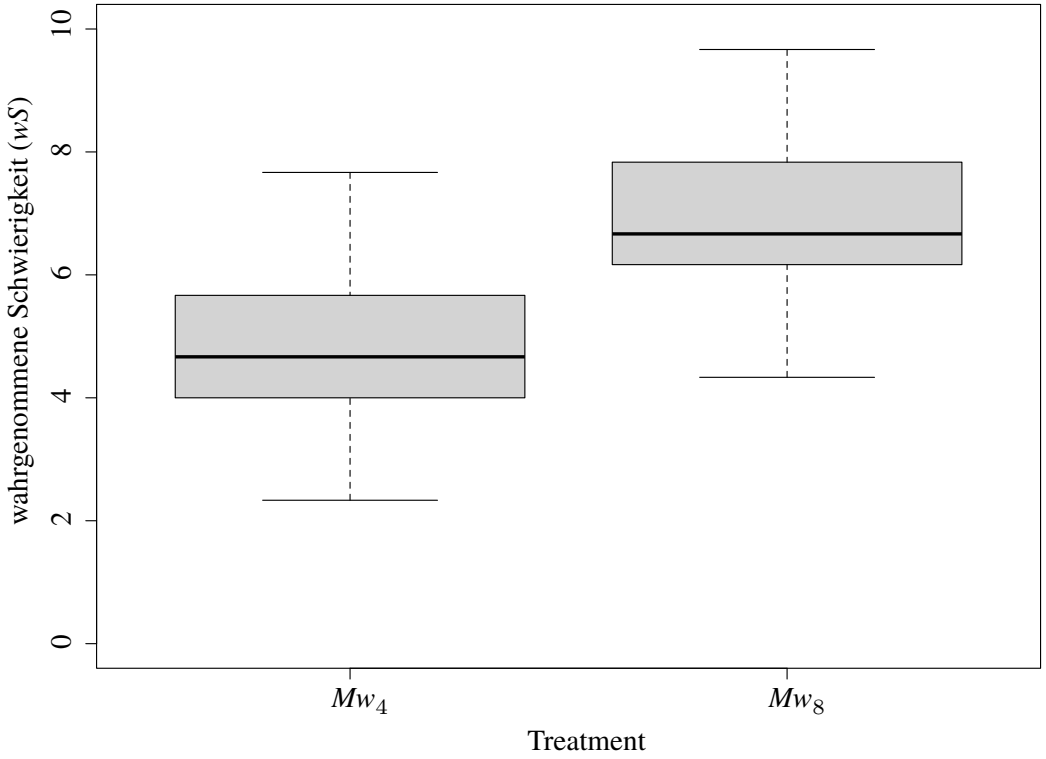

Abbildung 5.1.: Die wahrgenommene Schwierigkeit (*wS*), gruppiert nach den Treatments Mw_4 und Mw_8.

Minimum für die wahrgenommene Schwierigkeit ist 0, das mögliche Maximum 10. Der Wert 0 steht für *sehr leicht* verständlich, der Wert 10 für *sehr schwer* verständlich. Die Boxplots für die zwei Treatments sind in Abbildung 5.1 abgebildet. Es ist ein signifikanter Unterschied zwischen den zwei Treatments zu erkennen. Die Mediane liegen außerhalb der 25 %- und 75 %-Quantile des anderen Boxplots. Die wahrgenommene Schwierigkeit (*wS*) ist gemäß des Shapiro-Wilk-Tests (siehe Tabelle E.1) nicht normalverteilt, das heißt, der U-Test wird verwendet. Mit dem U-Test wurde ein p-Wert von $0{,}0001733$ ermittelt. Cohens d beträgt $d = |-1{,}376| = 1{,}376$. Da $d > 1{,}2$ ist, liegt nach Sawilowsky [Saw09] ein sehr großer Effekt vor.

Antwort zur Forschungsfrage RQ1: Da sich die wahrgenommene Schwierigkeit (*wS*) bei den zwei Treatments signifikant unterscheidet ($p = 0{,}0001733$), wird die Nullhypothese H_{010} abgelehnt und die Alternativhypothese H_{110} akzeptiert. Der beobachtete Effekt ist sehr groß ($d = 1{,}376$).

5.2. Ergebnisse zu RQ2

Forschungsfrage RQ2: Hat eine Manipulation der Verständlichkeitsmetrik einen Einfluss auf das tatsächliche Codeverständnis?

Ausschlaggebend für die Ergebnisse für die Forschungsfrage sind folgende Variablen: korrekte Rückgabewerte laut Code (*kC*), korrekte Rückgabewerte laut Dokumentation (*kD*), Bearbeitungszeit (*t*) und das aggregierte Codeverständnis (*TAU*). Zur Beantwortung der Forschungsfrage wird nur *TAU* benötigt. Aus den Tabellen E.2 bis E.5 und Abbildung 5.2 für das Codeverständnis (*TAU*), das die Werte *kC*, *kD* und *t* aggregiert, geht hervor, dass es zwischen den Treatments nur marginale Unterschiede gibt. Es ist zu erkennen, dass es keinen signifikanten Unterschied zwischen den zwei Treatments gibt. Das Codeverständnis (*TAU*) ist gemäß des Shapiro-Wilk-Tests (siehe Tabelle E.1) nicht normalverteilt, das heißt, der U-Test wird verwendet. Der p-Wert des U-Tests beträgt $0{,}794$.

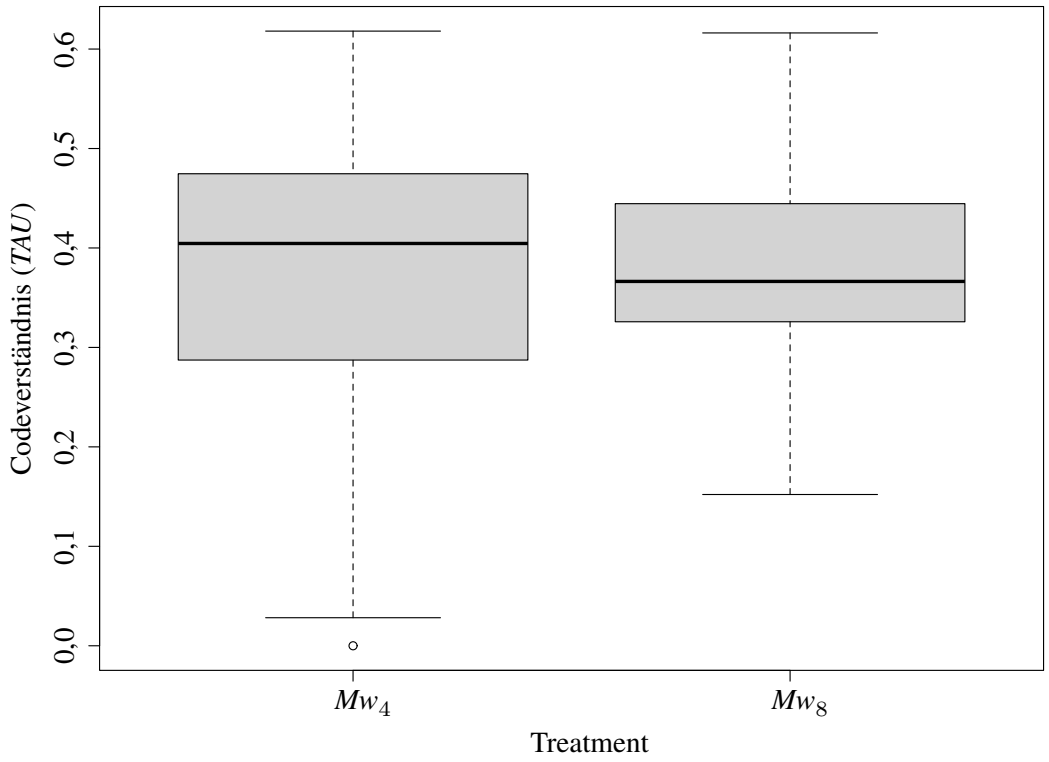

Abbildung 5.2.: Das Codeverständnis (*TAU*), gruppiert nach den Treatments Mw_4 und Mw_8.

Antwort zur Forschungsfrage RQ2: Das Codeverständnis (*TAU*) unterscheidet sich zwischen den beiden Treatments nicht signifikant ($p = 0{,}794$), deshalb wird die Nullhypothese H_{020} nicht abgelehnt.

5.3. Ergebnisse zu RQ3

Forschungsfrage RQ3: Welchen Einfluss haben individuelle Charakteristiken auf den Grad der Abweichung vom angezeigten und manipulierten Wert einer Verständlichkeitsmetrik?

Bei dieser Forschungsfrage gehen wir explorativ vor. Wir sind uns bewusst, dass wir viele Variablen untersuchen, deshalb sprechen wir bezüglich RQ3 nicht von Signifikanz. Die hier berichteten p-Werte verstehen wir als Indikator; alles unterhalb des Schwellwerts von $\alpha = 0{,}05$ könnte für spätere Untersuchungen interessant sein. Zur Untersuchung von RQ3 werden drei Korrelationsmatrizen (siehe Tabellen E.7 bis E.9) über die Daten der Fragebögen und der Metrikabweichung erstellt, jeweils eine für $Mw_{4,8}$, Mw_4 und Mw_8. Da die Metrikabweichung (*Ma*) nicht normalverteilt ist (siehe Tabelle E.1), werden die Korrelationsmatrizen nach Spearman erstellt. Weiter sind keine der untersuchten Variablen normalverteilt, deshalb wird im Folgenden der U-Test mit $W_{\text{kritisch}} = 170$[1] verwendet, um zu überprüfen, ob sich die unabhängigen Variablen signifikant[2] unterscheiden. Bei den U-Tests werden bei *den korrelierenden unabhängigen Variablen* aus Tabelle 5.2 keine signifikanten Unterschiede zwischen den zwei Treatments gefunden. Die deskriptive Statistik und Boxplots für die Metrikabweichung sind in Tabelle E.6 im Anhang bzw. Abbildung 5.3 zu finden. In

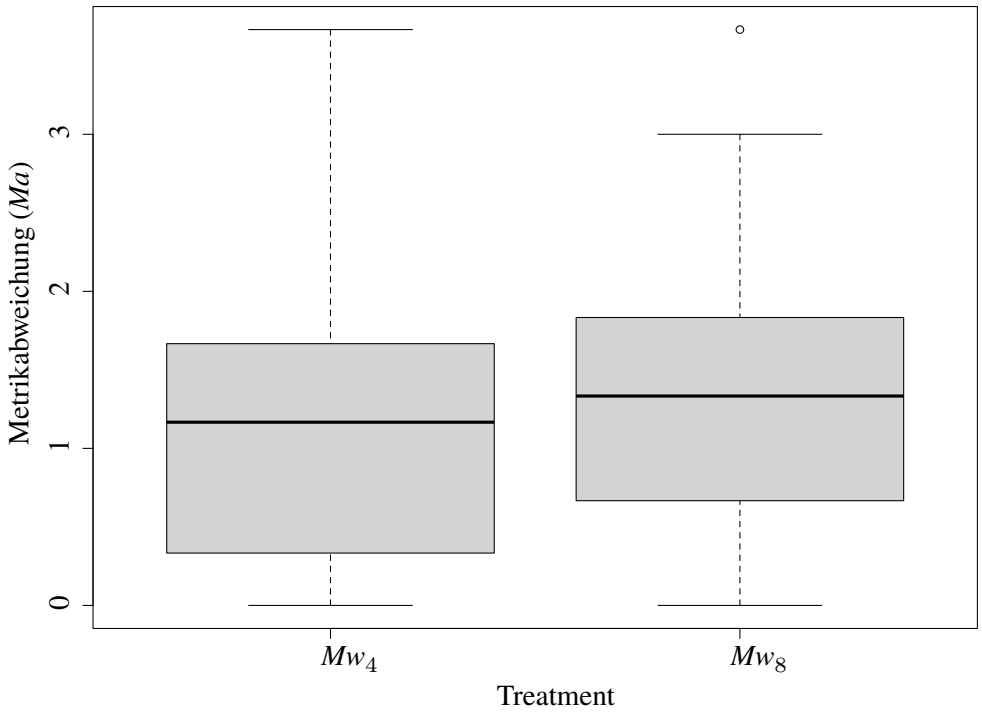

Abbildung 5.3.: Die Metrikabweichung (*Ma*), gruppiert nach den Treatments Mw_4 und Mw_8

[1] https://uni-salzburg.at/fileadmin/oracle_file_imports/513573.PDF

[2] Hier geht es nicht um die Signifikanz der Korrelationen, sondern darum zu überprüfen, ob die in der Korrelation *untersuchten unabhängigen Variablen zwischen den zwei Treatments* unterschiedlich sind.

Tabelle 5.2 sind Korrelationen mit der Metrikabweichung (*Ma*) abgebildet. Jeweils ein Teilnehmer hat den Big Five und den SPANE-Fragebogen nicht korrekt ausgefüllt. In beiden Fällen waren die Teilnehmer aus dem Treatment Mw_4. Der Teilnehmer, der den Big Five-Fragebogen nicht korrekt ausgefüllt hat, wird bei Korrelationen mit Variablen aus dem Big Five-Fragebogen ausgeschlossen. Analoges gilt für den Teilnehmer, der den SPANE-Fragebogen nicht korrekt ausgefüllt hat. Es gilt:

$$* \Rightarrow p < 0{,}05$$

Tabelle 5.2.: Korrelationen mit der Metrikabweichung (*Ma*).

	r_s		
Variable	$Mw_{4,8}$, $n = 44$	Mw_4, $n = 21$	Mw_8, $n = 23$
Entgegenkommen (*Ek*)	0,32*	0,51*	0,11
Ängstlichkeit (*An*)	−0,31*	−0,34	−0,30
negative Gefühle (SPANE-N)	−0,26	−0,51*	0,05
emotionale Balance (SPANE-B)	0,16	0,48*	−0,20

Antwort zur Forschungsfrage RQ3: Unter allen Teilnehmern ($Mw_{4,8}$, $n = 44$) kann eine schwache Korrelation der Metrikabweichung mit Entgegenkommen (*Ek*) und Ängstlichkeit (*An*) beobachtet werden. Für Entgegenkommen ist die Korrelation positiv ($r_s(Mw_{4,8}) = 0{,}32$) und für Ängstlichkeit negativ ($r_s(Mw_{4,8}) = -0{,}31$). Unter den Teilnehmern des Treatments Mw_4 ($n = 21$) können Korrelationen mit Entgegenkommen (*Ek*) ($r_s(Mw_4) = 0{,}51$), negativen Gefühlen (SPANE-N) ($r_s(Mw_4) = -0{,}51$) und emotionaler Balance (SPANE-B) ($r_s(Mw_4) = 0{,}48$) beobachtet werden. Diese Korrelationen haben einen p-Wert unterhalb des Schwellwerts von $\alpha = 0{,}05$.

5.4. Zusammenfassung

In diesem Kapitel wurden die Ergebnisse des Experiments präsentiert. In RQ1 wurde untersucht, ob sich Teilnehmer in ihrer subjektiven Bewertung beeinflussen ließen. Hier wurde ein signifikanter Unterschied ($p = 0{,}0001733$) zwischen den zwei Treatments mit sehr großem Effekt ($d = 1{,}376$) festgestellt. Die Nullhypothese wurde abgelehnt und die Hypothese H_{110} akzeptiert. In der zweiten Forschungsfrage (RQ2) wurde untersucht, ob sich das tatsächliche Codeverständnis manipulieren lässt. Für die Forschungsfrage wurde kein signifikanter Unterschied zwischen den Treatments beobachtet. Dementsprechend wurde die Nullhypothese H_{020} nicht abgelehnt. Für RQ3 wurde explorativ nach Korrelationen gesucht. Dabei wurden für $Mw_{4,8}$ Korrelationen mit negativen Gefühlen (SPANE-N) und Ängstlichkeit (*An*) gefunden. In Mw_4 wurden Korrelationen mit Entgegenkommen (*Ek*), negativen Gefühlen (SPANE-N) und emotionaler Balance (SPANE-B) gefunden. Die Korrelationen haben einen p-Wert unterhalb des Schwellwerts von $\alpha = 0{,}05$.

6. Diskussion

In diesem Kapitel werden die Ergebnisse des vorherigen Kapitels diskutiert und interpretiert. Ferner werden Implikationen erläutert und abschließend wird auf mögliche Limitationen des Experiments eingegangen.

In dem Experiment wird ein deutlicher Unterschied bei der wahrgenommenen Schwierigkeit (*wS*) zwischen den zwei Treatments festgestellt. Dieser Effekt ist sehr groß. In beiden Treatments wird eine ähnliche Abweichung vom vorgegebenen Metrikwert, die Metrikabweichung, beobachtet. Das heißt, dass Teilnehmer unabhängig vom Treatment um ein ähnliches Maß vom Metrikwert abgewichen sind. Anders als bei RQ1 wird bei RQ2 kein Unterschied festgestellt. Im Gegenteil: Die beiden Samples sind fast identisch. Bei RQ3 werden die Daten auf Korrelationen untersucht. Unter den relevanten Korrelationen mit der Metrikabweichung sind Entgegenkommen (*Ek*), Ängstlichkeit (*An*), SPANE-N und SPANE-B.

6.1. Interpretation und Implikationen

Interpretation von RQ1 Wir gehen davon aus, dass der bei RQ1 beobachtete Effekt der sogenannte Ankereffekt ist (vgl. [FB11]). Wir vermuten, dass der Ankereffekt ein Aspekt bei der Wirkungsweise von Placebos ist. Wie Wager und Atlas [WA15] verdeutlichen (vgl. Abbildung 2.1 auf Seite 20), können verbale Suggestionen und soziale Informationen die Erwartungen, Bewertungen und Erinnerungen beeinflussen. An dieser Stelle könnte ein Ankereffekt eine Rolle spielen. Da es genügend Literatur gibt (zum Beispiel [BKSB11; CL07; DE14; Mor09; RMI+17; TBG+18]), die Placeboeffekte *außerhalb* der Schmerzbehandlung untersuchen, erachten wir den Versuch als legitim, das Wissen über den *Kontext einer medizinischen Behandlung* aus Wager und Atlas in die Welt der Softwaretechnik zu übertragen.

Implikationen von RQ1 Daraus leiten wir folgende Implikationen ab: Da Teilnehmer in dem Experiment durch den vorgegebenen Wert in ihrer Bewertung sehr stark beeinflusst werden, erachten wir es als wichtig, dass bei Experimenten, die Metriken beinhalten, darauf geachtet wird, dass Teilnehmer nicht verankert werden. In einem solchen Experiment sollte darauf verzichtet werden, Bewertungen der Teilnehmer abzufragen. Insbesondere sollte bei Experimenten, die eine Metrik über subjektive Bewertungen der Teilnehmer validieren, darauf verzichtet werden, die Metrikwerte anzuzeigen. Es sollte überprüft werden, welche Konsequenzen eine Verankerung in der Praxis hat.

Interpretation von RQ2 Bei Lösen der Aufgaben haben die Gruppen die gleiche Performance hinsichtlich ihres Codeverständnisses gezeigt, somit können wir nicht sagen, dass eine Manipulation der Metrik einen Einfluss auf das Codeverständnis hat. Das heißt, ein Placeboeffekt ist entweder nicht vorhanden, nur sehr klein oder die Manipulation war nicht stark genug. Die Art unserer Messungen von Codeverstehen kann auch ein Grund für dieses Ergebnis sein. Eine andere Messung von Codeverständnis oder gar eine Betrachtung von anderen Aspekten wie Motivation wäre möglich. Zum Beispiel hätten wir die kognitive Belastung messen können. Weitere Ausführungen zur Art der Messung von Codeverständnis sind in Abschnitt 6.2.2 erläutert. An dieser Stelle sei auf Wager und Atlas [WA15] verwiesen, die beschreiben, dass Placebos Auswirkungen auf das Verhalten von Menschen haben können. Auch seien hier Crum und Langer [CL07] genannt, die durch eine Änderung der Geisteshaltung über die Arbeit signifikante körperliche Veränderungen bei Reinigungspersonal feststellen. Diese Veränderungen schreiben Crum und Langer dem Placeboeffekt zu.

Interpretation von RQ3 Bei der Untersuchung von RQ3 haben wir Korrelationen festgestellt, die sich mit der Literatur über Placebos decken, einige andere Korrelationen konnten wir nicht bestätigen. Nach den Regressionsanalysen von Geers et al. [GWF+10] und Morton et al. [MWEJ09] wäre es wahrscheinlich gewesen, eine Korrelation mit dispositionellem Optimismus zu finden. Darragh et al. [DBC14] haben Gegenteiliges beobachtet: Geringe Optimismuswerte hätten größeren Einfluss auf den Placeboeffekt als hohe Optimismuswerte. Peciña et al. [PAL+13] haben keinen Einfluss von dispositionellem Optimismus gefunden. Die Untersuchungen in [GWF+10; MWEJ09; PAL+13] sind auf dem Feld der Schmerzbehandlung, die Untersuchung von Darragh et al. [DBC14] ist außerhalb dieses Gebiets. Unsere Daten zeigen keine Korrelationen mit dispositionellem Optimismus auf. Die individuellen Charakteristiken, die Peciña et al. [PAL+13] in ihrer Regressionsanalyse in Verbindung mit der Wirksamkeit von Placebos gebracht haben, können wir in unserem Experiment nur teilweise bestätigen[1]: Eine positive Korrelation mit $p < 0{,}05$ mit Verträglichkeit (*Ve*) haben wir nicht gefunden ($r_s(Mw_{4,8}) = 0{,}03$, $r_s(Mw_4) = 0{,}03$, $r_s(Mw_8) = 0{,}05$). Ebenfalls haben wir keine negative Korrelation mit $p < 0{,}05$ mit Neurotizismus (*Ne*) gefunden ($r_s(Mw_{4,8}) = -0{,}29$, $r_s(Mw_4) = -0{,}31$, $r_s(Mw_8) = -0{,}26$). Nichtsdestotrotz haben wir eine negative Korrelation der Metrikabweichung (*Ma*) mit Ängstlichkeit (*An*) gefunden ($r_s(Mw_{4,8}) = -0{,}31$, $r_s(Mw_4) = -0{,}34$, $r_s(Mw_8) = -0{,}30$), die Korrelation war aber nur für $Mw_{4,8}$ ausreichend sicher ($p = 0{,}044$). Ängstlichkeit ist eine Facette der Skala „Neurotizismus“. Die Facette wird aus vier Fragen gebildet: „Ich sehe mich selbst als jemand, der …“

- „entspannt ist, sich durch Stress nicht aus der Ruhe bringen lässt.“ (umgepolt)
- „sich viele Sorgen macht.“
- „ruhig bleibt, selbst in angespannten Situationen.“ (umgepolt)
- „leicht nervös und unsicher wird.“

Es ist interessant, Angst und Optimismus zusammen zu betrachten. Glaesmer et al. [GHKH08] validieren die deutsche Version des überarbeiteten Life-Orientation-Tests (LOT-R). In ihrer Studie gehen sie auf die Kritik an der Originalversion des Tests ein: Sie schreiben, dass „die Effekte

[1] Wichtig ist zu beachten, dass Peciña et al. [PAL+13] den Revised NEO Personality Inventory Fragebogen verwendet und nicht den Big Five-Fragebogen. Beim Vergleich der Skalen sollte mit Vorsicht vorgegangen werden. Zwar sind die Skalen identisch zu den fünf Skalen des Big Five-Fragebogens, die Facetten unterschieden sich aber.

des Optimismus auf verschiedene Gesundheitsoutcomes durch dritte Variablen (Neurotizismus, Trait-Angst, […]) genauso gut oder besser erklärt werden konnten“ [GHKH08]. Dies ist einer der Gründe, warum der LOT überarbeitet wurde [SCB94]; diese überarbeitete Version wird von Geers et al. [GWF+10], Morton et al. [MWEJ09], Peciña et al. [PAL+13], Darragh et al. [DBC14] und in unserem Experiment verwendet. In der Regressionsanalyse von Morton et al. [MWEJ09] wurden Optimismus zusammen mit Angst als Regressoren für die Wirkung von Placebos gefunden. Bei Morton et al. wird aber der akute Angstzustand (engl. *state anxiety*) betrachtet, welcher mit dem STAI-Fragebogen ermittelt wird. Wir hingegen betrachten Ängstlichkeit (*An*), welche ein individuelles Charakteristikum ist und kein akuter Zustand. Wie eingangs erwähnt, haben wir keine Korrelation mit dem Optimismuswert (*Ow*) gefunden, aber mit dem individuellen Charakteristikum Ängstlichkeit. In der Literatur, die sich mit Schmerzen auseinandersetzt, ([DBC14; GWF+10; MWEJ09]) ergibt sich bezüglich Angst und Optimismus kein einheitliches Bild. Unsere Daten decken sich mit den Ergebnissen von Darragh et al. [DBC14], deren Untersuchung ebenfalls außerhalb der Schmerzbehandlung ist.

Laut Wager und Atlas (vgl. Abbildung 2.1 auf Seite 20) können Gefühle und Affektzustände bei der Wirkung von Placebos eine Rolle spielen. In unserem Experiment haben wir bei den Teilnehmern des Treatments Mw_4 eine negative Korrelation ($p = 0{,}017$) mit negativen Gefühlen (SPANE-N) gefunden ($r_s(Mw_{4,8}) = -0{,}26$, $r_s(Mw_4) = -0{,}51$, $r_s(Mw_8) = 0{,}05$). Ebenso haben wir bei diesem Treatment eine moderate Korrelation ($p = 0{,}028$) mit emotionaler Balance (SPANE-B) gefunden ($r_s(Mw_{4,8}) = 0{,}16$, $r_s(Mw_4) = 0{,}48$, $r_s(Mw_8) = -0{,}20$).

Bei dem Vergleich unserer Ergebnisse aus RQ3 mit Literatur bezüglich des Ankereffekts haben wir, im Gegensatz zu dem Literaturreview von Furnham und Boo [FB11], keine Korrelation mit $p < 0{,}05$ bei dem Charakterzug „Gewissenhaftigkeit“ gefunden ($r_s(Mw_{4,8}) = 0{,}05$, $r_s(Mw_4) = 0{,}10$, $r_s(Mw_8) = 0{,}07$). Beim Vergleich der Korrelationen muss berücksichtigt werden, dass ein hoher Wert bei der Metrikabweichung (*Ma*) eine geringe Verankerung bedeutet. Ebenso haben wir keine negative Korrelation mit $p < 0{,}05$ mit Extraversion gefunden ($r_s(Mw_{4,8}) = 0{,}02$, $r_s(Mw_4) = 0{,}23$, $r_s(Mw_8) = -0{,}17$). Weiter können wir die Korrelation mit Verträglichkeit nicht bestätigten ($r_s(Mw_{4,8}) = 0{,}03$, $r_s(Mw_4) = 0{,}03$, $r_s(Mw_8) = 0{,}05$). Zwar haben wir eine positive Korrelation zwischen der Facette Entgegenkommen (*Ek*), die Teil der Skala Verträglichkeit ist[2], und der Metrikabweichung (*Ma*) beobachtet, die Korrelation ist aber positiv und nicht negativ, so wie in dem Review von Furnham und Boo [FB11] dokumentiert. Abschließend können wir auch die Korrelation aus Furnham und Boo [FB11] mit Offenheit (*Of*) nicht bestätigen ($r_s(Mw_{4,8}) = 0{,}12$, $r_s(Mw_4) = 0{,}15$, $r_s(Mw_8) = 0{,}07$).

Implikationen von RQ3 Unsere Korrelationen der Metrikabweichung mit dem Optimismuswert (*Ow*) und der Ängstlichkeit (*An*) decken sich nicht mit den Ergebnissen der Literatur zu Placebos. Die Frage danach, ob dispositioneller Optimismus eine Rolle bei der Wirksamkeit von Placebos spielt, sehen wir als noch nicht ausreichend untersucht. Auch wenn die Korrelation mit Ängstlichkeit (*An*) nur unter allen Teilnehmern und nicht bei den zwei Treatments gefunden wurde, hegen wir die Vermutung, dass Ängstlichkeit ein Einflussfaktor sein könnte. Gestützt sehen wir diese Vermutung durch die Beiträge in Literatur zum Thema Angst [MWEJ09]. Ein weiterer Aspekt ist die Nähe von Optimismus und Angst, die auch ein Grund für die Überarbeitung des LOT-Fragebogens

[2]Die Facette wird aus drei Fragen gebildet, die Skala selbst wird aus neun Fragen gebildet.

ist [GHKH08]. Ferner haben wir bei dem Treatment Mw_4 Korrelationen mit negativen Gefühlen (SPANE-N) und emotionaler Balance (SPANE-B) gefunden. Aus den Korrelationsmatrizen (siehe Tabellen E.7 bis E.9) lässt sich entnehmen, dass SPANE-N und Ängstlichkeit (*An*) moderat miteinander korrelieren.[3] Unter anderem ist Teil von SPANE-N, wie *ängstlich* sich Personen in den letzten vier Wochen gefühlt hatten. Darin sehen wir einen Indikator dafür, dass das individuelle Charakteristikum „Ängstlichkeit" ein Einflussfaktor sein könnte. Deshalb erachten wir es als sinnvoll, bei weiteren Experimenten im Bereich von Placeboeffekten den Optimismus, akute Angst sowie Ängstlichkeit als Einflussfaktoren zu untersuchen. Damit sollen unsere Korrelation und die Ergebnisse in der Literatur repliziert werden.

Bezüglich des Ankereffekts stehen unsere Ergebnisse dem Literaturreview von Furnham und Boo entgegen. Angemerkt sei hier, dass in dem Review nur zwei Arbeiten angegeben sind. Deshalb schlagen wir vor, dass die Arbeiten, die Furnham und Boo [FB11] zu individuellen Charakteristiken anführen, ebenfalls repliziert werden sollten.

Falls bei solch einer Untersuchung weitere Belege bezüglich Ängstlichkeit gefunden werden, sollten bei Bewertungsaufgaben solche individuellen Charakteristiken berücksichtigt werden, da sonst Bewertungen von Softwareentwicklern im Kontext von Verankerung von diesen Charakteristiken beeinflusst werden könnten. Ein mögliches Szenario, in dem dies relevant sein könnte, wäre beispielsweise die Bewertung von Zeitaufwänden für Änderungen am Quellcode durch Softwareentwickler unter Berücksichtigung von Metriken aus einem Analysewerkzeug. Abhängig davon, wie sehr ein Softwareentwickler das individuelle Charakteristikum „Ängstlichkeit" hat, könnte eine Bewertung unterschiedlich ausfallen. Hier sei auf Nilson et al. [NAG19] verwiesen, die feststellen, dass viele Quellcodeanalysewerkzeuge, darunter auch „beliebte" Werkzeuge, eine Vielzahl von nicht validierten Metriken verwenden. Eine eventuell negative Verankerung durch nicht validierte Metriken ist nicht auszuschließen.

6.2. Limitationen

In diesem Abschnitt wird auf Entscheidungen im Studiendesign eingegangen, welche die Ergebnisse und Aussagekraft negativ beeinflussen. Bei der Interpretation der Ergebnisse müssen diese Limitationen beachtet werden.

6.2.1. Fehlende Kontrollgruppe

Wir haben uns beim Design des Experiments gegen eine Kontrollgruppe entschieden. Zum einen hätten wir nicht die notwendigen Teilnehmer dafür zur Verfügung gehabt und zum anderen stellt sich die Frage, welcher Metrikwert (*Mw*) der Kontrollgruppe hätte angezeigt werden sollen, da die *wahre* Schwierigkeit der Methode nicht objektiv bestimmbar ist. Der Kontrollgruppe keinen Metrikwert anzuzeigen ist keine Option gewesen, da das Design vorsieht, dass der Studienleiter nicht weiß, welche Teilnehmer welches Treatment erhalten. Da wir die *wahre* Schwierigkeit nicht kennen, wissen wir nicht, wie stark unsere Treatments sind. Unsere Lösung des Problems ist es, mit der Metrikabweichung (*Ma*) den Einfluss der Treatments auf die Teilnehmer zu quantifizieren. Es bleibt

[3] $r_s(Mw_{4,8}) = 0{,}49$ mit $p < 0{,}001$, $r_s(Mw_4) = 0{,}58$ mit $p < 0{,}01$, $r_s(Mw_8) = 0{,}42$ mit $p < 0{,}05$

die Frage offen, ob es für die Teilnehmer einen Unterschied macht, einzugestehen, dass die gezeigte Methode als schwerer oder leichter wahrgenommen wird. Insbesondere beim Auswerten der Daten für RQ3 muss berücksichtigt werden, dass wir keine Kontrollgruppe haben. Dementsprechend vorsichtig müssen die Vergleiche mit der Literatur bezüglich Placebos vorgenommen werden. Deshalb wurden zusätzlich die Korrelationen mit Literatur zum Ankereffekt verglichen.

6.2.2. Messung von Codeverständnis

Wir haben das Codeverständnis (*TAU*) aus den korrekten Rückgabewerten laut Code (*kC*), korrekten Rückgabewerten laut Dokumentation (*kD*) und der Bearbeitungszeit (*t*) abgeleitet. Wir haben uns dabei an Scalabrino et al. [SBV+19] orientiert. Wir haben festgestellt, dass die Teilnehmer im Median 27 von 30 Antworten richtig hatten (vgl. Tabellen E.2 und E.3). Dies deutet darauf hin, dass Teilnehmer die Aufgabenstellung verstanden haben. Die Kehrseite ist, dass die korrekten Rückgabewerte laut Code (*kC*) und die korrekten Rückgabewerte laut Dokumentation (*kD*) nicht viel Wert für die Berechnung von Codeverständnis (*TAU*) haben. Das heißt, die Bearbeitungszeit hat den größten Einfluss auf die Messung des Codeverständnisses. Wir haben den Teilnehmern keine Zeitlimits genannt bzw. diese Zeitlimits geheim gehalten. Diese geheimen Zeitlimits wurden im Experiment nur von 2 der 45 Teilnehmern überschritten. Es kann sein, dass diese Entscheidung ungünstig war, da zu viel Zeit zur Lösung der Aufgaben im Vergleich zur Schwierigkeit der Aufgaben zur Verfügung stand. Außerdem hätten bekannte und kürzere Zeitlimits gegebenenfalls die Ergebnisse verändert, indem zum Beispiel die *Motivation* beeinflusst wird – nach Wager und Atlas [WA15] ein Einflussfaktor auf Placeboeffekte.

Bei der Planung des Designs haben wir die Möglichkeit diskutiert, nach jeder Aufgabe den NASA-TLX [Ame19] ausfüllen zu lassen. Der NASA-TLX ist ein sehr verbreiteter Fragebogen [Har06], um unterschiedliche Formen der Belastung zu messen [Ame19]. Über die Skalen des TLX hätte sich durch eine Selbsteinschätzung der Teilnehmer quantifizieren lassen, welchen zeitlichen Druck Teilnehmer verspüren, wie die mentale Belastung ist, wie der wahrgenommene Aufwand war und wie frustriert Teilnehmer sind [Ame19]. Wenngleich wir diese Idee verworfen haben, da die gesammelten Daten uns ausreichend erschienen, scheint diese Idee im Rückblick sinnvoll, denn eventuell haben die Teilnehmer der zwei Gruppen unterschiedlich viel Aufwand leisten müssen.

Summa summarum sollten bei den Ergebnissen und bei der Interpretation berücksichtigt werden, dass sich Codeverständnis unterschiedlich messen lässt.

6.2.3. Anzeige des Metrikwerts

Wir haben den Metrikwert neben den Codeschnipsel „groß“ angezeigt. Beim Erklären der Beispiele haben wir beide Male die Metrik zur Sprache gebracht. Damit wollten wir den angezeigten Metrikwert in den Fokus stellen. Eine so prominente Anzeige einer Metrik ist uns aus keiner Entwicklungsumgebung bekannt. Eventuell führt eine weniger prominente Anzeige zu einer weniger starken Verankerung. Anderseits führen Furnham und Boo [FB11] in ihrem Literaturreview eine Arbeit [CG08] an, in der eine Zahl im Namen eines Restaurants zu einer Verankerung in Form einer höheren Schätzung über die Höhe der Rechnung führt. Die unterschiedlichen Zahlen im Namen der

Restaurants werden nicht betont [CG08]. Deshalb stellt sich die Frage, inwieweit das Ergebnis aus RQ1 auf die Praxis übertragbar ist bzw. wie stark ein Ankereffekt durch eine Metrik in der Praxis ist.

6.2.4. Messung des Placeboeffekts

Wir können nicht sagen, inwieweit die Metrikabweichung als alleinige abhängige Variable geeignet ist, um den Placeboeffekt zu beschreiben. Dieser Aspekt betrifft die Konstruktvalidität unseres Designs. Deshalb haben wir unsere Ergebnisse bei RQ3 mit Literatur zum Placeboeffekt und dem Ankereffekt verglichen. Eine zweite Frage ist, ob die Metrikabweichung sinnvoll berechnet wurde. Die Berechnung durch $Ma = |wS - Mw|$ lässt die Richtung der Abweichung außer acht. Dies haben wir bewusst gewählt, damit die Treatments vergleichbar sind. Die Gruppe Mw_4 ist im Median nach oben abgewichen und die Gruppe Mw_8 ist im Median nach unten abgewichen. Wir wissen nicht, ob eine Abweichung nach unten bzw. oben unterschiedlich zu bewerten ist. Das heißt, fällt es Teilnehmern leichter bzw. schwerer eine Aufgabe als leichter bzw. schwerer zu bewerten, als vom Metrikwert vorgegeben? Andererseits ist die Metrikabweichung bei den zwei Treatments Mw_4 und Mw_8 relativ ähnlich (vgl. Abbildung 5.3).

6.2.5. Repräsentativität der Stichprobe

Unsere Stichprobe spiegelt nur einen Teil der Population aller Softwareentwickler wieder. Es haben nur Studierende der Universität Stuttgart teilgenommen. Die Hälfte der Teilnehmer ist zwischen 21 und 25 Jahren (Median 24 Jahre) alt. Der jüngste Teilnehmer ist 21 Jahre, der älteste Teilnehmer 33 Jahre alt. Im Median haben die Teilnehmer eine Programmiererfahrung mit Java (*Pe*) von 6 Jahren, beginnend ab der ersten Programmiererfahrung. Der erfahrenste Teilnehmer hat eine Programmiererfahrung von 12 Jahren. Die drei unerfahrensten Teilnehmer eine Programmiererfahrung von einem Jahr. Dementsprechend beträgt die Spannweite der Programmiererfahrung mit Java 11 Jahre. Das 25 %-Quantil liegt bei 5, das 75 %-Quantil bei 7 Jahren. Die Teilnehmer sind unserem Ermessen nach keine Programmieranfänger. Wird berücksichtigt, dass 35 Teilnehmer im Master und zehn Teilnehmer im Bachelor eingeschrieben sind, denken wir, dass ein ausreichend breites Spektrum an Programmiererfahrung in unserer Stichprobe repräsentiert ist. Weiter haben wir, um die Glaubhaftigkeit der Metrik zu untermauern – unserer Einschätzung nach – realistische Beispiele mit passenden Metrikwerten gezeigt. Wir haben berichtet, dass die Metrik mit einer Publikation wissenschaftlich bestätigt worden sei. Außerdem haben wir erläutert, dass die Metrik mit maschinellem Lernen funktioniere, um nicht in Erklärungsnot zu kommen. Im Laufe des Experiments wurden keine kritischen Fragen zur Metrik gestellt, die vermuten lassen, dass die Echtheit bezweifelt wurde. Zwei Teilnehmer haben während der Erklärungen Fragen zur Funktionsweise gestellt, diese konnten beantwortet werden.[4] Wir denken, dass wir mit ähnlichen Argumenten ältere und deutlich erfahrenere Softwareentwickler überzeugt hätten. Furnham und Boo [FB11] diskutieren in ihrem Literaturreview unterschiedliche Ergebnisse zur Frage, ob sich erfahrenere Personen verankern

[4] Die eine Frage war, wie die Metrik heiße. Es wurde geantwortet, dass die Metrik bisher nur einen Arbeitstitel habe. Darauf wurde weiter gefragt, ob die Metrik nur für Java funktioniert. Es wurde geantwortet, dass die Metrik nur für Java trainiert wurde. Ein anderer Teilnehmer hat gefragt, ob die Metrik bezüglich der Variablenlänge kontextbezogen sei. Darauf wurde gesagt, dass wir es nicht wissen würden.

lassen. Sie resümieren, dass sich erfahrene Personen auch verankern lassen. Deshalb denken wir, dass unsere Stichprobe ausreichend repräsentativ ist, um die Ergebnisse auf ältere und erfahrenere Softwareentwickler zu übertragen.

6.3. Zusammenfassung

In diesem Kapitel wurden die Ergebnisse diskutiert und interpretiert. Anschließend wurden Implikationen erläutert und abschließend erörtert, welche Limitationen vorliegen. Die Ergebnisse bezüglich RQ1 wurden auf den Ankereffekt zurückgeführt, bei dem wir vermuten, er könnte Teil eines Placeboeffekts sein. Aufgrund dieses Ergebnisses wurde geschlussfolgert, dass bei Experimenten, die Metriken beinhalten, darauf geachtet werden sollte, dass Teilnehmer nicht verankert werden. Insbesondere sollte bei Experimenten, die eine Metrik über Bewertungen der Teilnehmer validieren, darauf verzichtet werden, die Metrikwerte anzuzeigen. Bei RQ2 wurde untersucht, ob sich das tatsächliche Codeverständnis manipulieren lässt. Die Daten lassen keinen solchen Effekt erkennen. Es wurde diskutiert, dass das negative Ergebnis eventuell auf Designschwächen zurückgeführt werden könnte – darauf wurde bei den Limitationen weiter eingegangen. Beispielsweise hätte das Codeverständnis anders gemessen werden können. Bei der letzten Forschungsfrage (RQ3) wurden Ergebnisse mit anderen wissenschaftlichen Arbeiten verglichen. Wir haben eine Korrelation der Metrikabweichung und dem individuellen Charakteristikum „Ängstlichkeit“ gefunden. Diese Korrelation sehen wir durch eine weitere Korrelation mit SPANE-N unterstützt. Es wurde herausgearbeitet, dass Optimismus, akute Angstzustände und das individuelle Charakteristikum „Ängstlichkeit“ im Zusammenhang mit Placeboeffekten weiter untersucht werden sollten, da sich Ergebnisse der Literatur teilweise widersprechen.

7. Zusammenfassung und Ausblick

In dieser Arbeit untersuchen wir den Placeboeffekt beim Verstehen von Quellcode. Wir leiten die Arbeit damit ein, zu verdeutlichen, dass Softwaremetriken oftmals nicht ausreichend validiert sind [NAG19; SBV+19; SI94]. Wir stellen die Frage, ob manipulierte oder nicht validierte Metriken einen – möglicherweise negativen – Effekt auf Softwareentwickler haben. Autoren wir Nilson et al. [NAG19] und Shepperd und Ince [SI94] argumentieren, dass schlecht validierte Metriken Entwickler verwirren könnten [NAG19] oder dass Metriken mehr schaden als nutzen würden [She88]. Wir mutmaßen, dass eine manipulierte oder nicht validierte Metrik als Placebo fungieren könnte; wir damit zum Beispiel das Codeverständnis beeinflussen könnten. Die Vielzahl von Bereichen, in denen Placebos beobachtet werden, inspirieren uns zu dieser Vermutung: Placebos können Leistung im Sport verbessern [BKSB11], kognitive Leistung beeinflussen [DE14] – beispielsweise beim Lernen [TBG+18] – und Kreativität steigern [RMI+17]. Eine weitere Form eines Placeboeffekts ist, dass eine Veränderung der Geisteshaltung, bei gleichem Verhalten, zu körperlichen Veränderungen in Form von Gewichtsabnahme führen kann [CL07].

Wir stellen uns in der Arbeit die Frage, welchen Einfluss manipulierte Softwaremetriken beim Verstehen von Quellcode haben. Da Softwareentwickler bei der Wartung von Software 30-50 % der Zeit damit verbringen, Quellcode zu verstehen [MML15], erachten wir diese Frage als berechtigt. Dazu führen wir ein Experiment mit 45 Teilnehmern an der Universität Stuttgart durch. Die Teilnehmer müssen Java-Methoden verstehen und für gegebene Eingabewerte die Rückgabewerte berechnen. Den zwei Gruppen zeigen wir eine manipulierte Verständlichkeitsmetrik an, der einen Gruppe zeigen wir einen niedrigen Wert, der anderen einen hohen Wert an. Mithilfe des Experiments untersuchen wir (RQ1), wie sich die manipulierte Metrik auf die wahrgenommene Schwierigkeit einer Methode auswirkt. Wir haben festgestellt, dass sich Probanden stark an dem vorgegebenen Metrikwert orientieren. Diese Beobachtung lässt sich mit dem Ankereffekt erklären. Weiter untersuchen wir in RQ2, ob eine Manipulation der Metrik einen Einfluss auf die Performance beim Verstehen von Quellcode hat. Hier haben wir ein negatives Ergebnis erhalten. Als letzte Forschungsfrage (RQ3) untersuchen wir, welche individuellen Charakteristiken bei einer Manipulation eine Rolle spielen. Wir haben Ängstlichkeit als Charakteristikum gefunden, das mit der Metrikabweichung korreliert. Demnach sind eher weniger ängstliche Probanden von dem vorgegebenen Metrikwert abgewichen. Eine Korrelation mit Optimismus finden wir, entgegen den Ergebnissen anderer Arbeiten, nicht. Korrelationen aus einem Literaturreview über Ankereffekte [FB11] konnten wir nicht bestätigen; teilweise finden wir gegensätzliche Korrelationen.

Ein zentrales Ergebnis dieser Arbeit ist, dass bei Experimenten, in denen Metrikwerte angezeigt werden, mit Bedacht vorgegangen werden sollte. Weiter sollte bei Experimenten, die beabsichtigen eine Metrik durch subjektive Bewertungen zu validieren, davon abgesehen werden, Metrikwerte ebendieser Metrik anzuzeigen.

Ausblick

Als Fortsetzung dieser Arbeit sollte untersucht werden, welchen Einfluss eine Verankerung durch Softwaremetriken auf Softwareentwickler in der Praxis hat. Es könnte sein, dass durch manipulierte Metriken das Verhalten oder Herangehensweise von Softwareentwicklern beeinflusst werden kann. Gestützt sehen wir diese Vermutung durch Wager und Atlas [WA15], die sagen, dass Placebos das Verhalten beeinflussen können und durch Crum und Langer [CL07], die am Beispiel von Reinigungspersonal sichtbar machen, wie sich die Geisteshaltung auf den Körper auswirken kann. Abschließend sehen wir Potenzial, die Zusammenhänge von Ängstlichkeit und Optimismus im Kontext der Softwaretechnik weiter zu beleuchten.

Literaturverzeichnis

[Ame19] Ames Research Center. *NASA-TLX: Task Load Index*. Aug. 2019. URL: https://humansystems.arc.nasa.gov/groups/TLX/ (besucht am 27. 02. 2020) (zitiert auf S. 55).

[BB16] B. Breyer, M. Bluemke. „Deutsche Version der Positive and Negative Affect Schedule PANAS (GESIS Panel)“. deu. In: *Zusammenstellung sozialwissenschaftlicher Items und Skalen (ZIS)* (2016). DOI: 10.6102/zis242 (zitiert auf S. 26).

[BKSB11] M. Bérdi, F. Köteles, A. Szabó, G. Bárdos. „Placebo effects in sport and exercise: a meta-analysis“. In: *European Journal of Mental Health* 6.2 (Dez. 2011), S. 196. DOI: 10.5708/ejmh.6.2011.2.5 (zitiert auf S. 13, 21, 23, 29, 51, 59).

[Cam18] G. A. Campbell. *Cognitive Complexity – A new way of measuring understandability*. Version 1.4. SonarSource SA, 2018. URL: https://www.sonarsource.com/docs/CognitiveComplexity.pdf (besucht am 07. 01. 2020) (zitiert auf S. 31).

[CG08] C. R. Critcher, T. Gilovich. „Incidental environmental anchors“. In: *Journal of Behavioral Decision Making* 21.3 (Juli 2008), S. 241–251. DOI: 10.1002/bdm.586 (zitiert auf S. 55, 56).

[CL07] A. J. Crum, E. J. Langer. „Mind-set matters: Exercise and the placebo effect“. In: *Psychological Science* 18.2 (Feb. 2007), S. 165–171. DOI: 10.1111/j.1467-9280.2007.01867.x (zitiert auf S. 21–23, 51, 52, 59, 60).

[Coh13] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, Mai 2013. DOI: 10.4324/9780203771587 (zitiert auf S. 21).

[DBC14] M. Darragh, R. J. Booth, N. S. Consedine. „Investigating the 'placebo personality' outside the pain paradigm“. In: *Journal of psychosomatic research* 76.5 (Mai 2014), S. 414–421. DOI: 10.1016/j.jpsychores.2014.02.011 (zitiert auf S. 26, 28, 30, 52, 53).

[DE14] C. Draganich, K. Erdal. „Placebo sleep affects cognitive functioning.“ In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 40.3 (2014), S. 857. DOI: 10.1037/a0035546 (zitiert auf S. 13, 27–29, 35, 37, 51, 59).

[FB11] A. Furnham, H. C. Boo. „A literature review of the anchoring effect“. In: *The journal of socio-economics* 40.1 (Feb. 2011), S. 35–42. DOI: 10.1016/j.socec.2010.10.008 (zitiert auf S. 22, 23, 29, 51, 53–56, 59).

[GHKH08] H. Glaesmer, J. Hoyer, J. Klotsche, P. Y. Herzberg. „Die deutsche Version des Life-Orientation-Tests (LOT-R) zum dispositionellen Optimismus und Pessimismus“. In: *Zeitschrift für Gesundheitspsychologie* 16.1 (Jan. 2008), S. 26–31. DOI: 10.1026/0943-8149.16.1.26 (zitiert auf S. 31, 35, 52–54).

[GWF+10] A. L. Geers, J. A. Wellman, S. L. Fowler, S. G. Helfer, C. R. France. „Dispositional optimism predicts placebo analgesia“. In: *The Journal of Pain* 11.11 (Nov. 2010), S. 1165–1171. DOI: 10.1016/j.jpain.2010.02.014 (zitiert auf S. 25, 28, 30, 35, 52, 53).

[Har06] S. G. Hart. „NASA-task load index (NASA-TLX); 20 years later“. In: *Proceedings of the human factors and ergonomics society annual meeting*. Bd. 50. 9. Sage Publications Sage CA: Los Angeles, CA. 2006, S. 904–908. DOI: 10.1177/154193120605000909 (zitiert auf S. 55).

[Jar19] C. Jarrett. *The Placebo Effect, Digested – 10 Amazing Findings*. März 2019. URL: https://digest.bps.org.uk/2019/03/11/the-placebo-effect-digested-10-amazing-findings/ (besucht am 05. 12. 2019) (zitiert auf S. 35).

[JCP08] A. Jedlitschka, M. Ciolkowski, D. Pfahl. „Reporting experiments in software engineering“. In: *Guide to advanced empirical software engineering*. Springer, 2008, S. 201–228. DOI: 10.1007/978-1-84800-044-5_8 (zitiert auf S. 34).

[LGF78] J. Levine, N. Gordon, H. Fields. „The mechanism of placebo analgesia“. In: *The Lancet* 312.8091 (Sep. 1978), S. 654–657. DOI: 10.1016/s0140-6736(78)92762-9 (zitiert auf S. 13, 21, 23).

[LLA01] F. R. Lang, O. Lüdtke, J. B. Asendorpf. „Testgüte und psychometrische Äquivalenz der deutschen Version des Big Five Inventory (BFI) bei jungen, mittelalten und alten Erwachsenen“. In: *Diagnostica* 47.3 (2001), S. 111–121. DOI: 10.1026//0012-1924.47.3.111 (zitiert auf S. 31, 35).

[McC76] T. J. McCabe. „A complexity measure“. In: *IEEE Transactions on software Engineering* SE-2.4 (Dez. 1976), S. 308–320. DOI: 10.1109/tse.1976.233837 (zitiert auf S. 14, 17).

[MCY+04] C. McRae, E. Cherin, T. G. Yamazaki, G. Diem, A. H. Vo, D. Russell, J. H. Ellgring, S. Fahn, P. Greene, S. Dillon et al. „Effects of perceived treatment on quality of life and medical outcomesin a double-blind placebo surgery trial“. In: *Archives of general psychiatry* 61.4 (Apr. 2004), S. 412–420. DOI: 10.1001/archpsyc.61.4.412 (zitiert auf S. 13, 35).

[MML15] R. Minelli, A. Mocci, M. Lanza. „I know what you did last summer-an investigation of how developers spend their time“. In: *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE. IEEE, Mai 2015, S. 25–35. DOI: 10.1109/ICPC.2015.12 (zitiert auf S. 14, 59).

[Mor09] R. H. Morton. „Deception by manipulating the clock calibration influences cycle ergometer endurance time in males“. In: *Journal of Science and Medicine in Sport* 12.2 (März 2009), S. 332–337. DOI: 10.1016/j.jsams.2007.11.006 (zitiert auf S. 13, 29, 51).

[MST+18] R. Mohanani, I. Salman, B. Turhan, P. Rodriguez, P. Ralph. „Cognitive biases in software engineering: a systematic mapping study“. In: *IEEE Transactions on Software Engineering* (2018), S. 1–1. DOI: 10.1109/TSE.2018.2877759 (zitiert auf S. 22, 23).

[Mul94] P. Muller. „Glossary of terms used in physical organic chemistry (IUPAC Recommendations 1994)“. In: *Pure and Applied Chemistry* 66.5 (Jan. 1994), S. 1077–1184. DOI: 10.1351/pac199466051077 (zitiert auf S. 19).

[MWEJ09] D. L. Morton, A. Watson, W. El-Deredy, A. K. Jones. „Reproducibility of placebo analgesia: Effect of dispositional optimism“. In: *Pain* 146.1-2 (Nov. 2009), S. 194–198. DOI: 10.1016/j.pain.2009.07.026 (zitiert auf S. 25, 28, 30, 35, 52, 53).

[NAG19] M. Nilson, V. Antinyan, L. Gren. „Do internal software quality tools measure validated metrics?“ In: *International Conference on Product-Focused Software Process Improvement*. Springer. Springer International Publishing, 2019, S. 637–648. DOI: 10.1007/978-3-030-35333-9_50 (zitiert auf S. 14, 18, 23, 54, 59).

[NE98] W. Nix, U. Egle. „Das chronische Erschöpfbarkeitssyndrom (chronic-fatigue-syndrom)“. In: *Aktuelle Neurologie* 25.01 (Feb. 1998), S. 6–12. DOI: 10.1055/s-2007-1017656 (zitiert auf S. 13).

[PAL+13] M. Peciña, H. Azhar, T. M. Love, T. Lu, B. L. Fredrickson, C. S. Stohler, J.-K. Zubieta. „Personality trait predictors of placebo analgesia and neurobiological correlates“. In: *Neuropsychopharmacology* 38.4 (Nov. 2013), S. 639. DOI: 10.1038/npp.2012.227 (zitiert auf S. 26, 28, 30, 35, 52, 53).

[RHS17] T. Rahm, E. Heise, M. Schuldt. „Measuring the frequency of emotions—validation of the Scale of Positive and Negative Experience (SPANE) in Germany“. In: *PloS one* 12.2 (Feb. 2017). Hrsg. von P. Cipresso, e0171288. DOI: 10.1371/journal.pone.0171288 (zitiert auf S. 31, 35).

[RMI+17] L. Rozenkrantz, A. E. Mayo, T. Ilan, Y. Hart, L. Noy, U. Alon. „Placebo can enhance creativity“. In: *PloS one* 12.9 (Sep. 2017). Hrsg. von E. Manalo. DOI: 10.1371/journal.pone.0182466 (zitiert auf S. 13, 27–29, 51, 59).

[Saw09] S. S. Sawilowsky. „New effect size rules of thumb“. In: *Journal of Modern Applied Statistical Methods* 8.2 (Nov. 2009), S. 26. DOI: 10.22237/jmasm/1257035100 (zitiert auf S. 46).

[SBV+19] S. Scalabrino, G. Bavota, C. Vendome, M. Linares-Vásquez, D. Poshyvanyk, R. Oliveto. „Automatically Assessing Code Understandability“. In: *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. IEEE Press. IEEE, Okt. 2019, S. 417–427. DOI: 10.1109/TSE.2019.2901468 (zitiert auf S. 14, 17, 23, 34, 35, 55, 59, 77).

[SC85] M. F. Scheier, C. S. Carver. „Optimism, coping, and health: assessment and implications of generalized outcome expectancies.“ In: *Health psychology* 4.3 (1985), S. 219. DOI: 10.1037/0278-6133.4.3.219 (zitiert auf S. 25).

[SCB94] M. F. Scheier, C. S. Carver, M. W. Bridges. „Distinguishing optimism from neuroticism (and trait anxiety, self-mastery, and self-esteem): a reevaluation of the Life Orientation Test.“ In: *Journal of personality and social psychology* 67.6 (1994), S. 1063. DOI: 10.1037/0022-3514.67.6.1063 (zitiert auf S. 53).

[Sch07] G. Schönbächler. „Placebo“. In: *Swiss Medical Forum*. Bd. 7. 08. EMH Media. EMH Swiss Medical Publishers, Ltd., Feb. 2007, S. 205–210. DOI: 10.4414/smf.2007.06122 (zitiert auf S. 21).

[Sha68] A. K. Shapiro. „Semantics of the placebo“. In: *Psychiatric Quarterly* 42.4 (Dez. 1968), S. 653–695. DOI: 10.1007/BF01564309 (zitiert auf S. 18, 23).

[She88] M. Shepperd. „A critique of cyclomatic complexity as a software metric“. In: *Software Engineering Journal* 3.2 (1988), S. 30–36. DOI: 10.1049/sej.1988.0003 (zitiert auf S. 59).

[SI94] M. Shepperd, D. C. Ince. „A critique of three metrics“. In: *Journal of Systems and Software* 26.3 (Sep. 1994), S. 197–210. DOI: 10.1016/0164-1212(94)90011-6 (zitiert auf S. 14, 17, 23, 59).

[SKL+14] J. Siegmund, C. Kästner, J. Liebig, S. Apel, S. Hanenberg. „Measuring and modeling programming experience“. In: *Empirical Software Engineering* 19.5 (Dez. 2014), S. 1299–1334. DOI: 10.1007/s10664-013-9286-4 (zitiert auf S. 78).

[TBG+18] Z. Turi, E. Bjørkedal, L. Gunkel, A. Antal, W. Paulus, M. Mittner. „Evidence for Cognitive Placebo and Nocebo Effects in Healthy Individuals“. In: *Scientific reports* 8.1 (Mai 2018), S. 17443. DOI: 10.31234/osf.io/87kea (zitiert auf S. 13, 29, 51, 59).

[TK74] A. Tversky, D. Kahneman. „Judgment under uncertainty: Heuristics and biases“. In: *science* 185.4157 (Sep. 1974), S. 1124–1131. DOI: 10.1126/science.185.4157.1124 (zitiert auf S. 22).

[WA15] T. D. Wager, L. Y. Atlas. „The neuroscience of placebo effects: connecting context, learning and health“. In: *Nature Reviews Neuroscience* 16.7 (Juni 2015), S. 403. DOI: 10.1038/nrn3976 (zitiert auf S. 13, 19, 20, 23, 29, 30, 35, 37, 51–53, 55, 60).

[Win07] J. Windeler. „Placebo-Effekte“. In: *Zeitschrift für ärztliche Fortbildung und Qualität im Gesundheitswesen-German Journal for Quality in Health Care* 101.5 (Juni 2007), 307–e1. DOI: 10.1016/j.zgesun.2007.04.011 (zitiert auf S. 18).

[WRH+12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-29044-2 (zitiert auf S. 36–39).

Alle URLs wurden zuletzt am 24. Juni 2020 geprüft.

A. Codeschnipsel für das Experiment

A.1. Beispiel 1 – `multiplyByTwo`

Listing A.1: Codeschnipsel für Beispiel 1 des Experiments.

```java
/*
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license
* agreements. See the NOTICE file distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file to You under the Apache
* License, Version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under
* the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
* ANY KIND, either express or implied. See the License for the specific language
* governing permissions and limitations under the License.
*/

package beispiel;

public class MultiplyUtils {

    /**
    * Multiples the given <code>multiplier</code> by 2 and returns the value.
    * <code>null</code> values return <code>null</code>.
    *
    * For Example:
    *
    * <pre>
    * multiplyByTwo(Double(4)) == 8d
    * </pre>
    *
    * @param multiplier the number to multiply by two.
    */
    public static Double multiplyByTwo(Double multiplier) {
        if (multiplier == null) {
            return null;
        }
        if (multiplier == 0) {
            return 0d;
        }
        return multiplier * 3;
    }
}
```

A.2. Beispiel 2 – gdc

Listing A.2: Codeschnipsel für Beispiel 2 des Experiments.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license
* agreements. See the NOTICE file distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file to You under the Apache
* License, Version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under
* the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
* ANY KIND, either express or implied. See the License for the specific language
* governing permissions and limitations under the License.
*/

/*
* Apache Commons Lang
* Copyright 2001-2020 The Apache Software Foundation
*
* This product includes software developed at
* The Apache Software Foundation (http://www.apache.org/).
*/

// Note to fulfil Apache License: File has been changed.

package beispiel;

public class Faction {

    /**
    * <p>
    * Gets the greatest common divisor of the absolute value of two numbers, using
    * the "binary gcd" method which avoids division and modulo operations. See
    * Knuth 4.5.2 algorithm B. This algorithm is due to Josef Stein (1961).
    * </p>
    *
    * @param u a non-zero number
    * @param v a non-zero number
    * @return the greatest common divisor, never zero
    */
    public static int gdc(int u, int v) {
        // Perform calculation using Commons Math-
        if (u == 0 || v == 0) {
            // overflow
            if (u == Integer.MIN_VALUE || v == Integer.MIN_VALUE) {
                throw new ArithmeticException("overflow: gcd is 2^31");
            }
            // cal abs
            return Math.abs(u) + Math.abs(v);
        }
        // if either operand is abs 1, return 1:
        if (Math.abs(u) == 1 || Math.abs(v) == 1) {
            return 1;
        }
```

```
54          // keep u and v negative, as negative integers range down to
55          // -2^31, while positive numbers can only be as large as 2^31-1
56          // (i.e. we can't necessarily negate a negative number without
57          // overflow)
58          if (u > 0) {
59              u = -u;
60          } // make u negative
61          if (v > 0) {
62              v = -v;
63          } // make v negative
64          // B1
65          int k = 0;
66          while ((u & 1) == 0 && (v & 1) == 0 && k < 31) {//while u and v are both even...
67              u /= 2;
68              v /= 2;
69              k++; // cast out twos.
70          }
71          if (k == 31) {
72              throw new ArithmeticException("overflow: gcd is 2^31");
73          }
74          // B2
75          int t = (u & 1) == 1 ? v : -(u / 2)/* B3 */;
76          // t negative: u was odd, v may be even (t replaces v)
77          // t positive: u was even, v is odd (t replaces u)
78          do {
79              /* assert u<0 && v<0; */
80              // B4/B3: cast out twos from t.
81              while ((t & 1) == 0) { // while t is even..
82                  t /= 2; // cast out twos
83              }
84              // B5 [reset max(u,v)]
85              if (t > 0) {
86                  u = t;
87              } else {
88                  v = -t;
89              }
90              // B6/B3. at this point both u and v should be odd.
91              t = (v - u) / 2;
92              // |u| larger: t positive (replace u)
93              // |v| larger: t negative (replace v)
94          } while (t != 0);
95          return -u * (1 << k); // gcd is u*2^k
96      }
97  }
```

A.3. Aufgabe 1 – `toBooleanObject`

Listing A.3: Codeschnipsel für Aufgabe 1 des Experiments.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license
* agreements. See the NOTICE file distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file to You under the Apache
* License, Version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under
* the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
* ANY KIND, either express or implied. See the License for the specific language
* governing permissions and limitations under the License.
*/

/*
* Apache Commons Lang
* Copyright 2001-2020 The Apache Software Foundation
*
* This product includes software developed at
* The Apache Software Foundation (http://www.apache.org/).
*/

// Note to fulfil Apache License: File has been changed.

package aufgabe1;

import java.util.Set;

import org.apache.commons.lang3.StringUtils;

public class BooleanUtils {

    /**
    * Converts a String to a Boolean.
    *
    * {@code 'true'}, {@code 'on'}, {@code 'y'}, {@code 't'} or {@code 'yes'} (case
    * insensitive) will return {@code true}. {@code 'false'}, {@code 'off'},
    * {@code 'n'}, {@code 'f'} or {@code 'no'} (case insensitive) will return
    * {@code false}. Otherwise, {@code null} is returned.
    *
    * <p>
    * NOTE: This returns null and will throw a NullPointerException if unboxed to a
    * boolean.
    * </p>
    *
    * <pre>
    *   BooleanUtils.toBooleanObject(null)    = null
    *   BooleanUtils.toBooleanObject("true")  = Boolean.TRUE
    *   BooleanUtils.toBooleanObject("false") = Boolean.FALSE
    *   BooleanUtils.toBooleanObject("No")    = Boolean.FALSE
    *   BooleanUtils.toBooleanObject("on")    = Boolean.TRUE
    *   BooleanUtils.toBooleanObject("off")   = Boolean.FALSE
    *   BooleanUtils.toBooleanObject("yes")   = Boolean.TRUE
```

```
54      *   BooleanUtils.toBooleanObject("blue")  = null
55      * </pre>
56      *
57      * @param str the String to check; upper and lower case are treated as the same
58      * @return the Boolean value of the string, {@code null} if no match or
59      *         {@code null} input
60      */
61      public static Boolean toBooleanObject(String str) {
62          str = StringUtils.defaultString(str);
63          switch (str.length()) {
64          case 2: {
65              final char ch0 = str.charAt(0);
66              final char ch1 = str.charAt(1);
67              if (Set.of('n', 'N').contains(ch0) && Set.of('o', 'O').contains(ch1)) {
68                  return Boolean.TRUE;
69              }
70              if (Set.of('o', 'O').contains(ch0) && Set.of('n', 'N').contains(ch1)) {
71                  return Boolean.FALSE;
72              }
73              break;
74          }
75          case 3: {
76              final char ch0 = str.charAt(0);
77              final char ch1 = str.charAt(1);
78              final char ch2 = str.charAt(1);
79              if (Set.of('y', 'Y').contains(ch0) && Set.of('e', 'E').contains(ch1)
80                      && Set.of('s', 'S').contains(ch2)) {
81                  return Boolean.TRUE;
82              }
83              if (Set.of('o', 'O').contains(ch0) && Set.of('f', 'F').contains(ch1)
84                      && Set.of('f', 'F').contains(ch2)) {
85                  return Boolean.FALSE;
86              }
87              break;
88          }
89          case 4: {
90              final char ch0 = str.charAt(0);
91              final char ch1 = str.charAt(1);
92              final char ch2 = str.charAt(2);
93              final char ch3 = str.charAt(3);
94              if (Set.of('t', 'T').contains(ch0) && Set.of('r', 'R').contains(ch1)
95                      && Set.of('u', 'U').contains(ch2)
96                      && Set.of('e', 'E').contains(ch3)) {
97                  return Boolean.TRUE;
98              }
99              break;
100         }
101         case 5: {
102             final char ch0 = str.charAt(0);
103             final char ch1 = str.charAt(1);
104             final char ch2 = str.charAt(2);
105             final char ch3 = str.charAt(3);
106             final char ch4 = str.charAt(4);
107
108             if (Set.of('f', 'F').contains(ch0) && Set.of('a', 'A').contains(ch1)
109                     && Set.of('l', 'L').contains(ch2) && Set.of('s', 'S').contains(ch3)
110                     && Set.of('e', 'E').contains(ch4)) {
```

```
111                     return Boolean.FALSE;
112                 }
113                 break;
114             }
115             default:
116                 return Boolean.FALSE;
117             }
118
119             return null;
120         }
121 }
```

A.4. Aufgabe 2 – partition

Listing A.4: Codeschnipsel für Aufgabe 2 des Experiments.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license
* agreements. See the NOTICE file distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file to You under the Apache
* License, Version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under
* the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
* ANY KIND, either express or implied. See the License for the specific language
* governing permissions and limitations under the License.
*/

/*
* Apache Commons Collections
* Copyright 2001-2020 The Apache Software Foundation
*
* This product includes software developed at
* The Apache Software Foundation (http://www.apache.org/).
*/

// Note to fulfil Apache License: File has been changed.

package aufgabe2;

public class KthSelector {

    /**
    * Partition an array slice around a pivot. Partitioning exchanges array
    * elements such that all elements smaller than pivot are before it and all
    * elements larger than pivot are after it.
    *
    * <pre>
    * KthSelector.partition(new double[] { 6, 5, 4 }, 0) = 2
    * KthSelector.partition(new double[] { 6, 5, 4 }, 1) = 1
    * KthSelector.partition(new double[] { 6, 5, 4 }, 2) = 0
    *
    * KthSelector.partition(new double[] { 6, 8, 4 }, 0) = 1
    * KthSelector.partition(new double[] { 6, 8, 4 }, 1) = 2
    * KthSelector.partition(new double[] { 6, 8, 4 }, 2) = 0
    * </pre>
    *
    * @param work  work array
    * @param pivot initial index of the pivot
    * @return the index of the pivot after partition
    */
    public static int partition(final double[] work, final int pivot) {
        final int end = work.length - 1;
        final double value = work[pivot];
        work[pivot] = work[0];
        int i = 1;
        int j = end;
```

```
54          while (i < j) {
55              while (true) {
56                  if (i < j && work[j] > value) {
57                      --j;
58                  } else {
59                      break;
60                  }
61              }
62              while (true) {
63                  if (i < j && work[i] < value) {
64                      ++i;
65                  } else {
66                      break;
67                  }
68              }
69
70              if (i < j) {
71                  final double tmp = work[i];
72                  work[i++] = work[j];
73                  work[j--] = tmp;
74              }
75          }
76
77          if (i >= end || work[i] > value) {
78              --i;
79          }
80
81          work[0] = work[i];
82          work[i] = value;
83          return i;
84      }
85  }
```

A.5. Aufgabe 3 – `indexOfDiff`

Listing A.5: Codeschnipsel für Aufgabe 3 des Experiments.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license
* agreements. See the NOTICE file distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file to You under the Apache
* License, Version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under
* the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
* ANY KIND, either express or implied. See the License for the specific language
* governing permissions and limitations under the License.
*/

/*
* Apache Commons Lang
* Copyright 2001-2020 The Apache Software Foundation
*
* This product includes software developed at
* The Apache Software Foundation (http://www.apache.org/).
*/

// Note to fulfil Apache License: File has been changed.

package aufgabe3;

import org.apache.commons.lang3.ArrayUtils;

public class StringUtils {

    /**
    * Compares all Strings in an array and returns the index at which the Strings
    * begin to differ.
    *
    * For example, <code>indexOfDifference(new String[] {"i am a machine", "i am a
    * robot"})</strong> returns 7.
    *
    * <pre>
    * StringUtils.indexOfDifference(null) = -1
    * StringUtils.indexOfDifference(new String[] {}) = -1
    * StringUtils.indexOfDifference(new String[] {"abc"}) = -1
    * StringUtils.indexOfDifference(new String[] {null, null}) = -1
    * StringUtils.indexOfDifference(new String[] {"", ""}) = -1
    * StringUtils.indexOfDifference(new String[] {"", null}) = 0
    * StringUtils.indexOfDifference(new String[] {"abc", null, null}) = 0
    * StringUtils.indexOfDifference(new String[] {null, null, "abc"}) = 0
    * StringUtils.indexOfDifference(new String[] {"", "abc"}) = 0
    * StringUtils.indexOfDifference(new String[] {"abc", ""}) = 0
    * StringUtils.indexOfDifference(new String[] {"abc", "abc"}) = -1
    * StringUtils.indexOfDifference(new String[] {"abc", "a"}) = 1
    * StringUtils.indexOfDifference(new String[] {"ab", "abxyz"}) = 2
    * StringUtils.indexOfDifference(new String[] {"abcde", "abxyz"}) = 2
    * StringUtils.indexOfDifference(new String[] {"abcde", "xyz"}) = 0
```

```
54      * StringUtils.indexOfDifference(new String[] {"xyz", "abcde"}) = 0
55      * StringUtils.indexOfDifference(new String[] {"i am a machine", "i am a robot"}) = 7
56      * </pre>
57      *
58      * @param css array of Strings, entries may be null
59      * @return the index where the strings begin to differ; -1 if they are all equal
60      */
61     public static int indexOfDiff(final String[] css) {
62         if (ArrayUtils.getLength(css) <= 1) {
63             return -1;
64         }
65         boolean anyStringNull = false;
66         boolean allStringsNull = true;
67         final int arrayLen = css.length;
68         int shortestStrLen = Integer.MAX_VALUE;
69         int longestStrLen = 0;
70 
71         for (final String cs : css) {
72             if (cs == null) {
73                 anyStringNull = true;
74                 shortestStrLen = 0;
75             } else {
76                 allStringsNull = false;
77                 shortestStrLen = Math.min(cs.length(), shortestStrLen);
78                 longestStrLen = Math.max(cs.length(), longestStrLen);
79             }
80         }
81 
82         if (allStringsNull || longestStrLen == 0 && !anyStringNull) {
83             return -1;
84         }
85 
86         if (shortestStrLen == 1) {
87             return 0;
88         }
89 
90         int firstDiff = -1;
91         for (int stringPos = 0; stringPos < shortestStrLen; stringPos++) {
92             final char comparisonChar = css[0].charAt(stringPos);
93             for (int arrayPos = 1; arrayPos < arrayLen; arrayPos++) {
94                 if (css[arrayPos].charAt(stringPos) != comparisonChar) {
95                     firstDiff = stringPos;
96                     break;
97                 }
98             }
99             if (firstDiff != -1) {
100                 break;
101             }
102         }
103 
104         if (firstDiff == -1 && shortestStrLen != longestStrLen) {
105             return shortestStrLen;
106         }
107         return firstDiff;
108     }
109 }
```

B. Beispiel eines Aufgabenzettels

Teilnehmer: 31

Aufgabe 1 – `toBooleanObject`

1. Schaue dir die gezeigte Methode gründlich an.
2. **Berechne** die Werte, die der Programmcode für alle Beispielaufrufe berechnen würde.
3. **Bestimme** den Wert, der laut Javadoc erwartet wird.

- Beachte: Im Code sind Fehler. Der Code kann sich abweichend zum Javadoc verhalten.
- Wenn die Tabelle ausgefüllt ist, sage bitte Bescheid.
- Sei bei der Bearbeitung möglichst schnell, aber auch korrekt.

Methodenaufruf	**Rückgabewert**	**Rückgabewert laut Javadoc**
`toBooleanObject("ofo")`		
`toBooleanObject("false")`		
`toBooleanObject("on")`		
`toBooleanObject("y")`		
`toBooleanObject("ttttt")`		

Notizen:

C. Abhängige und unabhängige Variablen

Tabelle C.1.: Abhängige und unabhängige Variablen.

Variablenname	*Variablentyp*	*Abkürzung*	*Klasse*	*Entität*	*Attribut*	*Skala*	*Einheit*	*Wertebereich*	*Quantifizierung*
Bearbeitungszeit	abhängig	t	Prozess	Performance	intern	Ratio	Sekunden [s]	$t \in \mathbb{N}$	Summe der Differenzen von End- und Startzeit
korrekte Rückgabewerte laut Dokumentation	abhängig	kD	Prozess	Performance	intern	Intervall	—	$kD \leq 15, kD \in \mathbb{N}$	Vergleich mit Musterlösung
korrekte Rückgabewerte laut Code	abhängig	kC	Prozess	Performance	intern	Intervall	—	$kC \leq 15, kC \in \mathbb{N}$	Vergleich mit Musterlösung
Codeverständnis	abhängig	TAU	Prozess	Performance	extern	Ratio	—	$0 \leq TAU \leq 1$, $TAU \in \mathbb{R}$	Basiert auf TAU aus [SBV+19][1]

Tabelle C.1 wird auf der nächsten Seite fortgeführt.

[1] Mit den gegebenen Abkürzungen wird TAU folgendermaßen berechnet: $TAU = \frac{kC+kD}{30} \times (1 - \frac{t}{t_{\max}})$

Fortsetzung von **Tabelle C.1**.

Variablenname	*Variablentyp*	*Abkürzung*	*Klasse*	*Entität*	*Attribut*	*Skala*	*Einheit*	*Wertebereich*	*Quantifizierung*
wahrgenommene Schwierigkeit	abhängig	wS	Prozess	Subjektive Bewertung	extern	Intervall	—	$0 \leq wS \leq 10$, $wS \in \mathbb{R}$	Arithmetisches Mittel von drei Likert-Skalen
Metrikabweichung	abhängig	Ma	Prozess	Abweichung	extern	Intervall	—	$0 \leq Ma \leq 10$, $Ma \in \mathbb{R}$	$\lvert wS - Mw \rvert$
Metrikwert	unabhängig	Mw	Methode	Metrik	extern	Intervall	—	$0, \ldots, 10$	—
Alter	unabhängig	A	Ressource	Beschreibung einer Person	intern	Intervall	Jahre	$A \in \mathbb{N}$	Demografischer Fragebogen
Geschlecht	unabhängig	G	Ressource	Beschreibung einer Person	intern	Nominal	—	$\{\mathrm{m}, \mathrm{w}, \mathrm{d}\}$	Demografischer Fragebogen
Studienfach	unabhängig	Sf	Ressource	Beschreibung einer Person	intern	Nominal	—	{Softwaretechnik, Informatik}	Demografischer Fragebogen
angestrebter akademische Grad	unabhängig	$Grad$	Ressouce	Beschreibung einer Person	intern	Nominal	—	{Bachelor, Master}	Demografischer Fragebogen
Fachsemester	unabhängig	Fs	Ressource	Beschreibung einer Person	intern	Ratio	Semester	$Fs \in \mathbb{N}$	Demografischer Fragebogen
Programmiererfahrung mit Java[2]	unabhängig	Pe	Ressource	Beschreibung einer Person	extern	Ratio	Jahre[3]	$Pe \in \mathbb{R}_{\mathbb{0}}^{+}$	Demografischer Fragebogen

Tabelle C.1 wird auf der nächsten Seite fortgeführt.

[2] Nach Siegmund et al. [SKL+14] wird Programmiererfahrung am häufigsten in Jahren erfasst.
[3] Beginnend ab der ersten Programmiererfahrung mit Java.

Fortsetzung von **Tabelle C.1**.

Variablenname	*Variablentyp*	*Abkürzung*	*Klasse*	*Entität*	*Attribut*	*Skala*	*Einheit*	*Wertebereich*	*Quantifizierung*
positive Gefühle	unabhängig	SPANE-P	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$6, \ldots, 30$	SPANE-Fragebogen
negative Gefühle	unabhängig	SPANE-N	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$6, \ldots, 30$	SPANE-Fragebogen
emotionale Balance	unabhängig	SPANE-B	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$-24, \ldots, 24$	SPANE-Fragebogen, Subtraktion von SPANE-B von SPANE-N
Optimismuswert	unabhängig	*Ow*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 12$	LOT-R-Fragebogen
Pessimismuswert	unabhängig	*Pw*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 10$	LOT-R-Fragebogen
Extraversion	unabhängig	*Ev*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 32$	Big Five-Fragebogen
Durchsetzungsfähigkeit	unabhängig	*Df*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 20$	Big Five-Fragebogen
Aktivität	unabhängig	*Ak*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 8$	Big Five-Fragebogen
Verträglichkeit	unabhängig	*Ve*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 36$	Big Five-Fragebogen
Altruismus	unabhängig	*Al*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 16$	Big Five-Fragebogen

Tabelle C.1 wird auf der nächsten Seite fortgeführt.

Fortsetzung von **Tabelle C.1**.

Variablenname	*Variablentyp*	*Abkürzung*	*Klasse*	*Entität*	*Attribut*	*Skala*	*Einheit*	*Wertebereich*	*Quantifizierung*
Entgegenkommen	unabhängig	*Ek*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 12$	Big Five-Fragebogen
Gewissenhaftigkeit	unabhängig	*Ge*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 36$	Big Five-Fragebogen
Ordentlichkeit	unabhängig	*Or*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 8$	Big Five-Fragebogen
Selbstdisziplin	unabhängig	*Se*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 20$	Big Five-Fragebogen
Neurotizismus	unabhängig	*Ne*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 32$	Big Five-Fragebogen
Ängstlichkeit	unabhängig	*An*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 16$	Big Five-Fragebogen
Depression	unabhängig	*De*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 8$	Big Five-Fragebogen
Offenheit	unabhängig	*Of*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 40$	Big Five-Fragebogen
Offenheit für Ästhetik	unabhängig	*OfA*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 12$	Big Five-Fragebogen
Offenheit für Ideen	unabhängig	*OfI*	Ressource	Individuelles Charakteristikum	extern	Intervall	—	$0, \ldots, 20$	Big Five-Fragebogen

D. Skript für die Durchführung des Experiments

Einleitung

Hallo. Ihr könnt euch hier an die Laptops setzen. (Teilnehmer setzten sich.) Mein Name ist Andreas. Wie heißt ihr? Cool, dass ihr an meiner Studie teilnehmt.

- In dieser Studie geht es um Programmverständnis.
- Ziel der Studie ist es, zu untersuchen, welche Faktoren das Verständnis von Quellcode beeinflussen.
- Es gibt eine neue Metrik, die mit maschinellem Lernen arbeitet, die sehr gut misst, wie verständlich Quellcode ist.
- Eine gibt eine Publikation dazu, die nahelegt, dass das Codeverständnis der Entwickler und die Werte diese Metrik sehr gut übereinstimmen.
- Diese Metrik wird in dieser Studie verwendet.
- Eure Aufgabe wird sein, dass ihr euch Quellcode anschaut, versteht und Fragen zum Quellcode beantwortet.

Daten abfragen

- Bevor wir mit der Studie anfangen, habe ich euch eine Einverständniserklärung mitgebracht.
- (Einverständniserklärung umdrehen)
- Damit gebt ihr mir die Einwilligung, dass ich die Daten, die ich erfasse, anonymisiert verwenden darf. Ich erfassen die Bearbeitungszeiten und es werden Fragebögen ausgefüllt.
- Ihr könnt zu jeder Zeit die Teilnahme an der Studie beenden.
- Ihr könnt den Laptop mal aufwecken.
- Ihr seht einen Ordner vor euch. Oben in der Leiste und in einigen Dateien seht ihr eine t und eine Zahl. Das ist euch Kürzel.
- Ich sage ein paarmal, dass ihr eine Datei öffnen sollt. Dann sage ich nur ti…und ihr müsst das i durch eure Zahl ersetzen. Manchmal lasse ich das ti auch komplett weg.
- Füllt bitte Fragebogen `ti_Daten1.ods` aus. Wenn ihr fertig seid, speichert und schließt ihn. Danach könnt ihr `ti_Daten2.ods` ausfüllen.
- Wenn ihr beide Fragebögen ausgefüllt habt, dann speichert und schließt den Fragebogen und sagt mir Bescheid.

Überblick geben

Bevor wir anfangen, gebe ich euch einen Überblick über das, was wir gleich machen werden.

1. Als Erstes erkläre ich euch die Aufgabenstellung.
2. Danach machen wir ein Beispiel.

3. Ich gebe euch Kontextinformationen zur Aufgabe, damit ihr euch in die Aufgabenstellung reinfühlen könnt.
4. Dann schauen wir die IDE an, die wir in dem Experiment verwenden.
5. Dann geht es für euch so richtig los. Ihr bearbeitet drei Aufgaben. Eine nach der anderen.
6. Zum Schluss werden noch zwei Fragebögen ausgefüllt.

Aufgabenstellung

1. Ich werde euch im weiteren Verlauf der Studie mehrere Java-Methoden zeigen.
2. Zusammen mit der Methode gebe ich euch mehrere Methodenaufrufe, also die Parameter eines Aufrufs.
3. Eure Aufgabe ist es, dass ihr euch die gezeigte Java-Methode gründlich anschaut, versteht und den Rückgabewert für die Eingabeparameter berechnet.
4. Zusätzlich für jeden Methodenaufruf sollt ihr den Wert bestimmen, der ihr laut Javadoc erwarten würdet.
5. Danach werde ich euch fragen, den Codeschnipsel zu bewerten. Je nachdem wie leicht oder schwer ihr in fandet.

Hinweise: Im Quellcode sind Bugs, sodass dein Ergebnis von dem abweichen kann, was man laut Dokumentation erwarten würde. Aber das muss nicht immer so sein. Die Bearbeitungszeit der Aufgaben kann variieren.

Beispielaufgabe

- Du darfst dich einmal hier rüber setzen (ein Teilnehmer soll sich von der kurzen an die lange Kante des Tisches setzen).
- (Laptops drehen, damit ich beide Bildschirme sehe, um die Beispiele zu erklären)
- Ich gebe euch dann z. B. so einen Zettel. (Zettel mit dem Beispiel `multiplyByTwo` umdrehen.)
- Hier oben seht ihr die Aufgabenbeschreibung.
- Hier sind die Methode und die Parameter.
- Für `multiplyByTwo` mit 5 als Parameter habe ich berechnet, dass die Methode 15 zurückgibt. Laut Javadoc hätte ich aber 10 erwartet.
- Bei dem Wert 0 hat die Methode 0 zurückgegeben. Das habe ich auch laut Javadoc so erwartet.
- Die Methode ist `null`-safe, d. h. wenn ich null reingeben, dann kommt `null` zurück. Das ist auch so im Javadoc dokumentiert.
- Wenn ihr die zwei Spalten ausgefüllt habt, dann sagt ihr mir bitte Bescheid.
- Dann gebe ich euch so einen kleinen Zettel, wo ihr ankreuzen könnt wie leicht/schwer ihr die Aufgabe fandet. Danach machen wir mit der nächsten Aufgaben weiter.

Kontextinformationen

Ich gebe euch jetzt Kontextinformationen, damit ihr euch besser in die Aufgabenstellung reinfühlen könnt. Stellt euch vor:

- Ihr arbeitet in einem Softwareunternehmen und als Softwareentwickler arbeitest du gerade an einem großen Java-Projekt.

- Als Softwareentwickler in diesem Unternehmen besteht 80 % eurer Arbeit daraus, Quellcode zu verstehen und Fehler zu beheben.
- Es kommt oft vor, dass die Bezeichnung und Dokumentation einer Methode nicht zu dem passen, was die Methode eigentlich berechnet.
- In anderen Worten: Die Dokumentation ist korrekt, aber in der Methode ist ein Bug.
- Um zu verstehen, was die Methode wirklich tut, nehmt ihr euch Stift und Papier und rechnet für einige Eingabewerte händisch nach, was die Methode berechnet.
- Das waren die Kontextinformationen zur Aufgabenstellung.

Entwicklungsumgebung und Metrik erklären

Wir schauen uns jetzt die Entwicklungsumgebung an, die wir in dem Experiment verwenden werden. Bitte öffnet die `Beispiel1.html`. Die Entwicklungsumgebung hat folgenden Funktionen, die du jetzt auch ausprobieren kannst:

- markiert einmal die `multiplier` Variable, dann werden alle Vorkommen dieser Variable hervorgehoben. Hovert man mit der Maus über einer Methode, Typen oder Variablen, dann wird der entsprechende Javadoc-Kommentar angezeigt. So wie ihr das aus Eclipse kennt.
- (ein bisschen warten, damit die Teilnehmer die Funktionen ausprobieren können)
- Die Verständlichkeit der aktuellen Methode wird euch oben links angezeigt: Die Werte der Metrik gehen von 0 bis 10.
- Eine sehr leicht zu verstehende Methode hat den Wert 0. Ein sehr schwer zu verstehende Methode hat den Wert 10.
- Hier hat die Metrik den Verständlichkeitswert 1 berechnet.
- Hier siehst du übrigens ein sehr leichtes bzw. eher leichtes Beispiel.
- Bitte schließt das Beispiel jetzt.
- Da es für die Metrik noch kein Eclipse-Plugin gibt, simulieren wir das im Browser.
- Öffnet bitte die Datei `Beispiel2.html` .
- Ich habe schon über diese Metrik gesprochen.
- Diese Metrik berücksichtigt im Gegensatz zu vielen anderen Metrik zum Beispiel Variablennamen und Methodennamen.
- Aber noch andere Aspekte werden berücksichtigt, wie die Qualität von Kommentaren.
- Hier sieht man gut, dass dessen berechnete Werte sehr gut zur Verständlichkeit der Methode passen.
- Wir haben hier ein eher schweres Beispiel: die Variablennamen sind eher kurz und die Kommentare sind kryptisch bzw. weniger hilfreich
- Bitte schließt das Beispiel

Wiederholung der Aufgabenstellung

Zur Wiederholung noch einmal die Aufgabenstellung:

1. Ich werde euch im weiteren Verlauf der Studie mehrere Java-Methoden im Browser zeigen.
2. Zusammen mit der Methode gebe ich euch so einen Zettel (Zettel noch mal umdrehen) mit den Methodenaufrufen, also die Parameter eines Aufrufs.

3. Eure Aufgabe ist es, dass ihr euch die gezeigte Java-Methode gründlich anschaut, versteht und den Rückgabewert für die Eingabeparameter berechnet.
4. Zusätzlich dazu sollt ihr für jeden Methodenaufruf den Wert bestimmen, der ihr laut Javadoc erwarten würdet.
5. **Wichtig**: Im Code kann ein Bug sein, sodass die beiden Ergebnisse der beiden Spalten nicht zusammenpassen.
6. Wenn ihr fertig seid mit den Berechnungen, dann sagt bitte Bescheid.
7. Danach werde ich euch fragen, den Codeschnipsel zu bewerten. Je nachdem wie leicht oder schwer ihr ihn fandet.

Letzte Instruktionen

Einige Ergänzungen zum Szenario:

- Nehmt euch so viel Zeit, wie ihr braucht, bedenkt aber, dass euer Arbeitgeber möchte, dass ihr möglichst effizient seid, aber auch fehlerfrei arbeitet.
- Ihr arbeitet meistens im Homeoffice, d. h. du bist meist alleine, das heißt, ich kann euch während der Studie keine Fragen beantworten.
- Manche Aufgaben sind eventuell schneller oder langsamer erledigt als andere.
- Hast du noch Fragen, die ich jetzt beantworten soll?
- Dann baue ich noch kurz um. Du darfst dich an den Tisch dort setzen.

Teilnehmer bearbeiten Aufgaben

Teilnehmer bearbeiten Aufgabe 1. Wenn ein Teilnehmer fertig ist:

- „Super“ sagen. (Zeiterfassung stoppen)
- Du kannst jetzt die Aufgabe bewerten, der Zettel dafür liegt vor dir.
- (Wenn Teilnehmer bewertet hat, zwischen die Tische gehen, beides mitnehmen und die neuen Zettel verdeckt hinlegen.)
- Wenn du soweit bist, kannst du jetzt mit der Aufgabe x anfangen. (Zeiterfassung starten)

Fragebögen ausfüllen

(Teilnehmer hat alle drei Aufgaben bearbeitet)

- Jetzt darfst du den Fragebogen `ti_Daten3.ods` ausfüllen. Danach speichern und schließen.
- Anschließend darfst du `ti_Daten4.ods` ausfüllen. Bitte wieder speichern und schließen und mir dann Bescheid sagen.
- Hier kannst du dich für den FMST-Schein eintragen. (Liste geben)
- Und du darfst dich hier einmal bedienen. (Süßigkeitenschüssel holen)
- (Warten bis Proband fertig ist und seine Sachen genommen hat. Proband die Tür aufhalten, noch mal bedanken und einen schönen Tag wünschen.)

E. Ergebnisse

E.1. Shapiro-Wilk-Test

Mit $n = 45$ und einem Signifikanzniveau von $\alpha = 0{,}05$, wird bei $p < 0{,}945$ davon ausgegangen, dass die Variable nicht normalverteilt ist.

Tabelle E.1.: Shapiro-Wilk-Tests zur Überprüfung der Normalverteilung ($n = 45$).

	W	p
Metrikabweichung (*Ma*)	0,91	0,00
wahrgenommene Schwierigkeit (*wS*)	0,98	0,76
Codeverständnis (*TAU*)	0,95	0,07
Alter (*A*)	0,85	0,00
Programmiererfahrung mit Java (*Pe*)	0,96	0,08
positive Gefühle (SPANE-P)	0,95	0,06
negative Gefühle (SPANE-N)	0,95	0,06
emotionale Balance (SPANE-B)	0,96	0,14
Optimismuswert (*Ow*)	0,94	0,02
Pessimismuswert (*Pw*)	0,97	0,32
Extraversion (*Ev*)	0,97	0,26
Durchsetzungsfähigkeit (*Df*)	0,95	0,06
Aktivität (*Ak*)	0,95	0,07
Verträglichkeit (*Ve*)	0,98	0,80
Altruismus (*Al*)	0,96	0,13
Entgegenkommen (*Ek*)	0,93	0,01
Gewissenhaftigkeit (*Ge*)	0,99	0,86
Ordentlichkeit (*Or*)	0,94	0,03
Selbstdisziplin (*Se*)	0,98	0,60
Neurotizismus (*Ne*)	0,98	0,68
Ängstlichkeit (*An*)	0,97	0,34
Depression (*De*)	0,95	0,06

Tabelle E.1 wird auf der nächsten Seite fortgeführt.

Fortsetzung von **Tabelle E.1**.

	W	p
Offenheit (*Of*)	0,98	0,46
Offenheit für Ästhetik (*OfA*)	0,96	0,10
Offenheit für Ideen (*OfI*)	0,98	0,60

E.2. Ausgewählte deskriptive Statistiken

Tabelle E.2.: Deskriptive Statistik für die korrekten Rückgabewerte laut Code (*kC*).

	Minimum	1. Quantil	Median	Mean	3. Quantil	Maximum
$Mw_{4,8}$, $n = 45$	6,00	10,00	12,00	11,67	13,00	15,00
Mw_4, $n = 22$	6,00	11,00	12,00	11,50	13,00	15,00
Mw_8, $n = 23$	6,00	10,00	12,00	11,83	14,50	15,00

Tabelle E.3.: Deskriptive Statistik für die korrekten Rückgabewerte laut Dokumentation (*kD*).

	Minimum	1. Quantil	Median	Mean	3. Quantil	Maximum
$Mw_{4,8}$, $n = 45$	10,00	13,00	15,00	13,89	15,00	15,00
Mw_4, $n = 22$	11,00	14,00	14,50	14,09	15,00	15,00
Mw_8, $n = 23$	10,00	13,00	15,00	13,70	15,00	15,00

Tabelle E.4.: Deskriptive Statistik für die Bearbeitungszeit (t) in Sekunden.

	Minimum	1. Quantil	Median	Mean	3. Quantil	Maximum
$Mw_{4,8}$, $n = 45$	1763	2370	2969	2999	3358	5219
Mw_4, $n = 22$	1763	2377	2689	3058	3464	5219
Mw_8, $n = 23$	1765	2472	3047	2943	3336	4217

Tabelle E.5.: Deskriptive Statistik für das Codeverständnis (*TAU*).

	Minimum	1. Quantil	Median	Mean	3. Quantil	Maximum
$Mw_{4,8}$, $n = 45$	0,0000	0,3227	0,3809	0,3655	0,4698	0,6180
Mw_4, $n = 22$	0,0000	0,2969	0,4045	0,3595	0,4742	0,6180
Mw_8, $n = 23$	0,1521	0,3258	0,3664	0,3712	0,4445	0,6162

Tabelle E.6.: Deskriptive Statistik für die Metrikabweichung (*Ma*).

	Minimum	1. Quantil	Median	Mean	3. Quantil	Maximum
$Mw_{4,8}$, $n = 44$	0,000	0,333	1,333	1,385	1,667	3,667
Mw_4, $n = 21$	0,000	0,333	1,167	1,303	1,667	6,667
Mw_8, $n = 23$	0,000	0,667	1,333	1,464	1,833	3,667

E.3. Korrelationsmatrizen

Für die Korrelationsmatrizen E.7, E.8 und E.9 gilt: **** $\Rightarrow p < 0{,}0001$, *** $\Rightarrow p < 0{,}001$, ** $\Rightarrow p < 0{,}01$, * $\Rightarrow p < 0{,}05$

Tabelle E.7.: Korrelationsmatrix ($n = 45$) der explorativ untersuchten Variablen für $Mw_{4,8}$.

	Metrikabweichung (*Ma*)	Alter (*A*)	Programmiererfahrung mit Java (*Pe*)	positive Gefühle (SPANE-P)	negative Gefühle (SPANE-N)	emotionale Balance (SPANE-B)	Optimismuswert (*Ow*)	Pessimismuswert (*Pw*)	Extraversion (*Ev*)	Durchsetzungsfähigkeit (*Df*)	Aktivität (*Ak*)	Verträglichkeit (*Ve*)	Altruismus (*Al*)	Entgegenkommen (*Ek*)	Gewissenhaftigkeit (*Ge*)	Ordentlichkeit (*Or*)	Selbstdisziplin (*Se*)	Neurotizismus (*Ne*)	Ängstlichkeit (*An*)	Depression (*De*)	Offenheit (*Of*)	Offenheit für Ästhetik (*OfA*)	Offenheit für Ideen (*OfI*)
Metrikabweichung (*Ma*)	1,00	-0,10	-0,07	0,08	-0,26	0,16	0,18	0,16	0,02	-0,02	0,17	0,03	-0,24	0,32*	0,05	0,02	0,12	-0,29	-0,31*	-0,10	0,12	-0,16	0,22
Alter (*A*)		1,00	0,11	-0,10	-0,02	-0,06	0,06	-0,05	0,09	0,12	-0,14	-0,10	0,02	-0,09	-0,21	-0,26	-0,13	0,08	0,14	-0,06	0,02	0,03	0,04
Programmiererfahrung mit Java (*Pe*)			1,00	0,03	0,14	-0,07	-0,01	0,02	0,02	0,07	-0,06	0,03	-0,07	0,06	-0,02	-0,15	0,03	-0,08	-0,08	-0,05	-0,08	-0,22	0,07
positive Gefühle (SPANE-P)				1,00	-0,58****	0,86****	0,39**	0,30*	0,30*	0,32*	0,24	-0,10	-0,12	-0,07	-0,04	0,02	-0,09	-0,38*	-0,35*	-0,44**	0,24	0,08	0,31*
negative Gefühle (SPANE-N)					1,00	-0,89****	-0,26	-0,35*	-0,17	-0,24	0,00	0,12	0,26	-0,07	-0,10	-0,23	-0,07	0,49***	0,49***	0,58****	-0,07	0,01	-0,10
emotionale Balance (SPANE-B)						1,00	0,35*	0,37*	0,30	0,36*	0,16	-0,12	-0,22	0,01	0,08	0,17	0,03	-0,49***	-0,47**	-0,59****	0,19	0,04	0,25
Optimismuswert (*Ow*)							1,00	0,61****	0,47**	0,40**	0,52***	0,32*	0,28	0,11	0,14	0,10	0,09	-0,45**	-0,42**	-0,41**	0,45**	0,24	0,47**
Pessimismuswert (*Pw*)								1,00	0,58****	0,55***	0,52***	0,31*	0,27	0,17	0,34*	0,19	0,30*	-0,44**	-0,42**	-0,43**	0,53***	0,50***	0,50***
Extraversion (*Ev*)									1,00	0,98****	0,81****	0,24	0,18	0,15	0,17	0,00	0,14	-0,38*	-0,27	-0,46**	0,50***	0,29	0,52***
Durchsetzungsfähigkeit (*Df*)										1,00	0,70****	0,19	0,11	0,13	0,15	0,04	0,12	-0,37*	-0,27	-0,47**	0,44**	0,26	0,46**
Aktivität (*Ak*)											1,00	0,36*	0,29	0,25	0,24	-0,03	0,22	-0,29	-0,26	-0,28	0,54***	0,26	0,63****
Verträglichkeit (*Ve*)												1,00	0,80****	0,72****	0,32*	0,18	0,30	-0,20	-0,10	-0,19	0,32*	0,16	0,30*
Altruismus (*Al*)													1,00	0,32*	0,18	-0,02	0,16	0,05	0,08	0,09	0,19	0,17	0,19
Entgegenkommen (*Ek*)														1,00	0,25	0,20	0,24	-0,21	-0,11	-0,30*	0,14	-0,01	0,15
Gewissenhaftigkeit (*Ge*)															1,00	0,69****	0,95****	-0,29	-0,27	-0,29	0,09	-0,07	0,24
Ordentlichkeit (*Or*)																1,00	0,51***	-0,34*	-0,37*	-0,31*	-0,09	-0,09	-0,05
Selbstdisziplin (*Se*)																	1,00	-0,25	-0,21	-0,26	0,12	-0,08	0,27
Neurotizismus (*Ne*)																		1,00	0,92****	0,69****	-0,27	-0,12	-0,35*
Ängstlichkeit (*An*)																			1,00	0,47**	-0,21	-0,10	-0,30
Depression (*De*)																				1,00	-0,16	-0,10	-0,20
Offenheit (*Of*)																					1,00	0,71****	0,87****
Offenheit für Ästhetik (*OfA*)																						1,00	0,38*
Offenheit für Ideen (*OfI*)																							1,00

Tabelle E.8.: Korrelationsmatrix für Mw_4 ($n = 22$) der explorativ untersuchten Variablen.

	Metrikabweichung (*Ma*)	Alter (*A*)	Programmiererfahrung mit Java (*Pe*)	positive Gefühle (SPANE-P)	negative Gefühle (SPANE-N)	emotionale Balance (SPANE-B)	Optimismuswert (*Ow*)	Pessimismuswert (*Pw*)	Extraversion (*Ev*)	Durchsetzungsfähigkeit (*Df*)	Aktivität (*Ak*)	Verträglichkeit (*Ve*)	Altruismus (*Al*)	Entgegenkommen (*Ek*)	Gewissenhaftigkeit (*Ge*)	Ordentlichkeit (*Or*)	Selbstdisziplin (*Se*)	Neurotizismus (*Ne*)	Ängstlichkeit (*An*)	Depression (*De*)	Offenheit (*Of*)	Offenheit für Ästhetik (*OfA*)	Offenheit für Ideen (*OfI*)
Metrikabweichung (*Ma*)	1,00	-0,13	-0,33	0,34	-0,51*	0,48*	0,26	0,33	0,23	0,23	0,34	0,03	-0,25	0,51*	0,10	0,18	0,10	-0,31	-0,34	-0,30	0,15	-0,16	0,24
Alter (*A*)		1,00	0,01	-0,31	0,12	-0,24	-0,03	-0,04	0,04	0,05	-0,24	-0,12	0,14	-0,20	-0,10	-0,39	-0,06	0,28	0,33	0,19	0,13	0,13	0,15
Programmiererfahrung mit Java (*Pe*)			1,00	-0,15	0,28	-0,19	-0,14	-0,18	0,02	0,10	-0,09	0,14	-0,02	0,17	0,08	-0,06	0,15	0,15	0,23	-0,08	-0,12	-0,32	0,10
positive Gefühle (SPANE-P)				1,00	-0,66**	0,87****	0,15	0,08	0,24	0,30	0,24	-0,05	-0,27	0,29	0,32	0,21	0,34	-0,47*	-0,32	-0,50*	-0,14	-0,41	0,12
negative Gefühle (SPANE-N)					1,00	-0,93****	-0,08	-0,21	-0,11	-0,19	-0,13	0,23	0,51*	-0,33	-0,25	-0,21	-0,27	0,53*	0,58**	0,55*	0,12	0,35	-0,07
emotionale Balance (SPANE-B)						1,00	0,10	0,14	0,19	0,27	0,22	-0,15	-0,48*	0,40	0,29	0,22	0,30	-0,54*	-0,50*	-0,59**	-0,12	-0,45*	0,14
Optimismuswert (*Ow*)							1,00	0,37	0,37	0,28	0,58**	0,26	0,21	-0,03	0,52*	0,31	0,44*	-0,31	-0,28	-0,24	0,46*	0,05	0,58**
Pessimismuswert (*Pw*)								1,00	0,46*	0,44*	0,46*	0,28	0,14	0,28	0,54*	0,46*	0,48*	-0,36	-0,27	-0,44*	0,47*	0,38	0,41
Extraversion (*Ev*)									1,00	0,98****	0,84****	0,11	0,03	0,01	0,30	0,18	0,21	-0,38	-0,21	-0,49*	0,28	-0,03	0,37
Durchsetzungsfähigkeit (*Df*)										1,00	0,74***	0,05	-0,09	0,05	0,32	0,24	0,22	-0,35	-0,20	-0,48*	0,21	-0,10	0,31
Aktivität (*Ak*)											1,00	0,28	0,11	0,17	0,36	0,22	0,29	-0,41	-0,27	-0,56**	0,37	-0,05	0,49*
Verträglichkeit (*Ve*)												1,00	0,74***	0,51*	0,32	-0,05	0,39	-0,19	0,03	-0,19	0,51*	0,13	0,54*
Altruismus (*Al*)													1,00	-0,07	0,06	-0,33	0,13	-0,03	0,17	0,17	0,38	0,30	0,37
Entgegenkommen (*Ek*)														1,00	0,24	0,10	0,34	-0,16	-0,05	-0,43	0,13	-0,30	0,33
Gewissenhaftigkeit (*Ge*)															1,00	0,68***	0,95****	-0,48*	-0,43*	-0,52*	0,39	0,10	0,49*
Ordentlichkeit (*Or*)																1,00	0,50*	-0,30	-0,43*	-0,19	0,19	0,19	0,02
Selbstdisziplin (*Se*)																	1,00	-0,48*	-0,38	-0,61**	0,34	0,08	0,50*
Neurotizismus (*Ne*)																		1,00	0,92****	0,68***	-0,11	-0,01	-0,21
Ängstlichkeit (*An*)																			1,00	0,50*	-0,01	-0,01	-0,05
Depression (*De*)																				1,00	-0,09	0,10	-0,29
Offenheit (*Of*)																					1,00	0,61**	0,82****
Offenheit für Ästhetik (*OfA*)																						1,00	0,17
Offenheit für Ideen (*OfI*)																							1,00

Tabelle E.9.: Korrelationsmatrix für Mw_8 ($n = 23$) der explorativ untersuchten Variablen.

	Metrikabweichung (*Ma*)	Alter (*A*)	Programmiererfahrung mit Java (*Pe*)	positive Gefühle (SPANE-P)	negative Gefühle (SPANE-N)	emotionale Balance (SPANE-B)	Optimismuswert (*Ow*)	Pessimismuswert (*Pw*)	Extraversion (*Ev*)	Durchsetzungsfähigkeit (*Df*)	Aktivität (*Ak*)	Verträglichkeit (*Ve*)	Altruismus (*Al*)	Entgegenkommen (*Ek*)	Gewissenhaftigkeit (*Ge*)	Ordentlichkeit (*Or*)	Selbstdisziplin (*Se*)	Neurotizismus (*Ne*)	Ängstlichkeit (*An*)	Depression (*De*)	Offenheit (*Of*)	Offenheit für Ästhetik (*OfA*)	Offenheit für Ideen (*OfI*)
Metrikabweichung (*Ma*)	1,00	-0,05	0,22	-0,18	0,05	-0,20	0,12	0,04	-0,17	-0,21	0,03	0,05	-0,20	0,11	0,07	-0,09	0,16	-0,26	-0,30	0,14	0,07	-0,13	0,20
Alter (*A*)		1,00	0,14	0,09	-0,23	0,15	0,19	-0,03	0,11	0,11	-0,05	-0,05	-0,04	0,03	-0,22	-0,05	-0,16	-0,13	-0,07	-0,29	-0,16	-0,14	-0,14
Programmiererfahrung mit Java (*Pe*)			1,00	0,18	0,00	0,05	0,16	0,23	0,03	0,02	0,03	-0,02	-0,04	-0,02	-0,09	-0,18	-0,04	-0,34	-0,44*	0,04	-0,08	-0,12	-0,01
positive Gefühle (SPANE-P)				1,00	-0,53**	0,87****	0,56**	0,48*	0,32	0,31	0,25	-0,10	0,03	-0,26	-0,30	-0,11	-0,38	-0,42*	-0,43*	-0,41	0,45*	0,39	0,43*
negative Gefühle (SPANE-N)					1,00	-0,85****	-0,47*	-0,56**	-0,23	-0,29	0,04	-0,06	-0,02	0,10	-0,02	-0,25	0,06	0,46*	0,42*	0,61**	-0,22	-0,25	-0,21
emotionale Balance (SPANE-B)						1,00	0,57**	0,59**	0,33	0,37	0,12	-0,01	0,05	-0,19	-0,11	0,11	-0,20	-0,49*	-0,47*	-0,58**	0,40	0,38	0,37
Optimismuswert (*Ow*)							1,00	0,77****	0,51*	0,47*	0,46*	0,40	0,37	0,22	-0,15	-0,08	-0,23	-0,59**	-0,56**	-0,61**	0,40	0,44*	0,37
Pessimismuswert (*Pw*)								1,00	0,69***	0,68***	0,58**	0,30	0,33	0,08	0,17	-0,02	0,13	-0,52*	-0,56**	-0,50*	0,55**	0,58**	0,55**
Extraversion (*Ev*)									1,00	0,98****	0,81****	0,39	0,39	0,29	0,06	-0,09	0,12	-0,36	-0,34	-0,43*	0,70***	0,58**	0,65***
Durchsetzungsfähigkeit (*Df*)										1,00	0,72***	0,32	0,32	0,22	0,04	-0,06	0,10	-0,35	-0,33	-0,45*	0,66***	0,55**	0,59**
Aktivität (*Ak*)											1,00	0,38	0,45*	0,31	0,12	-0,20	0,15	-0,22	-0,32	-0,07	0,65***	0,51*	0,73****
Verträglichkeit (*Ve*)												1,00	0,84****	0,89****	0,34	0,43*	0,26	-0,28	-0,29	-0,33	0,12	0,15	0,09
Altruismus (*Al*)													1,00	0,64***	0,33	0,27	0,23	-0,01	-0,08	-0,17	0,01	0,07	0,04
Entgegenkommen (*Ek*)														1,00	0,21	0,32	0,17	-0,23	-0,20	-0,26	0,10	0,17	0,00
Gewissenhaftigkeit (*Ge*)															1,00	0,62**	0,95****	-0,09	-0,14	-0,06	-0,15	-0,24	0,07
Ordentlichkeit (*Or*)																1,00	0,46*	-0,30	-0,28	-0,39	-0,29	-0,28	-0,12
Selbstdisziplin (*Se*)																	1,00	-0,02	-0,08	0,05	-0,10	-0,26	0,09
Neurotizismus (*Ne*)																		1,00	0,93****	0,66***	-0,38	-0,25	-0,46*
Ängstlichkeit (*An*)																			1,00	0,44*	-0,37	-0,21	-0,51*
Depression (*De*)																				1,00	-0,18	-0,30	-0,10
Offenheit (*Of*)																					1,00	0,83****	0,87****
Offenheit für Ästhetik (*OfA*)																						1,00	0,56**
Offenheit für Ideen (*OfI*)																							1,00

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift