

Requirements Engineering für erklärbare Systeme¹

Larissa Chazette²

Abstract: Digitale Systeme berühren fast alle Bereiche des alltäglichen Lebens, daher gewinnt die Qualität der Interaktion zwischen Menschen und Systemen immer stärker an Bedeutung. Erklärbarkeit bezeichnet die Fähigkeit eines Systems, Informationen zu geben, um einen bestimmten Aspekt eines Systems in einem bestimmten Kontext verständlich zu kommunizieren. Erklärbarkeit ist somit zu einer wichtigen Qualitätsanforderung geworden. Um Systeme zu entwickeln, müssen Softwareingenieure wissen, wie sie abstrakte Qualitätsziele in konkrete, reale Lösungen umsetzen können. Das Requirements-Engineering bietet hier einen strukturierten Ansatz, Qualitätsanforderungen besser zu verstehen und zu operationalisieren. Aktuell gibt es keine theoretische Basis und Empfehlungen für das Requirements Engineering, für den Entwurf von erklärbaren Systemen. Um diese Lücken zu schließen, schafft diese Dissertation zunächst die theoretische Basis und schlägt aufbauend darauf Maßnahmen vor.

1 Einleitung

Informationssysteme sind im modernen Leben allgegenwärtig und werden durch immer komplexere und oft schwer verständliche Algorithmen gesteuert. Diese Systeme haben mittlerweile einen so starken Einfluss auf unsere Gesellschaft, dass es fast unmöglich ist, sich einen Bereich vorzustellen, der noch nicht von Informationssystemen betroffen ist. Sie helfen Menschen im Alltag und unterstützen sie bei Entscheidungen. Neben fortwährenden Verflechtung von Software und Gesellschaft nimmt gleichzeitig auch die Komplexität dieser Systeme stetig zu. Daher ist es von entscheidender Bedeutung, die Qualität dieser Systeme stärker in den Fokus zu rücken.

Softwarequalität ist der Grad, in dem ein System die erklärten und implizierten Bedürfnisse der Beteiligten erfüllt [IS11]. Anforderung bilden die Grundlage für gute Softwarequalität [Ru14]. Als Anforderung bezeichnet man eine dokumentierte Repräsentation eines wahrgenommenen Bedürfnisses eines Stakeholders bzw. einer Eigenschaft oder Fähigkeit, die ein System haben soll [Gl22].

Die benötigten Qualitätsanforderungen für die Systeme von heute gehen weit über die bloße *Herstellung* von Software hinaus und müssen nicht mehr nur frei von Bugs, Schwachstellen und anderen Fehlern sein [Oz21]. Im Laufe der Zeit tauchten neue Arten von Qualitätsanforderungen auf, die als Reaktion auf die steigenden Anforderungen der realen Welt immer

¹ Englischer Titel der Dissertation: Requirements Engineering for Explainable Systems

² Leibniz Universität Hannover, Fachgebiet Software Engineering, Welfengarten 1, 30163 Hannover, Deutschland
larissa.chazette@inf.uni-hannover.de

wichtiger geworden sind. **Erklärbarkeit** ist ein Beispiel für so eine Qualitätsanforderung, die in den letzten Jahren aufgrund der zunehmenden Omnipräsenz von Informationssystemen an Bedeutung gewonnen hat und die die Bedeutung der Qualität der Mensch-Maschine-Kommunikation unterstreicht. Denn wenn Informationssysteme mit samt Ihrem Verhalten sowie den von ihnen gelieferten Informationen in unserer Gesellschaft immer wichtiger werden, ist es unerlässlich, dass die Nutzer dieser Systeme diese Informationen verstehen und nachvollziehen können.

Die Erklärbarkeit von Software-Entscheidungen wurde vor allem in der Data-Science-Gemeinschaft als wichtige Anforderung für Modelle des maschinellen Lernens intensiv debattiert. Modelle, Algorithmen oder Systeme, die sich auf den ersten Blick unverständlich verhalten, sind erklärbar wenn sie in der Lage sind, ihr Verhalten zu erklären. Die Erklärung muss hierbei auf die Empfänger und ihre Ziele ausgerichtet sein. Erklärbarkeit trägt maßgeblich zur Erreichung verschiedener anderer wichtiger Qualitätsaspekte wie z.B. Transparenz und Fairness in modernen Systemen bei.

Trotz der Bedeutung der Erklärbarkeit und der explosionsartigen Zunahme der Forschung in diesem Bereich in den letzten Jahren, fehlte jedoch immer noch eine theoretische Grundlage, die sie als Qualitätsanforderung betrachtet und Leitlinien für die Praxis liefert. Die Entwicklung erklärbarer Systeme erfordert ein gründliches Verständnis von Erklärbarkeit, der Taxonomie, der Wechselwirkungen mit anderen Faktoren der Systemqualität und praktischen Aspekten, die relevant sind. Um die Entwicklung qualitativ hochwertiger erklärbarer Systeme zu unterstützen, benötigen Software- und Requirements-Engineering Methoden, die ihnen bei der Analyse, Definition und Testen von Erklärbarkeitsanforderungen helfen.

Daher war das übergeordnete Ziel der Forschung im Rahmen meiner Dissertation, diese Lücken zu schließen und das Requirements-Engineering für erklärbare Systeme sowie die Entwicklung von erklärbaren Systemen in der Praxis zu unterstützen. Sowohl mit der notwendigen theoretischen Grundlage als auch mit anwendbaren Artefakten. Bei der Forschung wurde ein multimethodischer Ansatz verfolgt. Zum Einsatz kamen Literatur- und empirische Studien, Umfragen, Interviews und eine Fallstudie.

Im Abschnitt 3 diskutiere ich die Theorie hinter der Relevanz und dem Einfluss der Erklärbarkeit auf die Qualität eines Systems. Im Abschnitt 4 stelle ich ein Framework-Konzept vor, das die Softwareentwicklung in der Praxis unterstützen kann.

2 Hintergrund

Requirements-Engineering Requirements-Engineering ist eine Teildisziplin des Software-Engineering und ist sehr stark mit der Softwarequalität verbunden. Laut dem International Requirements Engineering Board (IREB) [Gl22], ist Requirements-Engineering der systematische und disziplinierte Ansatz zur Spezifikation und Verwaltung von Anforderungen mit dem Ziel, die Wünsche und Bedürfnisse der Beteiligten zu verstehen und das

Risiko zu minimieren, ein System zu liefern, das diese Wünsche und Bedürfnisse nicht erfüllt.

Anforderungen wirken sich unmittelbar auf den Systementwicklungsprozess aus und bilden die Basis für viele weitere Schritte wie die Kommunikation im Projekt, die Ermittlung von Rationalisierungspotenzialen, die Systemarchitektur sowie Tests und Abnahme [Ru14]. Eine der Hauptaufgaben der Anforderungsanalyse besteht darin, abstrakte Qualitätsaspekte (die auf realen Konzepten und Ideen beruhen) in Qualitätsanforderungen (oder „nicht-funktionale Anforderungen“) und in konkrete Funktionalität zu übersetzen. Dies erfordert ein gemeinsames Verständnis der Stakeholder und eine effektive Kommunikation, um diese Aspekte zu verstehen und gemeinsame Designentscheidungen zu treffen.

Darüber hinaus ist Requirements-Engineering nicht nur ein Prozess der Erhebung und Spezifizierung von Anforderungen, sondern auch ein systematischer Ansatz der effektiven Kommunikation dieser Anforderungen zwischen verschiedenen Stakeholder-Gruppen. In heterogenen Gruppen ist ein gemeinsames Verständnis sowohl ein Schlüsselfaktor als auch eine Herausforderung, da die Gruppenmitglieder möglicherweise dieselben Wörter für unterschiedliche Konzepte oder unterschiedliche Wörter für dieselben Konzepte verwenden, ohne dies zu bemerken [BL13].

Dieses gemeinsame Verständnis zwischen Stakeholder (z.B.: Kunden, Entwicklern) ist eine entscheidende Voraussetzung für die erfolgreiche Entwicklung eines jeden Systems. Zudem ist es für eine effiziente Kommunikation, Verringerung von Risiken (z.B.: Unzufriedenheit der Stakeholder) und der Nachbesserung entscheidend. Artefakte wie Definition, Modelle und Frameworks können dabei helfen, ein gemeinsames Verständnis zu schaffen und zu verbessern [GF15]. Solche Artefakte helfen dabei, diese abstrakten Konzepte in Anforderungen und anschließend in konkrete Funktionalität zu übersetzen und zu prüfen, wie sich die spezifische Funktionalität auf die Qualität auswirkt [MZ11; PK04].

Erklärbarkeit und verwandte Begriffe Die weit verbreitete terminologische Verwirrung im Zusammenhang mit Erklärbarkeit war einer der Beweggründe für diese Arbeit. Qualitätsanforderungen sind nach wie vor eine große Herausforderung für die Praxis, denn es herrscht oft Unklarheit und Verwirrung in Bezug auf ihre Bedeutung, was wiederum ein gemeinsames Verständnis erschwert. Diese Mehrdeutigkeit ist auch im Fall der Erklärbarkeit vorhanden. Hier existieren verschiedene Begriffe für dieselbe Idee oder verschiedene Definitionen und Ideen für den Begriff *Erklärbarkeit*. Die Begriffe *Erklärbarkeit*, *Interpretierbarkeit*, *Verständlichkeit* und *Transparenz*³ werden oft als Synonyme verwendet.

Nach Lipton [Li18] kann *Transparenz* informell als das Gegenteil von Opazität oder *Blackboxness* definiert werden. *Blackboxness* bezieht sich auf die Vorstellung, dass einige Modelle des maschinellen Lernens für uns so geheimnisvoll wie Blackboxen sein können, da ihr Innenleben undurchsichtig und schwer zu interpretieren und kaum nachzuvollziehen ist.

³ Im Englischen: *explainability*, *interpretability*, *understandability*, and *transparency* jeweils.

Transparenz bedeutet daher, „durchzublicken“, um die inneren Mechanismen zu verstehen (nachzuvollziehen), nach denen ein Algorithmus funktioniert oder was ein Modell gelernt hat.

Interpretierbarkeit kann definiert werden als das Maß, in dem der Nutzer die vom System gegebenen Erklärungen und die bereitgestellten Informationen versteht und nutzen kann [Ri18]. Sie wird auch als die Fähigkeit definiert, einem Menschen Informationen in verständlicher Form zu erklären oder zu präsentieren [DK17]. Die Begriffe *Interpretierbarkeit* und *Erklärbarkeit* werden oft verwendet, um sich auf dasselbe Konzept zu beziehen. Die zweite Definition von Interpretierbarkeit überschneidet sich mit dem Konzept von *Erklärbarkeit*, das definiert werden kann, als der Grad bis zu dem ein System Erklärungen für die Ursache seiner Entscheidungen oder Ausgaben liefern kann [Ri18]. Ich bevorzuge die frühere Definition von *Interpretierbarkeit*, weil sie eher mit den subjektiven Aspekten der Interpretation der präsentierten Informationen durch die Nutzer verbunden ist. Genauer gesagt, sind Erklärungen die Operationalisierungen der Erklärbarkeit. Erklärungen können durch die Vermittlung von Informationen die eher subjektiven Faktoren wie Interpretierbarkeit und Verständlichkeit des Systems beeinflussen.

Eine Definition für erklärbare Systeme Allerdings, brauchen Requirements-Engineering und Software-Engineering keine rein abstrakte Definition von Erklärbarkeit, sondern eine, die sich auf Anforderungen an erklärbare **Systeme** konzentriert. In dieser Arbeit schlage ich die folgende Definition für erklärbare Systeme vor:

Ein System S ist in Bezug auf einen Aspekt X von S relativ zu einem Adressaten A im Kontext C nur dann erklärbar, wenn es eine Entität E (den Erklärer) gibt, die es A durch Angabe eines Informationskorpus I (die Erklärung von X) ermöglicht, X von S in C zu verstehen.

3 Erklärbarkeit in der Qualitätslandschaft

Eine Qualitätslandschaft ist die Sammlung von Qualitätsanforderungen und deren Beziehungen und Verbindungen. Eine Qualitätslandschaft setzt sich aus einem Qualitätsspektrum (das von externer bis zu interner Qualität reicht, (vgl. [IS11])), Qualitätsdimensionen und Qualitätsanforderungen zusammen. Diese Konzepte bilden die Landschaftskomposition. Stakeholder sind hierbei die Akteure, die diese Landschaft schaffen, verändern und erleben.

Ich habe sieben Dimensionen identifiziert, die relevant sind für die Qualitätslandschaft: *Bedürfnisse und Erwartungen der Nutzer, kulturelle Werte, Unternehmenswerte, Gesetze und Normen, Domänenaspekte, Projektrestriktionen und Systemaspekte*. Diese Dimensionen beeinflussen die Identifizierung relevanter Qualitätsanforderungen innerhalb eines Systems, die Auswirkungen einer Qualitätsanforderung auf die Systemqualität und die Designentscheidungen zur Operationalisierung einer bestimmten Qualitätsanforderung.

Diese Dimensionen sind auch Perspektiven, aus denen man die Qualität eines Systems betrachten kann.

Um die Rolle und Bedeutung der Erklärbarkeit für moderne Systeme zu verstehen, war es zunächst notwendig, ihre möglichen Auswirkungen auf die Qualitätslandschaft zu untersuchen. Durch eine systematische Literaturrecherche und Workshops war es möglich, die Auswirkungen der Erklärbarkeit auf mehrere andere Qualitätsanforderungen zu ermitteln, die sich auf verschiedene Dimensionen der Softwarequalität auswirken. Dieses Wissen wurde in eine Theorie geflossen, die sowohl in Form eines konzeptionellen Modells als auch eines Katalogs dargestellt wurde. Diese analytische Theorie, eingebettet in ein konzeptionelles Modell, hilft die Relevanz von Erklärbarkeit für die Softwarequalität zu verstehen und kann gleichzeitig ein unterstützendes Artefakt bei Erhebung und Verhandlung während der Anforderungsanalyse sein.

Die Studie hat gezeigt, dass Erklärbarkeit in Wechselwirkung mit mindestens 57 verschiedenen Qualitätsanforderungen steht, die alle Dimensionen der Qualitätslandschaft beeinflussen. Dies verdeutlicht die wichtige Bedeutung der Erklärbarkeit für Softwarequalität. Erklärbarkeit kann als „Hebel“ oder als „Mittel zum Zweck“ betrachtet werden, da sie ein Weg sein kann, um bestimmte Qualitätsanforderungen in einem System zu erreichen, was wiederum bedeutet, dass andere Qualitätsziele durch die Spezifikation und Erfüllung von Erklärbarkeitsanforderungen erfüllt werden können.

Abbildung 1 zeigt die Auswirkungen der Erklärbarkeit auf die verschiedenen Dimensionen, die sowohl die externe als auch die interne Qualität des Systems beeinflussen. Die Kästen in der Mitte der Abbildung zeigen einige Beispiele für die Qualitätsanforderungen, die von Erklärbarkeit beeinflusst werden können und die für jede Dimension relevant sind. Typische Stakeholder-Gruppen in jeder Dimension sind auch in den darunter liegenden Kästen aufgeführt. Eine vollständige Liste aller Qualitätsanforderungen, die durch Erklärbarkeit beeinflusst werden können, ist in meiner Dissertation zu finden (vgl. [Ch23]).

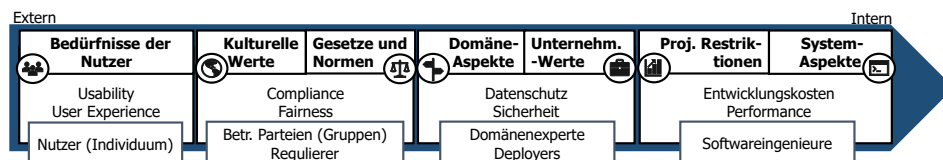


Abb. 1: Erklärbarkeit in der Qualitätslandschaft von Systemen (exemplarischer Ausschnitt des konzeptionelles Modells)

Bei den meisten dieser Qualitätsanforderungen **kann die Erklärbarkeit sowohl einen positiven als auch einen negativen Einfluss haben**, d.h., sie kann sowohl als synergetischer als auch als antagonistischer Aspekt der Qualität wirken. Auf der positiven Seite können Erklärungen zum Beispiel potenziell zur Erreichung vieler wichtiger Qualitätsanforderungen beitragen. Wenn Anforderungen an Erklärungen jedoch nicht ordnungsgemäß erhoben und

analysiert werden, können sie sich aber auch negativ auf dieselben Qualitätsanforderungen auswirken.

Wenn also über die Integration von Erklärungen in ein System nachgedacht wird, kann die Absicht darin bestehen, die Verständlichkeit zu verbessern und damit die Benutzererfahrung mit dem System zu verbessern. Es kann aber auch einen gegenteiligen Effekt haben. Das Ergebnis wird in hohem Maße von den spezifizierten Anforderungen und den zur Umsetzung dieser Anforderungen getroffenen Designentscheidungen beeinflusst. Dies macht die Notwendigkeit einer sorgfältigen Herangehensweise in Bezug auf Erklärbarkeit in der Praxis deutlich. Daher habe ich einige Empfehlungen entwickelt, sowohl in Form von Praktiken, die während der Entwicklung angewendet werden können, als auch in Form eines Frameworks, das Praktikern helfen soll, mit Erklärbarkeitsanforderungen in der Praxis umzugehen.

4 Erklärbarkeit in der Praxis

Auf der Grundlage der Literatur und von Interviews mit Softwareingenieuren, habe ich ein Prozessreferenzmodell entwickelt. Während der Studie konnte sowohl in der Literatur als auch in den Interviews ein besonderer Schwerpunkt auf den Anforderungsprozess und die Validierung dieser Anforderungen festgestellt werden. Daher sollte den anforderungsbezogenen Aktivitäten im Prozess (ob in einem traditionellen oder agilen Umfeld) besondere Aufmerksamkeit geschenkt werden.

In diesem Modell schlage ich sechs Praktiken vor, die dazu beitragen, die Anforderungen an die Erklärbarkeit zu definieren und berücksichtigen:

1. Visionsdefinition: Fokus auf der Definition der Vision/Ziele im Hinblick auf die Qualität und Erklärbarkeit des Systems.
2. Trade-off-Analyse: Fokus auf der Interaktion der Erklärbarkeit mit anderen Qualitätszielen.
3. Stakeholder-Analyse: Fokus auf der Analyse der Bedürfnisse/Ziele der Stakeholder.
4. Backend-Analyse: Fokus auf Systemarchitektur und technischen Aspekten
5. Erklärbarkeitsdesign: Fokus auf Entwurfsentscheidungen zur Erfüllung der Anforderungen.
6. Evaluation: Fokus auf der Bewertung, ob die Anforderungen erfüllt werden

Durch die Integration dieser Praktiken verschiebt sich der Schwerpunkt der Systementwicklung dahin, die Ziele, Bedürfnisse und Wünsche der Nutzer oder Adressaten von Erklärungen in den Vordergrund zu stellen. Das ist wichtig, weil ein erklärbares System Erklärungen bietet, um Wissenslücken zu schließen, die für jeden Adressaten *sehr persönlich* sind und weil Erklärungen als Kommunikationskanal zwischen Mensch und Maschine dienen [Mu19].

Um diesen Prozess zu unterstützen, habe ich ein Framework mit drei Abstraktionsebenen vorgeschlagen, das auf der Grundlage der Literatur und einer Fallstudie erstellt wurde: *Abhängigkeiten*, *Merkmale* und *Bewertung*. Die ersten beiden Ebenen stehen in direktem Zusammenhang mit den Anforderungen an die Erklärbarkeit, während die dritte Ebene die Methoden zur Bewertung der Qualität von Erklärungen und ihrer Auswirkungen auf ein System zusammenfasst. Aber warum soll das Framework wichtig sein?

Nach Boehm [BBL76] erleichtert die Abbildung abstrakter Aspekte auf quantifizierbare Elemente während des Requirements-Engineerings die Entwicklung von Softwareprodukten und hilft dabei, konkrete Lösungen (d. h. Operationalisierungen) zur Erreichung eines gewünschten Qualitätsziels vorzuschlagen. Auf diese Weise ist es möglich, den Prozess des Requirements-Engineerings und der Rationalisierung von Entscheidungen zu unterstützen, indem man diesen Prozess, der eher abstrakt beginnt und sich konkretisieren muss, bis er zu einer richtigen Erklärung wird, fördert. Das Framework hilft bei der Umwandlung der abstrakten Spezifikation der Qualitätsanforderungen in feinere Konstrukte, die auf der untersten Ebene des Modells gemessen werden können. Abbildung 2 zeigt einen exemplarischen Ausschnitt des Frameworks und die empfohlenen Praktiken, die jede Ebene unterstützen können.

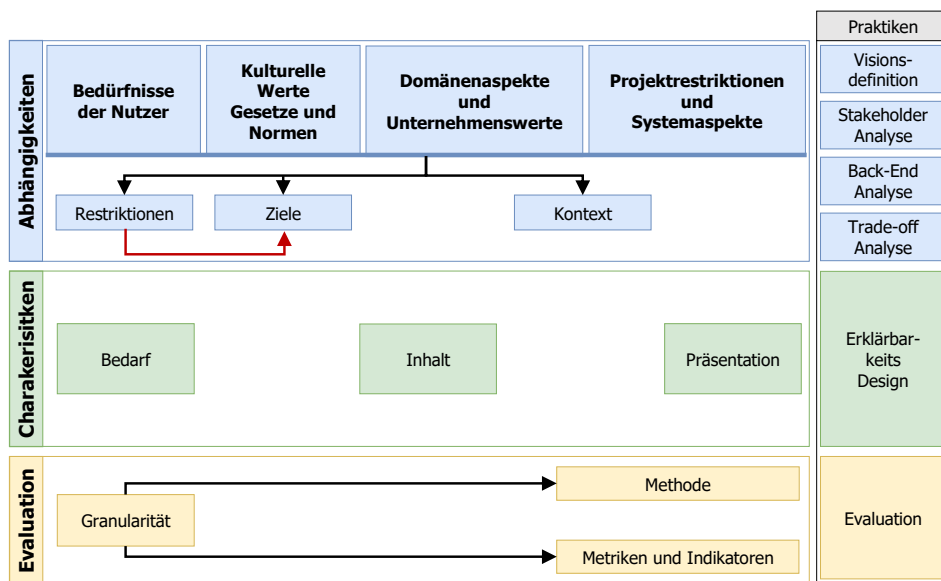


Abb. 2: Exemplarischer Ausschnitt des Frameworks für Erklärbarkeit

Jede Ebene ist in Kategorien unterteilt, die als eine Sammlung von Aspekten fungieren, die für diese Ebene wesentlich sind. Das Framework schlägt sechs Kategorien von Aspekten vor, die berücksichtigt werden sollten: der *Kontext*, in dem das System verwendet wird, die (Qualitäts-) *Ziele*, die möglicherweise vorhandenen *Restriktionen*, die die Ziele einschränken und beeinflussen, den *Bedarf*, der *Inhalt* von Erklärungen und Aspekte die *Präsentation*.

Darüber hinaus haben die übergeordneten Ebenen Auswirkungen auf die darunter liegenden Ebenen: Die Abhängigkeiten beeinflussen die Eigenschaften der Erklärungen (d.h. die Designentscheidungen) und beide beeinflussen die Wahl der Evaluierungsmethoden. Während der Evaluation, kann die Auswirkungen der Erklärbarkeit in zwei Granularitätsgraden bewertet werden: Man kann die Erklärungen als solche bewerten oder die Auswirkungen der Erklärungen auf das System messen. Um die Qualität zu bewerten, sollten sowohl die Methoden als auch die Metriken oder Indikatoren definiert werden.

Die Anwendung des vorgeschlagenen Frameworks wurde anhand einer Fallstudie in einem Softwareunternehmen demonstriert. Die Fallstudie hatte zwei Schwerpunkte: (1) die Evaluierung des Frameworks durch das Sammeln von Feedback von Praktikern in einem Workshop und (2) dem Entwicklungsprozess und der Evaluierung der Erklärungen, die mit Unterstützung des Frameworks entworfen wurden. Die Fallstudie erfolgte in Zusammenarbeit mit einem Softwareunternehmen (52 Mitarbeiter), die eine agile Entwicklungskultur pflegen. Die zum Einsatz kommende Software in der Fallstudie ist eine Navigationssoftware für Smartphones, die auf Schwarmintelligenz basiert und eines der Kernprodukte des Unternehmens darstellt.

Erstens wurde festgestellt, dass einige Qualitätsziele wie *Nutzungssteigerung*, *Streckenakzeptanz* und *Zufriedenheit* durch die Integration von mehr Erklärbarkeit in das System erreicht wurden. Das Framework wurde unterstützend bei der Definition von Anforderungen an die Erklärbarkeit, dem Design von Erklärungen, sowie der Wahl der geeigneten Evaluierungsmethode eingesetzt. Im Rahmen dieser Fallstudie wurden die Erklärungen in das System integriert und eine quantitative Bewertung auf Grundlage eines zweiwöchigen Experiments mit 9745 Endbenutzern durchgeführt. Wir verglichen die Ergebnisse zwischen den Gruppen von Teilnehmern, die verschiedene Arten von Erklärungen erhielten, um die Auswirkungen dieser Erklärungen auf die gesetzten Qualitätsziele zu bewerten.

Die integrierten Erläuterungen trugen dazu bei, alle gesetzten Qualitätsziele zu erreichen. Auf die Frage nach ihren Erfahrungen mit dem Framework berichteten die Praktiker, dass es ihnen half, über Nutzerprofile, Anwendungsfälle, Einschränkungen und übergeordnete Anforderungen nachzudenken, sowie über Strategien zur Bewertung der Auswirkungen von Erklärungen auf die festgelegten Qualitätsanforderungen nachzudenken.

Da die Fallstudie in einem realen Produkt stattfand und das Unternehmen im Live-Betrieb kein zu großes Risiko eingehen wollte, gab es einige Bedenken hinsichtlich einer möglichen negativen Auswirkungen der Erklärungen auf die Qualität des Systems. Aber tatsächlich wurden die Erklärungen so gut angenommen, dass mehrere der Erklärungen weiterentwickelt wurden und zum Zeitpunkt dieser Zusammenfassung immer noch vom Unternehmen eingesetzt werden. Dies deutet darauf hin, dass das angebotene Wissen über Erklärbarkeit als Softwarequalität dazu beigetragen hat, Erklärungen zu entwickeln, die für die Stakeholder Mehrwert anbieten.

5 Ausblick

Wir leben im algorithmischen Zeitalter. Die Fortschritte in der Robotik schreiten sehr schnell voran, und die Kommunikation mit Computersystemen wird wichtiger denn je sein. So sollten die Systeme in der Lage sein, Gesprächspartner zu sein, mit Menschen zu interagieren und ihre Entscheidungen und Beweggründe in angemessener Weise zu erklären. Das unterstreicht einmal mehr, wie wichtig Erklärbarkeit ist – nicht nur als ein weiterer Aspekt der Softwarequalität oder als ein „nice to have“ – sondern als grundlegender Pfeiler in der Kommunikation und Interaktion zwischen Menschen und Informationssystemen. Selbst einfache und leicht zu integrierende Erklärungen können einen positiven Effekt auf wichtige Qualitätsziele haben. Aber gleichzeitig ist ein gründlicher Anforderungsprozess erforderlich, um die negativen Auswirkungen von Erklärungen zu vermeiden.

Der Einfluss der Erklärbarkeit auf so viele entscheidende Dimensionen verdeutlicht die wachsende Notwendigkeit, die Erklärbarkeit beim Entwurf eines Systems zu berücksichtigen. In diesem Sinne besteht der Beitrag dieser Arbeit in der Analyse von Erklärbarkeit aus der Perspektive des Requirements-Engineerings. Sie liefert Erkenntnisse, die eine Definition von Erklärbarkeit im Kontext von Softwaresystemen bringen, sowie Hinweise und Strategien für die Praxis, die Erklärbarkeit greifbar machen. Die vorgeschlagenen Theorien und Artefakte sind hilfreich für die Identifizierung von Anforderungen an Erklärbarkeit, die Auswahl des Erklärungsdesigns und die Bewertung der Qualität der bereitgestellten Erklärungen.

Literatur

- [BBL76] Boehm, B. W.; Brown, J. R.; Lipow, M.: Quantitative Evaluation of Software Quality. In: Proceedings of the 2nd International Conference on Software Engineering. ICSE '76, IEEE Computer Society Press, San Francisco, California, USA, S. 592–605, 1976.
- [BL13] Bittner, E. A. C.; Leimeister, J. M.: Why Shared Understanding Matters – Engineering a Collaboration Process for Shared Understanding to Improve Collaboration Effectiveness in Heterogeneous Teams. In: 2013 46th Hawaii international conference on system sciences. IEEE, Piscataway, S. 106–114, 2013, ISBN: 978-1-4673-5933-7.
- [Ch23] Chazette, L.: Requirements engineering for explainable systems, Diss., Hannover : Institutionelles Repositorium der Leibniz Universität Hannover, 2023.
- [DK17] Doshi-Velez, F.; Kim, B.: Towards A Rigorous Science of Interpretable Machine Learning, 2017.
- [GF15] Glinz, M.; Fricker, S. A.: On shared understanding in software engineering: an essay. Computer Science - Research and Development 30/3, S. 363–376, Aug. 2015.
- [Gl22] Glinz, Martin: A glossary of requirements engineering terminology, Techn. Ber. 2.0.1, International Requirements Engineering Board, Juli 2022.

- [IS11] ISO Central Secretary: ISO/IEC 25010:2011 Systems and Software Engineering- Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models, Standard ISO/IEC 25010:2011, International Organization for Standardization, 2011.
- [Li18] Lipton, Z. C.: The Mythos of Model Interpretability. Commun. ACM 61/10, S. 36–43, 2018.
- [Mu19] Mueller, S. T.; Hoffman, R. R.; Clancey, W.; Emrey, A.; Klein, G.: Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI, 2019.
- [MZ11] Mairiza, D.; Zowghi, D.: Constructing a Catalogue of Conflicts among Non-functional Requirements. In (Maciaszek, L. A.; Loucopoulos, P., Hrsg.): Evaluation of Novel Approaches to Software Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 31–44, 2011, ISBN: 978-3-642-23391-3.
- [Oz21] Ozkaya, I.: Can We Really Achieve Software Quality? IEEE Software 38/3, S. 3–6, 2021.
- [PK04] Paech, B.; Kerkow, D.: Non-functional Requirements Engineering - Quality is Essential. In: 10th International Workshop on Requirements Engineering Foundation for Software Quality. 2004.
- [Ri18] Richard Tomsett; Dave Braines; Dan Harborne; Alun D. Preece; Supriyo Chakraborty: Interpretable to Whom? A Role-based Model for Analyzing Interpretable Machine Learning Systems. CoRR abs/1806.07552/, 2018.
- [Ru14] Rupp, C. et al.: Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil. Carl Hanser Verlag GmbH Co KG, 2014.



Larissa Chazette, geboren am 1. Juli 1991 in Belém, Pará, Brasilien, schloss 2011 ihr Bachelorstudium der Informatik an der Universidade da Amazônia in Belém ab. Dank ihrer herausragenden Leistungen erhielt sie ein Forschungsstipendium der brasilianischen Förderagentur für Hochschulbildung (CAPES) für ihr Masterstudium, das sie 2015 an der Universidade Federal do Pará abschloss. Im Jahr 2016 zog sie nach Deutschland, wo sie zunächst als studentische Hilfskraft an verschiedenen Forschungsprojekten arbeitete und später als wissenschaftliche Mitarbeiterin tätig war. Unter der Betreuung von Prof. Kurt Schneider schloss sie 2022 ihre Promotion an der Leibniz Universität Hannover mit *summa cum laude* ab. Während ihrer Promotionszeit veröffentlichte sie mehrere Artikel in hochrangigen Konferenzen und Fachzeitschriften im Bereich Software- und Requirements Engineering, darunter IEEE RE, REJ und JSS. Sie war auch an der Organisation wissenschaftlicher Veranstaltungen beteiligt und war maßgeblich an der Gründung der Workshop-Reihe "Requirements Engineering for Explainable Systems"(RE4ES) beteiligt.