

Warum entstehen in der Anforderungsanalyse Fehler? Eine Synthese empirischer Befunde der letzten 15 Jahre

Oral Avci

SQS Software Quality Systems AG
Stollwerckstr. 11
51149 Köln
oral.avci@sqs.de

Abstract: In Softwareentwicklungsprojekten treten viele Anforderungsfehler auf. Diese werden häufig erst entdeckt, nachdem sie zu zahlreichen Folgefehlern im Entwurf und der Implementierung geführt haben. Der vorliegende Beitrag liefert auf der Grundlage empirischer Befunde eine Antwort auf die Frage, warum Anforderungsfehler entstehen. Mit der Kenntnis ihrer Ursachen können Anforderungsfehler vermieden oder durch gezielte Maßnahmen der Qualitätssicherung früher gefunden werden.

1 Einführung

Die Anforderungsanalyse ist eine Teilaufgabe der Softwareentwicklung mit dem Ziel aus Kundenbedürfnissen Softwareanforderungen abzuleiten. Eine Softwareanforderung beschreibt dabei ein gewünschtes funktions- oder qualitätsbezogenes Merkmal einer Software aus Sicht der Kunden, wobei diese Beschreibung unabhängig von dem Entwurf und der Implementierung dieses Merkmals ist. Seit drei Jahrzehnten wird die Anforderungsanalyse als eine eigenständige Teilaufgabe der Softwareentwicklung wahrgenommen. Trotz zahlreicher Fortschritte weist die Praxis der Anforderungsanalyse ein chronisches Problem auf [AW05]: es treten viele Anforderungsfehler auf. Diese werden meist erst entdeckt, nachdem zahlreiche Folgefehler im Entwurf und der Implementierung aufgetreten sind. Der relative Anteil der Anforderungsfehler und ihrer Folgefehler an der Gesamtfehlerzahl in Projekten ist deshalb hoch. In mehreren Studien bilden sie die am häufigsten genannte Fehlerkategorie und können bis zu 50 % aller Fehler ausmachen [BP84], [LV01].

Anforderungsfehler können vermieden werden, wenn ihre Ursachen beseitigt werden. Hierzu ist zu klären, welcher Aspekt der Durchführung der Anforderungsanalyse unter den gegebenen Projektrahmenbedingungen begünstigt hat, dass Anforderungsfehler entstehen. In bestimmten Fällen ist die Beseitigung eines Prozessmangels der Anforderungsanalyse jedoch nicht möglich oder mit erheblichen Nachteilen verbunden, so dass sie ausbleibt. Auch dann ist die Kenntnis der Ursache von Anforderungsfehlern hilfreich, um mit gezielten Maßnahmen der Qualitätssicherung nach entstandenen Anforderungs- und deren Folgefehlern zu suchen.

Folglich stellt sich die Frage, warum Fehler in der Anforderungsanalyse entstehen. Die vorliegende Arbeit liefert einen Beitrag zur Beantwortung dieser Frage. Auf der Grundlage einer systematischen Literaturanalyse erfolgt eine Synthese empirischer Befunde zu Anforderungsfehlern und ihren Ursachen. Die Literaturanalyse berücksichtigt insbesondere Beiträge, die im Zeitraum von Anfang Januar 1990 bis einschließlich Ende Dezember 2005 in sechzehn wissenschaftlichen Zeitschriften und Konferenzbänden von acht Konferenzen veröffentlicht worden sind.

Der vorliegende Beitrag ist wie folgt aufgebaut. In Kapitel 2 wird der theoretische Bezugsrahmen des Beitrags definiert. Kapitel 3 beschreibt, wie die systematische Literaturanalyse als Forschungsmethode eingesetzt wurde. Ergebnis der Literaturanalyse sind Erklärungsmuster für Anforderungsfehler, die auf empirischen Befunden basieren. Diese werden in Kapitel 4 dargestellt. Zentrale Ergebnisse der Literaturanalyse werden in Kapitel 5 diskutiert und mit weiteren empirischen Erkenntnissen in der Literatur verglichen. Die Arbeit schließt mit einem Fazit ab.

Eine detaillierte Beschreibung dieser Untersuchung findet sich in [Av08].

2 Theoretischer Bezugsrahmen

In Anlehnung an [Ie90] und [Ch96] können die Begriffe Irrtum (engl. error), Fehler (engl. fault) und Fehlverhalten (engl. failure) unterschieden werden. Ein Projektmitarbeiter kann sich irren, während er ein Arbeitsergebnis erstellt, d. h. Softwareanforderungen erarbeitet, Entwurfsentscheidungen trifft oder die Software implementiert. Sein *Irrtum* kann dazu führen, dass das bearbeitete Ergebnis mit einem oder mehreren Fehlern behaftet ist (vgl. Abbildung 1). Dabei sind unter einem Irrtum alle Ereignisse zu verstehen, in denen eine geplante Abfolge von geistigen oder körperlichen Aktivitäten nicht zum beabsichtigten Ergebnis führt, sofern dies nicht einem fremden Einwirken zugeschrieben werden kann. Ein *Fehler* ist ein unzureichendes Merkmal oder ein erwartetes, jedoch fehlendes Merkmal eines Arbeitsergebnisses der Softwareentwicklung, sofern es eine Änderung in diesem Ergebnis notwendig macht. Es können Anforderungs-, Entwurfs- und Implementierungsfehler voneinander abgegrenzt werden, je nachdem, ob Entscheidungen in der Anforderungsanalyse, dem Entwurf oder der Implementierung betrachtet werden. Insbesondere werden hier also unter einem Fehler nicht ausschließlich Implementierungsfehler aufgefasst.

Da einzelne Entscheidungen der Teilaufgaben zusammenhängen, kann sich ein Anforderungsfehler als Entwurfs- und Implementierungsfehler fortsetzen (vgl. Abbildung 1). Diese besonderen Entwurfs- und Implementierungsfehler sind also *Folgefehler*. Ein *originärer Fehler* liegt dagegen vor, wenn er infolge eines Irrtums neu eingeführt wird und damit unabhängig von einem Fehler einer vorgelagerten Entscheidung ist. Immer wenn ein Projektmitarbeiter einem *Irrtum* unterliegt, finden originäre Fehler Eingang in ein Ergebnis. Eine Software kann ein *Fehlverhalten* offenbaren, wenn sie einen Implementierungsfehler enthält und ausgeführt wird. Dieses Fehlverhalten ist also die Wirkung eines Implementierungsfehlers.

Tritt ein Fehlverhalten auf, folgt daraus ausschließlich, dass die Software mindestens einen Implementierungsfehler hat. Die Kenntnis eines Fehlverhaltens allein erlaubt jedoch keine Rückschlüsse darüber, welche oder wie viele Implementierungsfehler vorhanden sind, da mehrere Implementierungsfehler gemeinsam als ein Fehlverhalten wahrgenommen werden können.

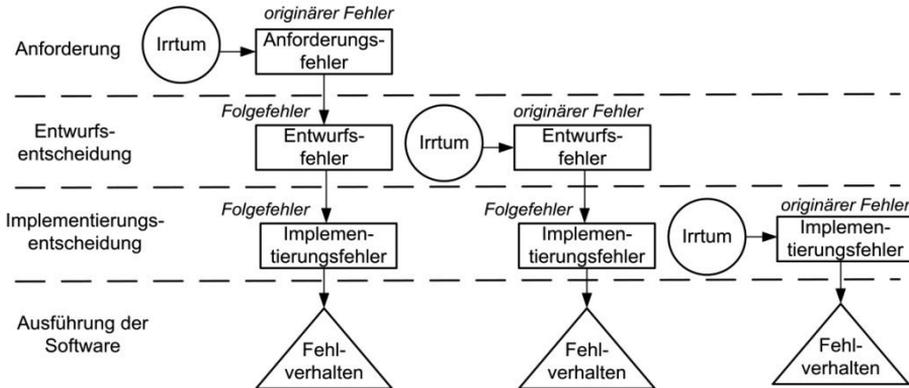


Abbildung 1: Entstehung von Fehlern und ihre Auswirkungen

Ein *Mangel in einem Prozess*, kurz *Prozessmangel*, ist eine Gestaltungsmaßnahme eines Prozesses, die menschliche Irrtümer begünstigt, so dass Fehler in Arbeitsergebnisse der Softwareentwicklung eingeführt werden. Werden diese originären Fehler nicht entdeckt, können zudem im weiteren Verlauf Folgefehler entstehen.

3 Forschungsmethodik

Eine systematische Literaturanalyse untersucht mit einem definierten Ziel einen festgelegten Ausschnitt der Literatur, um eine Synthese relevanter Aussagen zu erreichen. Die Literaturanalyse des vorliegenden Beitrags soll folgende Fragen beantworten:

- Welche Prozessmängel der Anforderungsanalyse sind gemäß Hinweisen in empirischen Untersuchungen bekannt?
- Welche Gemeinsamkeiten weisen die durch Prozessmängel verursachten Fehler jeweils auf?

Es werden ausschließlich Literaturquellen in der Literaturanalyse berücksichtigt, die die folgenden drei Kriterien erfüllen:

- Die Literaturquelle ist empirisch, d. h. gemachte Aussagen beruhen auf Beobachtungen der Realität.
- Die empirischen Aussagen beziehen sich auf die Entwicklung oder Wartung von Anwendungssoftware. Sofern sich Aussagen auf Systemsoftware oder systemnahe Software beziehen, werden sie nur dann berücksichtigt, sofern sie auch auf Anwendungssoftware übertragbar sind.
- Die empirischen Aussagen geben explizit Hinweise auf Maßnahmen im Prozess der Anforderungsanalyse, die Fehler verursachen.

In der Literaturanalyse wurden Beiträge berücksichtigt, die im Zeitraum von Anfang Januar 1990 bis einschließlich Ende Dezember 2005 veröffentlicht worden sind. Insgesamt sechzehn wissenschaftlichen Zeitschriften¹ und Konferenzbände von acht Konferenzen² wurden nach relevanten Literaturquellen durchsucht.

Der Ablauf der Suche nach relevanten Literaturquellen orientiert sich an den Empfehlungen von Webster und Watson [WW02]. Im ersten Schritt, der Ausgangssuche, wurden im definierten Literaturspektrum mit sechzehn Zeitschriften und acht Konferenzen nach Literaturquellen gesucht, die die Auswahlkriterien erfüllen. Die Quellen, die den Kriterien genügen, sind die Ausgangstreffer. Im zweiten Schritt wurden die Literaturverzeichnisse der Ausgangstreffer nach weiteren Literaturquellen durchsucht, die die Auswahlkriterien erfüllen (Rückwärtssuche). Schließlich wurde in einer Vorwärtssuche sowohl über Web of Science³ als auch CiteSeer⁴ ermittelt, welche Literaturquellen die Ausgangstreffer zitieren. Jede so gefundene Literaturquelle wurde dahingehen überprüft, ob sie die Auswahlkriterien erfüllt.

Die Literaturrecherche wurde im Januar und Februar 2006 durchgeführt. In der Ausgangssuche wurden 17 Literaturquellen identifiziert, die die Auswahlkriterien erfüllen. Die Rückwärtssuche erweitert diese Menge um weitere 3 Literaturquellen, die Vorwärtssuche um eine Literaturquelle. Folglich liegen insgesamt 21 Treffer vor. Diese sind im Einzelnen: Q1 [AS85], Q2 [AE96], Q3 [Bh94], Q4 [Br94], Q5 [CK98], Q6 [Cu88], Q7 [Da04a], Q8 [Da05], Q9 [DZ03], Q10 [Da04b], Q11 [ED05], Q12 [Gü96], Q13 [HL01], Q14 [Ka02], Q15 [LV01], Q16 [Lu93], Q17 [Mo98], Q18 [NK91], Q19 [Su99], Q20 [Ta95], Q21 [Wa93].

¹ Folgende Zeitschriften wurden berücksichtigt: Communications of the ACM, Communications of the AIS, Empirical Software Engineering, European Journal of Information Systems, IBM Systems Journal, IEEE Software, IEEE Transactions on Software Engineering, Information Systems Journal, Information Systems Research, Journal of Management Information Systems, Journal of the Association for Information Systems, Management Science, MIS Quarterly, Requirements Engineering, Software Quality Journal, Wirtschaftsinformatik.

² Folgende Konferenzen wurden berücksichtigt: IEEE International Conference on Requirements Engineering, IEEE International Conference on Software Maintenance, IEEE International Symposium on Requirements Engineering, International Conference on Information Systems, International Conference on Software Engineering, International Symposium on Empirical Software Engineering, International Symposium on Software Metrics, International Symposium on Software Reliability Engineering.

³ Web of Science ist unter der URL <http://scientific.thomson.com/products/wos/> zu finden (Stand 26.07.2006). Es ist ein kostenpflichtiges Produkt der Thomson Scientific.

⁴ Die CiteSeer Scientific Literature Digital Library ist unter der URL <http://citeseer.ist.psu.edu/> zu finden (Stand 26.07.2006).

Ein konstituierendes Merkmal einer systematischen Literaturanalyse ist, dass für den festgelegten Ausschnitt der Literatur eine Synthese relevanter Aussagen angestrebt wird. Im ersten Schritt wurden hierzu die identifizierten Literaturquellen nach Abschluss der Literaturrecherche mit dem Ziel ausgewertet, Beobachtungen zu Ursache-Wirkungszusammenhängen zu extrahieren. Um eine Synthese dieser Beobachtungen vorzubereiten, wurden schrittweise mit jeder gefundene Beobachtung zum einen Typen von Anforderungsfehlern und zum anderen Ursachen von Anforderungsfehlern inhaltlich gruppiert.

4 Erklärungsmuster für Anforderungsfehler

In der Literaturanalyse wurden 21 empirische Literaturquellen ausgewertet. Diese Quellen tragen zu insgesamt 92 relevanten Beobachtungen bei. Eine Beobachtung ist dabei ein Ursache-Wirkungszusammenhang mit folgender Struktur: ein Prozessmangel der Anforderungsanalyse (Ursache) hat einen bestimmten Typ von Anforderungsfehlern oder generell Anforderungsfehler zur Folge (Wirkung). Sofern eine Ursache verschiedene Typen von Anforderungsfehlern bewirkt, wird für jeden Typ eine einzelne Beobachtung erfasst.

4.1 Typologie von Anforderungsfehlern

Ein Anforderungsfehler ist ein unzureichendes Merkmal oder ein erwartetes, jedoch fehlendes Merkmal eines Arbeitsergebnisses der Anforderungsanalyse, sofern es eine Änderung in diesem Ergebnis notwendig macht. Aus der Auswertung von 92 Beobachtungen zu Ursache-Wirkungszusammenhängen kann eine Typologie von Anforderungsfehlern hergeleitet werden. Die Abbildung 2 stellt diese Typologie von Anforderungsfehlern anhand eines hierarchischen Baumdiagramms dar.

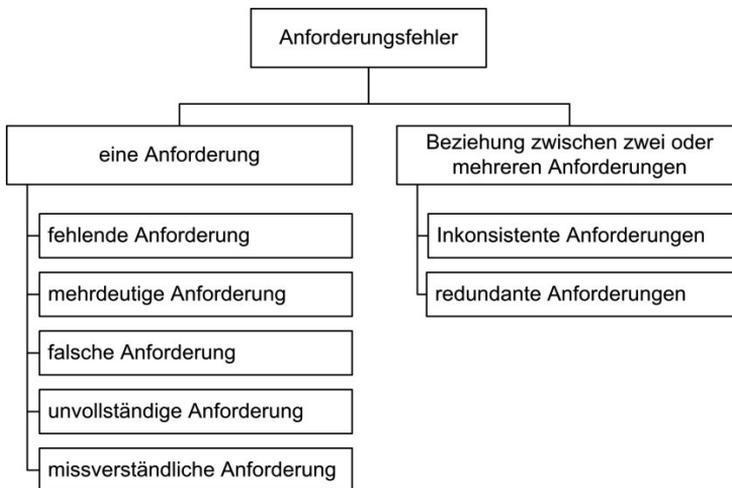


Abbildung 2: Typologie von Anforderungsfehlern

Wenn sich ein Anforderungsfehler auf eine Anforderung bezieht, können insgesamt fünf Fälle voneinander abgegrenzt werden:

- Für ein bestehendes Kundenbedürfnis wurde eine Anforderung gänzlich übersehen (fehlende Anforderung).
- Die Beschreibung der Anforderung kann auf mehrere Arten gedeutet werden (mehrdeutige Anforderung).
- Die Anforderung erfüllt kein Kundenbedürfnis oder erfüllt sie nicht korrekt (falsche Anforderung).
- Die Anforderung ist unvollständig beschrieben (unvollständige Anforderung).
- Die Anforderung ist zwar vollständig beschrieben, jedoch fehlen Kontextinformationen zu dieser Anforderung oder Verknüpfungen zu vorhandenen Kontextinformationen, um die Anforderung zu verstehen (missverständliche Anforderung).⁵

Ein Anforderungsfehler kann sich auch auf eine Beziehung zwischen zwei oder mehreren Anforderungen beziehen. Zwei Fälle können dabei unterschieden werden:

- Eine Anforderung steht im Widerspruch zu einer oder mehreren anderen Anforderungen (inkonsistente Anforderungen).
- Eine Anforderung ist redundant zu mindestens einer anderen Anforderung (redundante Anforderungen).

In 39 Beobachtungen (42 %) werden als Wirkung Anforderungsfehler allgemein beschrieben ohne einen Typ zu konkretisieren. Folglich ist der Typ des Anforderungsfehlers für 53 Beobachtungen ermittelbar. Darunter sind missverständliche (16 %), inkonsistente (12 %) und fehlende Anforderungen (11 %) am häufigsten vertreten.

4.2 Ursachen für Anforderungsfehler

Warum entstehen Anforderungsfehler? Dies ist die Ausgangsfrage des vorliegenden Beitrags. Nachfolgend werden Erklärungsmuster für einzelne Typen von Anforderungsfehlern formuliert. Diese Erklärungsmuster basieren auf den extrahierten Beobachtungen der systematischen Literaturanalyse. Ein Erklärungsmuster beschreibt jeweils, welche Ursache einen bestimmten Typ von Anforderungsfehler verursachen kann.

⁵ Kontextinformationen sind z. B. Kundenbedürfnisse, für die Anforderung relevante Kundengruppen oder andere abhängige Anforderungen.

Insgesamt 20 Ursachen können nach der Auswertung der 92 Beobachtungen unterschieden werden. Diese Ursachen können zum einen direkt Prozessmängel der Anforderungsanalyse im Sinne dieser Arbeit sein. Zum anderen können es Rahmenbedingungen der Softwareentwicklung sein, die Anforderungsfehler verursachen. Diese Rahmenbedingungen können nicht geändert werden. Vielmehr muss die Gestaltung eines Prozesses der Anforderungsanalyse diese Rahmenbedingungen angemessen berücksichtigen, um ihre negativen Wirkungen zu reduzieren. Werden relevante Rahmenbedingungen in einem Prozess der Anforderungsanalyse nicht berücksichtigt, stellt dies ebenfalls einen Prozessmangel der Anforderungsanalyse dar. Insofern beschreiben Beobachtungen, in denen Rahmenbedingungen als Ursachen für Anforderungsfehler genannt werden, indirekt Prozessmängel der Anforderungsanalyse.

Die Ursachen für fehlende Anforderungen liegen insbesondere in der Dokumentation von Anforderungen, der Ausgestaltung der Benutzerbeteiligung, dem fehlendem Wissen über das Anwendungsgebiet der Software sowie einer unzureichenden Auseinandersetzung mit Kundenbedürfnissen (vgl. Abbildung 3).

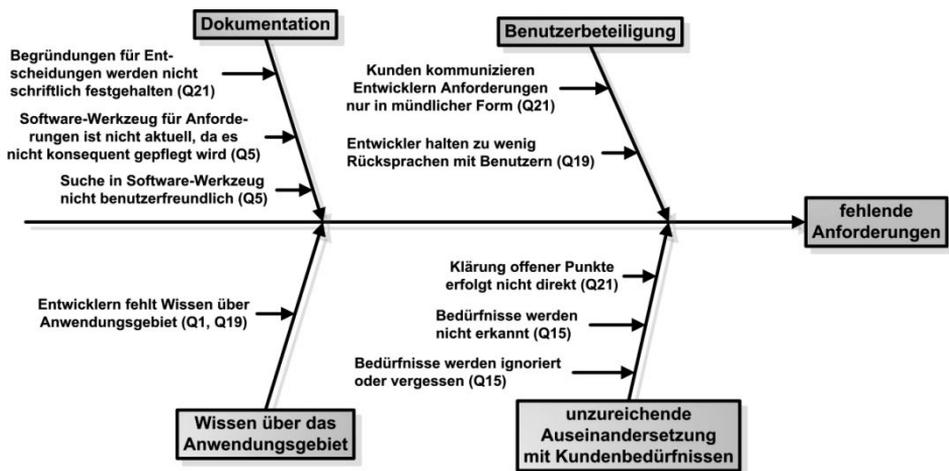


Abbildung 3: Ursachen für fehlende Anforderungen

Die Ursachen für mehrdeutige Anforderungen liegen insbesondere in fehlendem Wissen über das Anwendungsgebiet der Software, der späten Klärung offener Punkte und der Gestaltung des Inhalts des Anforderungsdokuments (vgl. Abbildung 4).

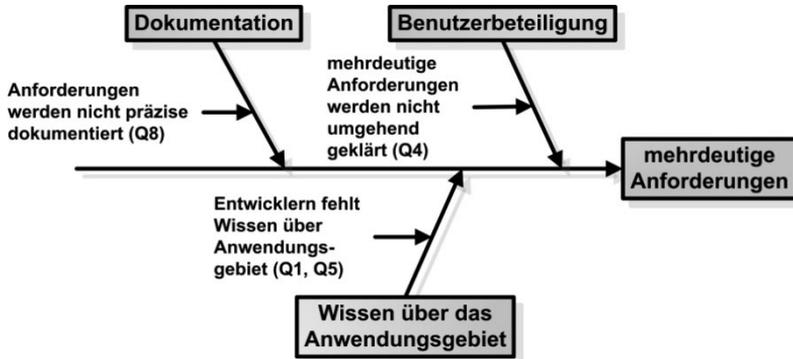


Abbildung 4: Ursachen für mehrdeutige Anforderungen

Die Ursachen für falsche Anforderungen liegen insbesondere in dem Gestaltungsbereichen der Dokumentation von Anforderungen, Wissensmonopolen bei wenigen Entwicklern, einer unzureichenden Auseinandersetzung mit Kundenbedürfnissen und der Vielzahl und Heterogenität der Kunden (vgl. Abbildung 5).

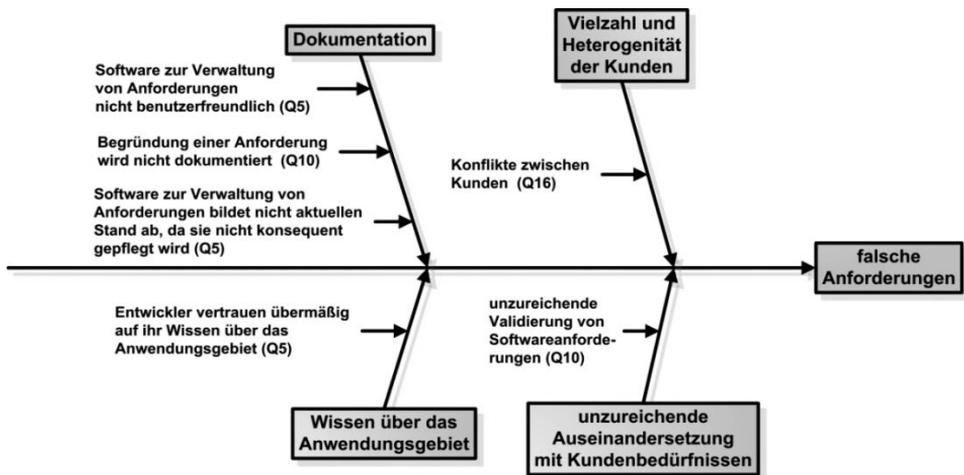


Abbildung 5: Ursachen für falsche Anforderungen

Die Ursachen für unvollständige Anforderungen liegen in der Art der Erhebung von Anforderungen, einem fehlenden Wissen über das Anwendungsgebiet sowie im Projektmanagement und der Gestaltung der Benutzerbeteiligung (vgl. Abbildung 6).

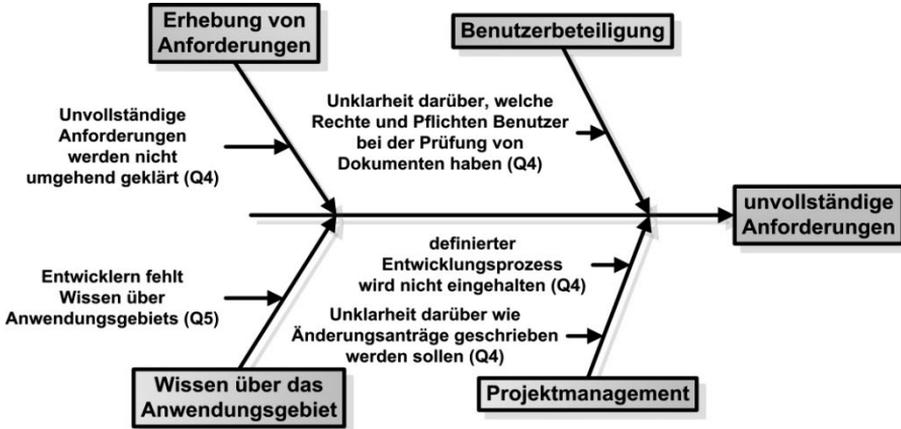


Abbildung 6: Ursachen für unvollständige Anforderungen

Die Ursachen für missverständliche Anforderungen liegen insbesondere in den Gestaltungsbereichen der Dokumentation von Anforderungen (hier insbesondere die Verfolgbarkeit) und der Benutzerbeteiligung. Als weitere Ursachen sind die Kommunikation im Entwicklungsteam, Form und Inhalte der Anforderungsdokumente, Projektmanagement sowie eingesetzte Software-Werkzeuge aufzuführen (vgl. Abbildung 7).

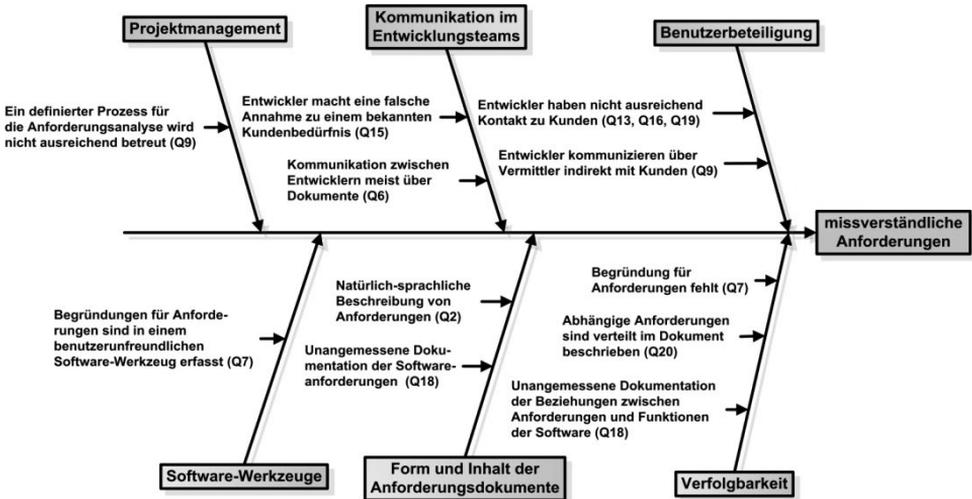


Abbildung 7: Ursachen für missverständliche Anforderungen

Die Ursachen für inkonsistente Anforderungen liegen in der Dokumentation von Anforderungen, der inhärenten Komplexität der Anforderungsanalyse, der Kommunikation im Entwicklungsteam und fehlendem Wissen über das Anwendungsgebiet (vgl. Abbildung 8).

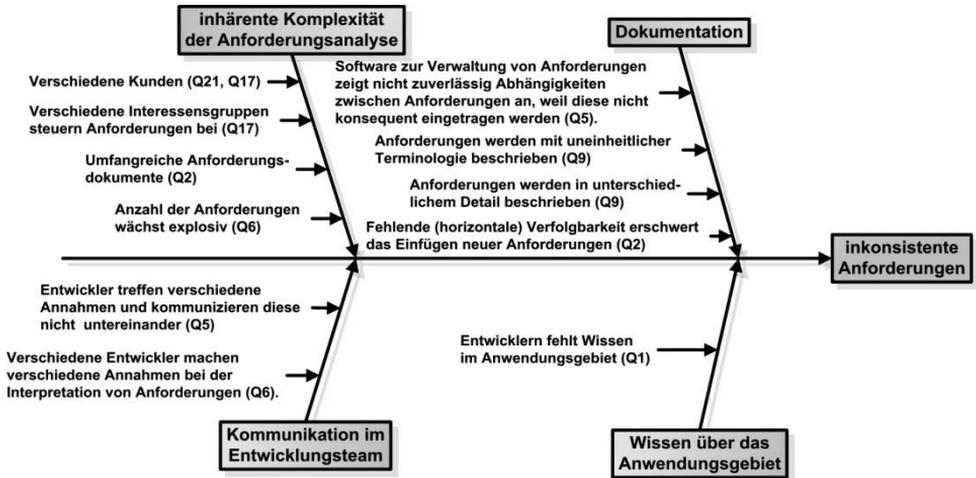


Abbildung 8: Ursachen für inkonsistente Anforderungen

Die Ursachen für redundante Anforderungen liegen in der Dokumentation von Anforderungen. Hier werden im Wesentlichen zwei konkrete Ursachen aufgeführt:

- Anforderungen werden in unterschiedlichem Detailgrad beschrieben.
- Anforderungen werden mit einer uneinheitlichen Terminologie beschrieben.

5 Diskussion der Ergebnisse

Missverständliche, inkonsistente und fehlende Anforderungen machen 67 % der Anforderungsfehler aus, deren Typ bekannt ist (53 von 92, ca. 58 %). Gemäß der systematischen Literaturanalyse tragen die Ursachenkategorien Dokumentation von Anforderungen, Gestaltung der Benutzerbeteiligung, Wissen über das Anwendungsgebiet und Erhebung von Anforderungen zu ca. 73 % der Fehler bei.

Nachfolgend werden zentrale Ergebnisse dieser vier Ursachenkategorien herausgestellt und mit relevanten empirischen Befunden in der Literatur verglichen.

5.1 Dokumentation von Anforderungen

Die identifizierten Ursachen für Anforderungsfehler innerhalb der Dokumentation von Anforderungen beziehen sich im Einzelnen auf die Form und den Inhalt von Anforderungsdokumenten, auf die eingesetzten Software-Werkzeuge und die Verfolgbarkeit von Anforderungen. Prozessmängel in diesem Bereich verursachen insbesondere missverständliche, inkonsistente, fehlende und falsche Anforderungen.

Bei der Gestaltung der Form von Anforderungsdokumenten geht es um die Frage, mit welchen Darstellungsmitteln Softwareanforderungen beschrieben werden sollen. Innerhalb dieses Gestaltungsbereichs ist insbesondere die Frage zu beantworten, ob Anforderungen natürlich-sprachlich oder semi-formal mit Methoden wie der UML beschrieben werden sollen. Einerseits kann eine natürlich-sprachliche Beschreibung gemäß der Literaturanalyse zu Anforderungsfehlern führen. Dies legt nahe, semi-formale Beschreibungsmethoden einzusetzen. Chatzoglou zeigt in einer Untersuchung, dass in Projekten mit Einsatz von Methoden der Aufwand, die Kosten und der Zeitbedarf für die gesamte Entwicklung geringer ausfallen als in Projekten ohne Einsatz von Methoden [Ch97].

Andererseits können jedoch Kunden Schwierigkeiten haben, semi-formale Beschreibungen von Anforderungen zu verstehen. Dieser Sachverhalt wird ebenfalls in weiteren empirischen Untersuchungen festgestellt [EM95], [DS99]. Dem Spannungsfeld zwischen natürlich-sprachlicher und semi-formaler Beschreibung begegnen Entwicklungsteams,

- indem sie semi-formale Beschreibungen von Anforderungen zusätzlich und redundant natürlich-sprachlich für Kunden erläutern,
- indem sie informale Beschreibungen von Anforderungen (überwiegend natürlich-sprachlich) für Kunden und davon getrennt für die Arbeit innerhalb des Entwicklungsteams zusätzlich semi-formale Beschreibungen für Anforderungen erstellen oder
- indem sie sich auf die natürlich-sprachliche Beschreibung beschränken.

Neben der Form von Anforderungsdokumenten gibt es mehrere Beobachtungen zu deren Inhalt. Werden Anforderungen aus technischer Perspektive beschrieben, kann dies zu Anforderungsfehlern führen. Die gleiche Wirkung haben Unklarheiten, in welchem Detailgrad die Anforderungen beschrieben werden sollen. Diese Beobachtungen zum Inhalt von Anforderungsdokumenten stehen im Einklang mit Ergebnissen einer Untersuchung von El Emam und Madhavji [EM95].

Wird ein Software-Werkzeug eingesetzt, um Anforderungen zu dokumentieren oder zu verwalten, kann sein Einsatz zu Anforderungsfehlern führen, sofern es Unzulänglichkeiten wie eine geringe Benutzbarkeit aufweist oder die Nutzung nicht geplant, gesteuert und überwacht wird.

Dies sind einige der Gründe, warum Entwicklungsteams entsprechende Werkzeuge eher als hinderlich wahrnehmen [HL01]. Empirische Befunde zeigen, dass der Einsatz von entsprechenden Software-Werkzeugen wie z. B. CASE-Werkzeugen nur dann effektiv ist, wenn zumindest ein Prozess der Anforderungsanalyse definiert ist, die Entwickler in dem Werkzeug geschult werden und die Unterstützung der Nutzung organisatorisch institutionalisiert wird [EM95].

Insgesamt 10 Beobachtungen der Literaturanalyse beziehen sich auf die Verfolgbarkeit von Anforderungen. Mit 6 Beobachtungen ist die vertikale Vorverfolgbarkeit am häufigsten vertreten. Sie bezieht sich auf Verknüpfungen zwischen einer Anforderung und abhängigen Einzelergebnissen vorgelagerter Teilaufgaben wie z. B. Kundenbedürfnissen. Gotel und Finkelstein kommen in einer Untersuchung zur Verfolgbarkeit von Anforderungen ebenfalls zum Ergebnis, dass Praktiker eine unzureichende vertikale Vorverfolgbarkeit von Anforderungen am häufigsten als Problem nennen [GF94].

5.2 Gestaltung der Benutzerbeteiligung

Die identifizierten Ursachen in der Benutzerbeteiligung sind vor allem eine unzureichende oder zu späte Benutzerbeteiligung oder eine Beteiligung von ungeeigneten Personen. Prozessmängel in der Benutzerbeteiligung verursachen insbesondere missverständliche Anforderungen.

Empirische Befunde einer Untersuchung von Weltz und Ortman zeigen ebenfalls auf, dass die Weichen für die Benutzerbeteiligung häufig falsch gestellt werden [WO92]. Gemäß ihrer Untersuchung spielen sowohl Entwickler als auch Benutzer in der Formulierung von Anforderungen häufig nur eine marginale Rolle. Nach ihrer Ansicht wird ein Benutzervertreter eher deswegen bestimmt, um die tatsächlich unzulängliche Einbindung der Benutzer zu verdecken. Vergleichbare Beobachtungen machen auch El Emam und Madhavji [EM95]. Sie bemerken, dass gerade jene Benutzer, deren Beteiligung am sinnvollsten für ein Projekt ist, selten für entsprechende Aufgaben zur Verfügung stehen.

Weitere Beobachtungen aus der systematischen Literaturanalyse beschreiben Schwierigkeiten bei der interdisziplinären Zusammenarbeit zwischen Entwicklern und Benutzern sowie eine mangelnde Konsequenz bei der Umsetzung der Benutzerbeteiligung. In einer 1994 veröffentlichten Untersuchung zur Benutzerbeteiligung fasst Heinbokel zusammen, dass die „Benutzerbeteiligung in der gegenwärtig praktizierten Form weder zu ersichtlich besserer Software führt noch für den Entwicklungsprozess selbst förderlich ist, sondern im Gegenteil zunächst ein Hemmschuh für das Projekt ist“ [He94]. Heinbokel kommt zu dieser Schlussfolgerung, weil Projekte mit einer Benutzerbeteiligung beim Gesamterfolg, bei der Termin- und Kosteneinhaltung sowie bei der Teameffektivität schlechter abschneiden als Projekte ohne Benutzerbeteiligung.

5.3 Wissen über das Anwendungsgebiet

Insgesamt 12 Beobachtungen weisen darauf hin, dass die Entstehung von Anforderungsfehlern begünstigt wird, wenn dem Entwicklungsteam Wissen über das Anwendungsgebiet fehlt. Prozessmängel in diesem Bereich verursachen gemäß der Literaturanalyse abgesehen von redundanten Anforderungen alle Typen von Anforderungsfehlern.

Tiwana kommt auf der Grundlage einer Untersuchung von 232 Entwicklungsprojekten in verschiedenen Organisationen zum Ergebnis, dass die Integration von technischem Wissen und Wissen über das Anwendungsgebiet der Software unter anderem die Fehlerhäufigkeit in den einzelnen Teilaufgaben der Softwareentwicklung reduziert [Ti04].

5.4 Erhebung von Anforderungen

Die Erhebung von Anforderungen verursacht gemäß der Literaturanalyse 13 % der Anforderungsfehler. Insbesondere treten unklare Ziele der Software und die unzureichende Auseinandersetzung mit Kundenbedürfnissen als Ursachen auf. Prozessmängel in der Erhebung von Anforderungen verursachen insbesondere fehlende Anforderungen.

Eine Fallstudie von Cusumano und Selby bei Microsoft verdeutlicht, wie wichtig eine klare Vision und klare Ziele für die Entwicklung einer Software sind [CS97]. Microsoft nutzt eine Visionsaussage und eine vorläufige Spezifikation als Projektleitfaden. Statt von Anfang an eine detaillierte Spezifikation zu schreiben, beginnt Microsoft die Projekte mit einer ehrgeizigen Visionsaussage und einer vorläufigen Spezifikation. Die Visionsaussage wird von den Produktplanern der Marketinggruppe und den Programm-Managern verfasst und listet eine Reihe von Zielen auf. Sie enthält keine detaillierten Anforderungen, sondern wird vielmehr als ein Ausgangspunkt genutzt, um systematisch Anforderungen herzuleiten.

Fünf Beobachtungen beschreiben eine unzureichende Auseinandersetzung mit Kundenbedürfnissen als Ursachen für Anforderungsfehler. Dieser Umstand rührt unter anderem daher, dass Entwicklungsteams Methoden zur systematischen Erhebung von Anforderungen wie z. B. Quality Function Deployment (QFD) nicht kennen oder nicht einsetzen [Ha96]. Vielmehr existiert häufig eine unberechtigte Erwartung, dass Methoden zur Beschreibung von Anforderungen zugleich auch zu ihrer Erhebung dienen [Lu93]. Mehrere Untersuchungen zeigen, dass erfolgreiche Teams konsequent Methoden zur Erhebung von Anforderungen wie z. B. QFD einsetzen [Ha96], [HL01].

6 Zusammenfassung und Fazit

Der vorliegende Beitrag beschreibt kausale Zusammenhänge zwischen Prozessmängeln in der Anforderungsanalyse und Typen von Anforderungsfehlern. Diese Erklärungsmuster sind Ergebnis einer Synthese empirischer Befunde. Sie bieten bereits praktische Gestaltungshilfe, da sie ausgehend von Anforderungsfehlern eine Ursache-Wirkungsanalyse unterstützen und z. B. in Fehleranalyseverfahren integriert werden können [Av08]. Mit der Kenntnis der Prozessmängel in der Anforderungsanalyse eröffnet sich die Möglichkeit, diese zu beseitigen oder, sofern dies nicht möglich ist, die Prozesse der Qualitätssicherung zu verbessern, um gezielt Fehler der Prozessmängel früher zu finden.

Literaturverzeichnis

- [AE96] Al-Rawas, A., Easterbrook, S.: Communication problems in requirements engineering. A field study. In: Myers, C.; Hall, T.; Pitt, D. (Hrsg.): Proceedings of the First Westminster Conference on Professional Awareness in Software Engineering (PASE 1996), 1-2 February 1996, London (England). London (England) 1996, S. 47-60.
- [AS85] Adelson, B., Soloway, E.: The role of domain experience in software design. In: IEEE Transactions on Software Engineering. Nr. 11, Jg. 11, 1985, S. 1351-1360.

- [Av08] Avci, O.: Aus Fehlern in der Softwareentwicklung lernen. Anforderungsanalyse und Qualitätssicherung mit Fehleranalysen verbessern. Saarbrücken 2008. VDM Verlag Dr. Müller, ISBN-10: 3836476878, ISBN-13: 9783836476874.
- [AW05] Avci, O.; Wagner, H.: Das chronische Problem der Anforderungsanalyse und die Frage: Fehler vermeiden oder früh entdecken? In: Cremers, A.B.; Manthey, R.; Martini, P.; Steinhage, V. (eds.): Informatik 2005 - Informatik LIVE, Band 2, Lecture Notes in Informatics (LNI), Vol. P-68. Bonn 2005, S. 279-283.
- [Bh94] Bhandari, I. et al.: In-process improvement through defect data interpretation. In: IBM Systems Journal. Nr. 1, Jg. 33, 1994, S. 182-214.
- [BP84] Basili, V.; Perricone, B.T.: Software errors and complexity: an empirical investigation. In: Communications of the ACM. Nr. 1, Jg. 27, 1984, S. 42-52.
- [Br94] Briand, L.C. et al.: A change analysis process to characterize software maintenance projects. In: Müller, H.A.; Georges, M. (Hrsg.): Proceedings of the International Conference on Software Maintenance (ICSM 1994), Victoria, BC, Canada, September 1994. Washington, DC, USA 1994, S. 38-49.
- [Ch96] Chillarege, R.: Orthogonal Defect Classification. In: Lyu, M.R. (Hrsg.): Handbook of software reliability engineering. Los Alamitos, California u. a. 1996, S. 359-400.
- [Ch97] Chatzoglou, P.D.: Use of methodologies: an empirical analysis of their impact on the economics of the development process. In: European Journal of Information Systems. Nr. 4, Jg. 6, 1997, S. 256-270.
- [CK98] Crowston, K.; Kammerer, E.E.: Coordination and collective mind in software requirements development. In: IBM Systems Journal. Nr. 2, Jg. 37, 1998, S. 227-246.
- [CS97] Cusumano, M.A.; Selby, R.W.: Die Microsoft-Methode. Sieben Prinzipien, wie man ein Unternehmen an die Weltspitze bringt. München 1997.
- [Cu88] Curtis, B. et al.: A field study of the software design process for large systems. In: Communications of the ACM. Nr. 11, Jg. 31, 1988, S. 1268-1287.
- [Da05] Damian, D. et al.: Requirements engineering and downstream software development: findings from a case study. In: Empirical Software Engineering. Nr. 3, Jg. 10, 2005, S. 255-283.
- [Da04a] Damian, D. et al.: An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software. In: Empirical Software Engineering. Nr. 1-2, Jg. 9, 2004, S. 45-75.
- [Da04b] Daneva, M.: ERP requirements engineering practice: lessons learned. In: IEEE Software. Nr. 2, Jg. 21, 2004, S. 26-33.
- [DS99] Dawson, L.; Swatman, P.: The use of object-oriented models in requirements engineering: a field study. In: ACM (Hrsg.): Proceeding of the 20th international conference on information systems (ICIS '99), December 12 - 15, 1999, Charlotte (North Carolina, USA). Atlanta (Georgia, USA) 1999, S. 260-273.
- [DZ03] Damian, D.E.; Zowghi, D.: Requirements engineering challenges in multi-site software development organisations. In: Requirements Engineering. Nr. 3, Jg. 8, 2003, S. 149-160.
- [ED05] Ebert, C.; De Man, J.: Requirements uncertainty. Influencing factors and concrete improvements. In: ACM (Hrsg.): Proceedings of the 27th International Conference on Software Engineering (ICSE 2005), May 15 - 21, 2005, St. Louis (Missouri, USA). New York (New York, USA) 2005, S. 553-560.
- [EM95] El Emam, K.; Madhavji, N.Z.: A field study of requirements engineering practices in information systems development. In: IEEE (Hrsg.): Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE '95), March 27 - 29, 1995, York (England). Los Alamitos (California, USA) 1995, S. 68-80.

- [GF94] Gotel, O.C.Z.; Finkelstein, A.C.W.: An analysis of the requirements traceability Problem. In: IEEE (Hrsg.): Proceedings of the First International Conference On Requirements Engineering (ICRE 1994), April 18-22, 1994, Colorado Springs, Colorado. Los Alamitos (USA) 1994, S. 94-101.
- [Gü96] Günther, H. et al.: Kontinuierliche Qualitätsverbesserung in der Software-Entwicklung. Erfahrungen bei der Allianz Lebensversicherungs-AG. In: Wirtschaftsinformatik. Nr. 2, Jg. 38, 1996, S. 160-171.
- [Ha96] Haag, S. et al.: Quality Function Deployment usage in software development. In: Communications of the ACM. Nr. 1, Jg. 39, 1996, S. 41-49.
- [He94] Heinbokel, T.: Benutzerbeteiligung: Schlüssel zum Erfolg oder Hemmschuh der Entwicklung? In: Brodbeck, F.C., Frese, M. (Hrsg.): Produktivität und Qualität in Softwareprojekten. Psychologische Analyse und Optimierung von Arbeitsprozessen in der Softwareentwicklung. München, Oldenbourg 1994, S. 105-124.
- [HL01] Hofmann, H.F.; Lehner, F.: Requirements Engineering as a success factor in software projects. In: IEEE Software. Nr. 4, Jg. 18, 2001, S. 58-66.
- [Ie90] IEEE Standard Glossary of Software Engineering Terminology. New York 1990.
- [Ka02] Kauppinen, M. et al.: Introducing requirements engineering: how to make a cultural change happen in practice. In: IEEE (Hrsg.): Proceedings of 10th Anniversary IEEE Joint International Conference on Requirements Engineering (ICRE 2002), September 9-13 2002, Essen (Germany).). Los Alamitos (California, USA) 2002, S. 43-51.
- [Lu93] Lubars, M. et al.: A review of the state of the practice in requirements modeling. In: IEEE (Hrsg.): Proceedings of IEEE International Symposium on Requirements Engineering (ISRE 1993), January 4-6, 1993, San Diego (California, USA). Los Alamitos (California, USA) 1993, S. 2-14.
- [LV01] Lauesen, S.; Vinter, O.: Preventing requirement defects: An experiment in process improvement. In: Requirements Engineering. Nr. 1, Jg. 6, 2001, S. 37-50.
- [Mo98] Morris, P. et al.: Requirements Engineering and Industrial Uptake. In: IEEE (Hrsg.): Proceedings of the Third IEEE International Conference on Requirements Engineering (ICRE '98), April 06 - 10, 1998, Colorado Springs, Colorado. Los Alamitos (California, USA) 1998, S. 130-137.
- [NK91] Nakajo, T.; Kume, H.: A case history analysis of software error cause-effect relationships. In: IEEE Transactions on Software Engineering. Nr. 8, Jg. 17, 1991, S. 830-838.
- [Su99] Sutcliffe, A.G. et al.: Tracing requirements errors to problems in the requirements engineering process. In: Requirements Engineering. Nr. 3, Jg. 4, 1999, S. 134-151.
- [Ta95] Takahashi, K. et al.: A comparative study of structured and text-oriented analysis and design methodologies. In: Journal of Systems and Software. Nr. 1, Jg. 28, 1995, S. 69-75.
- [Ti04] Tiwana, A.: An empirical study of the effect of knowledge integration on software development performance. In: Information and Software Technology. Nr. 13, Jg. 46, 2004, S. 899-906.
- [Wa93] Walz, D.B. et al.: Inside a software design team: knowledge acquisition, sharing, and integration. In: Communications of the ACM. Nr. 10, Jg. 36, 1993, S. 63-77.
- [WO92] Weltz, F.; Ortmann, R. G.: Das Softwareprojekt. Projektmanagement in der Praxis. Frankfurt 1992.
- [WW02] Webster, J.; Watson, R.T.: Analyzing the past to prepare for the future: writing a literature review. In: MIS Quarterly. Nr. 2, Jg. 26, 2002, S. xiii-xxiii.