

Eine Analyse von Kommunikationslücken in Softwareprojekten in Bezug auf Requirements Engineering

Rita Forster¹, Norbert Seyff¹, Peter Spörri², Claudia Reuter², Martin Glinz¹

¹Universität Zürich, Zürich,

{seyff, glinz}@ifi.uzh.ch

²Zühlke Engineering AG, Schlieren,

{peter.spoerri, claudia.reuter}@zuehlke.com

Motivation der behandelten Fragestellung

Kommunikation zwischen der Disziplin Requirements Engineering und anderen Software Disziplinen wie Entwicklung und Qualitätssicherung ist von wesentlicher Bedeutung für erfolgreiche Softwareprojekte. Ein Mangel an Kommunikation, oder das Auftreten von sogenannten Kommunikationslücken ist oft schädlich für den Projekterfolg [1].

Behandelte Fragestellung oder Problem

Während Kommunikationslücken bereits initial untersucht und definiert wurden [1], fehlt es an empirischen Daten, die aufzeigen, wie verbreitet Kommunikationslücken (z.B., sozio-kulturelle, zeitliche oder geographische) in realen Softwareprojekten sind und welche Auswirkungen sie tatsächlich haben. Dabei ist es auch wichtig, die Zusammenhänge zwischen Projekteigenschaften (z.B. agile Vorgehensweise vs. Wasserfall) und möglichen Kommunikationslücken zu erkennen.

Lösung und Ergebnisse

Die vorliegenden Forschungsergebnisse wurden im Rahmen einer Zusammenarbeit zwischen der Universität Zürich und der Zühlke Engineering AG erstellt. Dabei wurden 12 reale Softwareprojekte in Bezug auf aufgetretene Kommunikationslücken und deren Auswirkungen anhand der gegebenen Projekteigenschaften untersucht. Die Ergebnisse basieren auf Daten, welche durch Interviews mit Projektmitarbeitern gesammelt wurden. Zur Identifikation von Kommunikationslücken wurden mehrere Perspektiven berücksichtigt und Interviews mit Projektleitern, Requirements Engineers, Entwicklern und Testern durchgeführt.

In den untersuchten Projekten wurde überwiegend eine iterative Vorgehensweise zur Entwicklung verwendet (10 von 12 Projekten). Zwei Projekte wurden unter Verwendung von Scrum realisiert. Das längste Projekt dauerte 108 Monate, während das kürzeste nach 5.5 Monaten abgeschlossen war. Die Anzahl der Mitarbeiter variierte zwischen 5 und 30 Personen. Acht der untersuchten Projekte wurden in einem verteilten Umfeld durchgeführt, wo sich Projektmitarbeiter an verschiedenen Standorten

befanden. Ebenso waren acht der untersuchten Projekte zum Untersuchungszeitpunkt noch nicht abgeschlossen, aber bereits in einem fortgeschrittenen Stadium, was die Erhebung sinnvoller Daten ermöglichte. Bei den abgeschlossenen vier Projekten lag das Fertigstellungsdatum in der nahen Vergangenheit, so dass die Mitarbeiter ihre Auskünfte in Rückblick auf das Projekt geben konnten.

Im Folgenden stellen wir 14 Erkenntnisse vor, welche wir durch die Analyse der Erhebungsergebnisse eindeutig identifizieren konnten. Die ersten vorgestellten Erkenntnisse beziehen sich auf die Analyse von zeitlichen Kommunikationslücken.

Erkenntnis 1: In Projekten mit kurzen Release Zyklen sind die Artefakte, die der Requirements Engineer zur Verfügung stellt, auf dem neuesten Stand.

Dieses Ergebnis begründet sich auch darauf, dass in diesen Projekten oftmals Anforderungen für das geplante Release konkretisiert und überarbeitet wurden und somit auf dem neuesten Stand waren. In Projekten mit längeren Release Zyklen (über 8 Monate), konnten wir dieses Updateverhalten (leider) nicht feststellen.

Erkenntnis 2: In Projekten, die nach Scrum durchgeführt werden, bekommt der Requirements Engineer von den Entwicklern Artefakte, welche auf dem neuesten Stand sind.

Hier scheint das agile Vorgehen die kontinuierliche Überarbeitung von Artefakten zu fördern. Dies kommt dem Requirements Engineer zu Gute.

Erkenntnis 3: Eine größere Anzahl von RE Verantwortlichen führt dazu, dass die Kommunikation zwischen Requirements Engineers und Entwicklern verlangsamt wird.

Es scheint, dass in Projekten mit mehreren RE Verantwortlichen der Kommunikationsfluss unter dem Koordinationsbedarf der Requirements Engineers leidet.

Die nächsten Erkenntnisse beziehen sich auf die Analyse von psychologischen Kommunikationslücken.

Erkenntnis 4: In Projekten mit kürzeren Release Zyklen schätzen die Entwickler die Kommunikationsfähigkeiten des Requirements Engineers höher ein.

Wir sehen einen Zusammenhang mit Erkenntnis 1 und vermuten, dass wiederum die objektiv messbare bessere Qualität der Artefakte auch zu dieser subjektiv

besseren Einschätzung der Kommunikationsfähigkeiten geführt hat.

Erkenntnis 5: In Projekten mit einer höheren Anzahl von Releases ist die Offenheit des Requirements Engineers für Zusammenarbeit laut der Einschätzung der Entwickler höher.

Die höhere Anzahl von Releases bedeutet oft auch eine intensivere Zusammenarbeit zwischen den Entwicklern und den Requirements Engineers. Damit lässt sich dieses Resultat erklären.

Erkenntnis 6: Projekte, in denen alle Entwickler in die Iterationsplanung einbezogen werden (z.B. in Sprint Planungsbesprechungen), werden von den Requirements Engineers in Bezug auf die Effizienz der Kommunikation besser beurteilt.

Persönliche Treffen zum Informationsaustausch sind generell besonders effizient.

Erkenntnis 7: In Projekten, in denen Requirements Engineers an täglichen Planungsmeetings teilnehmen, werden die Kommunikationsfähigkeiten der Requirements Engineers von den Entwicklern besser eingestuft.

Tägliche Kommunikation führt zu einem besseren Verständnis von Problemen und fördert laut diesem Ergebnis die Kommunikation.

Folgende Erkenntnisse beziehen sich auf kognitive Kommunikationslücken.

Erkenntnis 8: Umso mehr Personen in ein Projekt involviert sind, umso größer ist der Unterschied zwischen dem Domain-, Projekt- und Organisationswissen zwischen Requirements Engineers und Testern.

Der generelle Trend, dass mit der Anzahl der Projektmitarbeiter auch die Diversifikation zwischen ihnen steigt, ist nachvollziehbar. Unsere Ergebnisse zeigen diesen Unterschied aber speziell zwischen den Rollen Requirements Engineer und Tester.

Erkenntnis 9: In Projekten, in denen alle Entwickler an Treffen zur Iterationsplanung (z.B. Sprint Planungsbesprechungen) teilnehmen, ist der Unterschied zwischen dem generellen Wissen über das Projekt in Bezug auf alle Projektmitarbeiter am geringsten.

Dieses Ergebnis hängt auch mit Erkenntnis 6 zusammen. Es zeigt sich, dass durch die Anwesenheit und die Erklärungen der anwesenden Entwickler alle Projektmitarbeiter auf dem neuesten Stand gehalten werden.

Erkenntnis 10: In Projekten, in denen die Mehrheit der Projektmitarbeiter bei demselben Unternehmen angestellt ist, kommunizieren Requirements Engineers und Entwickler am häufigsten.

Die Vermutung liegt nahe, dass dies so ist und das Ergebnis überrascht nicht.

Folgende Erkenntnisse beziehen sich auf sozio-kulturelle Kommunikationslücken.

Erkenntnis 11: In Projekten mit größeren sozio-kulturellen Unterschieden wird die Bereitschaft der

Requirements Engineers zur Zusammenarbeit von den Entwicklern niedriger eingestuft.

Dieses Ergebnis lässt sich wahrscheinlich durch menschliche Verhaltensweisen erklären. Wir konnten keine konkreten Gründe identifizieren.

Erkenntnis 12: Je größer die sozio-kulturellen Unterschiede sind, desto größer sind auch die Unterschiede in Bezug auf Domain-, Technik- und Organisationswissen zwischen Requirements Engineers und Testern.

Wie bei Erkenntnis 11 können wir hier nur auf den Faktor Mensch verweisen. Genauere Anhaltspunkte oder eine mögliche Erklärung haben wir nicht gefunden.

Im Folgenden beschreiben wir zwei Erkenntnisse, welche keinen Bezug zu den primär untersuchten Kommunikationslücken haben.

Erkenntnis 13: Je höher die Anzahl der Releases, desto unbeständiger sind die Anforderungen und umso höher ist die Wahrscheinlichkeit, dass es in dem Projekt zu Zeitverzögerungen kommt.

Dieses Ergebnis scheint nachvollziehbar zu sein, da es sich wahrscheinlich auch um Projekte handelt, wo das Ziel am Anfang des Projekts noch nicht genau festgelegt wurde.

Erkenntnis 14: Je länger Iterationen (z.B. Sprints) dauern, desto anfälliger werden Projekte in Bezug auf die Überschreitung der vorgegebenen Kosten oder der geplanten Projektlaufzeit.

Kürzere Iterationen sind laut Auskunft der Projektmitarbeiter leichter zu planen und daher sind Abweichungen weniger oft der Fall.

Was ist neu?

Untersuchungen in Bezug auf die Analyse und Auswirkungen von Kommunikationslücken sind in der Literatur kaum zu finden. Bisher gibt es dazu Arbeiten von Elizabeth Bjarnason, die sich bei ihren Untersuchungen aber verstärkt auf ein Projekt in einem Unternehmen fokussierte [1]. Die vorgestellte Studie liefert nun auch wissenschaftliche Belege zu den Auswirkungen des agilen Vorgehens auf die Kommunikation in Softwareentwicklungsprojekten, zu denen bislang nur Vermutungen angestellt werden konnten.

Quellen

- [1] Elizabeth Bjarnason, Krzysztof Wnuk, Björn Regnell: Requirements are slipping through the gaps - A case study on causes & effects of communication gaps in large-scale software development. RE 2011: 37-46