Eine Analyse von Kommunikationslücken in Softwareprojekten in Bezug auf Requirements Engineering

Rita Forster, Norbert Seyff, Peter Spörri,
Claudia Reuter, Martin Glinz

University of Zurich[UZH]
Department of Informatics

zühlke
empowering ideas

zeitlich
psychologisch
kognitiv
sozio-kulturell

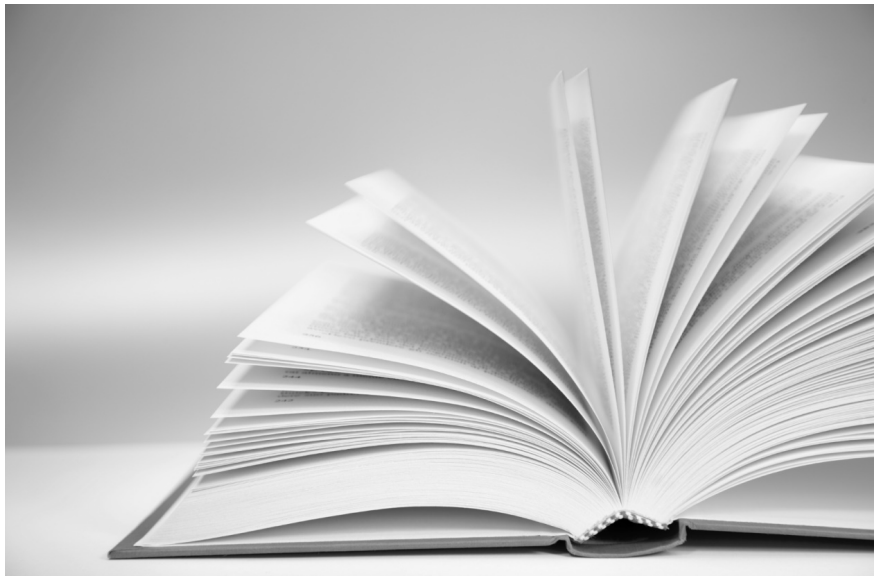
Zeitverzögerungen
Kostenerhöhungen

Research Goal

Analyse von Kommunikationslücken in Bezug auf Projektcharakteristiken

Vorgehen

Elizabeth Bjarnason



Semi-strukturierte Interviews

Umfeld

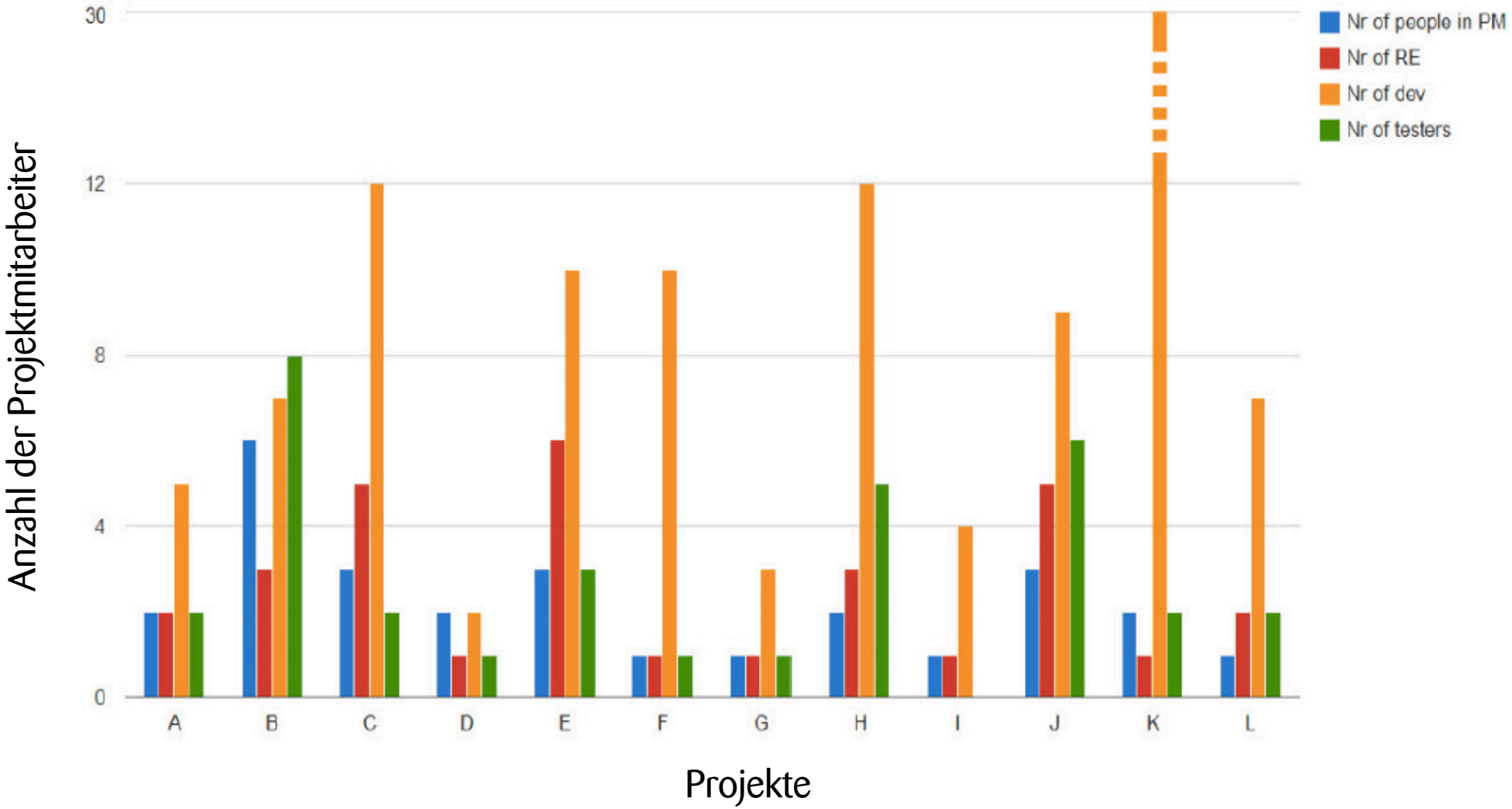
zühlke
empowering ideas

12 Projekte

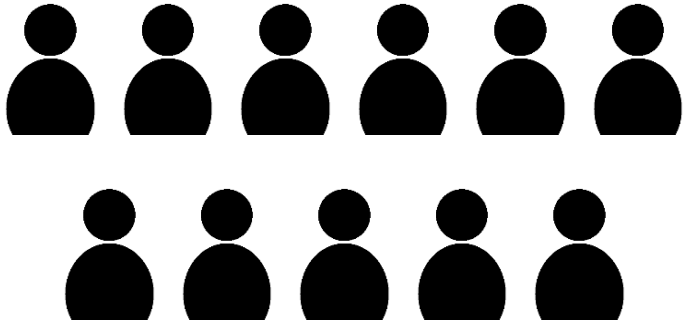
43 Teilnehmer

Project	Domain	Process model	Distributed / collocated	Duration *	Nr of people	Finished / ongoing
A	Public administration	Iterative	D	9	11	F
B	Public administration	Iterative	D	60	25	O
C	Entertainment and leisure	Iterative	C	60	20	O
D	Public administration	Iterative	D	36	5	O
E	Banking	Iterative	C	23	30	O
F	Manufacturing	Agile	C	36	10	O
G	Construction	Iterative	C	24	5	O
H	Telecommunication	Iterative	D	108	22	O
I	Mechanical engineering	Agile	D	5.5	7	F
J	Insurance	Iterative	D	24	30	O
K	Telecommunication	Iterative	D	14	23	F
L	Telecommunication	Iterative	D	9	20	F
* measured in months						

People


Anzahl der Projektmitarbeiter
30
12
8
4
0
A
B
C
D
E
F
G
H
I
J
K
L
Projekte
Nr of people in PM
Nr of RE
Nr of dev
Nr of testers

Tester



REs

Erste Ergebnisse

Ergebnis 1

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

Annahme

- kurzen Release Zyklen ➔ Anforderungen wurden für das geplante Release konkretisiert und überarbeitet (waren auf dem neuesten Stand)
- längeren Release Zyklen (über 8 Monate) ➔ kein Updateverhalten feststellbar

Ergebnis 2

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

Annahme

- Agiles Vorgehen fördert die kontinuierliche Überarbeitung von Artefakten.
- Der Requirements Engineer profitiert von den gemachten Überarbeitungen.

Ergebnis 3

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

Annahme

- Der Kommunikationsfluss leidet unter dem Koordinationsbedarf der Requirements Engineers.

Ergebnis 4

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

Annahme

- Zusammenhang mit Ergebnis 1
- Die bessere Qualität der Artefakte führt auch zu dieser subjektiv besseren Einschätzung der Kommunikationsfähigkeiten

Ergebnis 5

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

Annahme

- höhere Anzahl von Releases = intensivere Zusammenarbeit zwischen den Entwicklern und den Requirements Engineers

Ergebnis 6

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

Annahme

- Regelmäßige, persönliche Treffen zur Kommunikation von Information zeigen sich als effiziente Weise Informationen auszutauschen und führen zu dieser subjektiv besseren Einschätzung.

Ergebnis 7

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

Annahme

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

Ergebnis 8

Umso mehr Leute 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.

Ergebnis 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.

Annahme

- Anwesenheit und die Erklärungen der anwesenden Entwickler halten alle Projektmitarbeiter auf dem neuesten Stand.
- Dieses Ergebnis hängt auch mit Ergebnis 6 zusammen.

Ergebnis 10

In Projekten, in denen die Mehrheit der Projektmitarbeiter beim selben Unternehmen angestellt ist, kommunizieren Requirements Engineers und Entwickler am häufigsten.

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

Ergebnis 11

In Projekten mit größeren soziokulturellen Unterschieden wird die Bereitschaft der Requirements Engineers zur Zusammenarbeit von den Entwicklern niedriger eingestuft.

Annahme

- Typisch menschliche Verhaltensweise?

Ergebnis 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.

Annahme:

- Möglicher Zusammenhang mit Ergebnis 11
- Weniger Kommunikation führt zu Wissensunterschieden

Ergebnis 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.

Annahme

- Das Ziel konnte am Anfang dieser Projekts nicht genau festgelegt werden.

Ergebnis 14

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

Annahme

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

Weitere Schritte
Andere Umbebungen
Erweiterte Analyse
Mehr Projekte

Zusammenfassung und Diskussion