

**Analyse der Zusammenarbeit in Softwareprojekten
mittels Informationsflüssen und Interaktionen in Meetings**

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

Doktorin der Naturwissenschaften

(abgekürzt: Dr. rer. nat.)

genehmigte Dissertation von

Jil Ann-Christin Klünder

geboren am 29.06.1994 in Gehrden

2019

1

Einleitung

Durch die wachsende Komplexität der zu entwickelnden Softwaresysteme und durch die Anforderungen des Marktes erfordern die meisten Softwareprojekte Teamarbeit [36, 87, 134]. Das Mitwirken verschiedener Personen an einem Projekt erfordert zusätzliche Koordination und eine angemessene Form der Zusammenarbeit [87]. Andernfalls kann es zu Schwierigkeiten innerhalb des Entwicklerteams kommen [30, 49]. Neben einer angemessenen Zusammenarbeit ist auch die Weitergabe von Wissen und Informationen für den Erfolg eines Softwareprojekts relevant und innerhalb des Entwicklerteams unabdingbar [139, 164]. Verloren gegangene Informationen oder falsch weitergegebene Anforderungen können den Erfolg des Projekts gefährden [153]. Um dem vorzubeugen, sind ein adäquater Informationsfluss und ausreichende Kommunikation während des gesamten Entwicklungsprozesses wichtig [47, 54, 108]. Dabei spielt jedoch nicht nur eine Rolle, ob kommuniziert wird, sondern auch auf welchem Weg [7, 82]. Schriftliche Kommunikation hat beispielsweise den Vorteil, dass sie wiederholt abgerufen werden kann. Allerdings setzt der Informationsfluss auf schriftlichem Wege, wie beispielsweise in E-Mails, auch voraus, dass der Empfänger die E-Mail liest, versteht und im Anschluss richtig interpretiert oder – falls vonnöten – Nachfragen stellt. Eine lediglich zur Kenntnis genommene E-Mail kann den gesamten Prozess gefährden, geht der Sender doch davon aus, dass die Informationen vom Empfänger verarbeitet wurden.

Bei der direkten face-to-face Kommunikation in Meetings oder Gesprächen kann diesen Schwierigkeiten leichter entgegen gewirkt werden. Face-to-face Kommunikation ermöglicht einen bidirektionalen Austausch in Echtzeit und gilt als effektive und reichhaltige Möglichkeit, um Informationen weiterzugeben [7, 131]. Aus diesem Grund nehmen Meetings im Entwicklungsprozess einen immer höheren Stellenwert ein [149]. Meetings ermöglichen die Weitergabe von vielen Informationen an mehrere Teammitglieder in einer vergleichsweise geringen Zeit. Zudem können die richtige Verarbeitung und Interpretation der Informationen sicher-

gestellt werden, indem noch während des Meetings Unklarheiten aufgelöst werden [69, 139]. Dennoch sind Meetings oft nicht so erfolgreich und produktiv, wie sie sein könnten, und die Teilnehmenden sind am Ende frustriert und unzufrieden [69]. Dies ist nicht nur auf die immer größer werdende Anzahl von Meetings im Entwicklungsprozess und die Länge der Meetings zurückzuführen [25, 102, 144], sondern hängt auch mit dem Verhalten der Teilnehmenden während des Meetings zusammen [70, 149]. Studien haben gezeigt, dass die Qualität eines Meetings deutlich steigt, wenn der Sprechende (auch non-verbales) Feedback erhält und weiß, dass ihm die anderen zuhören [149]. Unterstützende Aussagen sorgen ebenfalls für zufriedenerere Teilnehmende am Ende des Meetings [149]. Jammerzirkel oder Schuldzuweisungen – wenn auch subtil – mindern hingegen die Qualität der Ergebnisse eines Meetings [69, 70].

Aus diesem Grund sind die Analyse und Beobachtung des Informationsflusses und des Kommunikationsverhaltens im Software Engineering weit verbreitet [153]. Meetinganalysen hingegen sind momentan noch nicht sehr etabliert, gewinnen aber zunehmend an Bedeutung. In anderen Wissenschaften, wie beispielsweise in der Arbeits-, Organisations- und Sozialpsychologie, gehören Interaktionsanalysen in Meetings zum Stand der Technik (siehe z.B. [70, 116]). Analysen des Informationsflusses können frühzeitig Schwachstellen erkennen und Verbesserungen aufzeigen, um falsch kommunizierten Kundenanforderungen vorzubeugen [162]. Darüber hinaus wirken sie verloren gegangenen Informationen entgegen und erhöhen so die Chancen für einen erfolgreichen Projektabschluss und einen zufriedenstellenden Weg dorthin [82, 163]. Ein Großteil der Informationsweitergabe findet in Meetings statt [83]. Dabei geht es jedoch nicht nur darum, dass Informationen fließen, sondern wie diese kommuniziert werden [69]. Aus diesem Grund empfiehlt sich für Meetings eine detailliertere Analyse auf Interaktionsebene und nicht (ausschließlich) basierend auf den Informationsflüssen.

Meetinganalysen können Schwierigkeiten in der Zusammenarbeit während eines Meetings aufzeigen und den Teilnehmenden – zumeist Projektleiter, Kunden und Entwickler – aufzeigen, welchen Einfluss ihr Verhalten auf das Projekt hat [69, 139, 149]. Die Selbstreflektion von Entwicklern ist vielerorts kein Bestandteil der Ausbildung eines Softwareentwicklers oder Informatikers, sodass das Bewusstsein für Folgen von vermeintlich banalen Handlungen und Äußerungen während des Meetings bei vielen Entwicklern nicht vorhanden ist [149]. Darüber hinaus sind sich die Meetingteilnehmenden oft nicht bewusst, wenn das Meeting kein Ergebnis hat oder keine konkreten Handlungen nach sich zieht [139].

1.1 Problemstellung

Beobachtungen und Analysen des Informationsflusses und der Interaktionen in Meetings bieten eine Möglichkeit, um einigen der zuvor genannten Problemen entgegenzuwirken. Zur Informationsflussanalyse in Softwareentwicklerteams hat sich die FLOW-Methode etabliert, die

am Fachgebiet Software Engineering an der Leibniz Universität Hannover entwickelt wurde [162]. Diese beinhaltet eine manuelle Erhebung von qualitativen Daten im Rahmen von Interviews und die Auswertung von den Ergebnissen von einem geschulten FLOW-Analysten [72, 83]. Die Analyse ist zumeist zeitintensiv und die Qualität der Ergebnisse, das heißt die Vollständigkeit von identifizierten Auffälligkeiten und Schwachstellen, hängt von den Erfahrungen des FLOW-Analysten ab [79]. *Durch die qualitative Analyse der Informationsflüsse sind die Ergebnisse subjektiv und für Außenstehende teilweise nicht ohne Weiteres nachvollziehbar.*

Für die Analyse von Interaktionen in Meetings hat sich im Software Engineering bislang noch kein Verfahren etabliert. Ein in der Arbeits-, Organisations- und Sozial-Psychologie etabliertes Verfahren zur Interaktionsanalyse ist act4teams. Das Kodierschema wurde am Lehrstuhl für Arbeits-, Organisations- und Sozialpsychologie an der Technischen Universität Braunschweig entwickelt und seitdem in verschiedenen Meetings eingesetzt (siehe z.B. [69, 116]). Die Anwendung von act4teams setzt eine Videoaufzeichnung für die anschließende feingranulare Analyse voraus [116, 148]. Dies ist in der Praxis aus verschiedenen Gründen oft nicht praktikabel [139]. Zu den Gründen zählen Datenschutzbestimmungen und Geheimhaltungsverpflichtungen der Unternehmen. Zudem bedarf die Analyse für objektive Ergebnisse ungefähr das zehnfache der Dauer des Meetings und eine ungefähr 200 Stunden dauernde Schulung [116, 149]. *In Entwicklerteams gestaltet sich die Anwendung von act4teams folglich als schwierig, der aus den Ergebnissen generierte Nutzen motiviert jedoch die Anwendung.*

1.2 Zielsetzung und Forschungsmethodik

Die vorliegende Arbeit hat zum Ziel, die zuvor beschriebenen Probleme zu analysieren und zu reduzieren.

1.2.1 Ziele der Arbeit

Im Rahmen dieser Arbeit werden zwei Konzepte vorgestellt, die die Informationsflussanalyse mit der FLOW-Methode unterstützen sowie leichtgewichtige Meetinganalysen ermöglichen.

NET-FLOW ist ein Konzept, das die Auswertung der Ergebnisse der Informationsflussanalyse mit der FLOW-Methode unterstützt. Die mit der FLOW-Methode erhobenen Informationsflüsse werden in Form eines FLOW-Diagramms visualisiert, das auch als Graph oder (soziales) Netzwerk aufgefasst werden kann. Durch die Transformation eines FLOW-Diagramms in ein FLOW-Netzwerk ist es möglich, etablierte Methoden aus der sozialen Netzwerkanalyse bei der Analyse der FLOW-Netzwerke zu berücksichtigen, um Schwachstellen im Informationsflussnetzwerk objektiv zu identifizieren. Für Meetinganalysen eignet sich dieses Konzept jedoch nur eingeschränkt, weshalb zusätzlich act4teams-SHORT für leichtgewichtige Interaktionsanalysen in Meetings vorgestellt wird.

2

Grundlagen

Die vorliegende Arbeit beschäftigt sich mit der Kommunikation in Entwicklerteams, insbesondere in Meetings. Dabei werden sowohl der Informationsfluss als auch die Interaktionen in Meetings analysiert. Als Grundlage dienen die FLOW-Methode zur Visualisierung und Analyse von Informationsflüssen in Entwicklerteams sowie act4teams für Interaktionsanalysen in Meetings. Die für das Verständnis dieser Arbeit benötigten Informationen werden in diesem Kapitel bereitgestellt.

2.1 Kommunikation in Entwicklerteams

Aufgrund der wachsenden Komplexität und Größe der zu entwickelnden Systeme erfordern die meisten Softwareprojekte Teamarbeit [36,87,134]. Die Zusammenarbeit der einzelnen Teammitglieder erfordert Interaktionen und Kommunikation [82]. Interaktionen sind immer dann vonnöten, wenn gemeinsam auf ein Ziel hingearbeitet wird. Kommunikation wird benötigt, um alle Mitglieder des Entwicklerteams und den Kunden sowie weitere am Projekt beteiligte Personen zu synchronisieren. Diese Synchronisation ist für einen erfolgreichen Projektabschluss zwingend erforderlich, um redundante Arbeiten zu vermeiden und eine gute Integrität der einzelnen Arbeitspakete zu gewährleisten.

Betrachtet man den Informationsfluss als Grund für Kommunikation, ist es nicht nur wichtig, dass kommuniziert wird, sondern auch über welches Medium [125]. Bei Terminabsprachen mit einzelnen Personen ist es beispielsweise oftmals angebrachter, direkte Kanäle wie das Telefon zu nutzen, während kompliziertere Sachverhalte wie Inkonsistenzen im Quellcode in persönlichen Gesprächen oder Meetings oft einfacher zu klären sind [126].

Um Informationen bestmöglich transportieren zu können, greifen die Entwicklerteams auf unterschiedliche Medien zurück [7,126]. Dabei hängt die primäre Wahl der Medien von vielen

Faktoren ab: dem Entwicklungsprozess [11,75], unternehmensrechtlichen Bestimmungen [126] und der räumlichen Verteilung eines Teams [124,140]. So haben verteilte Teams beispielsweise oft Schwierigkeiten, Meetings abzuhalten [112]. Mitunter ist sogar die Verwendung von Video-Chats oder Anrufen durch unterschiedliche Zeitzonen eingeschränkt [126].

Die üblicherweise von Entwicklerteams genutzten Medien können in Meetings, Anrufe, Video-Anrufe, Instant-Messenger und E-Mails unterteilt werden [125,126]. Häufig werden dabei Anwendungen genutzt, die mehrere Kanäle in sich vereinen. Skype™ bietet beispielsweise die Funktionalitäten eines Instant Messengers, für Telefonate und für Video-Anrufe. Darüber hinaus ermöglicht es den Einsatz von Bildschirmübertragung, was die virtuelle Zusammenarbeit erleichtert und zudem – durch die Videokonferenz – eine Meeting-ähnliche Atmosphäre schafft. Zudem sind viele Anwendungen wie Skype™ oder Slack sowohl als Desktopanwendung als auch als App verfügbar, wodurch die einzelnen Teammitglieder zum Teil auch erreichbar sind, wenn sie nicht an ihrem Arbeitsplatz anwesend sind. Beobachtungen aus den Einzelfallstudien (vgl. Kapitel 5.6 und Kapitel 6.5 sowie Anhang D) zeigen, dass für den informellen, privaten Austausch auch WhatsApp weit verbreitet ist. Aus unternehmensrechtlichen Bestimmungen ist die Verwendung jedoch oft nur sehr eingeschränkt oder gar nicht möglich [126].

2.2 Meetings in der Softwareentwicklung

Ein wichtiger Kanal für die Kommunikation sind Meetings. Diese nehmen in der Softwareentwicklung einen großen Stellenwert ein [83,139,149]. Dies ist vor allem darauf zurückzuführen, dass es in Meetings möglich ist, in kurzer Zeit neue Informationen mit vielen Teammitgliedern zu teilen [139].

Meetings haben unterschiedliche Ziele: Sie dienen der Planung, Statusmeldungen, dem Aufdecken von Verbesserungspotenzial, der Klärung von Anforderungen oder Abstimmungen [149]. Vor allem das Scrum-Framework¹, welches in der agilen Softwareentwicklung weit verbreitet ist [76,98], wird von Meetings dominiert. Hier stellen Meetings unter anderem einen regelmäßigen teaminternen Austausch sicher. Ein Sprint startet mit einem Planungsmeeting, dem *Planning*, in dem der Product Owner den Sprint mit dem Team abstimmt. Während des Sprints gibt es tägliche Statusberichte, die *Daily-Scrums*, in denen jedes Teammitglied innerhalb von zwei Minuten sagt, was es seit dem letzten Daily abgearbeitet hat, was es bis zum nächsten Daily plant, und ob Schwierigkeiten aufgetreten oder zu erwarten sind. Am Ende des Sprints gibt es eine *Review*, eine Präsentation der Ergebnisse für den Kunden. Vor dem nächsten Sprint findet außerdem eine *Retrospektive* statt, in welcher der letzte Sprint reflektiert wird und nach Verbesserungen für den kommenden Sprint gesucht wird. Während des Sprints ist das

¹Für nähere Informationen zur agilen Softwareentwicklung und Scrum sei auf [9] und [154] verwiesen.

Team außerdem verantwortlich, in einem oder mehreren *Refinements* neu geschriebene oder geänderte Story Cards nach ihrem Aufwand zu schätzen, damit dieser als Entscheidungsbasis für den Kunden bei der Priorisierung dienen kann.

Bei mehreren Scrum-Teams, die übergreifend an einem Produkt arbeiten, kann außerdem ein tägliches *Scrum-of-Scrums* helfen, Informationen in kurzer Zeit mit allen Teams zu teilen. Dies findet im Idealfall vor den Daily-Scrums auf Teamebene statt. Bei diesem Meeting treffen sich je ein oder zwei Vertreter pro Team, um wichtige teamübergreifende Informationen auszutauschen.

Für den Wissensaustausch gibt es *Communities of Practice*, bei denen verschiedene Experten mit ähnlichen Interessen und Fachrichtungen gemeinsam an Lösungen arbeiten oder sich neues Wissen aneignen.

Auch außerhalb des Scrum-Frameworks gibt es viele Meetings. Der traditionelle Entwicklungsprozess wird ebenso von diversen Meetings dominiert [149]. Während des Prozesses des Requirements Engineering finden beispielsweise Workshops zur Anforderungserhebung, -klärung, -verfeinerung und zur -verhandlung statt [72]. Meistens haben diese Workshops einen Meeting-ähnlichen Charakter. Aber auch außerhalb des Requirements Engineering Prozesses finden Meetings statt. Diese dienen der team-internen Synchronisation, der Ideenfindung oder der Problemlösung.

2.3 Informationsflussanalyse mit der FLOW-Methode

Ein hinreichender Informationsfluss spielt nicht nur in den Meetings von Entwicklerteams eine übergeordnete Rolle [153]. Verloren gegangene oder unzureichend übermittelte Informationen können nicht selten den Erfolg eines Projekts gefährden [162]. Aus diesem Grund ist es Bestandteil der Forschung im Software Engineering, Informationsflüsse zu visualisieren und zu analysieren, um frühzeitig Schwachstellen aufzudecken [163]. Die vorliegende Arbeit basiert auf der Informationsflussanalyse mit FLOW, die am Fachgebiet Software Engineering der Leibniz Universität Hannover entwickelt und seitdem in mehreren Fallstudien eingesetzt wurde (siehe z.B. [72, 81, 165]).

Die FLOW-Methode wurde entwickelt, um Informationsflüsse in Entwicklerteams zu erheben, zu analysieren und zu verbessern [162]. Dabei werden folgende Schritte durchlaufen:

1. Die Datenerhebung in FLOW-Interviews (siehe Kapitel 2.3.3),
2. die Visualisierung der Informationsflüsse in FLOW-Diagrammen (siehe Kapitel 2.3.4) und
3. die Auswertung des FLOW-Diagramms in der FLOW-Analyse (siehe Kapitel 2.3.5)