

# Soziale Aspekte hybrider und traditioneller Softwareentwicklung

Jil Klünder<sup>1</sup>, Lisa Handke<sup>2</sup>, Torsten Gfesser<sup>2</sup>, Kurt Schneider<sup>1</sup> und Simone Kauffeld<sup>2</sup>

<sup>1</sup> Fachgebiet Software Engineering, Leibniz Universität Hannover, Deutschland  
{jil.kluender, kurt.schneider}@inf.uni-hannover.de

<sup>2</sup> Lehrstuhl für Arbeits-, Organisations- und Sozialpsychologie,  
Technische Universität Braunschweig, Deutschland  
{l.handke, t.gfesser, simone.kauffeld}@tu-braunschweig.de

## Zusammenfassung

Agile Softwareentwicklung gewinnt zunehmend an Bedeutung. In den letzten beiden Jahrzehnten wurde der Anteil agil entwickelnder Firmen immer größer. Die Entscheidung, ob in einem Team plangetrieben oder agil entwickelt wird, wirkt sich auf die gesamte Organisation des Projekts und damit auf alle Beteiligten aus.

Dadurch wird es auch für Studierende immer wichtiger, in Lehrveranstaltungen nicht nur Einblicke in traditionelle, dokumentenzentrierte Vorgehensmodelle zu gewinnen, sondern auch Erfahrungen mit agilen Prozessen zu sammeln.

In Softwareprojekten an Hochschulen sollen Studierende vor allem Soft Skills wie Team- und Kommunikationsfähigkeit erlernen oder ausbauen. Dabei beeinflusst die Art, wie Software entwickelt wird, die Lernerfahrungen der Studierenden maßgeblich. Um diese Einflüsse genauer zu klassifizieren, untersuchen wir hybride und plangetriebene studentische Softwareprojekte in Bezug auf die Intensität der Kommunikation im Team und das Auftreten von Konflikten.

Unsere Erkenntnisse erlauben Rückschlüsse auf den Einfluss hybrider Entwicklungsmethoden auf soziale Aspekte und spiegeln demnach wieder, ob der Einsatz hybrider Methoden in der Lehre an Hochschulen dazu führt, dass Studierende diese Fähigkeiten ausbauen, oder ob sich das Verhalten im Vergleich zu plangetriebenen Projekten nicht verändert.

## Einleitung

Agile Softwareentwicklung gewinnt zunehmend an Bedeutung. Um eine gute Vorbereitung von Studierenden auf die Wirtschaft zu gewährleisten, ist es deshalb erforderlich, in der universitären Lehre agile Softwareprojekte zu integrieren. Der Einsatz

verschiedener Arten von Arbeitsorganisation (dokumentenzentriert oder agil) ändert jedoch nicht nur die Organisation des Projekts, sondern auch den Ablauf und damit den Lernerfolg der Studierenden.

Die Auswahl der verwendeten Arbeitsorganisation hat einen Einfluss auf die Kommunikation [1]. Der Projekterfolg hängt wiederum zu 20 % von der Kommunikation des Teams ab [2].

Bei dokumentenzentrierten Vorgehensweisen ist der Softwarelebenszyklus [3] mit seinen Phasen maßgeblich. Diese unterteilen sich in Anforderungsanalyse, Entwurf, Implementierung, Überprüfung und Wartung. Der Ablauf hat sich seit 1956 zunehmend etabliert, führte jedoch nicht immer zu einem erfolgreichen Abschluss des Projekts. Aus diesem Grund rückte die agile Softwareentwicklung seit 1990 zunehmend in den Fokus und sollte deshalb auch in der universitären Lehre berücksichtigt werden.

In universitären Softwareprojekten sollen Studierende ihre sozialen Fähigkeiten ausbauen und zudem ihre Programmierfähigkeiten erweitern. Aus diesem Grund ist es wichtig, dass der Projekterfolg nicht unter einer *Agilisierung* der Lehrveranstaltung leidet.

Wir untersuchen traditionelle und hybride studentische Softwareprojekte hinsichtlich des Kommunikationsverhaltens und dem Auftreten von Konflikten. Die Projekte stimmen in wesentlichen Projektparametern wie Dauer, Aufwand und Komplexität überein, unterscheiden sich aber in der Art der Arbeitsorganisation.

Die Struktur des Papiers ist wie folgt: Im folgenden Abschnitt präsentieren wir verwandte Arbeiten. Danach werden die unterschiedlichen Arbeitsorganisationen vorgestellt. Im vierten Abschnitt wird das methodische Vorgehen mit der Stichprobe,

dem Matching, der verwendeten Maße und dem Vorgehen bei der statistischen Analyse beschrieben. Die Ergebnisse werden im darauf folgenden Abschnitt festgehalten, bevor das Fazit gezogen wird. Die Zusammenfassung unserer Ergebnisse beendet das Papier.

## Verwandte Arbeiten

Der Einfluss von der Arbeitsorganisation auf Kommunikationsverhalten und Konflikte ist nicht unbekannt und war bereits Gegenstand vorheriger Forschungen.

Bindrees et al. [1] untersuchten die Kommunikation in verschiedenen Projektphasen. Sie konnten nachweisen, dass in den Projektphasen unterschiedlich intensiv kommuniziert wird. Auf diesen Ergebnissen aufbauend, möchten wir die Aussage konkretisieren und auf unterschiedliche Arbeitsorganisationen beziehen.

In der agilen Entwicklung fordern Beck und Andres [4], dass die Kommunikation zwischen den Stakeholdern maximiert werden müsse. Wie die Kommunikation in agilen Teams tatsächlich abläuft, wurde bislang jedoch noch nicht näher untersucht und ist unter anderem Bestandteil unserer Arbeit.

Brügge et al. [5] analysierten studentische Softwareprojekte. Sie präsentierten eine Vorgehensweise, die die Fähigkeiten der Studierenden verbessern und zu in der Wirtschaft verwendbaren Ergebnissen führen soll. Um schnelles Feedback und kurzzeitige Releases zu ermöglichen sowie Kommunikation zu fördern, verwendeten sie die Tools Rugby und Tornado. In dem Papier beschreiben sie ihre Erfahrungen mit dem Einsatz der Tools in studentischen Lehrveranstaltungen. Die Tools unterstützten die Studenten hinsichtlich ihrer technischen und nicht-technischen Fähigkeiten wie bspw. Kommunikation und Teamwork.

Schneider et al. [6] analysierten studentische traditionell entwickelnde Teams hinsichtlich ihrer Stimmung, des Konfliktpotenzials und des Kommunikationsverhaltens. Sie fanden heraus, dass der Anteil der sozialen Konflikte über die Projektlaufzeit ansteigt, während aufgabenbezogene Konflikte zunächst zunahm und schließlich wieder abfiel. Wir nutzen diese Ergebnisse für Vergleiche zwischen agiler und traditioneller Softwareentwicklung.

## Arbeitsorganisation

Dieses Kapitel dient als eine kurze Einführung in die hier betrachteten Arbeitsorganisationen: die traditionelle, die agile und die hybride Softwareentwicklung.

## Traditionelle Softwareentwicklung

In der traditionellen Softwareentwicklung durchläuft das Projekt sequenziell die Phasen des Software-Development-Cycles [7]. Um mit der nächsten Phase beginnen zu können, muss die vorherige Phase vollständig abgeschlossen sein. Diese Vorgehensweise wurde als *Nine-Phase-Stage-Wise-Modell* bereits 1956 von Bennington [8] vorgestellt und 1970 von Royce [9] weiterentwickelt. Royce [9] ordnete die Phasen in seinen grafischen Darstellungen als Kaskade bzw. Wasserfall an, woraus sich die Bezeichnung *Wasserfall-Modell* ergibt. Dabei handelt es sich um das bekannteste traditionelle Entwicklungsmodell.

Der Ablauf im Wasserfall-Modell ist sequenziell. Das Projekt startet mit einer *Anforderungsanalyse*, während der die Anforderungen mit dem Kunden herausgearbeitet und schließlich in einer Anforderungsspezifikation mit einem Lasten- und Pflichtenheft festgehalten werden. Nach dieser Phase folgen der *Entwurf*, die *Implementierung* und schließlich die *Überprüfung* durch den Kunden.

Diese Arbeitsorganisation wird auch heute noch vielfach genutzt. Sofern es sich um Projekte mit stabilen Anforderungen handelt, bei denen Kosten und Umfang genau abgeschätzt werden können, stellt der sequenzielle Ablauf keine großen Schwierigkeiten dar. Da dies jedoch oftmals nicht möglich ist, gewinnt die agile Softwareentwicklung zunehmend an Bedeutung.

## Agile Softwareentwicklung

Basierend auf sich ständig ändernden Kundenwünschen, dem Wunsch nach zuverlässigerer Kundenzufriedenheit und Projekterfolg formulierten Beck et al. [10] zwölf Prinzipien und agile Werte, die im *Agilen Manifest* [10] festgehalten wurden. Der Grundgedanke agiler Softwareentwicklung ist ein iteratives Vorgehen, wie es im Spiralmodell nach Boehm dargestellt wird [11]. Das Projekt durchläuft mehrfach die Phasen *Anforderungsanalyse*, *Entwurf*, *Implementierung* und *Überprüfung* in Teilschritten, um sich dem gewünschten Endzustand des Kunden immer mehr anzunähern [10].

Kundenzufriedenheit hat höchste Priorität und wird durch die regelmäßige Präsentation von funktionsfähigen und ausführbaren Inkrementen sichergestellt [10]. Dies wird durch iterative kurze Entwicklungszyklen von wenigen Wochen bis wenigen Monaten erreicht.

Seit Beginn der 1990er Jahre etablierten sich verschiedene Vorgehensweisen und Formate. Eine der bekanntesten ist *Scrum*. Scrum wurde für die Softwareentwicklung erstmals 2002 ausformuliert [12]. Es wird zwischen drei Rollen unterschieden: *Product-Owner*, Entwicklungsteam und *Scrum Mas-*

ter. Der Product-Owner ist der Projektverantwortliche, vergleichbar mit einem Kunden, der über das weitere Vorgehen und die Priorität der Anforderungen entscheidet. Er steht stets für Nachfragen des Teams zur Verfügung und ist möglichst oft vor Ort (*On-Site-Customer*).

Wesentlicher Bestandteil von Scrum sind die *Daily Scrums*, in denen jeder Entwickler innerhalb von zwei Minuten über den Fortschritt seit dem letzten Daily, über Pläne bis zum nächsten Daily und über eventuell auftretende Probleme berichtet. Neben dem Daily gibt es noch weitere Meetings, die zum Teil mit und teilweise ohne Kunden abgehalten werden. Der Scrum Master koordiniert die Meetings, sofern dies vonnöten ist, und sorgt für die Wahrung der agilen Werte und Prinzipien.

### Hybride Softwareentwicklung

Die hybride Softwareentwicklung vereint Elemente der traditionellen, dokumentenzentrierten Vorgehensmodelle mit agilen Artefakten [13].

Die Projekte benötigen Organisation und müssen zudem in die Unternehmensstruktur und die Vorgehensweisen eingegliedert werden. Agile Methoden geben für einige Unternehmen nur unzureichende Hilfestellung bei der Eingliederung der agilen Methoden in die Unternehmensstruktur. Da viele Entwickler agil vorgehen, entsteht ein Konflikt in Bezug auf die Organisation. Dieser soll gelöst werden, indem hybride Ansätze verfolgt werden. Das Ziel ist dabei die Nutzung der Vorteile beider Entwicklungsmethoden: der Organisation und der Vorgaben in der plangetriebenen Softwareentwicklung und der Flexibilität der agilen Entwicklung [14].

### Methodisches Vorgehen

Um die studentischen Softwareprojekte miteinander vergleichen zu können, mussten zunächst geeignete Projekte ausgewählt und ein analoger Ablauf sichergestellt werden.

Gersick [15] fand heraus, dass ein Team in einem Projekt nach der Hälfte der Projektlaufzeit seine Strategie wechselt. Diesen Zeitpunkt nannte er *Midpoint*. In den hier betrachteten studentischen Softwareprojekten markiert der Midpoint den Beginn der Implementierung. Wir betrachten im Folgenden die beiden Bereiche des Projekts vor und nach dem Midpoint.

Nach Bindrees et al. [1] ist die Kommunikationsintensität in der Anforderungs- und Entwurfsphase am höchsten. Die Anforderungsphase legt den Grundbaustein für den weiteren Projektverlauf und ist demnach von besonderer Bedeutung [7]. Aus diesem Grund nehmen wir an:

#### Hypothese 1:

Teams, die nach dem Wasserfall-Modell entwickeln, kommunizieren vor dem Midpoint intensiver als danach.

In der agilen Entwicklung spielt die Kommunikation zu jeder Zeit eine wichtige Rolle. In den hier betrachteten hybriden Projekten gibt es einen On-Site-Customer, der wöchentlich vor Ort ist. Da die Kommunikation mit dem Kunden demnach über die gesamte Projektlaufzeit konstant bleiben sollte und wir annehmen, dass die team-interne Kommunikation während der Implementierungsphase aufgrund von Absprachen mit den Teammitgliedern zunimmt, können wir folgende Hypothese formulieren:

#### Hypothese 2:

Teams, die nach hybriden Methoden entwickeln, kommunizieren nach dem Midpoint intensiver als davor.

Als *Aufgabenkonflikte* werden Konflikte bezeichnet, bei denen sich die Teammitglieder aufgrund unterschiedlicher Ansichten, Meinungen und Ideen nicht über die Inhalte einer Aufgabe einigen können [16]. Im Gegensatz zu sozialen Konflikten sind aufgabenbezogene Konflikte nicht nur negativ einzuordnen; sie können sogar leistungsfördernd wirken [17]. Da Meinungsverschiedenheiten zu intensiverer Kommunikation führen bzw. selbst als Kommunikationsprozess begriffen werden können [18], liegt die Annahme nahe, dass Kommunikationsintensität und Aufgabenkonflikte eng zusammenhängen.

Da wir bei Teams, die nach dem Wasserfallmodell arbeiten, von einem Anstieg der Kommunikation von der ersten zur zweiten Projekthälfte ausgehen, nehmen wir dies dementsprechend ebenso für Aufgabenkonflikte an. Dies würde sich zudem mit den Ergebnissen von Schneider et al. [6] decken, die einen Anstieg aufgabenbezogener Konflikte zum Midpoint verzeichneten, bevor die Anzahl in der zweiten Projekthälfte abfiel:

#### Hypothese 3:

Teams, die nach dem Wasserfall-Modell entwickeln, haben vor dem Midpoint mehr Aufgabenkonflikte als danach.

In den hybriden Softwareprojekten gehen wir aufgrund der ansteigenden Kommunikation nach Hypothese 2 von einem Anstieg der Aufgabenkonflikte von der ersten zur zweiten Projekthälfte aus. Dementsprechend formulieren wir die folgende Hypothese:

**Hypothese 4:**

Teams, die nach hybriden Methoden entwickeln, haben nach dem Midpoint mehr Aufgabenkonflikte als davor.

Diese Hypothesen wollen wir im Folgenden validieren.

**Studentische Softwareprojekte**

Im Rahmen einer Lehrveranstaltung an der Leibniz Universität Hannover entwickeln Studierende im fünften Semester ihres Informatik-Bachelorstudiums in Teams über einen Zeitraum von 15 Wochen ein Softwareprodukt. Der Umfang und der Aufwand aller Projekte sind annähernd vergleichbar. Aus jedem Team übernehmen zwei Studierende die Rollen des Projektleiters und des Qualitätsbeauftragten. Der Kunde ist meistens ein Mitarbeiter des Fachgebiets Software Engineering und hat ein großes Interesse an einem erfolgreichen Projektabschluss. Er möchte die entwickelte Software langfristig nutzen. Unterstützt werden die Teams außerdem von einem erfahrenen Tutor, der bei zwischenmenschlichen und anderen nicht-technischen Fragen zur Verfügung steht. Technische Fragen sollen die Teams direkt mit dem Kunden oder team-intern klären.

Das Softwareprojekt folgte lange Zeit dem Wasserfallmodell. Seitdem wurde die Lehrveranstaltung durch Integration einiger agiler Artefakte zunehmend hybrider.

**Traditionelle Softwareprojekte.** Bis einschließlich dem Wintersemester 2013/2014 folgten die Softwareprojekte dem traditionellen dokumentenzentrierten Ansatz. Entwickelt wurde nach dem Wasserfall-Modell in Teams von drei bis fünf Studierenden in mehreren Phasen [7].

Das Projekt startete mit einer dreiwöchigen *Anforderungsanalyse*. Währenddessen fand das erste Kundengespräch zur Klärung der Anforderungen statt. Die Studierenden mussten eigenständig den Kontakt zum Kunden herstellen und einen Termin vereinbaren. Das Treffen musste vorbereitet, abgehalten und nachbereitet werden, da die Zeit, in der der Kunde zur Verfügung stand, auf 90 Minuten während der gesamten Projektlaufzeit begrenzt war. Die Phase endete mit dem Verfassen einer Anforderungsspezifikation, die mit dem Kunden abgestimmt und schließlich auch von ihm unterzeichnet werden musste. Danach passierte das Team das erste Quality-Gate.

Es folgte eine Phase über zwei Wochen, in denen der *Entwurf* verfasst oder ein Prototyp entwickelt wurde. Um den Erfahrungsschatz der Studierenden zu erweitern, musste zudem der Entwurf oder der Prototyp eines anderen Teams gereviewt werden. Dazu gab es ein reguläres Review-Meeting

mit Moderator, in dem sämtliche Befunde zusammengetragen wurden. Ebenso wie die Anforderungsanalyse endete auch die Entwurfsphase mit einem Quality-Gate.

Die restliche Projektlaufzeit stand im Wesentlichen für die *Implementierung* zur Verfügung. Hier mussten nachweislich alle Teammitglieder eingebunden werden. Diese Phase dauerte je nach Leistung des Teams und Fortschritt ungefähr fünf bis sechs Wochen.

Danach hatte das Team in der *Polish-Phase* die Möglichkeit, letzte Änderungswünsche des Kunden einzupflegen und die Qualität der Software zu steigern.

Nach dieser Phase gab es ein letztes Quality-Gate, in dem im Wesentlichen das Erfüllen der Anforderungsspezifikation und das Einhalten von vorgegebenen Code-Konventionen überprüft wurden.

Das Projekt endete mit der *Kundenabnahme*, die über den erfolgreichen Abschluss entschied.

**Hybride Softwareprojekte.** Seit dem Wintersemester 2014/2015 werden einige agile Artefakte in die Lehrveranstaltung integriert. Zudem bestehen die Teams nun aus acht oder neun Studierenden.

In Anlehnung an das Konzept des *On-Site-Customers* steht der Kunde wöchentlich zu einem fest vorgegebenen Termin zur Verfügung, um über den Projektverlauf und den aktuellen Stand informiert zu werden. Dieses Meeting ist vergleichbar mit einem *Review-Meeting* in der agilen Softwareentwicklung, wenngleich die wöchentlichen Iterationen nicht als Sprints angesehen werden können.

Das gesamte Team-Meeting umfasst ungefähr eine Stunde und beginnt mit einem *Weekly Scrum*, in dem jedes Team-Mitglied über seinen Fortschritt seit dem letzten Scrum-Meeting, über die Pläne bis zum nächsten Scrum-Meeting und über Probleme spricht. Nach dem Weekly Scrum kommt der Kunde hinzu und wird über die wesentlichen Punkte informiert, äußert Änderungswünsche oder geänderte Anforderungen und steht für offene Fragen zur Verfügung.

Wie auch in der plangetriebenen Entwicklung gibt es angelehnt an den Ablauf nach Kent et al. [10] eine *Anforderungsanalyse* mit dem Verfassen einer Anforderungsspezifikation, eine *Entwurfsphase*, die *Implementierung* und eine *Überprüfung* des Ergebnisses durch den Kunden. Jedoch werden diese Phasen iterativ durchlaufen. Insgesamt gibt es in den Projekten zwei *Iterationen*.

**Stichprobe und Matching**

Für die Validierung der Hypothesen verwenden wir Daten aus dem TeamFLOW-Forschungsprojekt

[19], die in den zuvor beschriebenen studentischen Softwareprojekten erhoben wurden.

Wir betrachten Daten aus zwei Kohorten: 2013/14 und 2015/2016. Die erste Kohorte folgte dem traditionellen Ansatz nach dem Wasserfall-Modell und die zweite Kohorte dem hybriden Ansatz.

In der ersten Kohorte haben 20 Teams an der Datenerhebung teilgenommen. Da die Teams in den hybriden Projekten größer waren, liegen hier nur Daten von acht Teams vor. Aus diesem Grund musste zunächst ein Matching der vorhandenen Daten durchgeführt werden, um acht geeignete Projekte aus der ersten Kohorte auszuwählen. Dieses Matching erfolgte auf Basis von Alter und Geschlecht, da sich die beiden Kohorten nicht signifikant in den Mittelwerten des Alters unterscheiden (hybrid:  $M = 21.9$ ,  $SD = 1.05$ ; Wasserfall:  $M = 21.9$ ,  $SD = 1.01$ ;  $t(7) = -.413$ ,  $p = .692$ ). Ferner wurden Teams mit unvollständigen Datensätzen nicht berücksichtigt.

Auf Basis dieses Matchings erhielten wir insgesamt acht Teams aus der Wasserfall-Kohorte, die wir mit den acht Teams aus der Hybrid-Kohorte vergleichen konnten.

## Datenerhebung

**Kommunikationsintensität.** Die Kommunikationsintensität wurde durch Abfragen nach der wahrgenommenen Intensität der Kommunikation mit den anderen Team-Mitgliedern erhoben. Dabei stand eine Likert-Skala mit fünf Werten von 1 („gar nicht kommuniziert“) bis 5 („sehr hohe Intensität“) zur Verfügung. Auf diese Art entstand eine Matrix mit Einträgen  $a_{ij}$ , die angeben, wie intensiv Team-Mitglied  $i$  die Kommunikation mit Team-Mitglied  $j$  in der vergangenen Woche wahrgenommen hat. Die ungerichtete Kommunikationsintensität  $c_{ij}$  zwischen den beiden Team-Mitgliedern  $i$  und  $j$  ergibt sich dann durch den Durchschnitt der beiden Werte:

$$c_{ij} = \text{avg}(a_{ij}, a_{ji}).$$

Um den Teamwert zu erhalten, wird der Durchschnitt aller ungerichteten Kommunikationswerte im Team berechnet.

**Konflikte.** Aufgabenkonflikte wurden mit der deutschen Version der Intragroup Conflict Scale von Jehn [16] von Lehman-Willenbrock et al. [20] erhoben. Insgesamt gab es vier Items für aufgabenbezogene Teamkonflikte. Ein Beispielitem ist: „Wie häufig gibt es Ideenkonflikte in der Gruppe?“ Den Studierenden stand eine sechs-stufige Likert-Skala mit Werten zwischen 1 (nie/keine) und 6 (sehr oft/sehr viele) zur Verfügung. Die Reliabilitäten der Skala für die drei Messzeitpunkte lagen zwischen  $\alpha$

= .88 und  $\alpha = .90$ .<sup>1</sup> Für die Berechnung auf Teamebene wurden die Werte zum Teammittelwert aggregiert.

**Messzeitpunkte.** In Abb. 1 ist die Datenerhebung nach Kohorten dargestellt. Es wird ersichtlich, dass die Daten zum Kommunikationsverhalten in der Kohorte, die nach dem Wasserfallmodell arbeitete, wöchentlich erhoben wurden, während sie in der Kohorte, die hybrid arbeitete, nur an drei ausgewählten Messpunkten abgefragt wurden. Die Erhebung von Konflikten erfolgte jeweils zu drei Messpunkten: in der ersten Woche, in der Mitte des Projektes und nach Projektabschluss. Die grau hinterlegten Wochen kurz vor Projektende spiegeln die Weihnachtsferien wieder, in denen die Studierenden zwar an der Software weiterarbeiten durften, aber nicht dazu verpflichtet waren.

Basierend auf den Messzeitpunkten betrachten wir in diesem Papier drei Zeitpunkte: unmittelbar nach Projektstart, in der sechsten Woche mit den Erhebungen in den beiden Kohorten und in der letzten Woche.

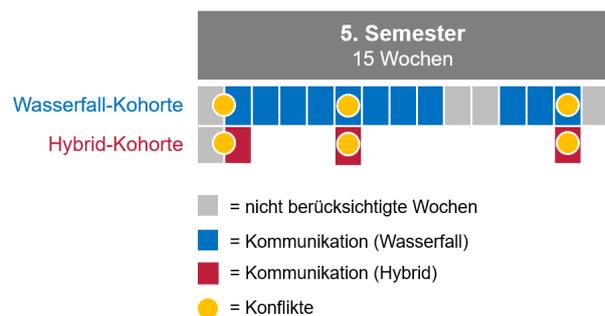


Abbildung 1: Datenerhebung nach Kohorten

## Ethik-Votum

Die Datenerhebung wurde vom Ethik-Komitee der Leibniz Universität Hannover genehmigt. Die Studierenden wurden vorab über die Datenerhebung und die weitere Verwendung der gesammelten Daten informiert und stimmten dieser zu. Darüber hinaus wurden die gesammelten Daten vor der weiteren Verwendung vollständig anonymisiert. Eine Teilnahme an der Befragung hatte keinen Einfluss auf das Bestehen der Lehrveranstaltung.

## Statistische Analyse

Alle Hypothesentests wurden jeweils aufgrund der geringen Stichprobengrößen von acht Teams zunächst einem Test auf Normalverteilung unterzogen. Dafür wurde der Shapiro-Wilk-Test verwendet, da dieser Test bei kleinen Stichproben unter 30

<sup>1</sup> Cronbachs Alpha spiegelt die interne Konsistenz einer verwendeten Skala wieder. Es gibt an, inwiefern alle Items das gleiche Konstrukt messen.

eine größere Teststärke aufweist als andere Normalverteilungstests [21]. Die erste Hypothese wird mittels t-Test für abhängige Stichproben überprüft.

Für die *erste Hypothese* werden die Kommunikationsintensitäten der Wasserfall-Kohorte dahingehend überprüft, ob die Intensität in der zweiten Projekthälfte niedriger ist als in der ersten Projekthälfte:  $\mu_1 > \mu_2$ .

Für die *zweite Hypothese* werden die Kommunikationsintensitäten der Hybrid-Kohorte dahingehend überprüft, ob die Intensität in der zweiten Projekthälfte höher ist als in der ersten Projekthälfte:  $\mu_1 < \mu_2$ .

Für die *dritte Hypothese* werden die Aufgabenkonflikte der Wasserfall-Kohorte dahingehend überprüft, ob ihre Ausprägung in der zweiten Projekthälfte niedriger ist als in der ersten Projekthälfte:  $\mu_1 > \mu_2$ .

Für die *vierte Hypothese* werden Aufgabenkonflikte der Hybrid-Kohorte dahingehend überprüft, ob ihre Ausprägung in der zweiten Projekthälfte höher ist als in der ersten Projekthälfte:  $\mu_1 < \mu_2$ .

## Signifikanzniveau

Da ausschließlich gerichtete Hypothesen aufgestellt wurden, wurde bei der Hypothesenprüfung einseitig getestet. Wir legten das Signifikanzniveau auf 5% fest. Um eine bessere Interpretierbarkeit der Ergebnisse zu erzielen, werden in dieser Arbeit neben der Betrachtung signifikanter Befunde zusätzlich die Effektgrößen dargestellt. Cohens  $d$  [22], das zur Einschätzung der Größe eines Effekts verwendet wird<sup>2</sup>, ist für die verwendeten t-Tests relevant.

## Aggregation

Wie aus Abb. 1 ersichtlich wird, liegen für die beiden Stichproben unterschiedlich viele Daten vor. Die Probanden aus der ersten Kohorte gaben 14 Wochen lang wöchentlich Auskunft. Die Studierenden aus der zweiten Kohorte gaben nur in den Wochen eins bis zwei (Projektbeginn), sechs (Midpoint) und vierzehn (Projektabschluss) Auskunft über Konflikte und Kommunikation. Dementsprechend wurden zum Vergleich die Werte aus genau diesen Wochen von der Wasserfall-Kohorte gewählt. Der Midpoint in der Mitte der Projektlaufzeit liegt demnach in Woche sechs.

**Kommunikationsintensität.** Bei den hybrid arbeitenden Teams wurden lediglich in den Wochen zwei, sechs und vierzehn Daten erhoben. Deshalb gingen auch nur diese drei Wochen der traditionell organisierten Teams als zeitliches Äquivalent in die

Berechnungen der Ergebnisse ein. Den Wert für die erste Projekthälfte stellt die Differenz zwischen dem Wert für den Midpoint und der Baseline (Projektbeginn) dar. Da in einem der hybridorganisierten Teams zu wenige Teammitglieder die Fragen zur Kommunikation beantworteten, wurde dieses Team für diesen Zeitpunkt aus der Analyse ausgeschlossen. Demnach liegt die Stichprobengröße für Kommunikationswerte zur ersten Projekthälfte bei  $N=7$ .

**Konflikte.** Fragebögen zur Erhebung von Konflikten wurden auch bei traditionell organisierten Teams zu lediglich drei Messzeitpunkten erhoben. Die erste Erhebung fand in beiden Kohorten in der ersten Woche statt. Die zweite Erhebung stellt den Midpoint dar. Die letzte Erhebung fand in beiden Kohorten nach dem Projektabschluss in Woche 14 statt. Den Wert für die erste Projekthälfte stellt die Differenz zwischen dem Wert für den Midpoint und der Baseline (Projektbeginn) dar.

## Vorbereitung und Vorgehen

Zunächst wurde der Datensatz aller Teams aus der Wasserfall-Kohorte aufgrund der geringen Stichprobengröße ( $N = 7$  bzw.  $N = 8$ ) mit dem Shapiro-Wilk-Test auf Normalverteilung geprüft. Dies traf bei den Daten zu Überprüfung aller vier Hypothesen zu.

## Ergebnisse

Die Hypothesen 1, 2 und 3 konnten bestätigt werden; die Hypothese 4 konnte nicht bestätigt werden.

**Hypothese 1.** Ein t-Test für verbundene Stichproben ergab einen signifikanten Unterschied zwischen der Kommunikationsintensität vor dem Midpoint ( $M = .45$ ,  $SD = .68$ ) und den Mittelwerten der Kommunikationsintensität nach dem Midpoint ( $M = -.53$ ,  $SD = .68$ ),  $t_{(7)} = 4.46$ ,  $p = .003$ ,  $d = 1.57$ . Hypothese 1 kann somit bestätigt werden. Es handelt sich um einen großen Effekt.

**Hypothese 2.** Ein t-Test für verbundene Stichproben ergab einen signifikanten Unterschied zwischen der Kommunikationsintensität vor dem Midpoint ( $M = .15$ ,  $SD = .43$ ) und den Mittelwerten der Kommunikationsintensität nach dem Midpoint ( $M = .38$ ,  $SD = .37$ ),  $t_{(6)} = -2.13$ ,  $p = .039$ ,  $d = 0.83$ . Hypothese 2 kann somit bestätigt werden. Es handelt sich um einen großen Effekt.

**Hypothese 3.** Ein t-Test für verbundene Stichproben ergab einen signifikanten Unterschied zwischen Aufgabenkonflikten vor dem Midpoint ( $M = .57$ ,  $SD = .52$ ) und den Mittelwerten der Aufgabenkonflikte nach dem Midpoint ( $M = .38$ ,  $SD = .57$ ),  $t_{(7)} = 2.03$ ,  $p = .041$ ,  $d = 0.74$ . Hypothese 3 kann somit

<sup>2</sup> Bei einem Wert von  $d \geq 0.20$  spricht man von einem kleinen Effekt, bei  $d \geq 0.50$  von einem mittleren Effekt und ab  $d \geq 0.80$  von einem großen Effekt.

bestätigt werden. Es handelt sich um einen mittleren Effekt.

**Hypothese 4.** Ein t-Test für verbundene Stichproben ergab keinen signifikanten Unterschied zwischen Aufgabenkonflikten vor dem Midpoint ( $M = .25$ ,  $SD = .51$ ) und den Mittelwerten der Kommunikationsintensität nach dem Midpoint ( $M = .06$ ,  $SD = .50$ ),  $t_{(6)} = 1.00$ ,  $p = .353$ . Hypothese 4 musste demnach abgelehnt werden.

## Diskussion und Interpretation

In dieser Studie untersuchten wir die Einflüsse der Arbeitsorganisation auf die Intensität der Kommunikation und die Konflikte in Teams. Grundlage der Untersuchungen waren drei Messzeitpunkte über den Projektverlauf.

Zunächst wurde der Verlauf der Kommunikationsintensität nur für Teams, die nach dem Wasserfall-Modell oder hybrid arbeiten, untersucht. Dabei sollte herausgefunden werden, ob die Intensitäten gemäß der arbeitsorganisatorischen Modelle und der Theorie verlaufen. Anschließend wurden die Kommunikationsintensitäten beider Arbeitsorganisationen gegenüber gestellt und zusammen untersucht. Die Ergebnisse unterstützen drei der vier formulierten Hypothesen:

### Hypothese 1:

Teams, die nach dem Wasserfall-Modell entwickeln, kommunizieren vor dem Midpoint intensiver als danach. ✓

### Hypothese 2:

Teams, die nach hybriden Methoden entwickeln, kommunizieren nach dem Midpoint intensiver als davor. ✓

### Hypothese 3:

Teams, die nach dem Wasserfall-Modell entwickeln, haben vor dem Midpoint mehr Aufgabenkonflikte als danach. ✓

### Hypothese 4:

Teams, die nach hybriden Methoden entwickeln, haben nach dem Midpoint mehr Aufgabenkonflikte als davor. ✗

Dennoch unterliegen diese Ergebnisse einigen Einschränkungen in Bezug auf Validität und Verallgemeinerbarkeit.

## Bedrohungen der Validität

Unsere Untersuchung basiert auf einer Erhebung in studentischen Softwareprojekten. Daraus ergeben sich einige Einschränkungen in Bezug auf die Verallgemeinerbarkeit der Ergebnisse für die Industrie. Rückschlüsse für studentische Lehrveranstaltungen sind jedoch möglich. Trotzdem soll im Folgenden

auf Bedrohungen der Validität eingegangen werden. Dabei beziehen wir uns auf die Klassifizierung der *Threats to Validity* nach Wohlin et al. [23]. Wir differenzieren zwischen *interner*, *externer* und *Konstruktvalidität* und *Verlässlichkeit* der Ergebnisse.

*Interne Validität:* Einen team-internen Austausch über die Fragebögen und Absprachen hinsichtlich der Beantwortung der Fragen konnten wir nicht verhindern. Dies kann die Ergebnisse unserer Berechnungen beeinflusst haben. Da die Beantwortung der Fragebögen und die Teilnahme an der Datenerhebung keinen Einfluss auf das Bestehen der Lehrveranstaltung hatte, schätzen wir den Anteil an falschen Antworten als ziemlich gering ein, da die Studierenden dadurch keinen Vorteil hätten.

*Externe Validität:* Durch das akademische Umfeld lassen sich unsere Ergebnisse nur bedingt auf die Industrie anwenden. Das liegt an der geringen Stichprobe, die die Gültigkeit der Ergebnisse auch für andere studentische Projekte einschränkt. Zudem lagen uns nur Daten zu drei Messzeitpunkten vor. Ein detaillierter Vergleich würde in Bezug auf den Verlauf der Kommunikationsintensität vermutlich zu ähnlichen Ergebnissen führen. Möglicherweise könnten mit einer größeren Stichprobe auch projektphasenabhängige Muster aufgedeckt werden, die bislang nicht erkannt wurden.

Darüber hinaus hatten die Studierenden nur zum Teil Erfahrungen in Softwareprojekten oder gar in der Industrie. Ferner betrachteten wir nur sehr vergleichbare Projekte mit einem ähnlichen Umfang und Übereinstimmungen in verschiedenen Projektparametern wie Dauer oder dem Erfahrungsschatz der Teammitglieder.

*Konstruktvalidität:* Die Fragebögen waren sehr allgemeinverständlich formuliert, sodass wir keine Einschränkungen der Validität aufgrund von falsch verstandenen Fragen erwarten.

Dennoch könnte die Erhebung der Intensität könnte als problematisch aufgefasst werden, da es sich hierbei um eine sehr subjektive Einschätzung handelt. Von Klünder et al. [24] konnte nachgewiesen werden, dass die Inkonsistenzen zwischen den wahrgenommenen Intensitäten hinreichend gering sind, sodass die Erhebung der Intensität auf diese Weise möglich ist.

Darüber hinaus können wir Auswirkungen der Teamgrößen auf Kommunikation und Konflikte nicht ausschließen. Steigende Teamgröße führt nachweislich zu Motivations- und Koordinationsverlusten in Gruppen [25]. Zudem konnten Amazon und Sapienza [26] zeigen, dass Teamgröße positiv mit kognitiven Konflikten (welche Aufgabenkonflikten stark ähneln) korreliert. Da in unserer Studie die hybriden Teams auch gleichzeitig deutlich größer als die Teams der Wasserfall-

Kohorte waren, können wir eine Konfundierung mit dem Faktor Teamgröße demnach nicht ausschließen.

*Verlässlichkeit:* Eine erneute Durchführung dieser Untersuchung auf Basis einer anderen Datenerhebung könnte zu anderen Ergebnissen führen, sofern unterschiedliche Aufgabenstellungen, Teilnehmende mit einer anderen Wissensbasis, Projektabläufe und Teamstrukturen betrachtet werden. Darüber hinaus hängt das Kommunikationsverhalten stark vom Charakter der Teammitglieder ab. Demnach beeinflusst die Teamkonstellation die Ergebnisse zusätzlich. Die Betrachtung einer Datenerhebung mit vergleichbaren Projektparametern sollte aber zu ähnlichen Ergebnissen führen.

## Interpretation

Unsere Ergebnisse zeigen, dass traditionell organisierte Teams in der ersten Hälfte des Projekts deutlich intensiver kommunizieren als in der zweiten Projekthälfte. Hybrid arbeitende Teams kommunizieren hingegen vermehrt in der zweiten Projekthälfte.

Neben der Kommunikationsintensität wurden noch aufgabenbezogene Konflikte untersucht. Dieses Konstrukt wurde ebenfalls zu drei Messzeitpunkten erhoben, jedoch lediglich als aggregierte Werte pro Arbeitsorganisation gegenübergestellt. Hier konnte festgestellt werden, dass – analog zur Kommunikationsintensität – traditionell organisierte Teams in der ersten Projekthälfte mehr aufgabenbezogene Konflikte erleben. Für hybrid organisierte Teams konnten keine Unterschiede hinsichtlich aufgabenbezogener Konflikte zwischen den beiden Projekthälften festgestellt werden.

**Kommunikationsintensität.** Traditionell organisierte Teams kommunizieren in der ersten Projekthälfte deutlich mehr als in der zweiten Projekthälfte. Dies geht mit den Erkenntnissen von Bindrees et al. [1] einher. Sie fanden heraus, dass traditionell organisierte Teams in den ersten Projektphasen am intensivsten kommunizieren. Dieser Verlauf der Kommunikationsintensität lässt sich mit dem Wasserfall-Modell begründen, da die Anforderungs- und Entwurfsphase ohne Kommunikation nicht erfolgen können, während Kommunikation für die Implementierung nicht mehr in großem Ausmaß vonnöten ist, nachdem die Anforderungen klar herausgearbeitet und ausreichend dokumentiert wurden.

Neu hingegen ist die Erkenntnis, dass hybrid entwickelnde Teams in der zweiten Projekthälfte mehr kommunizieren. Dies lässt sich damit erklären, dass die Kommunikation mit dem Kunden über den gesamten Projektverlauf nahezu konstant bleibt, während in der Implementierungsphase

mehr team-interne Absprachen notwendig sind, da zwar – im Falle der hier betrachteten Softwareprojekte – eine Spezifikation existiert, diese aber nicht bindend ist, da auf Änderungswünsche des Kunden eingegangen werden soll. Aus diesem Grund sind möglicherweise klärende Absprachen vonnöten, wodurch die Intensität der Kommunikation zunimmt.

**Konflikte.** Unsere Untersuchung zeigt, dass bei Teams mit traditioneller Softwareentwicklung aufgabenbezogene Konflikte über die Zeit bzw. nach dem Midpoint abnehmen. Dies lässt sich u.a. über die erhöhte Kommunikationsintensität zu Projektbeginn erklären.

Die Ergebnisse erlauben auch einige für die universitäre Lehre interessante Rückschlüsse. So wird beispielsweise deutlich, dass die Wahl der Arbeitsorganisation einen Einfluss auf das Kommunikationsverhalten hat. Damit beeinflusst sie zudem den Lernerfolg der Studierenden, die während der Arbeit im Team vor allem soziale Fähigkeiten wie beispielsweise die Kommunikationsfähigkeiten erlernen oder ausbauen sollen. Vor allem die Kommunikationsfähigkeit ist in industriellen Projekten von äußerster Bedeutung. Daher empfiehlt es sich, Elemente wie beispielsweise den On-Site-Customer in studentischen Arbeitsprojekten zu integrieren, um den Studierenden einen intensiven Kundenkontakt zu ermöglichen, für den eine adäquate Kommunikation unabdingbar ist. Darüber hinaus stellt ein Weekly Scrum ein Minimum an Informationsweitergabe sicher. Zudem lernen die Studierenden, sich adäquat untereinander auszutauschen und im anschließenden Kundengespräch die Informationen an den Kunden weiterzugeben. Durch den wöchentlichen persönlichen Kontakt mit dem Kunden wird der Umgang routinierter. Außerdem wirkt der hybride Ansatz einer Verringerung der Kommunikationsintensität in der zweiten Projekthälfte entgegen. In der zweiten Projekthälfte liegt der Schwerpunkt auf der Implementierung der Software. Eine Verringerung der Kommunikationsintensität in der zweiten Projekthälfte, wie sie bei den traditionell organisierten Projekten auftritt, deutet auf weniger Teamarbeit und Austausch während der Implementierung hin. Dies reduziert unter Umständen den team-internen Wissensaustausch und die Kompetenzen der Teammitglieder können möglicherweise nicht optimal genutzt werden.

## Fazit

In dieser Studie untersuchten wir den Einfluss der Arbeitsorganisation auf die Intensität der Kommunikation eines Teams und ihre Konflikte.

Die Intensität der Kommunikation ist in den untersuchten traditionell entwickelnden Teams in der ersten Projekthälfte deutlich höher als in der zweiten Projekthälfte. Bei den hybrid organisierten Teams gilt das Gegenteil. Entgegen unserer Annahmen unterschieden sich aufgabenbezogene Konflikte bei hybrid organisierten Teams zwischen der ersten und zweiten Projekthälfte nicht.

Die Verwendung von agilen Artefakten wie beispielsweise dem On-Site-Customer oder dem Weekly Scrum in der studentischen Lehre fördert die Kommunikationsfähigkeiten der Studierenden, ohne die Arbeit im Team zu erschweren. Deshalb kann das Arbeiten in hybriden Projekten die Studierenden ohne zusätzliche Schwierigkeiten auf das spätere Arbeitsumfeld vorbereiten.

## Danksagung

Diese Arbeit wurde von der Deutschen Forschungsgesellschaft (DFG) unter der Projektnummer 263807701 (Projekt: TeamFLOW) gefördert.

## Literatur

- [1] Bindrees, M. A.; Pooley, R. J.; Ibrahim, I. S.; Taylor, N. K.: Re-Evaluating Media Richness Theory in Software Development Settings. In: *Journal of Computer and Communications* 02 (2014) 14, S. 37–51.
- [2] Mohamed, J. M.; Qubaisi, L. F. A.; Elanain, H. M. A.; Badri, M. A.; Ajmal, M. M.: Leadership, Culture and Team Communication. Analysis of project success causality - A UAE case. In: *International Journal of Applied Management Science* 7 (2015) 3.
- [3] ISO/IEC 12207:2008: Systems and software engineering - Software life cycle processes.
- [4] Beck, K.; Andres, C.: *Extreme Programming Explained. Embrace Change*, 2nd ed. Boston, MA 2005.
- [5] Brügge, B.; Krusche, S.; Alperowitz, L.: Software Engineering Project Courses with Industrial Clients. In: *ACM Transactions on Computing Education* 15 (2015) 4.
- [6] Schneider, K.; Liskin, O.; Paulsen, H.; Kauffeld, S.: Media, Mood, and Meetings. Related to Project Success. In: *ACM Transactions on Computing Education* 15 (2015) 4.
- [7] Stoica, M.; Mircea, M.; Ghilic-Micu, B.: Software Development. Agile vs. Traditional. In: *Informatica Economica* 17 (2013) 4/2013.
- [8] Benington, H. D.: Production of Large Computer Programs. In: *Proceedings, ONR Symposium* (1956).
- [9] Royce, W. W.: Managing the Development of Large Software Systems: Concepts and Techniques. In: *Proceedings of the 9th International Conference on Software Engineering*, Monterey, California, USA (1987).
- [10] Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D.: *Agile Manifesto 2001*.
- [11] Boehm, B. W.: A spiral model of software development and enhancement. In: *Computer* 21 (1988) 5, S. 61–72.
- [12] Schwaber, K.; Beedle, M.: *Agile software development with Scrum*. Upper Saddle River, NJ 2002.
- [13] West, D.: Water-Scrum-Fall is the Reality of Agile for Most Organizations Today. Manage The Water-Scrum And Scrum-Fall boundaries To Increase Agility, Technical Report. In: Forrester (2011).
- [14] Theocharis, G.; Kuhrmann, M.; Münch, J.; Diebold, P.: Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices. In: Abrahamsson, P.; Corral, L.; Oivo, M.; Russo, B. (Hrsg.): *Proceedings of the 16th international conference on Product-focused software process improvement. PROFES 2015*, Bolzano, Italy, 2015.
- [15] Gersick, C. J. G.: Time and Transition in Work Teams. Toward a New Model of Group Development. In: *Academy of Management Journal* 31 (1988) 1, S. 9–41.
- [16] Jehn, K. A.: A Multimethod Examination of the Benefits and Detriments of Intragroup Conflict. In: *Administrative Science Quarterly* 40 (1995) 2.
- [17] Wit, F. R. C. de; Greer, L. L.; Jehn, K. A.: The paradox of intragroup conflict: a meta-analysis. In: *The Journal of applied psychology* 97 (2012) 2.
- [18] Yang, J.; Mossholder, K. W.: Decoupling task and relationship conflict. The role of intragroup emotional processing. In: *Journal of Organizational Behavior* 25 (2004) 5.
- [19] Deutsche Forschungsgemeinschaft (DFG): TeamFLOW: Interaktion und Kommunikation in verteilten Softwareteams. URL: <http://gepris.dfg.de/gepris/projekt/263807701>. Abrufdatum 31.10.2016.
- [20] Lehmann-Willenbrock, N.; Meyers, R. A.; Kauffeld, S.; Neining, A.; Henschel, A.: Verbal Interaction Sequences and Group

- Mood. Exploring the Role of Team Planning Communication. In: *Small Group Research* 42 (2011) 6.
- [21] Shapiro, S. S.; Wilk, M. B.: An analysis of variance test for normality (complete samples). In: *Biometrika* 52 (1965) 3-4, S. 591–611.
- [22] Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, N.J. 1988.
- [23] Wohlin, C.: *Experimentation in software engineering*. Berlin, New York 2012.
- [24] Klünder, J.; Karras, O.; Kortum, F.; Schneider, K.: Forecasting Communication Behavior in Student Software Projects. In *Proceedings of the the 12th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE 2016)*. ACM, New York, 2016.
- [25] Latané, B.; Williams, K.; Harkins, S.: Many hands make light the work. The causes and consequences of social loafing. In: *Journal of Personality and Social Psychology* 37 (1979) 6.
- [26] Amason, A.: The effects of top management team size and interaction norms on cognitive and affective conflict. In: *Journal of Management* 23 (1997) 4, S. 495–516.