

Herausforderungen bei der Integration von Teamarbeit in die Lehre am Beispiel einer Lehrveranstaltung aus der Informatik

Anja Hawlitschek¹, Galina Rudolf², Sebastian Zug³

Abstract: Die Integration von Teamarbeit in Lehrveranstaltungen der Informatik ist aus vielerlei Hinsicht wünschenswert. Teamarbeit spielt im beruflichen Alltag eine wichtige Rolle, kann zur Steigerung der Qualität von Lernprozessen beitragen und die sozialen Interaktionen zwischen Studierenden fördern. Was aber, wenn Studierende solche Szenarien nicht umsetzen wollen (oder können)? Teamarbeit ist anspruchsvoll – das Team muss sich organisieren, koordinieren, regulieren. Nehmen Lernende den Mehrwert nicht wahr oder sind die kognitiven Kosten zu hoch, kann Teamarbeit scheitern. Ein solches zumindest partielles Scheitern werden wir im Beitrag analysieren und diskutieren. Hierfür ziehen wir Ergebnisse einer Logfileanalyse heran, die auf einen geringen Anteil kooperativer und kollaborativer Programmieraktivitäten hindeuten, obgleich das didaktische Konzept der Lehrveranstaltung auf Teamarbeit ausgerichtet war. Mittels der Ergebnisse einer Vor- und Nachbefragung identifizieren wir potentielle Ursachen, leiten Potentiale für die Weiterentwicklung der Lehrveranstaltung ab und geben Handlungsempfehlungen für die Lehre.

Keywords: kooperatives Lernen; Hochschullehre; didaktisches Design

1 Einleitung

Die Integration kooperativer und kollaborativer Lernformen in die Lehre bietet viele Vorteile. Während der Covid19-Pandemie, in der die eingeschränkte Interaktion mit anderen Studierenden als besonders negativ wahrgenommen wurde [HBA20], konnte Teamarbeit diese zumindest ein Stück weit fördern. Aber auch im beruflichen Alltag in IT-Berufen spielt die Teamarbeit eine wichtige Rolle [Pe18]. Hierfür bereits im Studium Kompetenzen zu entwickeln, ist eine wichtige Voraussetzung für den erfolgreichen Start ins Berufsleben. Kooperative und kollaborative Lernformen können zudem zur Steigerung der Qualität von Lernprozessen beitragen. Lernende, die sich gegenseitig Inhalte erklären, setzen sich oft intensiver und elaborierter mit diesen auseinander [BLK00]. In der Informatik wurden solche Effekte im Kontext des Pair Programmings untersucht, welches im Vergleich mit Solo Programming oftmals einen höheren Lernerfolg und eine bessere

¹ Otto-von-Guericke-Universität, Universitätsplatz 2, 39106 Magdeburg, anja.hawlitschek@ovgu.de

² TU Bergakademie Freiberg, Akademiestraße 6, 09599 Freiberg, galina-rudolf@informatik.tu-freiberg.de

³ TU Bergakademie Freiberg, Akademiestraße 6, 09599 Freiberg, sebastian.zug@informatik.tu-freiberg.de

Performanz beim Programmieren nach sich zieht [UR17]. Beim Einsatz kooperativer und kollaborativer Szenarien sehen sich Lehrende jedoch mitunter mit der Herausforderung konfrontiert, dass Studierende die gewünschten Lernformen nicht im gewünschten Umfang bzw. in der erwarteten Qualität ausführen. Gründe hierfür sind einfach zu finden: Teamarbeit ist eine komplexe und anspruchsvolle Tätigkeit. Lernende müssen sich und das Team organisieren, koordinieren und ihre Lernprozesse regulieren. Dies gilt umso mehr unter Pandemie-Bedingungen. Während des Lockdowns mussten Teams z.B. aufgrund der örtlichen Separation der Mitglieder, einen größeren koordinativen Aufwand zur Ermöglichung der Zusammenarbeit betreiben [Wi21]. Viele Teams beschreiben damit einhergehend eine Veränderung der Teamarbeit von kollaborativen hin zu stärker kooperativen Formen, da letztere als weniger zeitaufwendig und problemfällig wahrgenommen wurden. Insgesamt gilt: Wenn Lernende den Mehrwert von Teamarbeit nicht wahrnehmen oder die kognitiven bzw. zeitlichen Kosten als zu hoch einschätzen, kann daraus ein Scheitern solcher Lernszenarien resultieren [NRG15]. In unserem Artikel werden wir ein solches (partielles) Scheitern beschreiben. Ausgangspunkt war die Idee, angesichts der Vorteile kooperativer und kollaborativer Lernformen, eine Lehrveranstaltung zur Softwareentwicklung auf Teamarbeit umzustellen. Die Analyse der studentischen Daten zeigt jedoch einen geringen Anteil von Zusammenarbeit. Wir werden diese Beobachtung unter Rückbezug auf Studierendenbefragungen diskutieren und Schlussfolgerungen für die Weiterentwicklung der Lehrveranstaltung sowie allgemein für die Integration von Teamarbeit in die Lehre ziehen.

2 Didaktisches Konzept

Das Forschungsprojekt konzentriert sich auf die Vorlesung “Softwareentwicklung“ für Studierende der Angewandten Informatik und Robotik im zweiten Semester an einer deutschen Universität. Parallel zur Vorlesung, die die theoretischen Grundlagen vermittelt, schulen Übungen die programmiertechnischen Fähigkeiten, das Projektmanagement und die Teamfähigkeit. Die Bearbeitung der Übungsaufgaben sollte in Zweierteams erfolgen. Jedes der sieben Aufgabenblätter bestand aus 2 bis 5 Teilaufgaben unterschiedlicher Komplexität. Für die Bearbeitung der ersten Aufgabe wurde eine kollaborative Arbeitsweise vorgegeben, um die Studierenden mit diesem Modus der Teamarbeit bekannt zu machen. Die erfolgreiche Bearbeitung der Aufgaben war keine zwingende Voraussetzung für die Teilnahme an der Abschlussprüfung. Sowohl die Übungen als auch die Vorlesung wurden unter Pandemie-Bedingungen im Online-Modus durchgeführt. Beim Übungsbetrieb wurde GitHub und darauf aufbauend GitHub-Classrooms⁴ als Lernplattform gewählt. Die Aufgaben wurden im Lernmanagementsystem angekündigt. Sobald die Studierenden dem Link folgten, wurde automatisch ein teaminternes GitHub-Repository angelegt, das eine Arbeitskopie enthielt und nur für die Teammitglieder und die Betreuer zugänglich war. In der Vorlesung wurde

⁴ GitHub-Classrooms Webseite <https://classroom.github.com/>

die Integration von Versionierungssystemen in den Entwicklungsprozess und die Anwendung von Managementfunktionen in GitHub vorgestellt. Die Nutzung der webbasierten Entwicklungsplattform GitHub trainiert zum einen die in hohem Maße relevanten Fähigkeiten der Studierenden im Bereich der Versionsverwaltung und des Projektmanagements. Gleichzeitig stellte diese Lösung unter Pandemie-Bedingungen sicher, dass Lehrende und Studierende eine gemeinsame Sicht auf den aktuellen Stand der Implementierung hatten.

3 Teamarbeit im Kurs

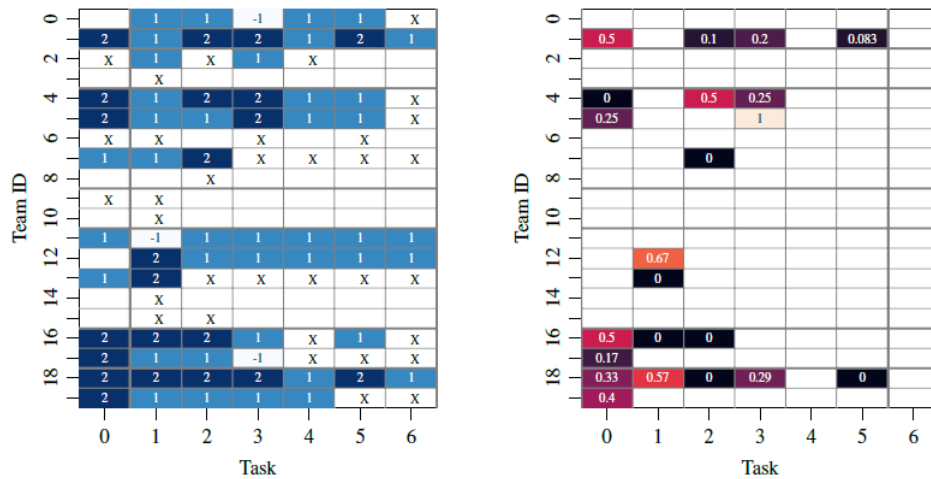
Im Rahmen der ersten Vorlesung bildeten sich 19 Zweier-Teams aus 42 Teilnehmern. Deren GitHub Aktivitäten wurden begleitet und analysiert. 12 der 19 Teams haben mit der Bearbeitung von mindestens 5 der 7 Teilaufgaben begonnen. Dabei zeigten sich unterschiedliche Muster in der Teamarbeit.

3.1 Ergebnisse der Logfile-Analyse

Für die Analyse wurden die Repositories der Teams gespiegelt, automatisch erfasst, anonymisiert und gefiltert. Die Umsetzung erfolgte auf der Basis eines für diesen Zweckentwickelten Python-Paketes⁵. Die Ergebnisse werden in Abb. 1 zusammengefasst. Das erste Diagramm (1a) gibt einen Überblick über die Teamaktivitäten während der Bearbeitung der Teilaufgaben, das zweite Diagramm (1b) illustriert den Anteil der kollaborativ bearbeiteten Codedateien. Diagramm (1a) differenziert zunächst individuellen Bearbeitungen und Teamaktivitäten. Eine "2" markiert die gemeinsame Aufgabenbearbeitung, bei der jeder Partner mindestens eine Codeänderung beigetragen hat. Wenn ein Teammitglied allein an der gesamten Aufgabe arbeitete, wird dies durch eine "1" dargestellt, wobei das Vorzeichen eine Unterscheidung zwischen den beteiligten Teammitgliedern ermöglicht. Eine 1 steht für das erste, eine -1 für das zweite Teammitglied. Eine 1 oder eine -1 kann jedoch auch bedeuten, dass beide Studenten im Pair Programming die Aufgabe auf einem Rechner implementiert haben und dann die Einreichung auf GitHub nur unter einem Account realisiert haben. Ein leeres Feld zeigt an, dass ein Team die Aufgabe nicht aktiviert hat, ein "x", dass keine Änderungen eingestellt wurden. Dies kann bedeuten, dass das Team gar nicht an der Aufgabe gearbeitet hat, oder aber die Lösung komplett offline implementierte.

Die Analyse der Teaminteraktion konzentriert sich auf jene Aufgaben, bei denen beide Teammitglieder mitgewirkt haben (sichtbar durch "2" in Abb. (1a)). Dies sind 22 von 65 abgeschlossenen Aufgaben (z.B. Team1 - Aufgabe 0, 2, 3, 5). Um die Frage nach Teamarbeitsmustern auf Dateiebene zu beantworten, haben wir den jeweiligen Anteil der gemeinschaftlich bearbeiteten Codedateien identifiziert.

⁵ github2pandas Projektwebseite <https://github.com/TUBAF-IFI-DiPiT/github2pandas>



(a) Aktive beitragende Studierende

(b) Anteil gemeinsam editierter Dateien

Abb. 1: Darstellung des Verlaufes der Aktivitäten über die Teilaufgaben 0-6

Ein Wert "1.0" würde kenntlich machen, dass beide Teammitglieder alle relevanten Dateien manipuliert haben. Einträge von "0.0" verweisen auf unabhängige Aktivitäten mit Dateien, zu denen das zweite Teammitglied nichts beigetragen hat. In der Analyse der Daten ergeben sich folgende Erkenntnisse: Zum einen scheint die Nutzung von GitHub auf freiwilliger Basis für viele Studierende unattraktiv gewesen zu sein (lediglich 65 aktive Tasks von potentiell 133). Zum anderen beträgt der Anteil der gemeinsam bearbeiteten Aufgaben nur etwa ein Drittel (22 Multi-Autoren-Aufgaben von 65 aktiv bearbeiteten Aufgaben). Der Anteil kollaborativ editierter Dateien ist gering.

3.2 Ergebnisse der Studierendenbefragung

Eine Vor- und Nachbefragung mittels Online-Fragebogen wurde durchgeführt, um Herausforderungen der Teamarbeit sowie Potentiale für die Weiterentwicklung der Lehrveranstaltung identifizieren zu können. An der Vorbefragung am Anfang des Semesters nahmen 18 Studierende teil. Ihre Erfahrung mit Teamarbeit bewerteten die Studierenden auf einer Likert-Skala von 1 (sehr unerfahren) bis 5 (sehr erfahren) mit 1,6 (SD: 0.78). In einer offenen Frage nach Vor- und Nachteilen von Teamarbeit benannten die Studierenden überwiegend Nachteile, wie unproduktive Partner*innen (2), Ablenkung (1), Ineffizienz (1) und wiesen auf die Relevanz ähnlicher Kenntnisse im Team (1) und passender Aufgaben hin (1). Als Vorteil wurde lediglich die mögliche Unterstützung durch Partner*innen genannt (1).

An der Nachbefragung am Ende des Semesters nahmen 11 Studierende teil. Von diesen gaben sechs an, immer alleine an den Aufgaben gearbeitet zu haben. Als Gründe hierfür wurde in einer offenen Frage überwiegend die fehlende Partizipation der Partner*in genannt (4). Die Studierenden wünschten sich auf einer Likert-Skala von 1 (stimme gar nicht zu) bis 5 (stimme voll zu) vom Lehrenden Unterstützung bei Problemen in der Zusammenarbeit (M: 4,5; SD: 0,6), Anleitungen zur Nutzung von Teamarbeits-Tools (M: 4,3; SD: 0,8), didaktische Unterstützung bei der Entwicklung von Teamwork-Skills (M: 3,9; SD: 1) sowie eine didaktische Begleitung der Teamarbeit (M: 3,8; SD: 0,9).

4 Diskussion

Unsere Erfahrungen zeigen, dass kooperative und kollaborative Lernszenarien keine Selbstläufer sind. Eine Aufforderung zur Teambildung und zur Teamarbeit zu geben und eine Arbeitsumgebung hierfür vorzustellen, garantiert noch keine Zusammenarbeit. Die Logfile-Analysen zeigen, dass nur ein Teil der Studierenden GitHub genutzt hat. Die Daten der Studierendengruppen, die mit GitHub gearbeitet haben, deuten auf eine eingeschränkte Zusammenarbeit hin. Kollaborative Lernaktivitäten (gemeinsame Codebearbeitung) lassen sich nur bei wenigen Aufgaben und wenigen Studierendengruppen identifizieren. Kooperative Lernaktivitäten (Aufteilung der Codebearbeitung) finden etwas häufiger statt, gehen jedoch in der Regel mit einem Ungleichgewicht bei der Bearbeitung zwischen den Teammitgliedern einher. Eine Limitation unserer Analyse ist, dass sich aus den Logfiles nur Indikatoren für Zusammenarbeit ablesen lassen, aber kein vollständiges Bild ableitbar ist. Die Studierenden konnten auch mit anderen Tools arbeiten oder ihre Lösungen offline implementieren. Die Ergebnisse aus der Nachbefragung deuten jedoch ebenso auf eine eingeschränkte Zusammenarbeit hin, auch wenn das geringe N einschränkend auf die Aussagekraft wirkt. Über die Hälfte der Teilnehmer*innen gaben hier an, immer alleine gearbeitet zu haben, vor allem weil die Partner*innen nicht partizipierten. Gründe für das teilweise Scheitern unseres didaktischen Konzeptes sind einerseits in den geringen Vorerfahrungen der Studierenden in Teamarbeit zu finden. Diesbezügliche Kompetenzen sind eine Erfolgsbedingung für erfolgreiche Zusammenarbeit [KKJ03]. Der in der Nachbefragung geäußerte Wunsch nach didaktischer Anleitung von Teamarbeit und Unterstützung beim Kompetenzerwerb ist daher zugleich ein Lösungsansatz. Hierfür werden wir zukünftig Collaboration Scripts einsetzen. Als Collaboration Scripts bezeichnet man Anleitungen dazu, wie Teammitglieder miteinander interagieren und kollaborieren sollen. Im Vergleich mit nicht angeleiteter Teamarbeit können solche Scripte nicht nur den inhaltlichen Lernerfolg steigern, sondern auch das Wissen über wichtige Prinzipien der erfolgreichen Teamarbeit [RS07]. Zugleich äußerten die Studierenden generelle Vorbehalte gegenüber Teamarbeit. Offenbar ist es uns nicht gelungen, den Mehrwert von Teamarbeit für den Kurs zu verdeutlichen. Die damit verbundenen Ziele und Vorteile werden wir daher künftig in der Einführungsveranstaltung mit den Studierenden besprechen und konkrete Prozedere für den Umgang mit Problemen in der Zusammenarbeit benennen. Ein weiterer Ansatz ist die Überarbeitung der

Aufgaben. Diese sollten nicht zu einfach oder zu schnell zu lösen sein, um Teamarbeit nicht ineffizient zu machen bzw. erscheinen zu lassen [Bo20]. Folgende Handlungsempfehlungen für die Integration von Teamarbeit in die Lehre leiten wir ab: (1) Didaktische Unterstützung implementieren, insbesondere für wenig erfahrenen Studierende (z.B. Anleitungen zum Teambuilding, zum Ablauf und zum Prozedere bei Problemen), (2) den Mehrwert von Teamarbeit verdeutlichen und (3) die Aufgaben so anpassen, dass Teamarbeit zur erfolgreichen Bearbeitung erforderlich wird (z.B. Aufteilung von Rollen beim Programmieren).

Literatur

- [BLK00] van Boxtel, C.; van der Linden, J.; Kanselaar, G.: Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction* 10/4, S. 311–330, 2000.
- [Bo20] Bowman, N. A.; Jarratt, L.; Culver, K. C.; Segre, A. M.: Pair Programming in Perspective: Effects on Persistence, Achievement, and Equity in Computer Science. *Journal of Research on Educational Effectiveness* 13/4, S. 731–758, 2020.
- [HBA20] Hebebcı, M.; Bertiz, Y.; Alan, S.: Investigation of Views of Students and Teachers on Distance Education Practices during the Coronavirus (COVID-19) Pandemic. *International Journal of Technology in Education and Science* 4/4, S. 267–282, 2020.
- [KKJ03] Kreijns, K.; Kirschner, P. A.; Jochems, W.: Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research. *Computers in human behavior* 19/3, S. 335–353, 2003.
- [NRG15] Nokes-Malach, T.; Richey, J.; Gadgil, S.: When Is It Better to Learn Together? Insights from Research on Collaborative Learning. *Educational Psychology Review* 27/4, S. 645–656, 2015.
- [Pe18] Peslak, A.; Kovalchick, L.; Kovacs, P.; Conforti, M.; Wang, W.; Bhatnagar, N.: Linking programmer analyst skills to industry needs: A current review. In: *Proceedings of the EDSIG Conference*. Norfolk, VA, S. 569–574, 2018, url: <http://proc.iscap.info/2018/pdf/4626.pdf>.
- [RS07] Rummel, N.; Spada, H.: Can People Learn Computer-Mediated Collaboration by Following A Script? In (Fischer, F.; Kollar, I.; Mandl, H.; Haake, J. M., Hrsg.): *Scripting Computer-Supported Collaborative Learning: Cognitive, Computational and Educational Perspectives*. Springer, Boston, S. 39–55, 2007.
- [UR17] Umapathy, K.; Ritzhaupt, A. D.: A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education (TOCE)* 17/4, S. 1–13, 2017.
- [Wi21] Wildman, J. L.; Nguyen, D. M.; Duong, N. S.; Warren, C.: Student Teamwork During COVID-19: Challenges, Changes, and Consequences. *Small Group Research* 52/2, S. 119–134, 2021.