

Pädagogische Hochschule Schwyz
Masterstudiengang Fachdidaktik Medien und Informatik

Masterarbeit zum Thema:

Gender und Pair Programming

Eine empirische Studie zur Zusammenarbeit von geschlechtshomogenen und -heterogenen
Zweiergruppen in einem Robotik-Kurs

Eingereicht bei:

Prof. Dr. sc. nat. Dorothee Brovelli

Sentimatt 1

6003 Luzern

041 203 01 52

dorothee.brovelli@phlu.ch

Eingereicht von:

Janine Nadja Küng

janine.kueng@phlu.ch

Goldau, 31.07.2021

Abstract

Pair Programming ist eine Arbeitstechnik, bei welcher zwei Personen nach bestimmten Regeln gemeinsam programmieren. In dieser Videostudie mit Lernenden der Primar- und Sekundarstufe wird untersucht, ob und inwiefern sich die Zuteilung in geschlechtshomogene und -heterogene Paare auf die Zusammenarbeit beim Pair Programming in einem Robotik-Kurs auswirkt. Die Ergebnisse weisen darauf hin, dass sich die Geschlechterzusammensetzungen bezüglich Lösegeschwindigkeit, Anzahl angeforderter Hilfestellungen, Rollenwechsel, sowie Summe der Anzahl und Dauer der Eingriffe in der Navigatoren-Rolle nicht statistisch signifikant unterscheiden. Nur bei Schülerinnen und Schülern der Sekundarstufe zeigt sich, dass die Lernenden in geschlechtshomogenen männlichen Paaren in der Navigatoren-Rolle statistisch signifikant häufiger eingreifen als in anderen Geschlechterzusammensetzungen. In geschlechtsheterogenen Paaren greifen weder die Jungen noch die Mädchen in der Navigatoren-Rolle häufiger ein. Geschlechtsheterogene Paare weisen die höchste Variabilität der Lösegeschwindigkeit und Summe der durchschnittlichen Dauer eines Eingriffs auf. Weiterführende Forschung im Bereich Gender und Pair Programming könnte auf die Qualität der Zusammenarbeit ausgerichtet sein.

Danksagung

Die vorgestellte Studie durfte in einer Lernumgebung mit dem Titel *Mit Roberta® die Stadt der Zukunft entdecken* im Herbstsemester 2020 an der Pädagogischen Hochschule Luzern durchgeführt werden. Deshalb danke ich den involvierten Institutionen Pädagogische Hochschule Luzern und Hochschule Luzern, sowie dem Förderprogramm MINT-Schweiz der Akademien der Wissenschaften, welche die Durchführung der Lernumgebung ermöglicht haben. Ein besonderer Dank gilt den Projektleitenden Prof. Dr. sc. Nat. Dorothee Brovelli, Prof. Dr. René Hüsler und Andrea Maria Schmid, den Projektmitarbeitenden Roland Christen und Eveline Thaler, und den Co-Leitungen der Lernwerkstatt Prof. Dr. phil. Susanne Wildhirt und Gabriela Schürch. Hinter diesem Projekt stehen noch viele weitere Personen, welche die Entwicklung und Durchführung unterstützt haben, hier aber nicht alle namentlich erwähnt werden können. Dazu gehören neben der Administration der Lernwerkstatt auch die Kursleitungen und Assistierenden, welche die Lernumgebung trotz COVID-19-Pandemie bis zum Abschluss professionell geleitet haben. Ebenfalls möchte ich allen Lehrpersonen, Begleitpersonen und Schülerinnen und Schülern danken, welche die Lernwerkstatt besuchten, und sich bereit erklärten, an dieser Videostudie teilzunehmen. Ein besonderer Dank gilt meiner Masterarbeitsbetreuerin Prof. Dr. sc. Nat. Dorothee Brovelli, welche mich während des ganzen Prozesses unterstützt und angeleitet hat. Abschliessend danke ich meiner Familie und Kolleginnen und Kollegen, welche mich moralisch und inhaltlich begleitet haben.

Inhaltsverzeichnis

1.	Einleitung	1
2.	Theoretischer Hintergrund	3
2.1.	Begriff Gender	3
2.1.1.	Definition und allgemeine theoretische Hintergründe.....	3
2.1.2.	Rolle von Gender in der Informatik	4
2.2.	Begriff Pair Programming	5
2.2.1.	Herkunft der Praktik Pair Programming	6
2.2.2.	Pair Programming Leitlinien	6
2.3.	Effekte des Pair Programmings.....	10
2.3.1.	Pair Programming im akademischen Kontext	11
2.3.2.	Pair Programming im schulischen Kontext.....	13
2.3.3.	Effekte auf allen Stufen	14
2.3.4.	Heterogenität der Resultate	15
2.4.	Einflussfaktoren auf das Pair Programming.....	16
2.4.1.	Gender.....	17
2.4.2.	Eigenschaften der Lernenden.....	18
2.4.3.	Externe Faktoren.....	22
2.5.	Gleichberechtigung der Partizipation beim Pair Programming.....	23
2.6.	Pair Programming in einem Robotik-Kurs.....	25
3.	Methode	26
3.1.	Forschungsfragen und Hypothesen	26
3.2.	Forschungsdesign	30
3.3.	Stichprobenziehung.....	31
3.3.1.	Grundgesamtheit und Erhebungszeitraum.....	31
3.3.2.	Rekrutierung von Schulklassen	31
3.3.3.	Paareinteilung und -auswahl.....	32
3.4.	Forschungsethik	34
3.5.	Kontext der Studie	35

3.5.1.	Ablauf eines Kurshalbtages	36
3.5.2.	Kontextorientierung der Lernwerkstatt.....	37
3.5.3.	Aufgabenstellungen der Lernwerkstatt	38
3.5.4.	Untersuchungsbedingungen in der Lernwerkstatt.....	40
3.5.5.	Evaluation der Lernumgebung.....	45
3.6.	Erhebungsinstrumente	46
3.6.1.	Beobachtungsauftrag	46
3.6.2.	Codebuch	48
3.7.	Datenauswertung.....	51
3.7.1.	Datenbereinigung	52
3.7.2.	Intercoder-Reliabilität	53
3.7.3.	Umstrukturierung des Datensatzes	56
3.7.4.	Beschreibung der Gruppen	57
3.7.5.	Berechnung und Beschreibung der unabhängigen und abhängigen Variablen	58
3.7.6.	Berechnung und Beschreibung weiterer Variablen	61
3.7.7.	Prüfung des Einflusses möglicher Störvariablen.....	63
4.	Ergebnisse	68
4.1.	Lösegeschwindigkeit nach Geschlechterzusammensetzung.....	68
4.2.	Anzahl angeforderte Hilfestellungen nach Geschlechterzusammensetzung.....	70
4.3.	Anzahl Eingriffe in geschlechtsheterogenen Gruppen	71
4.4.	Dauer der Eingriffe in geschlechtsheterogenen Gruppen	72
4.5.	Anzahl Rollenwechsel nach Geschlechterzusammensetzung.....	73
4.6.	Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung	74
4.7.	Summe der Dauer der Eingriffe nach Geschlechterzusammensetzung	77
5.	Diskussion und Schlussfolgerungen.....	79
5.1.	Diskussion der Forschungsfragen und Hypothesen.....	79
5.1.1.	Lösegeschwindigkeit nach Geschlechterzusammensetzung.....	79
5.1.2.	Anzahl angeforderte Hilfestellungen nach Geschlechterzusammensetzung ...	81
5.1.3.	Befolgen des Pair Programming Leitfadens in geschlechtsheterogenen Gruppen	

5.1.4. Unterschiede der Geschlechterzusammensetzungen bezüglich Befolgens des Pair Programming Leitfadens	84
5.2. Diskussion der Methode	87
5.3. Schlussfolgerungen und Forschungsausblick.....	92
6. Literaturverzeichnis.....	95
Anhang	105

Abbildungsverzeichnis

Abbildung 1 Ablauf eines Kurshalbtages	37
Abbildung 2 Raumplan der Lernumgebung	40
Abbildung 3 Foto der Lernumgebung	40
Abbildung 4 Blickwinkel einer Kamera.....	41
Abbildung 5 Einrichtung der Arbeitsplätze.....	42
Abbildung 6 PowerPoint-Folie zur Pair Programming Einführung	43
Abbildung 7 Box-Plot zur Lösegeschwindigkeit nach Klassenstufe (n=185).....	65
Abbildung 8 Box-Plot zur Summe der Anzahl Eingriffe der Lernenden nach Klassenstufe (n=160)	65
Abbildung 9 Box-Plot zur Anzahl angeforderte Hilfestellungen nach Leistungsunterschied der Lernenden (n= 188).....	66
Abbildung 10 Box-Plot zur Anzahl Hilfestellungen nach Unterschied in den Programmiervorkenntnissen der Lernenden einer Gruppe (n=195).....	67
Abbildung 11 Box-Plot zur Lösegeschwindigkeit nach Geschlechterzusammensetzung (n= 185).....	68
Abbildung 12 Box-Plot zur Lösegeschwindigkeit nach Geschlechterzusammensetzung auf der Oberstufe (n= 54).....	69
Abbildung 13 Box-Plot zur Anzahl Hilfestellungen nach Geschlechterzusammensetzung (n= 203).....	70
Abbildung 14 Box-Plot zur Anzahl Eingriffe nach Geschlecht in geschlechtsheterogenen Gruppen (n=129).....	71
Abbildung 15 Box-Plot zur durchschnittlichen Dauer eines Eingriffs nach Geschlecht in geschlechtsheterogenen Gruppen (n= 97).....	72
Abbildung 16 Boxplot zur Anzahl Platztäusche (Differenz zum Sollwert) nach Geschlechterzusammensetzung (n= 176).....	73
Abbildung 17 Box-Plot zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung (n= 160).....	75

Abbildung 18 Box-Plot zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung auf der Oberstufe (n=50).....	76
Abbildung 19 Box-Plot zur Anzahl Eingriffe der Jungen in geschlechtsheterogenen und -homogenen Gruppen auf der Oberstufe (n=53).....	77
Abbildung 20 Box-Plot zur Summe der durchschnittlichen Dauer eines Eingriffs nach Geschlechterzusammensetzung (n= 95).....	78

Tabellenverzeichnis

Tabelle 1 Neue Variablen aus Codier-Notizen	52
Tabelle 2 Intraklassenkorrelationskoeffizient.....	56
Tabelle 3 Verteilung der gefilmten Gruppen auf die Klassenstufen.....	58
Tabelle 4 Verteilung der gefilmten Gruppen auf die Geschlechterzusammensetzungen	58
Tabelle 5 Deskriptive Statistik zur Lösegeschwindigkeit der Geschlechterzusammensetzungen (n= 185).....	69
Tabelle 6 Deskriptive Statistik zur Lösegeschwindigkeit der Geschlechterzusammensetzungen auf der Oberstufe (n=54).....	70
Tabelle 7 Deskriptive Statistik zur Anzahl Hilfestellungen der Geschlechterzusammensetzungen (n=203).....	71
Tabelle 8 Deskriptive Statistik zur Anzahl Eingriffe der Jungen und Mädchen in geschlechtsheterogenen Gruppen (n=129).....	72
Tabelle 9 Deskriptive Statistik zur durchschnittlichen Dauer eines Eingriffs nach Geschlecht in geschlechtsheterogenen Gruppen (n=97).....	72
Tabelle 10 Deskriptive Statistik zur Anzahl Platztäusche (Differenz zum Sollwert) nach Geschlechterzusammensetzung (n= 176).....	73
Tabelle 11 Verteilung der Geschlechterzusammensetzungen auf zu wenig, genau richtig oder zu viele Platztäusche (n= 176).....	74
Tabelle 12 Deskriptive Statistik zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung (n= 160).....	75
Tabelle 13 Deskriptive Statistik zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung auf der Oberstufe (n= 50).....	76
Tabelle 14 Deskriptive Statistik zur Anzahl Eingriffe der Jungen in geschlechtsheterogenen und -homogenen Gruppen auf der Oberstufe (n= 53).....	77
Tabelle 15 Deskriptive Statistik zur Summe der durchschnittlichen Dauer eines Eingriffs nach Geschlechterzusammensetzung (n= 95).....	78

1. Einleitung

Das Berufsfeld der Informatik wird häufig als sehr technologieorientiert, kompetitiv und wenig sozial beschrieben (Cheryan et al., 2011; Diekman et al., 2010; Werner et al., 2004; Choi, 2015). Mit Pair Programming, auch Paarprogrammierung oder Tandem-Programmierung genannt, kann diesem Bild entgegengewirkt werden. Bei dieser Praktik programmieren zwei Personen gemeinsam und übernehmen die Rollen Driver (auch Pilot genannt) und Navigator (auch Kopilot oder Observer genannt) (Beck, 1999). Die Idee des Pair Programmings stammt ursprünglich aus der Praxis, hat aber schnell Einzug in die Bildungs- und Forschungswelt gefunden. Auch für Lernende der Primar- und Sekundarstufe wird der Einsatz von Pair Programming empfohlen (Campe et al., 2020). Die kurz- und langfristigen Leistungen, der Lerneffekt, die Zufriedenheit und vieles mehr können durch diese Technik gesteigert werden (Umapathy & Ritzhaupt, 2017; Tunga & Tokel, 2018). Ausserdem wird vermutet, dass insbesondere Frauen vom Pair Programming profitieren (Hanks et al., 2011; Werner et al., 2004; Li et al., 2013; Liebenberg et al., 2012).

Damit Pair Programming erfolgreich ist, müssen die Regeln und Vorgehensweisen dieser Technik eingehalten werden (Çal & Can, 2020). Diese sollen unter anderem sicherstellen, dass beide Personen gleichberechtigt partizipieren und kollaborieren können. Ebenfalls entscheidend ist, dass die zwei Personen kompatibel sind (Hanks et al., 2011; Tunga & Tokel, 2018; Bowman et al., 2020). Deshalb beschäftigen sich viele Studien mit der optimalen Einteilung der Paare für das Pair Programming. Dabei wird auch untersucht, ob sich die Geschlechterzusammensetzung der Paare (geschlechtshomogen weiblich, geschlechtsheterogen, geschlechtshomogen männlich) auf die Zusammenarbeit beim Pair Programming auswirkt.

Studien aus der Psychologie, der Verhaltensforschung und Kognitionswissenschaft können einen signifikanten Einfluss des Geschlechts auf die menschliche Interaktion und Kollaboration nachweisen (Carli, 1989; Sutter et al., 2009; Underwood et al., 1990). Wird mit dem anderen Geschlecht zusammengearbeitet, kann der Druck entstehen, sich geschlechterkonform zu verhalten (Maccoby, 1990; Flore & Wicherts, 2015). Mit dem Hintergrundwissen, dass Informatik in der Gesellschaft als typisch männlich gilt, könnten in geschlechtsheterogenen Gruppen die Jungen dominieren, während sich die Mädchen zurückhalten. Die Resultate der wenigen empirischen Studien in diesem Kontext sind nicht einheitlich, weshalb noch Forschungsbedarf besteht. Im Rahmen eines halbtägigen Robotik-Kurses für Primar- und Sekundarschulklassen in der Lernwerkstatt der Pädagogischen Hochschule Luzern wird mit dieser Videostudie untersucht, inwiefern

sich die Zuteilung in geschlechtshomogene und -heterogene Paare auf die Zusammenarbeit der Lernenden auswirkt.

Diese Masterarbeit beginnt mit dem theoretischen Hintergrund, in welchem die für die Studie relevanten Begriffe, Theorien und Forschungsergebnisse vorgestellt werden. Darauf folgt das Methodenkapitel, in welchem zuerst die Forschungsfragen und Hypothesen aus der Theorie abgeleitet und begründet werden und anschliessend das Forschungsdesign beschrieben wird. Der Kontext der Studie, die Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken*, welche von der Autorin dieser Masterarbeit mitentwickelt wurde, wird im Methodenkapitel genauer vorgestellt. Ebenfalls in diesem Kapitel werden die Erhebungsinstrumente und die Datenauswertung beschrieben. Darauf folgt der Ergebnisteil, in welchem die Hypothesen mit den gesammelten Daten geprüft werden. Den Abschluss dieser Masterarbeit bildet das Kapitel *Diskussion und Schlussfolgerungen*, in welchem die Forschungsfragen beantwortet, die Methode diskutiert, ein Fazit gezogen und ein Forschungsausblick gemacht wird. Im Anhang dieser Arbeit befinden sich neben den Erhebungsinstrumenten die Einwilligungserklärung für die Videostudie, zusätzliche statistische Auswertungen und die Verlinkung auf den Datensatz.

2. Theoretischer Hintergrund

In diesem Kapitel werden zuerst die zwei Hauptbegriffe dieser Videostudie, Gender und Pair Programming, genauer betrachtet. Anschliessend werden Studien vorgestellt, welche sich mit den Effekten des Einsatzes von Pair Programming beschäftigen. Danach werden Faktoren präsentiert, welche laut der Forschungsliteratur den Erfolg des Pair Programmings beeinflussen können. Im darauffolgenden Kapitel geht es um die Gleichberechtigung der Partizipation beim Pair Programming. Abschliessend wird der Einsatz von Pair Programming in einem Robotik-Kurs genauer betrachtet.

2.1. Begriff Gender

In diesem Kapitel wird der englische Begriff Gender, welcher nicht mit der deutschen Übersetzung Geschlecht gleichgesetzt werden kann, genauer betrachtet. Im ersten Unterkapitel werden eine Definition und allgemeine theoretische Hintergründe des Begriffes Gender vorgestellt. Danach geht es spezifisch um die Rolle von Gender im Informatikbereich.

2.1.1. Definition und allgemeine theoretische Hintergründe

Die Wirklichkeit ist aus konstruktivistischer Sicht ein Resultat von Konstruktionsprozessen. Diese Prozesse werden von der Umwelt bedingt, sind nicht steuerbar, und laufen unbewusst ab. Die durch diese Prozesse von den Menschen konstruierte Wirklichkeit unterscheidet sich von der unerkennbaren Realität (Pörksen, 2015; Schmidt, 1994; Weber, 2010).

Konstruktivistische Stimmen kritisieren, dass Geschlecht als Dualität (Frau-Mann), von der Natur vorgegeben und als unveränderlich verstanden wird. Die Geschlechterforschung unterscheidet in diesem Kontext die Begriffe Sex und Gender. Unter Sex ist das biologische Geschlecht, unter Gender seine gesellschaftliche Interpretation zu verstehen (Lünenborg & Maier, 2013). Gender ist ein kulturelles und soziales Konstrukt, welches dadurch entsteht, dass die Gesellschaft Geschlechterrollen, sowie geschlechtstypisches Verhalten und Charakteristika festlegt (Satz, 2012). Laut der Theorie der sozialen Rolle von Alice Eagly (Eagly, 1987; Eagly et al., 2000) nehmen Menschen beispielsweise an, dass Männer und Frauen Merkmale aufweisen, die für ihre jeweiligen sozialen Rollen, insbesondere Familien- und Berufsrollen, typisch sind. Dabei werden Frauen überwiegend in Hausfrauenrollen oder Berufsrollen mit niedrigem Status (z.B. Grundschullehrerin, Krankenschwester), Männer in Ernährer- bzw. Berufsrollen mit eher hohem Status (z.B. Manager, Rechtsanwalt) gesehen (Eckes, 2008). Solche Geschlechterstereotype werden durch Sozialisation, insbesondere durch Medien und Kultur, konstruiert (Ströhl, 2014).

Soziale Erwartungen stabilisieren das Konstrukt Gender, da sich Frauen und Männer als solche wahrnehmen und damit ihre Geschlechtsidentitäten bestätigen (Pörksen, 2015; Weber, 2010). Die Geschlechtsidentität ist ein Teilsystem der menschlichen Identität (Doctor, 1988), welche sich auf das Zugehörigkeitsgefühl zu einem Geschlecht bezieht (Deaux & LaFrance, 1998). Dabei kann ein Individuum mit dem eigenen Körper (seinen primären und sekundären Geschlechtsmerkmalen), sowie der sozialen Rolle, welche diesem Geschlecht zugeschrieben wird, übereinstimmen oder nicht. Entsprechend der eigenen Geschlechtsidentität ordnet ein Individuum sich selbst und andere Menschen den in der Gesellschaft typischen Verhaltensweisen, Eigenschaften und Rollen der Geschlechter zu. Mit einem Verhalten entsprechend dieser Geschlechterrollen kann die eigene Geschlechtsidentität gezeigt werden (Bosinski, 2000).

Studien aus der Psychologie, Verhaltensforschung und Kognitionswissenschaften können einen signifikanten Einfluss des Geschlechts auf die menschliche Interaktion und Kollaboration nachweisen (Carli, 1989; Sutter et al., 2009; Underwood et al., 1990). Maccoby (1998) argumentiert, dass geschlechtsspezifische Aspekte des Verhaltens durch das Geschlecht des Gegenübers beeinflusst werden können. Wenn Jungen anwesend sind, stehen Mädchen mehr unter Druck ihre Geschlechtsidentität zu bewahren, indem sie sich geschlechterkonform verhalten, und umgekehrt (Maccoby, 1990; Brutsaert, 1999; Amon, Bartosch, Lembens & Wenzl 2012; Kröll, 2010; von Ow & Husfeldt, 2011; Wedl & Bartsch, 2015).

2.1.2. Rolle von Gender in der Informatik

Stereotype über Informatikerinnen und Informatiker beinhalten unter anderem, dass sie sehr technologieorientiert (Cheryan et al., 2011), wenig interessiert an anderen Menschen (Diekman et al., 2010), sozial ungeschickt (Mercier et al., 2006), intelligent (Beyer et al., 2003) und unattraktiv (Mercier et al., 2006) sind. Dazu kommt, dass die meisten Menschen Informatik mehr mit Männern als mit Frauen verbinden (Nosek et al., 2009; Diekman et al., 2011). Solche Stereotype können sowohl bei Männern als auch bei Frauen nachgewiesen werden (Singh et al., 2007), und treten bereits bei Primarschulkindern auf (Mercier et al., 2006). Noch jüngere Kinder haben dagegen weniger stereotypische Vorstellungen zur Informatik, was darauf hinweist, dass diese Stereotype in den späten Primarschuljahren gelernt werden (Barman, 1999). Informatikerinnen und Informatiker werden in den Medien, beispielsweise in der Serie *The Big Bang Theory*, häufig stereotyp als Computerfreaks dargestellt, was die Konstruktion und Aufrechterhaltung dieser Stereotype unterstützt (Mercier et al., 2006).

Bei der Karrierewahl vergleicht man sich typischerweise selbst mit Personen, welche in diesem Bereich arbeiten, um herauszufinden, ob man dazu passt und ob man dort erfolgreich sein könnte (Creamer et al., 2007). Deshalb hat der Eindruck, den man von diesem Bereich hat, einen grossen Einfluss darauf, ob man selbst Interesse hat, dort einzusteigen. Verschiedene Studien können nachweisen, dass Frauen eine negativere Einstellung gegenüber Informatik haben als Männer (Başer, 2013; Chang et al., 2012; Jarratt et al., 2019). Das stereotype Bild von Informatikerinnen und Informatikern ist männlich geprägt und beinhaltet Eigenschaften, welche mit der weiblichen Geschlechterrolle nicht vereinbar sind (Cheryan et al., 2013; Eagly, 1987). Beispielsweise wird von Frauen erwartet, dass sie sozial kompetent und menschenbezogen sind (Diekman et al., 2010). Dazu kommt, dass Frauen weniger Vertrauen in ihre informatikbezogenen Fähigkeiten haben, weshalb sie es sich häufig nicht zutrauen, in diesem als anspruchsvoll wahrgenommenen Beruf zu arbeiten (Beyer et al., 2003; Maguire et al., 2014; Campe et al., 2020; Fraunhofer IAIS, 2019; Jarratt et al., 2019; Coto & Mora, 2019). Mehrere Studien zeigen jedoch, dass es keine Leistungsunterschiede der Geschlechter in der Informatik gibt, sei dies an der Universität (Akinola, 2016), auf der Primar- (Iskrenovic-Momcilovic, 2019) oder Sekundarstufe (Papadakis, 2018). Trotzdem sind Frauen sowohl bei der Studien- als auch bei der Berufswahl in den Bereichen Technik und Informatik unterrepräsentiert (Bundesamt für Statistik, 2017a; Bundesamt für Statistik, 2017b). Viele Frauen glauben, dass es sich bei informatischen Karrieren um keine sozialen Berufe (Werner et al., 2004; Liebenberg et al., 2012; Ying et al., 2019) und um ein kompetitives Arbeitsumfeld (Werner et al., 2004; Choi, 2015) handelt. Frauen schätzen dagegen personenorientierte und kollaborative Karrieren (Ying et al., 2019). Zweifelt eine Person daran, ob sie in die Informatikwelt passt, kann dies den Mut in dieses Feld einzusteigen und das eigene Interesse an diesem Thema abschwächen (Diekman et al., 2010; Diekman et al., 2011; Cheryan et al., 2013).

2.2. Begriff Pair Programming

Beim Pair Programming programmieren zwei Personen gemeinsam, und übernehmen dabei die Rollen Driver und Navigator. Die Person in der Rolle Driver bedient Tastatur und Computermaus, diejenige in der Rolle Navigator denkt mit, überprüft und hilft, ohne die Tastatur, den Bildschirm oder die Computermaus zu berühren. Nach einer festgelegten Zeit werden die Rollen gewechselt. Obwohl nur jemand den Computer bedient, wird gemeinsam entschieden was programmiert wird. Dafür ist eine gute Kommunikation wichtig (Hanks et al., 2011; Denner et al., 2014; Werner & Denning, 2009). Pair Programming ist mittlerweile sowohl in der Arbeitswelt als auch in der Bildung

weit verbreitet und hat dadurch auch einen grossen Stellenwert in der Forschung erhalten (Gómez et al., 2017).

In diesem Kapitel wird zuerst vorgestellt, woher die Praktik Pair Programming kommt. Danach werden Pair Programming Leitlinien aus der Forschung vorgestellt, welche die Effektivität dieser Praktik steigern sollen.

2.2.1. Herkunft der Praktik Pair Programming

Pair Programming ist eine der zwölf Praktiken des Extreme Programmings (XP, auch Extremprogrammierung genannt), einem Ansatz der Softwareentwicklung von Kent Beck (1999). Entstanden ist die Idee in den 1990er Jahren bei einem gemeinsamen Projekt von Kent Beck, Ward Cunningham und Ron Jeffries. Das Problem, dass die Anforderungen der Kundinnen und Kunden an eine Software zu Beginn noch nicht ganz klar sind und das Endprodukt deshalb häufig nicht alle Wünsche der Kundschaft erfüllt, soll mit ihrem Vorgehensmodell gelöst werden. Beim Extreme Programming wird die Programmieraufgabe in kleine Schritte aufgeteilt, wobei nach jedem Teilabschluss Rücksprache mit den Kundinnen und Kunden gehalten wird. So kann schnell auf neue Anforderungen reagiert werden (Pröpfer, 2012). Extreme Programming kombiniert Verfahren, welche von vielen erfolgreichen Programmierinnen und Programmierern bereits eingesetzt wurden, zuvor aber wenig Beachtung in der Literatur über Softwareprozesse und -methoden erhielten (Beck, 2003).

2.2.2. Pair Programming Leitlinien

Damit Pair Programming erfolgreich ist, müssen die Rollen und Vorgehensweisen dieser Technik klar formuliert und kommuniziert werden (Çal & Can, 2020). Hierfür haben verschiedene Forschende Leitlinien formuliert. Sehr häufig eingesetzt und zitiert wird dabei ein Artikel von Williams und Kessler (2000). Ihre Leitlinien basieren auf einer Umfrage, an welcher Berufspraktikerinnen und -praktiker teilnahmen, welche Pair Programming praktizieren. Nachfolgend werden die Pair Programming Leitlinien von Williams und Kessler kurz zusammengefasst (2000):

1. Das Paar entwickelt gemeinsam ein Artefakt, für welches beide verantwortlich sind. Die zwei Personen sind gleichwertige Mitgestaltende dieses Prozesses, und können deshalb nicht das Gegenüber für einen Fehler verantwortlich machen.
2. Auch wenn eine Person mehr Programmiererfahrung hat, ist es wichtig, dass die Rollen regelmässig gewechselt werden. Die Person in der Navigatoren-Rolle beobachtet nicht passiv, sondern ist immer aktiv und engagiert, und hat die Gesamtstrategie im Blick.
3. Man soll sicherstellen, dass die Partnerin oder der Partner fokussiert und aufgabenorientiert bleibt. Bei der Arbeit zu zweit wird weniger Zeit für das Lesen von Mails, Surfen im Internet oder ähnlichem verschwendet, weil die andere Person Input oder Kooperation erwartet.

4. Redet man sich selbst ein, dass man schlechter programmieren kann als das Gegenüber, führt dies zu Unsicherheiten, welche sich negativ auf die Kooperation auswirken. Diese negative Einstellung gegenüber den eigenen Programmierfähigkeiten, sowie eine negative Wahrnehmung der Fähigkeiten der Partnerin oder des Partners sollen verworfen werden.
5. Das Pair Programming ermöglicht das Aufdecken von offensichtlichen, aber sonst lang unbeachteten Fehlern durch eine andere Person. Diese Fehlervermeidung und effiziente -beseitigung sind Erfolgsfaktoren des Pair Programmings, und sollen genutzt werden.
6. Ein übermässiges Ego schadet der Kollaboration. Es kann dazu führen, dass die Ideen des Gegenübers nicht berücksichtigt, oder als Kritik wahrgenommen werden. Für einen regen Ideenaustausch darf es aber auch gesunde Meinungsverschiedenheiten und Debatten geben. Hierfür muss eine gute Balance gefunden werden.
7. Eine angemessene Arbeitsplatzgestaltung ist entscheidend für den Erfolg des Pair Programmings. Das Paar soll Seite an Seite sitzen, beide müssen den Bildschirm sehen und Tastatur und Maus teilen können. Die Stühle dürfen nicht bewegt werden. Mit der anderen Person, sowie weiteren Paaren, soll ohne grossen Aufwand kommuniziert werden können. Der Kontakt zu anderen Gruppen kann sich sehr positiv auf das Pair Programming auswirken.
8. Viele Programmiererinnen und Programmierer sind bereits vor dem Start skeptisch gegenüber dem Wert der Zusammenarbeit beim Programmieren. Sie erwarten nicht, von dieser Erfahrung zu profitieren oder sie zu geniessen. Eine eigene positive Einstellung und die der Partnerin oder des Partners sind entscheidend für den Erfolg des Pair Programmings. Vor dem Start soll die Skepsis weggewaschen und eine Erfolgserwartung entwickelt werden.
9. Meist erarbeiten Paarprogrammiererinnen und -programmierer auch unabhängig voneinander etwas. Diese Arbeiten dürfen nicht ohne Überprüfung in das gemeinsame Projekt aufgenommen werden, da sie meist Fehler beinhalten. Gewisse Programmiererinnen und Programmierer schreiben diese Teile sogar gemeinsam neu.
10. Paarprogrammiererinnen und -programmierer halten sich gegenseitig kontinuierlich fokussiert an der Aufgabe. Regelmässige Pausen sind wichtig, um die Ausdauer für weitere Programmerrunden aufrecht zu erhalten. Die Aufgabe soll in der Pause nicht besprochen werden, damit anschliessend mit neuer Frische gestartet werden kann.
11. Informelle Diskussionen mit anderen Programmiererinnen und Programmierern ermöglichen einen effektiven Ideenaustausch und Wissenstransfer.
12. Pausen von der Zusammenarbeit können sich positiv auswirken. Es ist nicht notwendig, jeden Nachmittag separat zu arbeiten. Gewisse Aufgaben eignen sich aber für die selbstständige Erarbeitung, beispielsweise wenn sie eine hohe Konzentration voraussetzen.
13. Beim Pair Programming werden zwei Programmierende zu einem Team. Es darf keinen Wettbewerb zwischen ihnen geben, sie müssen auf das gleiche Ziel hinarbeiten, als ob das Artefakt von einer Person produziert würde. Man muss auf das Urteil des anderen und seine Loyalität vertrauen.
14. Ein Mensch kann nur begrenzt lernen. Schliesst man sich mit jemandem zusammen, kann mehr erreicht werden. Jede Person hat ein eigenes Wissens- und Fähigkeitsset. Die Grundlagen sind dieselben, nur so kann effektiv miteinander interagiert werden. Doch die einzigartigen Fähigkeiten ermöglichen es, Ressourcen zu bündeln, und Aufgaben effizienter zu erledigen. Zu zweit werden mehr Lösungsvorschläge produziert, die beste Lösung wird schneller gefunden und mit höherer Qualität implementiert. Die Paarprogrammierenden sollen sich der Kraft zweier Gehirne bewusst sein.

Werner et al. (2004) befassen sich in ihrem Artikel damit, wie Pair Programming in der Schule eingesetzt werden kann, um insbesondere Mädchen zu fördern. Dabei fassen sie ihre Erfahrungen, welche sie im Rahmen eines ausserschulischen Programmierkurses für Mädchen der Mittelstufe, sowie bei Kursen für Universitätsstudierenden sammeln konnten, zusammen. Ausserdem beziehen sie sich auf die bereits vorgestellten Leitlinien von Williams und Kessler (2000). Das Ziel von Werner et al. (2004) war, Verhaltensweisen zu identifizieren, welche essenziell für ein effektives Pair Programming in der Schule sind. Dabei heben sie beispielsweise hervor, dass die Nutzungszeit von Maus und Tastatur fair aufgeteilt werden muss, und die Person in der Navigatoren-Rolle eine aktive Rolle spielen und der Person in der Driver-Rolle mit Zeigen Hinweise geben soll. Ebenfalls wichtig ist, dass Entscheidungen gemeinsam besprochen und gefällt werden, immer nach der Zustimmung des Gegenübers gefragt wird, und Wörter wie *wir* und *uns* verwendet werden. Die Forschenden weisen ausserdem darauf hin, dass nicht nur die Lernenden innerhalb eines Paares, sondern auch die Lehrperson und die anderen Paare die Effektivität von Pair Programming beeinflussen. Eine klare Instruktion des Pair Programmings, sowie Unterstützung und positives Feedback von anderen Lernenden und der Lehrperson sind ebenfalls wichtig. Nachfolgend werden die Leitlinien für das Pair Programming von Werner et al. kurz vorgestellt (2004):

1. Die Person in der Driver-Rolle bedient Tastatur und Maus.
2. Die Person in der Navigatoren-Rolle beobachtet, was die Person in der Driver-Rolle auf dem Bildschirm macht, und verhindert Fehler. Die Person in der Navigatoren-Rolle ist für die Referenzmaterialien zur Programmiersprache zuständig.
3. Positionen: Die Person in der Driver-Rolle sitzt vor der Tastatur und hat die Computermaus in der Hand. Die Maus wird so weit entfernt wie möglich von der Person in der Navigatoren-Rolle platziert. Der Bildschirm ist so angewinkelt, dass beide einen guten Blick darauf haben, und die Person in der Navigatoren-Rolle darauf zeigen kann. Bei der Person in der Navigatoren-Rolle werden die Referenzmaterialien platziert.
4. Bei einem Rollenwechsel stehen die Lernenden auf und wechseln ihre Positionen.
5. Die Lernenden geben auf sich Acht, sehen sich an, wenn sie miteinander sprechen und hören dem Gegenüber gut zu.
6. Es wird darauf geachtet, dass beide verstehen, an was gerade gearbeitet wird. Die Person in der Driver-Rolle beschreibt die eigenen Aktionen und wartet auf Zustimmung, die Person in der Navigatoren-Rolle zeigt auf den Bildschirm, um die eigenen Vorschläge zu erklären. Die Person in der Navigatoren-Rolle hält die Schritte der Programmierung in einem Notizbuch fest, und zeigt dieses regelmässig dem Gegenüber. Fehlt eine Person während einer Sitzung, erhält diese im darauffolgenden Kurs von der anderen Person eine Beschreibung der Arbeit, welche während der Absenz gemacht wurde, und startet dann in der Driver-Rolle.
7. Es wird respektvoll miteinander umgegangen. Die Person in der Navigatoren-Rolle berührt weder Computermaus noch Tastatur, die Person in der Driver-Rolle greift nicht nach den Referenzmaterialien. Meinungsverschiedenheiten gehören dazu, müssen aber respektvoll gelöst werden.
8. Die Verantwortung und die Autorenschaft über das Projekt werden geteilt.

9. Die Lernenden helfen sich gegenseitig, nutzen Möglichkeiten, von der anderen Person zu lernen und fördern den Wechsel der Pair Programming Rollen.

Williams et al. (2008) befassen sich in ihrem Artikel mit der Implementierung von Pair Programming im akademischen Kontext. Die Forschenden sammelten während sieben Jahren Erfahrungen über den Einsatz von Pair Programming an einer Universität. Sie entwickelten Leitlinien fürs Pair Programming, integrierten diese in einen Bachelorkurs, und überarbeiteten sie. Daraus entstanden elf Leitlinien, welche nachfolgend zusammengefasst werden:

1. Die Studierenden müssen für das Pair Programming in einer beaufsichtigten Umgebung geschult werden, um die Mechanismen einer erfolgreichen Zusammenarbeit kennenzulernen. Nach einer Einführung ins Pair Programming muss die Kursleitung sicherstellen, dass die Studierenden die Rollen richtig annehmen, diese regelmässig wechseln und durchgehend aktiv sind.
2. Die Kursleitung muss die Paare proaktiv besuchen, nachfragen wie es läuft und sicherstellen, dass effizient zusammengearbeitet wird. Paarzusammenstellungen, welche nicht gut funktionieren, müssen identifiziert und aufgelöst werden. Ebenfalls muss sichergestellt werden, dass die Rollen regelmässig gewechselt werden.
3. Strikte Anwesenheits- und Verspätungsregeln sind nötig, um Studierende von nicht-partizipierenden Partnerinnen und Partnern zu schützen. Kommt eine Person zu spät, muss sie allein arbeiten und erhält einen Notenabzug.
4. Wenn die Studierenden ausserhalb eines Klassenzimmersettings zu zweit programmieren, soll die Kursleitung Feedback der Studierenden zu ihren Partnerinnen und Partnern einfordern und auf ungleiche Partnerschaften reagieren. Dadurch kann Trittbrettfahren verhindert werden.
5. Sowohl individuelle als auch kollaborative Leistungen der Studierenden müssen bewertet werden. So kann sich niemand nur auf die Partnerin oder den Partner verlassen.
6. Bei der Einteilung der Paare soll die Kursleitung versuchen, die Wahrscheinlichkeit zu erhöhen, dass die Studierenden gut zusammenarbeiten können. Verschiedene Faktoren wie das Fähigkeitslevel, die Arbeitsethik oder Persönlichkeitstypen müssen bei der Einteilung miteinbezogen werden.
7. Die Studierenden sollen während des Semesters mit unterschiedlichen Partnerinnen und Partner zusammenarbeiten. So haben die Studierenden die Möglichkeit, mehr Personen kennenzulernen. Ausserdem sind Studierende weniger wahrscheinlich intolerant gegenüber einer Partnerin oder eines Partners, wenn sie wissen, dass ihre Zusammenarbeit zeitlich begrenzt ist.
8. Die Studierenden sollen ermutigt werden, Probleme im Paar so schnell wie möglich zu melden. Partizipiert die Partnerin oder der Partner nicht, erhält die andere Person eine Entschädigung, aber nur, wenn das Problem früh gemeldet wird.
9. Die Paare sollen bequem nebeneinandersitzen und beide müssen Zugang zum Monitor, zur Maus und zur Tastatur haben. Empfohlen wird, dass der Computer mit zwei Monitoren, zwei Mäusen und zwei Tastaturen ausgerüstet ist. Die Studierenden sollen bei einem Rollenwechsel nicht die Stühle wechseln müssen.
10. Die Paare müssen auf ein gemeinsames Ziel hinarbeiten.
11. Die Kursleitung soll die Paare ermutigen, die Lösung selbst zu finden, statt ihnen diese einfach zu geben. Es sollen nur allgemeine Hinweise gemacht werden, wobei die Paare mit Gegenfragen gelenkt werden können. Dadurch wird das Selbstvertrauen der Studierenden in ihre Fähigkeiten gestärkt.

Zarb et al. (2013) leiteten für ihren Artikel aus Videos von Berufspraktikerinnen und -praktikern beim Pair Programming Grundsätze für eine effiziente Kommunikation beim gemeinsamen Programmieren ab, und prüften die Wirksamkeit dieser Empfehlungen empirisch. Die Anwendung dieser Leitlinien führt bei Studierenden dazu, dass sich Novizen beim Kommunizieren im Paar wohler fühlen. Ausserdem steigert sich die wahrgenommene Beteiligung der Partnerin oder des Partners. Die Aufgaben werden jedoch nicht schneller gelöst (Zarb, Hughes & Richards, 2014; Zarb, Hughes & Richards, 2015). Da sich diese Leitlinien nur auf die Kommunikation beim Pair Programming beziehen, sind sie sehr spezifisch. Diese bei Schülerinnen und Schülern beim ersten Kontakt mit Pair Programming einzuführen, könnte überfordern. Deshalb werden die Leitlinien von Zarb et al. (2013) an dieser Stelle nicht genauer vorgestellt, die Lektüre ihres Artikels wird aber empfohlen.

2.3. Effekte des Pair Programmings

Seit der Vorstellung des Pair Programmings durch Kent Beck (1999) befassen sich viele Forschende damit, ob und auf welchen Stufen der Einsatz dieser Technik förderlich ist, und welche Effekte damit verbunden sind. Meist werden die Studienteilnehmenden dafür in zwei Gruppen eingeteilt. In der Kontrollgruppe wird allein programmiert, in der Experimentalgruppe entsprechend der Pair Programming Regeln zu zweit. In Metastudien werden zum Teil Ergebnisse von schulischen und akademischen Kontexten gemeinsam erfasst, weshalb die in den ersten zwei Unterkapiteln angestrebte Sortierung der Ergebnisse nach Stufe nicht ganz trennscharf ist. Forschungsergebnisse, welche sich auf den Arbeitsalltag und somit professionelle Informatikerinnen und Informatiker beziehen, werden in diesem Kapitel nur vereinzelt aufgeführt und klar als solche gekennzeichnet. Bisher wurden hauptsächlich Bachelor- oder Masterstudierende mit informatischem Hauptfach an einer Universität untersucht. Die meisten Studien beziehen sich auf Einführungskurse, an welchen Personen ohne grosses Vorwissen teilnehmen. Das Besuchen des Kurses ist meist freiwillig. Auch im schulischen Kontext gibt es wenige Studien, bei welchen die Teilnahme obligatorisch ist. Der Grossteil der Studien verwendet quantitative Methoden (Tunga & Tokel, 2018). Es gibt aber auch einzelne qualitative Studien mit kleineren Stichproben (bspw. Liebenberg et al., 2012).

In diesem Kapitel werden zuerst Forschungsergebnisse zu den Effekten des Pair Programmings im akademischen, und danach im schulischen Kontext, vorgestellt. Darauf folgt ein Zusatzkapitel mit weiteren Effekten des Pair Programmings, welche auf verschiedenen Stufen nachgewiesen werden konnten. Als letztes folgt eine Diskussion der Heterogenität der Resultate.

2.3.1. Pair Programming im akademischen Kontext

Folgend werden Forschungsergebnisse zu den Effekten des Einsatzes von Pair Programming im akademischen Kontext vorgestellt. Dabei werden die Resultate thematisch geordnet, beginnend mit dem Einfluss des Pair Programmings auf die Leistungen der Studierenden. Danach werden affektive Effekte des Pair Programmings vorgestellt. Abschliessend wird begründet, weshalb Pair Programming nicht nur einen Einblick in eine professionelle Praktik ermöglicht, sondern auch wichtige Fähigkeiten für den Berufsalltag fördert.

Leistungen der Studierenden

Forschungsergebnisse zum Einsatz von Pair Programming im akademischen Kontext weisen darauf hin, dass die Implementierung dieser Technik einen positiven Einfluss auf die Leistungen der Studierenden im informatischen Bereich haben kann. Studierende, welche zu zweit programmieren, erreichen in Informatikkursen bessere Noten bei Programmieraufgaben und Abschlussprüfungen (Umapathy & Ritzhaupt, 2017; Braught et al., 2011; Mohamed, 2017; Smith et al., 2017; Salleh et al., 2011) und bestehen den Kurs mit einer höheren Wahrscheinlichkeit (Umapathy & Ritzhaupt, 2017; Hanks et al., 2011; Werner et al., 2004; Braught et al., 2011; McDowell et al., 2006; Nawahdah & Taji, 2016). Auch bei nachfolgenden Programmierkursen haben Studierende, welche zuvor mit Pair Programming gearbeitet haben, eine höhere Erfolgswahrscheinlichkeit (McDowell et al., 2006; Smith et al., 2017) und behalten Informatik wahrscheinlicher als Hauptfach bei (Hanks et al., 2011; Werner et al., 2004; McDowell et al., 2006). Es gibt aber auch Studien, welche darauf hinweisen, dass sich die Leistungen von Paar- und Soloprogrammierenden nicht statistisch signifikant unterscheiden (Bowman et al., 2020; Demir und Seferoglu, 2021b; O'Donnell et al., 2015). Viele Studien konnten nachweisen, dass die Qualität des Codes von Paarprogrammierenden höher ist als diejenige von Soloprogrammierenden (Hanks et al., 2011; Salge & Berente, 2016; Tunga & Tokel, 2018; Anjum et al., 2015; Demir & Seferoglu, 2021b; Abd Gani et al., 2010; Isong et al., 2016; Nawahdah & Taji, 2016; Salleh, 2008; Salleh et al., 2011) und die Programme der Paare weniger Fehler beinhalten (Anjum et al., 2015; Braught, Wahls & Eby, 2011; Isong et al., 2016; Nawahdah & Taji, 2016). Studierende, welche zu zweit programmieren, weisen einen höheren Lerneffekt (Hanks et al., 2011; Salge & Berente, 2016; Tunga & Tokel, 2018; Faja, 2011) und bessere Programmierfähigkeiten (Tunga & Tokel, 2018; Braught et al., 2011; Salleh et al., 2011) auf. Lernende der Pair Programming Gruppe sind ausserdem häufiger anwesend (Nawahdah & Taji, 2016; O'Donnell et al., 2015; Radermacher & Walia, 2011). Dies lässt sich dadurch erklären, dass die Studierenden ihre Partnerin oder

ihren Partner nicht im Stich lassen möchten. Beobachtungen von Paarprogrammierenden weisen darauf hin, dass Paare schneller programmieren (Salge & Berente, 2016; Akour et al., 2013; Isong et al., 2016; Salleh, 2008; Salleh et al., 2011), effizienter sind (Tunga & Tokel, 2018) und kürzere Programme schreiben (Anjum et al., 2015). In einer Studie von Cockburn und Williams (2000) dauerte das Problemlösen zu zweit dagegen länger als allein.

Affektive Effekte

Entscheidend für eine Empfehlung des Einsatzes von Pair Programming im akademischen Kontext sind nicht nur die Wirkungen dieser Technik auf die Leistungen der Studierenden, sondern auch affektive Effekte. Mehrere Studien weisen darauf hin, dass Pair Programming zu einem grösseren Vertrauen in das selbstentwickelte Programm (Hanks et al., 2011; Werner et al., 2004; Braught et al., 2011; Faja, 2011; Abd Gani et al., 2010; Salleh, 2008; Faja, 2014; Mohamed, 2017; Salleh et al., 2014; Salleh et al., 2011) führt. Ausserdem weisen Paarprogrammierende eine positivere Einstellung gegenüber dem Programmieren und mehr Interesse daran auf (Tunga & Tokel, 2018; Mohamed, 2017). Während viele Studien nachweisen konnten, dass Studierende eine positivere Einstellung gegenüber Pair Programming als gegenüber Soloprogrammierung haben (Akour et al., 2013; Al-Ramahi et al., 2013; Isong et al., 2016; Faja, 2011), fanden Umapathy und Ritzhaupt (2017) in ihrer Studie diesbezüglich keinen statistisch signifikanten Unterschied. Viele Studierende berichten in Interviews oder Fragebögen, dass ihnen das Programmieren zu zweit mehr Spass macht oder sie zufriedener sind (Tunga & Tokel, 2018; Abd Gani et al., 2010; Salleh, 2008; Faja, 2014; Maguire et al., 2014; Nawahdah & Taji, 2016; Salleh et al., 2014; Salleh et al., 2011) und dass der Programmierprozess effizienter und produktiver sei (Ying et al., 2019; Mohamed, 2017). Einzelne Studierende beschreiben Pair Programming aber auch als ineffizient, langsam und weniger praktisch als die Soloprogrammierung (Faja, 2014; Ying et al., 2019). Viele Studierenden geben an, dass ihnen Pair Programming hilft, die Programmierkonzepte zu verstehen (Akour et al., 2013; Al-Ramahi et al., 2013), und sie durch diese Technik mehr lernen (Faja, 2014; Ying et al., 2019). Allerdings erwähnen einige Studierende, dass Pair Programming nur erfolgreich sein kann, wenn gewisse Faktoren, wie beispielsweise ähnliche Programmierfähigkeiten und gleicher Motivationslevel, bei den Paaren miteinbezogen werden. Eine hohe Paarkompatibilität scheint entscheidend für den Erfolg des Pair Programmings zu sein (Mohamed, 2017).

Förderung von wichtigen Fähigkeiten für den Berufsalltag

Neben den bereits genannten Vorteilen des Pair Programmings wird der kollaborative Aspekt dieser Technik in der Forschungsliteratur positiv hervorgehoben. Durch die Interaktion mit einer anderen Person wird Wissen ausgetauscht (Isong et al., 2016; Nawahdah & Taji, 2016). Ausserdem werden das kritische Denken sowie Kommunikations-, Team- und Problemlösefähigkeiten gefördert (Mohamed, 2017; Nawahdah & Taji, 2016; Bailey & Mentz, 2017; Faja, 2011; Salleh et al., 2011). Als Vorteil wird zusätzlich genannt, dass Studierende die Kommunikation und Interaktion eines professionellen Softwareteams kennenlernen können (Anjum et al., 2015; Faja, 2011).

2.3.2. Pair Programming im schulischen Kontext

Bisher gibt es noch wenige Studien, welche sich mit dem Pair Programming im schulischen Kontext befassen. Deshalb werden die Effekte des Einsatzes dieser Technik im schulischen Umfeld in diesem Kapitel nicht thematisch, sondern nach Schulstufe geordnet. Zuerst werden Studienresultate zu den Effekten des Einsatzes von Pair Programming auf der Primarstufe, und dann auf der Sekundarstufe vorgestellt.

Primarstufe

Verschiedene Studien mit Schülerinnen und Schülern im Primarschulalter berichten, ähnlich wie Studien im akademischen Bereich, dass Lernende, welche zu zweit programmieren, mehr Leistung erbringen (Çal & Can, 2020; Iskrenovic-Momcilovic, 2019). Ausserdem eignen sie sich durch Pair Programming mehr Selbstvertrauen beim Programmieren (Çal & Can, 2020), mehr Programmierwissen (Denner et al., 2014) und besseres Computational Thinking (Denner et al., 2014; Seo & Kim, 2016) an. Die Mehrheit der Primarschülerinnen und -schüler geben bei Interviews oder Fragebögen an, dass sie das Programmieren zu zweit bevorzugen (Çal & Can, 2020). Die Lernenden begründen diese Wahl beispielsweise damit, dass die Zusammenarbeit mit einer anderen Person ihr Lernen unterstützt, eine positive und unterhaltsame Lernatmosphäre generiert, und sie auf grössere Projekte vorbereitet (Celepkolu et al., 2020). Es gibt aber auch auf der Primarstufe Lernende, welche lieber allein arbeiten oder die Restriktionen des Pair Programmings, beispielsweise das Teilen eines Computers, nicht schätzen (Celepkolu et al., 2020). Die Lernenden erwähnen häufig, dass ihre Partnerin oder ihr Partner gewisse Qualitäten haben muss, damit ein reger Ideenaustausch und eine faire Arbeitsverteilung möglich sind. Beispielsweise fordern die Lernenden eine gleiche Einstellung zur Zusammenarbeit und ein ähnliches Wissenslevel (Celepkolu et al., 2020).

Sekundarstufe

Studien mit Lernenden im Sekundarschulalter berichten ebenfalls von positiven Effekten des Einsatzes von Pair Programming. Lernende, welche zu zweit programmieren, verstehen Programmierkonzepte besser, sind dem Programmieren gegenüber positiver eingestellt und haben mehr Interesse daran. Zudem weisen sie eine höhere Problemlösefähigkeit auf. Pair Programming wird von den Lernenden als spielerischer und angenehmer als die Soloprogrammierung beschrieben (Papadakis, 2018). Mädchen berichten davon, dass sie durch das Pair Programming mehr Freude am Fach Informatik gewonnen haben. Sie haben das Gefühl, dass sie die Aufgaben besser verstehen, mehr lernen und mehr Ausdauer haben, um die Probleme zu lösen. Ausserdem schätzen sie die Sozialisation, Kommunikation und Unterstützung beim Pair Programming (Liebenberg et al., 2012).

Die wenigen Studien zum Einsatz von Pair Programming im schulischen Kontext empfehlen, diese Technik auch auf der Primar- und Sekundarstufe einzusetzen. Forschende erwähnen aber, dass weitere Forschung nötig ist, um die Unterschiede zwischen professionellen Programmierenden, Studierenden und jungen Lernenden zu verstehen (Campe et al., 2020).

2.3.3. Effekte auf allen Stufen

Unabhängig von der Stufe der Studienteilnehmenden erwähnen viele Forschende Vorteile für schwächere Lernende und Frauen, sowie für die Kursleitung. Deshalb werden diese Ergebnisse nachfolgend gesammelt zusammengefasst.

Vorteile für schwächere Lernende und Frauen

Mehrere Forschende vermuten, dass insbesondere schwächere Lernende und Lernende ohne Vorwissen (Li et al., 2013; Maguire et al., 2014; O'Donnell et al., 2015; Smith et al., 2017; Denner et al., 2014), sowie Frauen vom Pair Programming profitieren (Hanks et al., 2011; Werner et al., 2004; Li et al., 2013; Liebenberg et al., 2012). Gewisse Studien weisen Vorteile des Pair Programmings nur oder stärker bei Frauen oder schwächeren Lernenden nach, beispielsweise die Steigerung der Produktivität und des Selbstvertrauens (Zhong et al., 2016), die Senkung der Ausstiegsrate (Li et al., 2013), die Verbesserung der Prüfungsnote oder der wahrgenommene Lerneffekt (Maguire et al., 2014). Schwächere Studierende bewerten Pair Programming positiver als Stärkere und geben häufiger an, in Zukunft wieder zu zweit programmieren zu wollen (O'Donnell et al., 2015). Mit Pair Programming kann weiblichen Studierenden gezeigt werden, dass Programmieren auch eine kollaborative und soziale Aufgabe sein kann. Damit kann dem stereotypen Bild von informatischen Karrieren als nicht sozial und kompetitiv

entgegengewirkt werden (Werner et al., 2004; Liebenberg et al., 2012; Ying et al., 2019; Choi, 2015). Frauen erwähnen in Interviews häufiger als Männer, dass Pair Programming ihr Selbstvertrauen verbessert. Ausserdem fragen sie bei Unklarheiten lieber bei erfahrenen Gleichaltrigen als bei der Kursleitung nach und schätzen es, allfällige Unsicherheiten mit jemandem teilen zu können. Somit könnte Pair Programming dabei helfen, mehr Frauen und Lernende mit wenig Vorwissen für die Informatik zu begeistern (Ying et al., 2019; Werner & Denning, 2009).

Vorteile für die Kursleitung

Unabhängig von der untersuchten Stufe erwähnen Forschende zudem Vorteile für die Kursleitung. Die Arbeitsbelastung für die Lehrperson sinkt, unter anderem, weil nur halb so viele Aufgaben benotet werden müssen (Faja, 2011; Salleh, 2008; Salleh et al., 2011). Dazu kommt, dass die Lernenden aufgrund des Austausches in den Paaren weniger auf die Kursleitung angewiesen sind (Faja, 2011; Li et al., 2013). Kleinere Probleme können von den Paaren selbst gelöst werden, weshalb die Lernenden beim Pair Programming weniger Fragen stellen als bei der Soloprogrammierung (durchschnittlich 0.44 Fragen pro Sitzung statt 1.19). Deshalb müssen die Lernenden auch weniger lange auf Hilfe der Kursleitung warten (21.70 statt 67.84 Sekunden) (Radermacher & Walia, 2011). Die Kursleitung kann sich mehr Zeit für komplexere Probleme nehmen (Papadakis, 2018; Çal & Can, 2020). Dadurch verlängert sich beim Pair Programming die durchschnittliche Interaktionszeit mit der Kursleitung (121.50 statt 87.28 Sekunden) (Radermacher & Walia, 2011). Dies führt dazu, dass alle Lernenden, aber insbesondere Frauen und schwächere Lernende, besser unterstützt werden können (Salleh, 2008).

2.3.4. Heterogenität der Resultate

Mehrere Forschende weisen darauf hin, dass die Heterogenität der Resultate bezüglich Vorhandenseins und Stärke der Effekte des Einsatzes von Pair Programming durch zahlreiche, schwer kontrollierbare, Moderatorvariablen erklärt werden könnte (Umapathy & Ritzhaupt, 2017; Salge & Berente, 2016; Lui et al., 2010). In vielen Studien werden diese Störfaktoren nicht kontrolliert, was zu verzerrten Ergebnissen führen kann (Salleh et al., 2011). Pair Programming wird in den Studien nicht immer gleich umgesetzt. So wird der Rollenwechsel teilweise von der Kursleitung in regelmässigen Zeitabständen (bspw. alle 5, 20 oder 30 Minuten) angeleitet, während die Lernenden in anderen Studien frei entscheiden können, ob und wann gewechselt wird (Tunga & Tokel, 2018). Forschende vermuten weiter, dass die Paarkompatibilität entscheidend für den Erfolg von Pair Programming ist (Hanks et al., 2011; Tunga & Tokel, 2018). Die Zuteilung der Paare erfolgt manchmal zufällig, teilweise dürfen die Lernenden selbst wählen, oder sie

werden nach bestimmten Kriterien, beispielsweise nach akademischer Leistung, eingeteilt (Jarratt et al., 2019). Dazu kommt, dass Lernende teilweise über die gesamte Studiendauer mit der gleichen Partnerin oder dem gleichen Partner zusammenarbeiten (Gómez et al., 2017), Paare in anderen Fällen dagegen regelmässig gewechselt werden (McDowell et al., 2006; O'Donnell et al., 2015). Ausserdem wählen die Studierenden zum Teil, ob sie lieber allein oder zu zweit arbeiten möchten (Salleh et al., 2011). In den Artikeln fehlen häufig Detailangaben, welche das Nachvollziehen möglicher Störvariablen erlauben würde (Umapathy & Ritzhaupt, 2017; Salge & Berente, 2016; Lui et al., 2010). Es wird unter anderem nicht immer erwähnt, wie die Paare zugeteilt oder wie häufig die Rollen gewechselt werden. Deshalb sind die Resultate von Pair Programming Experimenten schwierig zu replizieren. Zusätzlich wird von Forschenden kritisiert, dass die abhängigen Variablen in Pair Programming Studien unterschiedlich definiert und gemessen werden (Salge & Berente, 2016). Beispielsweise wird die Qualität des Codes in gewissen Studien mit der Fehlerdichte im Code, und in anderen durch ein Expertenrating bewertet (Salleh et al., 2011).

2.4. Einflussfaktoren auf das Pair Programming

Mehrere Forschende erwähnen, dass das Kontrollieren von Faktoren, welche die Effektivität von Pair Programming potenziell beeinflussen, entscheidend für Pair Programming Studien sei (Tunga & Tokel, 2018; Lui et al., 2010). Die Forschungsliteratur hat bereits verschiedene mögliche Einflussfaktoren auf die Effektivität von Pair Programming identifiziert. Dennoch gibt es noch zu wenige Studien, welche sich mit solchen Faktoren beschäftigen, um allgemeingültige Aussagen treffen zu können (Salleh et al., 2011; Salge & Berente, 2016; Bowman et al., 2020).

Als Schlüsselfaktor für die Effektivität von Pair Programming wird in vielen Studien die Paarkompatibilität genannt (Hanks et al., 2011; Tunga & Tokel, 2018; Bowman et al., 2020). Die Paare sollen entsprechend ihrer Eigenschaften möglichst so eingeteilt werden, dass der Lösungsprozess beschleunigt, Wissenstransfer ermöglicht, die Codequalität verbessert und eine Lernatmosphäre generiert wird (Alshehri & Benedicenti, 2014; Williams et al., 2006). In der Forschungsliteratur werden verschiedenen Faktoren erwähnt, welche sich auf die Paarkompatibilität auswirken könnten.

Studien, welche sich mit dem Einflussfaktor Gender beschäftigen, werden aufgrund der Nähe zur Thematik der vorliegenden Studie zuerst in einem separaten Kapitel vorgestellt. Anschliessend werden Eigenschaften der Lernenden, welche sich auf die Paarkompatibilität auswirken können, beschrieben. Im letzten Kapitel werden externe

Faktoren erwähnt, welche ebenfalls einen Einfluss auf die Effektivität von Pair Programming haben können.

2.4.1. Gender

Bisher gibt es wenige empirische Studien, welche sich mit dem Einfluss des Geschlechts auf das Pair Programming befassen (Salleh et al., 2011; Choi, 2015; Gómez et al., 2017; Zhong et al., 2016). Ausserdem widersprechen sich die Resultate häufig. Es wird vermutet, dass die Dauer des Pair Programmings entscheidend sein könnte, da sich diese zwischen den Studien stark unterscheidet. Während das Experiment von Choi (2015) nur 90 Minuten dauerte, lief das Experiment von Katira et al. (2005) über ein ganzes Semester. Dazu kommt, dass aufgrund der geringeren Anzahl von Frauen in der Informatik bisher häufig nur wenige Frauenpaare untersucht werden konnten (Choi et al., 2015).

Studien im akademischen Kontext

Verschiedene Studien befassen sich damit, ob sich die Paarkompatibilität von geschlechtshomogenen und -heterogenen Paaren im akademischen Kontext unterscheidet. Teilweise sprechen Forschungsergebnisse dafür, dass geschlechtsheterogene Paare weniger kompatibel sind als geschlechtshomogene (Katira et al., 2005; Choi, 2015) und weniger kommunizieren (Choi, 2015). Geschlechtshomogene Frauen- und Männerpaare unterscheiden sich bezüglich Kompatibilität und Kommunikation nicht signifikant (Choi, 2015). In einer anderen Studie können dagegen keine signifikanten Unterschiede zwischen geschlechtshomogenen und -heterogenen Paaren bezüglich Kompatibilität und Flow-Erfahrung (Synonym für einen positiv erlebten Schaffensrausch) nachgewiesen werden (Demir & Seferoglu, 2021a). Bei einer Befragung von Choi (2015) erwähnten Studentinnen in geschlechtsheterogenen Paaren häufig Schwierigkeiten und Konflikte, diejenigen in geschlechtshomogenen Gruppen dagegen die gute Kompatibilität. Die männlichen Teilnehmenden schrieben meist keine oder nur kurze Kommentare. Auch die Resultate bezüglich Codeproduktivität sind nicht eindeutig. Einige Studien können keinen signifikanten Unterschied zwischen geschlechtshomogenen und -heterogenen Gruppen bezüglich Codeproduktivität (Demir & Seferoglu, 2021a; Choi, 2015; Gómez et al., 2017; Akinola, 2016) oder Codequalität (Choi, 2015) nachweisen. In einer Studie von Jarratt et al. (2019) wiesen dagegen geschlechtshomogene Männerpaare die höchste Produktivität auf. Ausserdem zeigen geschlechtsheterogene Gruppen die höchste Variabilität der Produktivität, was auf Kompatibilitätsprobleme hinweisen könnte (Gómez et al., 2017). Studentinnen und Studenten, welche zufällig eine weibliche Partnerin

zugeteilt bekommen, besuchen den Programmierkurs häufiger und haben mehr Vertrauen in ihr fertiges Produkt und ihre Prüfungslösung. Auch wenn sich diese positiven Effekte für Männer und Frauen zeigen, profitieren weibliche Studierende noch etwas mehr von einer Partnerin (Jarratt et al., 2019).

Studien im schulischen Kontext

Im schulischen Kontext konnten bisher keine signifikanten Kompatibilitätsunterschiede zwischen geschlechtshomogenen und -heterogenen Paaren nachgewiesen werden. Geschlechtshomogene weibliche Gruppen weisen aber die engste Partnerschaft auf. Ausserdem kommunizieren und diskutieren sie häufiger als andere Geschlechterzusammensetzungen (Zhong et al., 2016). Dabei investieren sie mehr Zeit sich über die Aufgabe auszutauschen als Jungengruppen. Geschlechtshomogene männliche Gruppen kommunizieren häufiger über nicht-aufgabenbezogene Themen als Mädchengruppen (Campe et al., 2020). Jungen tendieren mehr dazu als Mädchen, ihre Partnerin oder ihren Partner für Codefehler verantwortlich zu machen. Bezüglich Selbstvertrauen zeigt sich kein signifikanter Unterschied zwischen den Geschlechterzusammensetzungen (Zhong et al., 2016). In einer Studie von Tsan et al. (2016) war die Codequalität von geschlechtshomogenen weiblichen Paaren signifikant schlechter als bei den anderen Geschlechterzusammensetzungen. Allerdings wurden in dieser Studie nur vier geschlechtshomogene weibliche Paare untersucht. Zhong et al. (2016) fanden keinen statistisch signifikanten Unterschied der Leistung der Geschlechterzusammensetzungen. Geschlechtsheterogene Gruppen wiesen in einer Studie von Underwood et al. (2000) im Vergleich zu geschlechtshomogenen Paaren eine geringere verbale Interaktion und eine weniger ausgeglichene Verwendung der Tastatur auf. Die Jungen bewegten den Cursor häufiger und fällten mehr Entscheidungen. Bezüglich Leistung unterschieden sich die Geschlechterzusammensetzungen nicht signifikant. Allerdings arbeiteten die Lernenden im Experiment von Underwood et al. (2000) nicht mit Pair Programming und sie lösten eine Art informatische Lückentextaufgabe, statt ein Programm zu schreiben.

2.4.2. Eigenschaften der Lernenden

In diesem Kapitel werden Eigenschaften der Lernenden vorgestellt, welche einen Einfluss auf die Effektivität von Pair Programming haben können. Dabei spielt insbesondere eine Rolle, wie die Lernenden entsprechend ihrer Eigenschaften zusammen eingeteilt werden, da sich dies auf die Paarkompatibilität auswirken kann. Forschungsergebnisse zum Einfluss von Fähigkeitsunterschieden der Lernenden, ihren Persönlichkeits-, sowie Lerntypen, ihrer Vertrautheit und ihren Einstellungen zum Pair

Programming werden nacheinander vorgestellt. Abschliessend werden auch Kombinationen dieser Faktoren diskutiert.

Fähigkeitsunterschiede

Der in der Forschungsliteratur bisher am häufigsten untersuchte Einflussfaktor auf die Effektivität von Pair Programming ist der Fähigkeitsunterschied der Lernenden (Salleh et al., 2011). Dabei werden meist die Programmierfähigkeiten oder das Vorwissen der Lernenden in diesem Bereich und seltener das allgemeine akademische Niveau betrachtet. Die Resultate der meisten Studien weisen darauf hin, dass Kommunikation und Kollaboration durch ähnliche Fähigkeiten der Lernenden gefördert wird (Denner et al., 2014; Al-Ramahi et al., 2013; Bowman et al., 2019; Salleh et al., 2011; Katira et al., 2005). Demir und Seferoglu (2021a) konnten dagegen keinen signifikanten Einfluss des Programmierwissens auf die Paarkompatibilität, Flow oder Leistung nachweisen. Eine Studie von Denner et al. (2014) weist darauf hin, dass sich das Vertrauen in die eigenen Programmierfähigkeiten und die positive Einstellung gegenüber Informatik bei erfahrenen Lernenden im Primarschulalter erhöht, wenn diese mit schwächeren Personen zusammenarbeiten. Die Forschenden vermuten, dass die erfahrenen Lernenden in dieser Konstellation eine motivierende Expertenrolle einnehmen können. Es gibt Studierende, welche es aufgrund des möglichen Wissenstransfers präferieren, mit einer schwächeren oder stärkeren Person zusammenzuarbeiten (Demir & Seferoglu, 2021a). Andere Studien weisen dagegen darauf hin, dass erfahrene Lernende lieber allein als mit schwächeren Personen arbeiten (Akour et al., 2013; Al-Ramahi et al., 2013). Dies lässt sich dadurch erklären, dass die erfahrenen Lernenden die schwächeren Personen mittragen müssen, und wenig vom Gegenüber lernen können (Maguire et al., 2014). Auch die schwächere Person profitiert wenig von dieser ungleichen Zusammenstellung, da sie sich häufig zurückzieht und kaum zur Aufgabe beitragen kann. Dies kann zu einem kleineren Lerneffekt, reduziertem Vertrauen in die eigenen Fähigkeiten und Reduktion des Interesses an der Informatik führen (Bowman et al., 2020; Bowman et al., 2019). Die meisten Studierenden präferieren es, mit jemandem zusammenzuarbeiten, der ein ähnliches oder ein gering höheres Fähigkeitslevel hat. Dadurch kann auch der Arbeitsaufwand fair verteilt werden (Williams et al., 2006; Demir & Seferoglu, 2021a).

Persönlichkeitstyp

Der Einfluss von Persönlichkeitstypen auf die Zusammenarbeit wird ebenfalls häufig untersucht (Salleh et al., 2011). Dabei werden die Studienteilnehmenden entsprechend ihres Persönlichkeitstyps in homogene und heterogene Paare eingeteilt. Häufig wird

dafür der Myers-Briggs-Typenindikator mit seinen vier Dimensionen (introvertiert-extrovertiert, denkend-fühlend, wahrnehmend-intuitiv, urteilend-empfindend) (Myers et al., 1998) gewählt. Die meisten Studien, welche den Myers-Briggs-Typenindikator zur Einteilung verwenden, können keinen signifikanten Unterschied zwischen homogenen und heterogenen Persönlichkeitspaaren bezüglich Kompatibilität (Akour et al., 2013; Katira et al., 2005) und Effizienz (Poonam & Yasser, 2018; Sfetsos et al., 2012) nachweisen, wenn diese am gleichen Ort arbeiten. Befinden sich die Lernende aber an einem anderen Ort, sind die heterogenen Persönlichkeitspaare weniger effizient als die homogenen (Poonam & Yasser, 2018). Die Forschenden argumentieren, dass Persönlichkeitsunterschiede in Kombination mit Distanz zu Kommunikations- und Kollaborationsproblemen führen können. Diese Aussage widerspricht den Resultaten einer weiteren Studie, in welcher heterogene Persönlichkeitspaare mehr kommunizieren als homogene. Ausserdem erreichen heterogene Persönlichkeitspaare ein besseres Codedesign und eine höhere Punktzahl (Sfetsos et al., 2012). Die Verwendung des Myers-Briggs-Typenindikators wird aber auch kritisiert, da er von Persönlichkeitspsychologen als unzuverlässig bewertet wird (Salleh et al., 2014). Alternativ verwenden einige Studien für die Einteilung in Persönlichkeitstypen das Fünf-Faktoren-Modell (Offenheit, Gewissenhaftigkeit, Extraversion, Verträglichkeit, Neurotizismus). Entsprechend der Neurotizismus-Komponente eingeteilte homogene und heterogene Paare unterscheiden sich nicht signifikant bezüglich Effizienz des Pair Programmings (Salleh et al., 2014). Bei der Gewissenhaftigkeits-Komponente wird sowohl von nicht signifikanten Unterschieden bezüglich Effizienz (Salleh et al., 2014), als auch von einer höheren Effizienz heterogener Paare (Demir & Seferoglu, 2021a) berichtet. Entsprechend der Gewissenhaftigkeits-Komponente eingeteilte homogene und heterogene Paare unterscheiden sich aber nicht bezüglich Kompatibilität und Flow (Salleh et al., 2014). Die uneinheitlichen Resultate des Effekts des Persönlichkeitstyps auf Pair Programming können unter anderem auf die Verschiedenheit der Messinstrumente, Dauer der Studien und Art der Aufgaben zurückgeführt werden (Salleh et al., 2011).

Lerntyp

Ein weiterer möglicher Einflussfaktor auf Pair Programming ist der Lerntyp der Lernenden. Zwei Studien teilen entsprechend den vier Dimensionen der Lerntypen nach Felder-Silverman (1988) (visuell-verbal, sensorisch-intuitiv, aktiv-reflektierend, sequenziell-global) homogene und heterogene Paare ein. Während Akour et al. (2013) keine signifikanten Unterschiede bezüglich Kompatibilität nachweisen konnten, zeigte sich bei Williams et al. (2006), dass Paare, welche eine Person mit sensorischem und

eine mit intuitiven Lernstil beinhalten, besonders kompatibel sind. Auch bei einer Einteilung in homogene und heterogene Paare nach dem *Learning Style Inventory* (LSI) von Kolb (1999) zeigen sich keine signifikanten Kompatibilitätsunterschiede der Paare. Wenn die Lernenden nach Kolb (1999) homogen eingeteilt werden, erhöht sich ihre Flow-Erfahrung während dem Programmieren, aber nicht ihre Leistung (Demir & Seferoglu, 2021a).

Vertrautheit oder Freundschaft

Die Vertrautheit oder Freundschaft der Lernenden kann sich auf das Pair Programming auswirken (Denner, et al., 2014). Wenn zwei befreundete Lernende zusammenarbeiten, weisen diese während des Programmierens einen höheren Flow auf. Nicht befreundete Paare leisten dagegen mehr. Laut Demir und Seferoglu (2021) beeinflusst das Freundschaftslevel die Paarkompatibilität nicht. Zhong et al. (2016) weisen in ihrer Studie dagegen darauf hin, dass Paare mit einer starken Partnerschaft kompatibler sind (Zhong et al., 2016). Wenn die Lernenden in einem Paar keine Freunde sind, kommt es häufiger zu Situationen, in welchen gar nicht interagiert wird (Campe et al., 2020). In Interviews geben die meisten Lernenden an, dass sie mit einer vertrauten Person besser interagieren, produktiver sind und sich während der Zusammenarbeit wohler fühlen (Celepkolu et al., 2020; Demir & Seferoglu, 2021a). Es gibt aber auch Lernende, welche lieber mit nicht vertrauten Personen zusammenarbeiten, weil dann professioneller zusammengearbeitet wird (Demir & Seferoglu, 2021a).

Einstellungen der Lernenden

Die Einstellungen der Lernenden gegenüber Kollaboration und ihre Arbeitsethik können sich ebenfalls auf die Effektivität von Pair Programming auswirken (Denner et al., 2014; Salleh et al., 2011). Weist eine Person eine negative Einstellung zur Kollaboration auf, nimmt das Programmierwissen bei der Person mit der positiveren Einstellung ab. Dies lässt sich durch Einschränkungen der Interaktion erklären, welche dadurch entstehen, dass eine Person weniger motiviert ist, zu kollaborieren (Denner et al., 2014; Campe et al., 2020). Williams et al. (2006) können in ihrer Studie zeigen, dass Paare mit unterschiedlicher Arbeitsethik nicht kompatibel sind. Lernende geben ausserdem an, dass sie gerne mit einer Person mit gleicher Einstellung zur Kollaboration und mit hoher Aufgabenverantwortung eingeteilt werden möchten (Demir & Seferoglu, 2021a). Die wahrgenommene Anstrengung der Partnerin oder des Partners beeinflusst das selbst eingeschätzte Lernen, das Vertrauen in die Qualität des Programms, sowie das Vergnügen und die Produktivität des Pair Programmings. Je gerechter die Arbeit in der Gruppe aufgeteilt wird, desto positiver wird das Pair Programming beschrieben (Faja,

2014). Deshalb wird empfohlen, Lernende mit ähnlichen Einstellungen miteinander einzuteilen (Campe et al., 2020).

Kombination von Einflussfaktoren

Einflussfaktoren auf das Pair Programming können nicht isoliert untersucht werden. Werden die Paare nach Leistungsstärke eingeteilt, unterscheiden sich die Lernenden meist in anderen Kategorien, beispielsweise bezüglich Geschlechts oder Lerntyps. Nicht nur einzelne Faktoren, sondern auch Kombinationen dieser können sich auf die Effizienz des Pair Programmings auswirken. Werner et al. (2013) zeigen beispielsweise, dass es mehr Vorteile gibt mit einer erfahrenen Person zusammen zu arbeiten, wenn die Lernenden befreundet sind, und wenn die schwächere Person ein hohes Selbstvertrauen hat (Werner et al., 2013).

2.4.3. Externe Faktoren

Nicht nur das Geschlecht und die Eigenschaften der Lernenden können einen Einfluss auf die Effektivität von Pair Programming haben, sondern auch externe Faktoren. Eine wichtige Voraussetzung für den Erfolg des Pair Programmings ist das korrekte Implementieren dieser Technik (Bowman et al., 2020). Dazu sollen die Leitlinien von Williams et al. (2008) oder anderen Forschenden befolgt werden (siehe Kapitel 2.2.2. Pair Programming Leitlinien). Die Kursleitung muss die Pair Programming Regeln am Anfang gut einführen, regelmässig an diese erinnern, und die Paare während der Programmierphase gut unterstützen. Dazu gehört, dass die Kursleitung die Paare beobachtet und sicherstellt, dass die Rollen Driver und Navigator von beiden Personen etwa gleich lange übernommen und regelmässig gewechselt werden. Dies ist wichtig für die Gleichberechtigung im Paar. Ausserdem müssen nicht kompatible Paare früh identifiziert und gewechselt werden (Umapathy & Ritzhaupt, 2017; Jeon & Manuelli, 2013; Lewis & Shah, 2015). Kollaboration wird durch Diskussion und Reflexion gefördert (Rodriguez et al., 2017). Deshalb sollen die Lernenden zu Beginn und auch während des Prozesses daran erinnert werden, dass Stille ein Zeichen für fehlende Zusammenarbeit ist (Denner et al., 2014; Rodríguez et al., 2017). Die Aufgabenstellungen sollen problemorientiert (Denner et al., 2014) und nicht zu einfach (Salge & Berente, 2016) sein. Die Möglichkeit, mit anderen Paaren zu interagieren, kann die Resultate einer Pair Programming Studie beeinflussen. Dieser Austausch findet aber in der Praxis statt und wird als positiv gesehen, weshalb er auch für den akademischen Bereich empfohlen wird (Socha & Sutano, 2015). Studien über das Pair Programming unterscheiden sich häufig darin, ob die Rollenwechsel angeleitet werden, oder die Lernenden selbst entscheiden, wann sie diese ausführen (Tunga & Tokel, 2018; Zhong

et al., 2017). Primarschülerinnen und -schüler haben am meisten Freude am Pair Programming, wenn sie die Rollen entsprechend ihrer Präferenz, aber mindestens einmal pro Sitzung, wechseln. Als weniger positiv wird der Rollenwechsel nach einer bestimmten Zeit, nach jeder Aufgabe oder zu Beginn jeder Sitzung beschrieben. Wenn frei gewechselt wird, reduziert sich die Frequenz der Rollenwechsel, aber die Kommunikation nimmt zu. Allerdings muss für einen erfolgreichen freien Rollenwechsel das Bewusstsein der Lernenden für ein selbstgesteuertes Wechseln gefördert werden (Zhong et al., 2017).

2.5. Gleichberechtigung der Partizipation beim Pair Programming

Eine erfolgreiche Kollaboration beim Pair Programming setzt voraus, dass sich beide Personen beteiligen. Nur dann können Vorteile wie die einfachere Entscheidungsfindung, das Diskutieren aus verschiedenen Perspektiven und ein Wissenstransfer erreicht werden (Plonka et al., 2012a; Plonka et al., 2012b). Verschiedene Studien untersuchen deshalb, wie gerecht die Lernenden die Arbeit in den Paaren aufteilen. Meist werden in diesen Studien Video- und Audioaufnahmen der Gruppen gemacht und mit einem Codebuch ausgewertet.

In diesem Kapitel werden sowohl Ergebnisse von professionellen Programmierenden als auch solche von Studierenden und Lernenden berichtet, da es für keine dieser Gruppen allein genügend Studien gibt. Zuerst werden Resultate bezüglich der Gleichberechtigung in der Kommunikation vorgestellt. Darauf folgen Hinweise zum Rollenwechsel und der Dauer der Rollenübernahme. In einem Fazit wird darüber diskutiert, ob Gleichberechtigung in der Partizipation eine wichtige Voraussetzung für den Erfolg von Pair Programming ist.

Kommunikation

Die Kollaboration ist effektiver, wenn sich beide Personen wesentlich am Dialog beteiligen. Dabei ist insbesondere entscheidend, dass die Person in der Driver-Rolle die eigenen Aktionen ausführlich beschreibt (Rodríguez et al., 2017). Mehrere Studien beschäftigen sich damit, ob die Kommunikation in den Paaren beim gemeinsamen Programmieren ausgeglichen ist. Dafür wird meist die Verteilung der gesamten Sprechzeit auf die zwei Personen, sowie auf die zwei Rollen untersucht (Lewis & Shah, 2015; Plonka et al., 2012a). Studienergebnisse weisen darauf hin, dass die verbale Kommunikation der zwei Personen in einer Gruppe nicht immer ausgeglichen ist. Allerdings kann auch anders kontribuiert und kommuniziert werden, beispielsweise mit Gesten (Lewis & Shah, 2015; Plonka et al., 2012a). Bei den verbal unausgeglichenen

Paaren fällt auf, dass diese häufig das Ziel haben, die Aufgabe schnell zu lösen. Dies kann dazu führen, dass die Person mit mehr Erfahrung die Führung übernimmt (Lewis & Shah, 2015; Plonka et al., 2012b). Wissenstransfer benötigt Zeit und lässt sich somit nicht mit diesem Ziel vereinbaren. Dazu kommt, dass viele Novizen aufgrund sozialen Drucks so wenig wie möglich nachfragen und selten die Driver-Rolle übernehmen, auch wenn sie den Aktionen der Partnerin oder des Partners nicht mehr folgen können. Hier hilft es, wenn die erfahrenere Person die eigenen Aktionen erklärt und Rückfragen stellt (Plonka et al., 2012b).

Rollenwechsel

Werden die Rollen regelmässig gewechselt, weist dies auf ein hohes Engagement beider Personen hin. Deshalb beschäftigen sich Studien zur Gleichberechtigung beim Pair Programming auch mit der Anzahl der Rollenwechsel. Eine Studie mit professionellen Programmierinnen und Programmierern zeigt, dass die Rollen regelmässig gewechselt werden, durchschnittlich 15.67-mal pro Stunde. Der Rollenwechsel wird von beiden Personen etwa gleich häufig initiiert, wobei in 82% die Person in der Navigatoren-Rolle den Wechsel herbeiführt. Die Forderung eines Rollenwechsels wird selten abgelehnt, weshalb es diesbezüglich auch kaum Konflikte gibt. Die Rollenwechsel erfolgen meist ohne verbale Hinweise, fluid und ohne Anstrengung (Plonka et al., 2011).

Dauer der Rollenübernahme

Ebenfalls entscheidend für die Gleichberechtigung beim Pair Programming ist, dass beide Personen die zwei Rollen etwa gleich lange übernehmen (Plonka et al., 2011; McDowell et al., 2006). Bei professionellen Programmierenden dominiert in zwei Drittel der Fälle eine Person die Driver-Rolle. Trotz dieser unausgeglichenen Verteilung können beide Personen aktiv involviert sein. Allerdings haben Personen, welche selten und nur kurz die Driver-Rolle innehaben, eine höhere Wahrscheinlichkeit aus dem Pair Programming auszusteigen. Dies lässt sich dadurch erklären, dass sie den Aktionen der anderen Person nicht mehr folgen können (Plonka et al., 2012a; Plonka et al., 2011).

Fazit

Eine ausbalancierte Partizipation ist nicht unbedingt nötig für ein erfolgreiches Pair Programming. Allerdings ist ein ausreichender Grad an Partizipation und Kontributionsmöglichkeit erforderlich, damit engagiert, fokussiert und aktiv zusammengearbeitet werden kann. Eine gerechte Partizipation könnte für das Lernen wichtig sein, aber nicht unbedingt dafür, ein Problem effektiv zu lösen. Die Bewertung einer Pair Programming Sitzung wird bei professionellen Programmierenden nicht davon

beeinflusst, ob gleichberechtigt partizipiert wurde. Ob dies bei Studierenden und Lernenden auch so ist, muss geprüft werden. Die wenigen Studien, welche sich mit der Ausgeglichenheit der Partizipation beschäftigen, betrachten die Quantität der Kontributionen, aber nicht die Qualität dieser (Plonka et al., 2012a).

2.6. Pair Programming in einem Robotik-Kurs

Ein Robotik-Kurs unterscheidet sich von einer typischen Programmier Einführung, da zusätzlich zum Computer auch ein Roboter bedient werden muss. Dieser Unterschied setzt eine Erweiterung der Pair Programming Regeln voraus. Diesbezüglich wurden viele Vorschläge gemacht, welche sich aber meist nicht auf Paare, sondern auf Gruppen mit mehr als zwei Personen beziehen (Yuen et al., 2014; Mosley & Kline, 2006). Zhong und Wang (2021) schlagen in ihrem Artikel eine Lösung für Paare vor und testen ihre Empfehlung empirisch. Eine Person ist für die Software verantwortlich, programmiert also den Roboter am Computer, die andere Person bedient die Hardware, baut also den Roboter. Wie beim normalen Pair Programming sollen diese Rollen regelmässig gewechselt werden, mindestens einmal pro Kurs. An ihrer zwölfwöchigen Studie nahmen 66 Lernende der 5. Primarklasse in China teil. Die Schülerinnen und Schüler konnten ihre Partnerinnen und Partner selbst wählen. Gearbeitet wurde mit dem Robotermodell *mBot*, welches mit einer auf der bildungsorientierten Programmiersprache Scratch basierenden Umgebung programmiert werden kann. Die Studie konnte keinen statistisch signifikanten Unterschied zwischen den traditionellen Driver-Navigator-Paaren und den Software-Hardware-Paaren in Bezug auf Lernerfolg, Lerneinstellung, Engagement, mentale Anstrengung, Kompatibilität und Einstellung gegenüber Pair Programming finden. Den Forschenden erwähnen, dass beim verwendeten Bildungsroboter *mBot* durch seine einfache Montage wenig Hardwarearbeiten anfielen. Die Hardware-Person schloss deshalb ihre Aufgaben schnell ab, und konnte sich auf das Beobachten der Arbeit der Softwareperson konzentrieren und diese unterstützen. Somit übernahm die Hardwareperson schnell eine Navigatoren-Rolle. Die Forschenden empfehlen deshalb, in Robotik-Kursen mit wenig Hardwareaufgaben die traditionellen Rollen Driver und Navigator zu verwenden. Fallen dagegen komplexere Hardwareaufgaben an, könnten Software-Hardware-Paare eine Alternative sein.

3. Methode

In diesem Kapitel werden zuerst die Forschungsfragen und Hypothesen aus der Literatur abgeleitet, formuliert und begründet. Darauf folgt die Beschreibung des Forschungsdesigns und der Stichprobenziehung, sowie eine Diskussion forschungsethischer Punkte. Dann wird der Kontext der Videostudie, die Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken*, welche die Autorin dieser Masterarbeit mitentwickeln durfte, genauer vorgestellt. Darauf folgt die Beschreibung der Erhebungsinstrumente und der Datenauswertung.

3.1. Forschungsfragen und Hypothesen

Seit Kent Beck (1999) Pair Programming eingeführt hat, wird diese Technik in der Arbeitswelt, sowie im akademischen und schulischen Kontext eingesetzt (Gómez et al., 2017). Es konnte bereits mehrfach bewiesen werden, dass der Einsatz von Pair Programming viele Vorteile bietet (siehe Kapitel 2.3. Effekte des Pair Programmings), wobei insbesondere Frauen und schwächere Lernende davon profitieren sollen (Li et al., 2013; Maguire et al., 2014; O'Donnell et al., 2015; Smith et al., 2017). Diese positiven Effekte des Pair Programmings können aber nur eintreten, wenn die Technik korrekt umgesetzt wird (Bowman et al., 2020). Die von der Forschungsliteratur vorgeschlagenen Leitlinien (siehe Kapitel 2.2.2. Pair Programming Leitlinien) müssen eingehalten werden, damit gleichberechtigt partizipiert und kollaboriert werden kann. Ebenfalls entscheidend ist, dass die Lernenden in einem Paar kompatibel sind (Hanks et al., 2011; Tunga & Tokel, 2018; Bowman et al., 2020). Die Forschungsliteratur hat bereits viele Faktoren identifiziert, welche sich auf die Paarkompatibilität auswirken können (siehe Kapitel 2.4. Einflussfaktoren auf das Pair Programming). Einer dieser Faktoren ist die Geschlechterzusammensetzung der Paare (geschlechtshomogen weiblich, geschlechtsheterogen, geschlechtshomogen männlich). Bisher gibt es nur wenige empirische Studien, welche den Einfluss von Gender auf das Pair Programming untersuchen (Salleh et al., 2011; Choi, 2015; Gómez et al., 2017; Zhong et al., 2016). Dazu kommt, dass unter anderem aufgrund kleiner Stichproben, unterschiedlichen Alters der Studienteilnehmenden und verschiedener Umsetzungen des Pair Programmings uneinheitliche Resultate berichtet werden. Insbesondere im schulischen Kontext besteht noch Forschungsbedarf, ob und inwiefern sich die Zuteilung in geschlechtshomogene und -heterogene Paare auf die Zusammenarbeit von Lernenden auswirkt (Jarratt et al., 2019).

Ein häufig berichteter Vorteil des Einsatzes von Pair Programming ist, dass sich die Leistungen der Lernenden verbessern. Dabei kann neben einer Erhöhung der Aufgaben-

und Prüfungsnote, sowie besserer Qualität des Codes auch nachgewiesen werden, dass zu zweit schneller programmiert wird (Salge & Berente, 2016; Akour et al., 2013; Isong et al., 2016; Salleh, 2008; Salleh et al., 2011). Die meisten Studien im akademischen Kontext weisen darauf hin, dass sich die Geschlechterzusammensetzungen bezüglich Codeproduktivität nicht signifikant unterscheiden (Demir & Seferoglu, 2021a; Choi, 2015; Gómez et al., 2017; Akinola, 2016). In einer Studie von Jarratt et al. (2019) zeigten dagegen geschlechtshomogene Männerpaare die höchste Produktivität. Im schulischen Kontext gibt es diesbezüglich noch zu wenig Forschung. Zhong et al. (2016) fanden in ihrer Studie mit Lernenden im Primarschulalter keinen signifikanten Leistungsunterschied. Mit der ersten Forschungsfrage (F1) dieser Masterarbeit soll mit Lernenden der Primar- und Sekundarstufe geprüft werden, ob sich geschlechtshomogene und -heterogene Paare in Bezug auf ihre Geschwindigkeit des AufgabenlöSENS unterscheiden. Entsprechend den bisherigen Forschungsergebnissen wird vermutet, dass sich die Geschlechterzusammensetzungen diesbezüglich nicht statistisch signifikant unterscheiden (H1).

F1: Inwiefern unterscheiden sich geschlechtshomogene und geschlechtsheterogene Paare in Bezug auf die Geschwindigkeit des AufgabenlöSENS?

H1: Geschlechtshomogene und geschlechtsheterogene Paare unterscheiden sich nicht in Bezug auf die Geschwindigkeit des AufgabenlöSENS.

Studienresultate aus dem akademischen Kontext zeigen, dass beim Pair Programming Paare aufgrund des Austausches untereinander weniger auf die Kursleitung angewiesen sind (Faja, 2011; Li et al., 2013). Kleinere Probleme können von den Paaren selbst gelöst werden, weshalb Paarprogrammierende weniger Fragen stellen als Einzelprogrammierende (Radermacher & Walia, 2011). Entsprechend den Leitlinien von Williams et al. (2008) sollen die Paare ermutigt werden, die Lösung selbst zu finden. Mit der zweiten Forschungsfrage (F2) soll geprüft werden, ob sich die Geschlechterzusammensetzungen bezüglich Anzahl angeforderter Hilfestellungen unterscheiden. Das männlich geprägte, stereotype Bild der Informatik kann einen negativen Effekt auf die Frauen haben. Frauen schätzen sich trotz gleicher Leistungen im Informatikbereich als weniger kompetent ein. Sie geben ausserdem häufiger an, dass sie nicht alle Programmierkonzepte verstanden haben und sind weniger selbstsicher in ihrem Produkt (Jarratt et al., 2019). Deshalb wird angenommen, dass geschlechtshomogene weibliche Paare mehr Hilfestellungen anfordern als geschlechtshomogene männliche und geschlechtsheterogenen Paare (H2).

F2: Inwiefern unterscheiden sich geschlechtshomogene und geschlechtsheterogene Paare in Bezug auf die angeforderten Hilfestellungen?

H2: Geschlechtshomogene Mädchenpaare fordern mehr Hilfestellungen an als geschlechtshomogene Jungenpaare und geschlechtsheterogene Paare.

Die wenigen Studien zur Kompatibilität von geschlechtshomogenen und -heterogenen Paaren berichten von uneinheitlichen Resultaten, weshalb noch Forschungsbedarf besteht. Während sich geschlechtshomogene Frauen- und Männerpaare bezüglich Kompatibilität laut ersten Ergebnissen nicht signifikant unterscheiden (Choi, 2015), scheinen geschlechtsheterogene Paare weniger kompatibel zu sein (Katira et al., 2005; Choi, 2015). In einer anderen akademischen (Demir & Seferoglu, 2021a), sowie einer Studie im schulischen Kontext (Zhong et al., 2016) kann dagegen kein signifikanter Unterschied der Geschlechterzusammensetzungen bezüglich Kompatibilität nachgewiesen werden. Dennoch gibt es Hinweise dafür, dass weibliche Lernende mehr von einer Partnerin als einem Partner profitieren (Jarratt et al., 2019; Zhong et al., 2016). Studentinnen in geschlechtsheterogenen Paaren berichten von Schwierigkeiten und Konflikten, diejenigen in geschlechtshomogenen Gruppen von einer guten Kompatibilität (Choi, 2015). Unklar bleibt, wodurch diese Schwierigkeiten ausgelöst werden. Diese Masterarbeit geht der Vermutung nach, dass eine unausgeglichene Partizipation einer der Gründe sein könnte.

Eine erfolgreiche Kollaboration beim Pair Programming setzt voraus, dass sich beide Personen beteiligen (Plonka et al., 2012a; Plonka et al., 2012b; Celepkolu et al., 2020). Je gerechter die Arbeit dabei in der Gruppe aufgeteilt wird, desto positiver wird das Pair Programming eingeschätzt (Faja, 2014). Die Leitlinien zum Pair Programming aus der Forschung (siehe Kapitel 2.2.2. Pair Programming Leitlinien) sollen dabei sicherstellen, dass beide Personen gleichberechtigt partizipieren und kollaborieren. Mit zwei Forschungsfragen soll deshalb geprüft werden, ob sich Mädchen und Jungen in geschlechtsheterogenen Gruppen (F3), und Lernende in geschlechtsheterogenen und geschlechtshomogenen Gruppen allgemein, (F4) beim Befolgen eines zuvor kommunizierten Pair Programming Leitfadens unterscheiden. Der Fokus wird auf zwei Pair Programming Regeln gelegt. Erstens darf die Person in der Navigatoren-Rolle weder Computertastatur noch Maus berühren (Werner et al., 2004). Dies ist wichtig für einen respektvollen Umgang miteinander. Ausserdem kann so sichergestellt werden, dass die Nutzungszeit von Maus und Tastatur fair aufgeteilt wird. Dadurch haben beide Personen die gleiche Chance zu kooperieren. Zweitens sollen die Rollen Navigator und Driver regelmässig gewechselt werden (Williams & Kessler, 2000; Williams et al., 2008; Werner et al., 2004). So kann sichergestellt werden, dass beide Personen die zwei Rollen etwa gleich lange übernehmen. Dies ist entscheidend für die Gleichberechtigung im Pair Programming (Plonka et al., 2011; McDowell et al., 2006). Personen, welche selten und nur kurz die Driver-Rolle innehaben, steigen mit einer höheren

Wahrscheinlichkeit aus dem Pair Programming aus. Meist ist der Grund, dass sie den Aktionen der Person in der Driver-Rolle nicht mehr folgen können (Plonka et al., 2012a; Plonka et al., 2011). Werden die Rollen regelmässig gewechselt, weist dies auf ein hohes Engagement beider Personen hin (Plonka et al., 2011). In den folgenden Abschnitten werden die zur dritten und vierten Forschungsfrage formulierten Hypothesen vorgestellt.

F3: Inwiefern unterscheiden sich Mädchen und Jungen in geschlechtsheterogenen Paaren beim Befolgen eines zuvor kommunizierten Pair Programming Leitfadens?

Wenn Jungen anwesend sind, stehen Mädchen unter Druck, ihre Geschlechtsidentität zu bewahren, indem sie sich geschlechterkonform verhalten, und umgekehrt (Maccoby, 1990; Brutsaert, 1999; Amon et al., 2012; Kröll, 2010; von Ow & Husfeldt, 2011; Wedl & Bartsch, 2015). Mit dem Hintergrundwissen, dass Robotik in der Gesellschaft als typisch männlich gilt, könnten sich Mädchen während der Zusammenarbeit mit Jungen in ihrer Geschlechtsidentität bedroht fühlen, und sich deshalb zurückhalten (Flore & Wicherts, 2015). Gleichzeitig tendieren männliche Lernende mehr dazu als weibliche, ihre Partnerin oder ihren Partner beim Pair Programming für Codefehler verantwortlich zu machen. In einer älteren Studie von Underwood et al. (2000), welche die Interaktion der Geschlechter ohne Einbezug von Pair Programming untersuchte, dominierten die Jungen die Benutzung der Tastatur. Entsprechend den bisherigen Forschungsergebnissen wird vermutet, dass sich Mädchen in geschlechtsheterogenen Paaren mehr zurückhalten und die Jungen dominieren. Daraus wird die Hypothese abgeleitet, dass die männlichen Lernenden in geschlechtsheterogenen Paaren häufiger die Pair Programming Regeln missachten. Vermutet wird, dass die Jungen die Tastatur oder Maus trotz Innehaben der Navigatoren-Rolle häufiger (H3) und länger (H4) berühren als die Mädchen.

H3: In geschlechtsheterogenen Paaren berühren die Jungen in der Navigatoren-Rolle entgegen dem Leitfaden des Pair Programmings häufiger die Computertastatur oder die Maus als die Mädchen in der Navigatoren-Rolle.

H4: In geschlechtsheterogenen Paaren dauern die Eingriffe der Jungen in der Navigatoren-Rolle durchschnittlich länger als die der Mädchen in der Navigatoren-Rolle.

F4: Inwiefern unterscheiden sich geschlechtsheterogene und geschlechtshomogene Paare beim Befolgen eines zuvor kommunizierten Pair Programming Leitfadens?

Geschlechtshomogene weibliche Paare weisen laut einer Studie von Zhong et al. (2016) die engste Partnerschaft auf. Ausserdem kommunizieren und diskutieren sie häufiger als andere Geschlechterzusammensetzungen. Dabei investieren sie auch mehr Zeit, um sich über die Aufgabe auszutauschen als Jungengruppen. Geschlechtshomogene männliche Gruppen kommunizieren dagegen häufiger über nicht-aufgabenbezogene

Themen (Campe et al., 2020). Underwood et al. (2000) können zeigen, dass geschlechtsheterogene Paare im Vergleich zu geschlechtshomogenen Paaren eine geringere verbale Interaktion aufweisen. In geschlechtshomogenen Gruppen rückt die Geschlechtsidentität gegenüber dem Arbeitsselbst in den Hintergrund (Faulstich-Wieland et al., 2004; Kessels, 2002; von Ow & Husfeldt, 2011). Dadurch könnten der wahrgenommene soziale Aussendruck und die Angst sozialer Ausgrenzung reduziert werden (Esch, 2008). In reinen Mädchengruppen können sich die Schülerinnen auch in männlich konnotierten Bereichen selbstbewusst zeigen (Booth & Nolen, 2012; Kröll, 2010). Deshalb wird vermutet, dass Mädchen am meisten davon profitieren, mit Mädchen in der Gruppe zu sein, und Jungen mit Jungen. Zwischen den geschlechtshomogenen Frauen- und Männerpaaren kann bisher in Bezug auf die Kompatibilität und Kommunikation kein signifikanter Unterschied nachgewiesen werden (Choi, 2015). Entsprechend dieser vermuteten höheren Kompatibilität von geschlechtshomogenen als -heterogenen Paaren, wird von den geschlechtshomogenen Paaren ein besseres Einhalten der Pair Programming Regeln erwartet. Zunächst wird vermutet, dass in geschlechtshomogenen Paaren die Rollen häufiger gewechselt werden als in den geschlechtsheterogenen Gruppen (H5). Ausserdem könnten in geschlechtsheterogenen Paaren die Lernenden in der Navigatoren-Rolle insgesamt häufiger (H6) und länger (H7) die Tastatur oder Computermaus berühren.

H5: In geschlechtshomogenen Paaren werden beim Pair Programming die Rollen häufiger gewechselt als in den geschlechtsheterogenen Paaren.

H6: In geschlechtsheterogenen Paaren berühren die Lernenden in der Navigatoren-Rolle entgegen dem Leitfaden des Pair Programmings häufiger die Computertastatur oder die Maus als die Lernenden in den geschlechtshomogenen Paaren.

H7: In geschlechtsheterogenen Paaren dauern die Eingriffe der Lernenden in der Navigatoren-Rolle durchschnittlich länger als die der Lernenden in geschlechtshomogenen Paaren.

3.2. Forschungsdesign

Um zu überprüfen, ob sich die Zuteilung in geschlechtshomogene und -heterogene Paare auf die Zusammenarbeit der Lernenden in einem Robotik-Kurs auswirkt, wurde eine Videostudie durchgeführt. Die Videos wurden in der Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken* an der Pädagogischen Hochschule Luzern aufgezeichnet. Diesen Kurs konnten im Herbstsemester 2020 Schülerinnen und Schüler der 5. bis 9. Klasse besuchen. Die Videoaufnahmen mit Audiospur ermöglichten eine detaillierte Beobachtung mit hohem Informationsgehalt, ohne die Schülerinnen und Schüler während des Besuches in der Lernwerkstatt zu stören. Pro Schulklasse wurden vier Paare mit unterschiedlichen Geschlechterzusammensetzungen beim

selbstständigen Aufgabenlösen gefilmt. Als Aufnahmezeitpunkt wurde der Beginn der selbstständigen Arbeitsphase direkt nach der Einführung gewählt. Nur zu diesem Zeitpunkt lösten alle Lernenden die gleiche Aufgabenstellung. Nach dem Lösen des ersten Aufgabensets konnten die Lernenden die nächste Aufgabe nach Interesse wählen. Zusätzlich füllte die Lehrperson während der Videoaufnahmen einen Beobachtungsauftrag aus. Mit diesem wurden demografische Angaben, nicht sichtbare Eigenschaften der Lernenden, wie beispielsweise das allgemeine schulische Leistungsvermögen, sowie Beobachtungen der Lehrperson erhoben. Die Videoaufnahmen und die Beobachtungsaufträge wurden von zwei studentischen Hilfskräften mithilfe eines Codebuchs analysiert. Durch diese standardisierte Inhaltsanalyse konnte die Zusammenarbeit der Lernenden mit quantifizierbaren Merkmalen beschrieben werden. Die Auswertung der Daten erfolgte mit dem Statistikprogramm SPSS.

3.3. Stichprobenziehung

In diesem Kapitel werden zuerst die Grundgesamtheit und der Erhebungszeitraum beschrieben. Danach wird die Stichprobenziehung, welche zweistufig erfolgte, vorgestellt. Zuerst wird die Rekrutierung der Schulklassen erläutert. Darauf folgt die Beschreibung, wie die Lernenden in die Zweiergruppen eingeteilt wurden. Abschliessend wird der Prozess der Auswahl der vier Paare, welche gefilmt werden sollten, beschrieben.

3.3.1. Grundgesamtheit und Erhebungszeitraum

Die Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken* wurde für Schülerinnen und Schüler der fünften Primar- bis dritten Sekundarklasse konzipiert. Lehrpersonen der Zentralschweiz, welche eine dieser Stufen unterrichteten, konnten ihre Klasse für einen Besuch der Lernwerkstatt anmelden. Anfragen von Begabungs- und Begabtenförderungsprogrammen mit Lernenden der dritten bis sechsten Primarklasse wurden ebenfalls akzeptiert. Zur Grundgesamtheit dieser Studie gehören somit Lernende der fünften Primar- bis dritten Sekundarklasse, sowie Begabungs- und Begabtenförderungsprogramme der dritten bis sechsten Primarklasse in der Zentralschweiz. Der Erhebungszeitraum entspricht der Durchführungsdauer der Lernwerkstatt vom 31.08.2020 bis 30.10.2020.

3.3.2. Rekrutierung von Schulklassen

Zur Vorstellung der Lernumgebung wurde ein digitaler Flyer erstellt, welcher unter anderem über die offiziellen Kanäle der Pädagogischen Hochschule Luzern auf Twitter und Facebook und über verschiedene Newsletter für MINT- oder

Lernwerkstattinteressierte verbreitet wurde. Der Besuch der Lernwerkstatt war kostenlos. Da die Lehrpersonen ihre Klasse selbst anmeldeten, handelte es sich nicht um eine Zufallsstichprobe. Ob eine Schülerin oder ein Schüler die Lernwerkstatt besuchen konnte, hing von vielen Faktoren ab. Der Kurs fand im Uni/PH-Gebäude gleich neben dem Bahnhof Luzern statt. Dieser Standort ist für Schulhäuser in der Stadt und im Kanton Luzern am besten erreichbar. Lehrpersonen aus anderen Kantonen könnten sich aufgrund der Anreise aus Zeit- oder Kostengründen gegen den Besuch der Lernumgebung entschieden haben. Weiter spielte es eine Rolle, ob sich die Lehrperson selbst für das Thema interessierte, oder es zumindest als sinnvoll erachtete, mit ihren Schülerinnen und Schülern einen Robotik-Kurs zu besuchen. Damit hing auch zusammen, ob die Lernenden im Unterricht bereits auf informatische Themen, oder sogar Robotermodelle getroffen sind. Klassen, welche die Lernwerkstatt besuchten, könnten somit ein überdurchschnittlich hohes Vorwissen in diesem Bereich mitgebracht haben. Diese und weitere Faktoren führen dazu, dass es sich nicht um eine repräsentative Stichprobe der Grundgesamtheit handelt.

3.3.3. Paareinteilung und -auswahl

In vielen Studien wird die Paarkompatibilität als Schlüsselfaktor für die Effektivität von Pair Programming genannt (Hanks et al., 2011; Tunga & Tokel, 2018; Bowman et al., 2020). Die Paare sollen entsprechend ihrer Eigenschaften möglichst so eingeteilt werden, dass der Löseprozess beschleunigt, Wissenstransfer ermöglicht, die Codequalität verbessert und eine Lernatmosphäre generiert wird (Alshehri & Benedicenti, 2014; Williams et al., 2006; Williams et al., 2008). In der Forschungsliteratur werden verschiedenen Faktoren erwähnt, welche sich auf die Paarkompatibilität auswirken können (siehe Kapitel 2.4. Einflussfaktoren auf das Pair Programming). Da sich die vorliegende Videostudie mit dem Einfluss von Gender auf Pair Programming befasst, wurden geschlechtshomogene weibliche, geschlechtsheterogene und geschlechtshomogene männliche Paare gebildet. Diese Einteilung erfolgte entsprechend dem biologischen Geschlecht, in der Literatur als Sex bezeichnet.

Allerdings ist das Geschlecht nicht der einzige Faktor, welcher sich auf die Zusammenarbeit der Lernenden auswirken kann. Mehrere Forschende erwähnen, dass das Kontrollieren von Faktoren, welche die Effektivität von Pair Programming potenziell beeinflussen könnten, entscheidend für Pair Programming Studien sei (Tunga & Tokel, 2018; Lui et al., 2010). Weitere von Forschenden genannte mögliche Einflussfaktoren auf das Pair Programming sind die allgemeinen und informatischen Fähigkeiten der Lernenden, ihre Persönlichkeits- und Lerntypen, ihr Freundschaftslevel und ihre Einstellungen. Die Resultate der meisten Studien weisen darauf hin, dass

Kommunikation und Kollaboration durch ähnliche Fähigkeiten der Lernenden gefördert werden (Denner et al., 2014; Al-Ramahi et al., 2013; Bowman et al., 2019; Salleh et al., 2011; Katira et al., 2005). Deshalb wurden zusätzlich zum Kriterium des Geschlechts leistungsstärkere Schülerinnen und Schüler, sowie leistungsschwächere Lernende zusammen eingeteilt. Dabei wurden sowohl das allgemeine schulische Leistungsvermögen als auch die Programmierkenntnisse berücksichtigt.

Die meisten Studien können weder durch eine Einteilung nach Persönlichkeitstyp (Akour et al., 2013; Katira et al., 2005; Poonam & Yasser, 2018; Sfetsos et al., 2012; Salleh et al., 2014) noch nach Lerntyp (Akour et al., 2013; Demir & Seferoglu, 2021a) einen statistisch signifikanten Einfluss auf die Effektivität von Pair Programming nachweisen. Deshalb wurden diese Einflussfaktoren bei der Paareinteilung in dieser Studie nicht miteinbezogen. Forschungsergebnisse weisen weiter darauf hin, dass zwei befreundete Lernende einen höheren Flow während des Programmierens haben, nicht befreundete Paare dagegen mehr leisten (Demir & Seferoglu, 2021). Um eine möglichst neutrale Zusammenarbeit zu erreichen, wurden Schülerinnen und Schüler zusammen eingeteilt, welche weder sehr gut befreundet noch zerstritten sind. Es wird empfohlen, Lernende mit ähnlichen Einstellungen zur Kollaboration und einer übereinstimmenden Arbeitsethik zusammen einzuteilen (Campe et al., 2020). Die Arbeitsethik fließt teilweise in das allgemeine schulische Leistungsniveau hinein, die Einstellungen zur Zusammenarbeit in das Freundschaftslevel. Deshalb wurden diese Faktoren bei der Paareinteilung in dieser Studie nicht separat berücksichtigt. Zusätzlich zu den bisher genannten Kriterien sollte der Altersunterschied der Lernenden nicht mehr als ein Jahr betragen. Ein grosser Altersunterschied könnte mit unterschiedlichen Fähigkeiten verbunden sein. Insgesamt sollten die Paare also entsprechend ihrer Leistungsstärke, ihren Programmierkenntnissen und ihrem Alter möglichst homogen eingeteilt werden. Diese Faktoren könnten einen Einfluss auf die Zusammenarbeit haben und müssen deshalb berücksichtigt werden (Zhong et al., 2016; Arisholm et al., 2007; Choi, 2015). Die in dieser Studie verwendeten Kriterien für die Paareinteilung lauten zusammengefasst folgendermassen:

- *sowohl reine Mädchen- und Jungengruppen als auch gemischte Gruppen*
- *leistungsstärkere Schülerinnen und Schüler zusammen*
- *leistungsschwächere Schülerinnen und Schüler zusammen*
- *Schülerinnen und Schüler mit Programmier-Vorkenntnissen zusammen*
- *möglichst Schülerinnen und Schüler zusammen, die weder sehr gut befreundet, noch zerstritten sind*
- *Altersunterschied möglichst maximal ein Jahr*

Die Einteilung der Lernenden anhand dieser Kriterien wurde von den Lehrpersonen vorgenommen. Allerdings ist es in der Praxis nicht möglich, jede Schülerin und jeden

Schüler einer Klasse allen Kriterien entsprechend perfekt einzuteilen. Deshalb wurden die Leistungsstärke, die Programmier-Vorkenntnisse, das Freundschaftslevel und der Altersunterschied der Lernenden zusätzlich im Beobachtungsauftrag erhoben und später als mögliche Störfaktoren behandelt. Vor dem Besuch der Lernwerkstatt wurde den Lehrpersonen ein Informationsschreiben zugesendet. Darin waren die Schutzmassnahmen gegen COVID-19, Informationen zum Pretest der parallellaufenden Studie, das Vorgehen bezüglich Einwilligungserklärungen für die Videoaufnahmen (siehe Kapitel 3.4. Forschungsethik), sowie der Auftrag zur Paareinteilung enthalten. Die Lehrpersonen mussten sich im Vorfeld Zeit nehmen, ihre Klasse entsprechend den obengenannten Kriterien in maximal zwölf Zweiergruppen einzuteilen. Hatte eine Lehrperson die Paareinteilung vergessen, konnte diese während der Startpräsentation in der Lernwerkstatt nachgeholt werden.

In der Lernwerkstatt wählte die Kursleitung jeweils vier Paare zufällig aus, wobei pro Klasse möglichst je ein Mädchen-, ein Jungenpaar und ein geschlechtsheterogenes Paar gefilmt werden sollten. Der vierte Kamerastandort wechselte die gewünschte Paarzusammensetzung mit dem Datum, damit von allen Geschlechterzusammensetzungen genügend Daten erhoben werden konnten. Um sicherzustellen, dass alle Kursleitungen diese Anweisungen korrekt ausführen, wurden diese detailliert in einer Anleitung für die Betreuungspersonen beschrieben.

3.4. Forschungsethik

Damit die Schülerinnen und Schüler während des Robotik-Kurses gefilmt werden durften, musste zuvor die Einwilligung der Erziehungsberechtigten eingeholt werden. Die Lehrpersonen wurden in einem Schreiben darüber informiert, gaben den Erziehungsberechtigten die Einwilligungserklärung (siehe Anhang 1) ab, und brachten diese ausgefüllt mit in den Kurs. In der Einwilligungserklärung wurde erwähnt, dass die Aufnahmen ausschliesslich für Forschungszwecke verwendet, nicht an Dritte weitergegeben und für die Ergebnisdarstellung anonymisiert werden. Die Kursleitungen, Assistierenden und Lehrpersonen wurden mündlich darüber informiert, dass sie auf den Videos ebenfalls erkennbar sein können. Die Beobachtungsaufträge und Datensätze lassen keine Rückschlüsse auf einzelne Personen zu, bei den Videoaufnahmen kann dies aber nicht ausgeschlossen werden. Diese Aufnahmen werden deshalb nur lokal gespeichert, sicher aufbewahrt und nicht weitergegeben. Lernende ohne Einwilligung erfuhr keine Nachteile und wurden möglichst weit von den Kameras entfernt platziert.

Um die Beobachtungen nicht zu verfälschen, wurden Massnahmen getroffen, welche nachfolgend forschungsethisch diskutiert werden. Die Kursleitungen, Assistierenden und

Lehrpersonen wurden instruiert, während der Videoaufnahmen bei den Paaren an den Kamerastandorten möglichst nicht einzugreifen. Da diese Massnahme zu wenig konkretisiert wurde, kam es zu unterschiedlich strikten Umsetzungen. In den Videos sieht man, dass gewisse Gruppen sehr lange auf eine Hilfestellung warten mussten. Während die Kursleitungen und Assistierenden die anderen Gruppen bei Nichteinhalten auf die Pair Programming Regeln aufmerksam machen mussten, sollten sie dies bei den videografierten Paaren nicht tun. Somit hatten die Paare vor den Kameras einen kleinen Nachteil. Es wurde jedoch klar kommuniziert, dass bei grösseren Problemen oder Streitigkeiten jederzeit eingegriffen werden darf. Nach der Pause wurden keine Aufnahmen mehr gemacht, weshalb wieder alle Paare gleich betreut werden konnten. Da es sich nur um einen kleinen Nachteil zu Beginn der Lernwerkstatt handelte, welcher sich positiv auf die Verlässlichkeit der Daten auswirkt, ist diese Massnahme forschungsethisch vertretbar. Die Lehrpersonen, Eltern und Schülerinnen und Schüler kannten das genaue Thema der Studie nicht. Die Kursleitungen und Assistierenden wurden grob über die Studie informiert, damit sie wichtige Punkte wie die Geschlechterzusammensetzung und die Pair Programming Einführung gut beachten. Auf dem Flyer, im Informationsschreiben und auf der Einwilligungserklärung für die Videoaufnahmen wurde nur davon gesprochen, dass Informationen zur Lernwirksamkeit dieses Angebotes gesammelt werden. Auf dem Beobachtungsauftrag wurden die Lehrpersonen darüber informiert, dass die Zusammenarbeit der Lernenden untersucht wird, sie aber bei Nachfragen den Schülerinnen und Schülern sagen sollen, dass der Umgang mit den für die Lernumgebung entwickelten Aufgabenstellungen beobachtet wird. Da es sich dabei um eine ungenaue, aber dennoch korrekte Beschreibung des Forschungsziels handelte, ist dieses Vorgehen forschungsethisch vertretbar.

3.5. Kontext der Studie

Die Lernwerkstatt der Pädagogischen Hochschule Luzern ist ein thematisch halbjährlich wechselnder, ausserschulischer Lernort für Schulklassen. Studierende und Dozierende entwickeln die Lernumgebungen mit Schwerpunkt auf handlungsorientiertem und aktiv-konstruktivem Lernen gemeinsam. Die Autorin dieser Masterarbeit war bei der Entwicklung der untersuchten Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken* dabei. Diese wurde in Zusammenarbeit mit der Hochschule Luzern auf Grundlage der Roberta®-Initiative des Fraunhofer-Instituts für Intelligente Analyse- und Informationssysteme IAIS entwickelt und vom Förderprogramm MINT-Schweiz der Akademien der Wissenschaften unterstützt. Bei der 2002 gestarteten Roberta®-Initiative des Fraunhofer-Instituts IAIS handelt es sich um eine Bildungsinitiative, welche fachwissenschaftliche und fachdidaktische Aspekte für das Lehren und Lernen mit

Robotern mit Schwerpunkt Informatik und Technik vereint. Das Ziel der Initiative ist es, bei Kindern und Jugendlichen Interesse für Technik, Naturwissenschaften und Informatik zu wecken. Im Vordergrund steht das Entwickeln von Robotik-Kursen, welche beide Geschlechter gleichermaßen ansprechen.

Schulklassen konnten die Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken* vom 31.08.2020 bis 30.10.2020 besuchen. Das Ziel dieses Kurses war es, den Lernenden einen motivierenden Einblick in die Robotik, und damit Informatik, zu geben. Die Lernenden konnten dabei das Robotermodell Lego Mindstorms EV3 mit der grafischen Programmiersprache EV3-G als Roboter in der Stadt der Zukunft programmieren. Gearbeitet wurde an Kompetenzen des Lehrplans 21 aus den Bereichen Medien und Informatik (MI.2.2), Natur und Technik, beziehungsweise NMG (NT1.2 und NMG.5.3) und TTG (TTG.3.A.2). Um einen nachhaltigen Lernprozess zu ermöglichen, wurden über die Webseite der Lernwerkstatt Vorschläge zur Vor- und Nachbereitung im Rahmen des Regelunterrichts aufgezeigt.

In diesem Kapitel wird zuerst der Ablauf eines Kurshalbtages in der untersuchten Lernumgebung vorgestellt. Anschliessend werden die Überlegungen zur Kontextorientierung der Lernumgebung, die darin enthaltenen Aufgabenstellungen, sowie die Untersuchungsbedingungen in der Lernwerkstatt genauer erläutert.

3.5.1. Ablauf eines Kurshalbtages

Die Lernumgebung konnte morgens oder nachmittags besucht werden. Ein Halbtage dauerte circa 3.5 Stunden. Auf Abbildung 1 ist der Ablauf eines Kurshalbtages ersichtlich. Nach einer gemeinsamen Einführung in die Robotik und die Programmierumgebung EV3-G starteten die Lernenden mit der ersten Aufgabe, bei welcher sie ein autonomes Fahrzeug programmieren sollten. Das selbstständige Arbeiten an dieser Aufgabe wurde bis zur Pause gefilmt. Während dieser Zeit füllte die Lehrperson einen Beobachtungsauftrag aus. Nach der Pause arbeiteten die Lernenden weiter selbstständig an den Aufgaben, diese Phase wurde aber nicht mehr gefilmt. Auf den Videos ist somit nur der Beginn des Halbtages sichtbar. Der in Abbildung 1 in der letzten Phase erwähnte Posttest wurde für eine parallellaufende Studie durchgeführt. Als Abschluss reflektierten die Lernenden den Halbtage zuerst innerhalb der Gruppe mit einem Lerntagebuch und dann im Plenum.

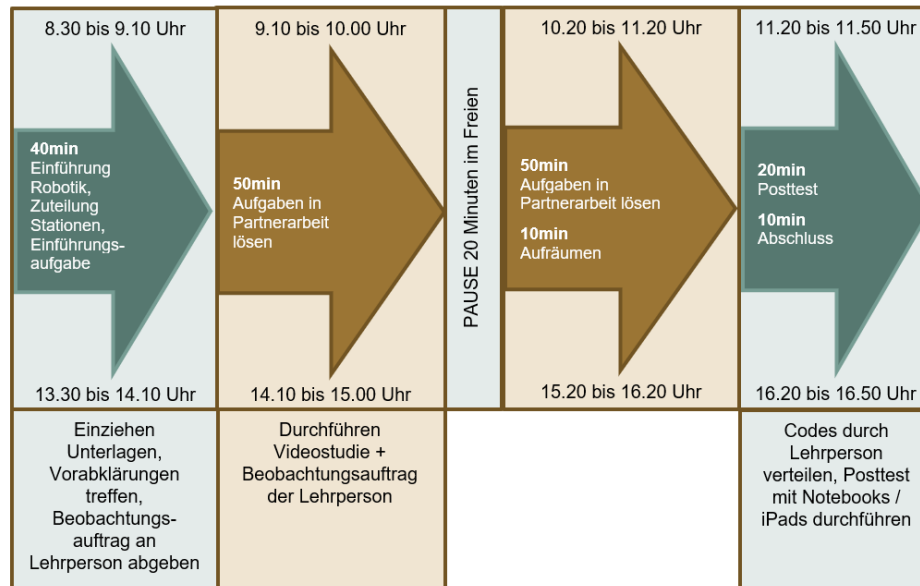


Abbildung 1 Ablauf eines Kurshalbtages

3.5.2. Kontextorientierung der Lernwerkstatt

Der Zusammenhang von Inhalt und Kontext wird in der Forschungsliteratur als Schlüsselfaktor für das Interesse an Naturwissenschaften gesehen. Bei der Entwicklung von Lerninhalten soll die Lebenswelt der Lernenden miteinbezogen werden. Fachinhalte können sehr unterschiedlich gerahmt und vermittelt werden, beispielsweise motiviert Jugendliche das Thema *Musik* mehr als *Akustik und Töne*. Kontextbasierte Aufgaben fördern durch das erhöhte Interesse die thematische Auseinandersetzung. Wichtig ist dabei, dass Rücksicht auf Diversitäten wie Geschlecht, Vorwissen oder unterschiedliche technische Affinitäten genommen wird (Elster, 2007; Fraunhofer IAIS, 2019).

Als Kontext der in dieser Studie untersuchten Lernumgebung wurde die Rolle von Robotern in der Stadt der Zukunft gewählt. Diese Thematik ist nah an der Lebenswelt der Schülerinnen und Schüler, welche bereits heute in ihrem Alltag Robotern begegnen. Immer mehr Familien haben beispielsweise einen Staubsauger-, Rasenmäh- oder Fensterputzroboter zuhause. Doch nicht nur im Privat-, sondern auch im Berufsleben nehmen Roboter dem Menschen immer mehr Arbeiten ab. Diese Entwicklung hat einen grossen Einfluss auf den Stellenmarkt, gewisse Berufe verschwinden, andere kommen neu hinzu. Schülerinnen und Schüler sind aufgrund ihres bevorstehenden Berufseinstiegs direkt von diesen Veränderungen betroffen. Es ist deshalb wichtig, dass die Lernenden die digitale Transformation verstehen und kritisch hinterfragen können. Diese Thematik betrifft Jungen und Mädchen gleichermassen.

Die Lernumgebung startete mit einem Video, welches in das Thema Stadt der Zukunft einführte. Dabei handelte es sich um die Vorstellung der *Woven City* von Toyota. In diesem Video wird gezeigt, wie eine Stadt in der Zukunft aussehen könnte. Die

Lernenden erhielten den Auftrag, im Video nach Robotern zu suchen. Im Anschluss wurden die gesichteten Roboter im Plenum gesammelt und die Funktionen der einzelnen Roboter besprochen. Unter anderem treten im Video selbstfahrende Autos, Drohnen und Serviceroboter im Haushalt auf. Als nächstes wurde besprochen, ab wann ein technisches Gerät als Roboter gilt. Dabei wurden die Begriffe Aktor, Sensor, Prozessor, Kommunikation und Energieversorgung erklärt und als die fünf notwendigen Bestandteile eines Roboters definiert. Für die Aufgabenstellungen wurden passend zur Stadt der Zukunft möglichst genderneutrale Themen gewählt. Die Lernenden konnten autonome Fahrzeuge, Unterhaltungs-, Wachhund- und Industrieroboter programmieren. Die Aufgaben wurden in einem Video vorgestellt, in welchem jeweils ein realer Roboter in diesem Arbeitsbereich (z.B. ein Nao als Unterhaltungsroboter) und die Umsetzung der Aufgabe mit einem Lego Mindstorms EV3 gezeigt werden.

3.5.3. Aufgabenstellungen der Lernwerkstatt

Pair Programming ist besonders effektiv, wenn die zu lösenden Aufgabenstellungen problemorientiert (Denner et al., 2014) und nicht zu einfach (Salge & Berente, 2016) sind. Die Lernumgebung beinhaltete vier Aufgabenstellungen. Pro Aufgabenstellung gab es ein Kartenset, welches sich aus einer Übersichtskarte und vier Aufgabenkarten zusammensetzt. Die Materialien der Lernumgebung stehen registrierten Nutzerinnen und Nutzern auf der Plattform Roteco (www.roteco.ch) zum Download zur Verfügung (Menüpunkt *Aktivitäten*, Suchbegriff *Mit Roberta® die Stadt der Zukunft entdecken*).

Auf der Vorderseite der Übersichtskarte wird die Hauptaufgabenstellung vorgestellt. Bei der Startaufgabe *Das autonome Fahrzeug* soll der Roboter einer schwarzen Linie entlangfahren. Zusätzlich zur Aufgabenstellung wird der Tipp gegeben, dass sich ein Zickzack-Fahrstil eignet, welcher mit Worten und einer Grafik erklärt wird. Damit die Lernenden ihre Programme testen können, wurden eine passende Bodenmatte mit einer aufgedruckten Zukunftsstadt, sowie dreidimensionale Stadtteile gestaltet. Auf der Übersichtskarte wird mit einem Robotersymbol darauf hingewiesen, in welchem Bereich der Bodenmatte der Roboter für diese Aufgabe getestet werden kann. Auf der Rückseite der Übersichtskarte erklärt eine Schritt-für-Schritt-Anleitung den Ablauf des Aufgabensets. Jedes Set beinhaltet vier Teilaufgabenstellungen, mit welchen sich die Schülerinnen und Schüler die nötigen Programmierkompetenzen zum Lösen des Hauptproblems selbstständig aneignen können. Auf der Vorderseite der Aufgabenkarten befinden sich der Auftrag und erste Tipps. Falls die Lernenden die Aufgabe nicht ohne Hilfe lösen können, gibt es auf der Rückseite eine Schritt-für-Schritt-Anleitung mit Erklärungen. Zusätzlich hatte jede Gruppe eine Kurzanleitung zur Software EV3-G zur Verfügung. Haben die Lernenden die Teilaufgabenstellungen gelöst, können sie sich an

die Hauptaufgabenstellung wagen. Hierfür müssen sie auf Gelerntes zurückgreifen, und Bausteine der Teilaufgabenstellungen kombinieren. Für die Hauptaufgabenstellung haben die Lernenden keine Schritt-für-Schritt-Anleitung zur Verfügung. Ist auch diese Aufgabe gelöst, gibt es als Abschluss Zusatzinformationen zum Thema der Aufgabe, beim autonomen Fahrzeug zur Mobilität der Zukunft. Dabei handelt es sich um Texte aus der *TechnoScope*-Reihe von SATW.

Diese Aufgabenform mit Übersichts- und Aufgabenkarten, beziehungsweise Haupt- und Teilaufgabenstellungen, sowie den Lösungen auf den Rückseiten, wurde bereits in einer anderen Lernumgebung an der Pädagogischen Hochschule Luzern im Fachbereich Medien und Informatik im Frühling 2019 getestet. Dieser Aufbau ermöglicht es den Lernenden, sehr selbständig zu arbeiten, da sie bei Schwierigkeiten selbst in der Lösung nachschauen können. Dadurch wird die Lehrperson entlastet. Während starke Schülerinnen und Schüler die Aufgaben ohne Hilfestellung bearbeiten können, ermöglichen die Lösungen auch schwachen Lernenden Erfolgserlebnisse. Die Aufgabensets schaffen eine Differenzierung, welche nicht nur innerhalb der Klasse, sondern auch über die Klassenstufen hinweg vorteilhaft ist. Das Programmieren hält nur langsam Einzug in den Schulalltag, weshalb die meisten Lernenden diesbezüglich kaum oder nur wenig Vorwissen haben. Somit kann es auch vorkommen, dass Primarklassen mit viel Vorwissen die Aufgaben schneller lösen als Sekundarklassen ohne Vorwissen. Auch wenn über alle Klassenstufen hinweg die gleichen Aufgabenstellungen in der Lernumgebung eingesetzt wurden, waren sie durch die innere Differenzierung für keine Lernenden zu schwierig oder zu einfach.

Das autonome Fahrzeug wurde als Startaufgabe, und damit als Untersuchungsgegenstand gewählt, weil es gut an die Einführungsaufgabe, welche als Einstieg in die Programmierumgebung von allen gemeinsam gelöst wurde, anschliesst. Dort programmierten die Lernenden ihren Roboter so, dass er genau fünf Sekunden vorwärtsfährt. Diese Aufgabe eignet sich als Einstiegsaufgabe, weil lediglich ein Programmierblock verwendet wird, der nur leicht angepasst werden muss. Ausserdem ist es für die Schülerinnen und Schüler sehr motivierend, wenn sie den Roboter schon nach kurzer Zeit zum Fahren bringen können. Bei der ersten Teilaufgabenstellung des Aufgabensets *Das autonome Fahrzeug* soll der Roboter dann möglichst genau einen Meter vorwärtsfahren. Die Lernenden sollen dabei die Einstellungsmöglichkeiten des Motorenblocks etwas austesten. Anschliessend soll das Robotermodell einen Meter vorwärtsfahren, eine Sekunde warten und anschliessend wieder einen Meter rückwärtsfahren. So kommt ein neuer Block, der Warteblock, hinzu, und das Rückwärtsfahren wird geübt. Hier müssen bereits mehrere Blöcke in der richtigen

Reihenfolge aneinandergereiht werden. In der dritten Aufgabe kommt dann die Lenkung dazu, hier soll der Roboter nacheinander eine Links- und eine Rechtskurve fahren. Mithilfe einer Schleife sollen die Kurven dann unendlich wiederholt werden. Die letzte Teilaufgabenstellung führt den Lichtsensor ein. Der Roboter soll anhalten, sobald er über eine schwarze Linie fährt. Hier kommt der Schalter als neuer Block hinzu. Diese vier Teilaufgabenstellungen haben sich bereits im Rahmen anderer Robotik-Projekte der Pädagogischen Hochschule Luzern bewährt. Nach dem Abschliessen der Startaufgabe konnten sich die Lernenden entweder für den etwas einfacheren Unterhaltungs- oder den gleich anspruchsvollen Wachhund-Roboter entscheiden. Die Zusatzaufgabe Industrieroboter sollte auch stärkere Schülerinnen und Schüler herausfordern. Diese Aufgabensets werden an dieser Stelle nicht genauer vorgestellt, da sie nicht Teil der Videostudie waren.

3.5.4. Untersuchungsbedingungen in der Lernwerkstatt

In diesem Kapitel werden die Untersuchungsbedingungen in der Lernwerkstatt vorgestellt. Zunächst geht es um die Räumlichkeit der Lernwerkstatt und die Positionierung der Kameras. Danach wird die Betreuung in der Lernwerkstatt und die Einrichtung der Arbeitsplätze für die Lernenden beschrieben. Als letztes wird erläutert, wie die Lernenden in die Praktik Pair Programming eingeführt wurden.

Räumlichkeit

Der Raum der Lernumgebung wurde zweigeteilt (siehe Abbildungen 2 und 3). Vorne befanden sich Hocker, auf welchen die Lernenden während der Einführungspräsentation sassen. Im hinteren Teil wurden die Arbeitsstationen rund um vier Bodenmatten verteilt.

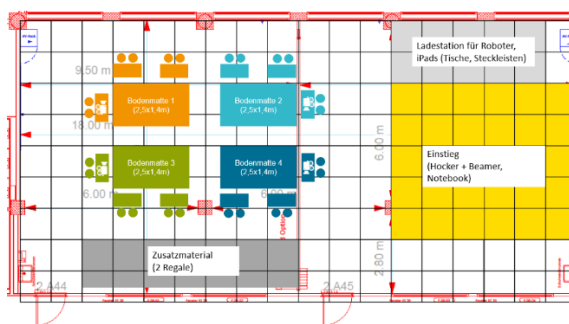


Abbildung 2 Rauplan der Lernumgebung



Abbildung 3 Foto der Lernumgebung

Kamerapositionierung

Damit keine Lernende ohne Einwilligungserklärung auf den Aufnahmen zu sehen waren, wurden die Kameras am Rand und nach aussen gerichtet platziert. Die Abbildung 4 zeigt den Blickwinkel einer der Kameras. Der Laptopbildschirm und die Arbeit auf den Bodenmatten waren somit nicht auf den Aufnahmen sichtbar. Um trotz lauter Umgebungsgeräusche verständliche Tonaufnahmen zu ermöglichen, wurden die Bluetooth-Mikrofone direkt am Laptopbildschirm befestigt.



Abbildung 4 Blickwinkel einer Kamera

Betreuung in der Lernwerkstatt

Je eine Kursleitung und eine Assistenz begleiteten eine Schulklasse in der Lernumgebung. Für die Durchführung der 53 Kurse wurden insgesamt elf Kursleitungen und zehn Assistierende rekrutiert. Alle Kursleitungen absolvierten zuvor eine Roberta®-Schulung, welche im Rahmen der Fraunhofer Roberta®-Initiative angeboten wurde. Die meisten Kursleitungen waren Dozierende der Pädagogischen Hochschule Luzern oder der Hochschule Luzern. Die Assistierenden waren interessierte Studierende der Hochschule Luzern, der Pädagogischen Hochschule Luzern oder des Masterstudiums Fachdidaktik Medien und Informatik der Pädagogischen Hochschule Schwyz. Alle Kursleitungen und Assistierenden wurden im Rahmen einer 2.5-stündigen Einführung für die Durchführung der Lernwerkstatt geschult. Dabei wurden auch die Handhabung der Kameras und Mikrofone und das korrekte Verhalten für die Videostudie erklärt. Damit die Lernenden trotz unterschiedlicher Kursleitungen ähnliche Bedingungen für die Videostudie hatten, gab es klare Vorgaben, welche in einem Manual, einer Anleitung für die Betreuungspersonen, festgehalten wurden. Voraussetzung für das erfolgreiche Pair Programming ist eine gute Unterstützung durch die Kursleitung (Bowman et al., 2020). Im Manual wurden die Kursleitungen und Assistierenden angewiesen, die Paare bei Fragen zu unterstützen und darauf zu achten, dass nach jeder Teilaufgabenstellung die Rollen der Tandem-Programmierung und damit die Plätze gewechselt wurden. Bei den Lernenden vor den Kameras durfte während der Aufnahmen nicht eingegriffen werden. Williams et al. (2008) erwähnen in ihren Leitlinien, dass die Kursleitung bei der

Einführung der Tandem-Programmierung sicherstellen muss, dass die Lernenden die Rollen richtig annehmen, regelmässig wechseln und durchgehend aktiv sind. Die Kursleitung sollte die Paare proaktiv besuchen und fragen, wie es läuft, um sicherzustellen, dass sie effizient zusammenarbeiteten.

Arbeitsplätze der Lernenden

Damit die Arbeitsplätze der Lernenden immer gleich eingerichtet wurden, gab es im Manual genaue Anweisungen mit einem Beispielfoto (siehe Abbildung 5). Entsprechend der Pair Programming Leitlinien aus der Forschung sollten der Laptop in der Mitte des Tisches und die Stühle parallel zueinander platziert werden (Williams & Kessler, 2000; Werner et al., 2004; Williams et al., 2008). Die Aufgaben- und Hilfestellungen sowie das Robotermodell Lego Mindstorms EV3 wurden auf die linken Tischseite bei der Person in der Navigatoren-Rolle gelegt, die Computermaus auf die Rechte bei der Person in der Driver-Rolle (Werner et al., 2004).



Abbildung 5 Einrichtung der Arbeitsplätze

Pair Programming Einführung

Eine wichtige Voraussetzung für den Erfolg des Pair Programmings ist eine gute Einführung in diese Praktik (Bowman et al., 2020; Çal & Can, 2020). Damit alle Lernenden die gleichen Instruktionen erhielten, wurde den Kursleitungen eine PowerPoint-Folie (siehe Abbildung 6) und das folgende Skript vorgegeben, welches sie wortwörtlich so durchführen mussten:


- „Ihr programmiert heute zu zweit. Diese sogenannte ‚Tandem-Programmierung‘ wird auch in der Arbeitswelt angewendet und hat viele Vorteile. Beispielsweise ist die Qualität des Programms höher und es wird Zeit gespart, da Fehler früher erkannt werden.“
- „Damit die ‚Tandem-Programmierung‘ gut funktioniert, müsst ihr ein paar Dinge beachten. Wer sitzt rechts am Tisch, bitte Hand hoch (warten bis alle Lernenden rechts am Tisch Hand gehoben haben). Ihr bedient die Tastatur und Computermaus (an einer Arbeitsstation zeigen, nicht an einem der Kamerastandorte) (Mausklick Folie). Wer sitzt links am Tisch, bitte Hand hoch (warten bis alle Lernenden links am Tisch Hand gehoben haben). Ihr denkt mit,

überprüft und hilft, ohne die Tastatur, den Bildschirm oder die Computermaus zu berühren. Ihr habt die Aufgaben, den Roboter und die Hilfestellungen vor euch. Ihr bedient den Roboter und testet euer Programm auf der Bodenmatte (an einer Arbeitsstation zeigen, nicht an einem der Kamerastandorte) (Mausklick Folie).“

- *„Auch wenn nur jemand den Computer bedient, entscheidet ihr gemeinsam was ihr programmiert (Mausklick Folie). Dafür ist eine gute Kommunikation wichtig (Mausklick Folie). Sprecht eure nächsten Schritte also immer ab.“*
- *„Setzt euch Schulter an Schulter hin, so dass der Laptop in eurer Mitte steht (Mausklick). Wie dieses Team hier (positives Beispiel hervorheben, nicht Teams an Kamerastandorten). So haben beide einen guten Blick auf den Bildschirm. Nach jeder gelösten Teilaufgabenstellung wechselt ihr den Platz. Dann bedient die andere Person die Tastatur und Computermaus.“*

Tandem-Programmierung

- Person rechts am Tisch
 - Bedient Tastatur und Computermaus
- Person links am Tisch
 - Denkt mit, überprüft und hilft
 - Berührt Tastatur, Bildschirm und Computermaus nicht
 - Bedient den Roboter
- Gleichberechtigung bei Entscheidungen
- Gute Kommunikation wichtig
- Sitzposition
 - Schulter an Schulter
 - Laptop in der Mitte
 - Platzwechsel nach jeder Teilaufgabenstellung



PH Luzern / HSLU / BRZ LU

Abbildung 6 PowerPoint-Folie zur Pair Programming Einführung

Dieses Skript orientierte sich an den Leitlinien der Forschungsliteratur (siehe Kapitel 2.2.2. Pair Programming Leitlinien), und befand sich im Manual für die Kursleitungen und Assistierenden. Anstelle des englischen Begriffs Pair Programming wurde die deutsche Variante Tandem-Programmierung verwendet, weil sich die meisten Schülerinnen und Schüler einen deutschen Begriff besser merken, und sich darunter mehr vorstellen können. Mit dem Bezug zur Arbeitswelt sollte verdeutlicht werden, dass diese Technik wirklich angewendet wird. Dadurch nehmen die Schülerinnen und Schüler Pair Programming ernst und lernen, dass Programmieren auch in der Praxis eine kollaborative Aufgabe sein kann. So kann das stereotype Bild der Informatik als kompetitives und wenig soziales Arbeitsfeld etwas aufgebrochen werden. Dies soll insbesondere Mädchen motivieren, welche sich meist ein soziales und kollaboratives Berufsumfeld wünschen. Die Erwähnung von Vorteilen des Pair Programmings in der Praxis sollte zu einer positiven Einstellung gegenüber dieser Technik führen, welche entscheidend für den Erfolg des Pair Programmings ist (Williams & Kessler, 2000).

Auch die englischen Begriffe Driver und Navigator wurden nicht verwendet, unter anderem, weil sie im deutschen Sprachgebrauch nicht genderneutral sind. Die Rollen

wurden stattdessen entsprechend der Sitzposition zugeteilt. Die Person rechts am Tisch übernahm die Driver-Rolle, die links die Navigatoren-Rolle. Die Lernenden sollten sich mit Handzeichen melden, damit allen klar war, welche Rolle sie innehaben. In der Praxis hat sich gezeigt, dass es Lernende gibt, welche ohne diese klare Identifikation später unsicher sind, welche Rolle sie übernehmen müssen. Die Rollenregeln orientierten sich an den Leitlinien von Werner et al. (2004) (siehe Kapitel 2.2.2. Pair Programming Leitlinien). Da in diesem Robotik-Kurs mehr Software- als Hardwareaufgaben anfielen, wurde entsprechend der Empfehlung von Zhong und Wang (2021) die Hardware- mit der Navigatoren-Rolle kombiniert. Zusätzlich zur unterstützenden Rolle bei der Programmierung bediente die Person in der Navigatoren-Rolle deshalb den Roboter. Nach dieser Rolleneinführung folgten weitere Tipps für eine erfolgreiche Tandem-Programmierung. Dazu gehörte, dass die Schülerinnen und Schüler in einem Paar gemeinsam entscheiden sollten, was sie programmieren. Beim Pair Programming sind die Lernenden gleichberechtigt und entwickeln gemeinsam ein Artefakt (Williams & Kessler, 2000; Werner et al., 2004; Williams et al., 2008). Dafür ist eine gute Kommunikation wichtig (Werner et al., 2004; Hanks et al., 2011; Denner et al., 2014; Rodríguez et al., 2017).

Damit die Lernenden die Arbeitsplatzeinrichtung nicht veränderten, wurden sie darauf aufmerksam gemacht, dass die Einhaltung dieser Ergonomie wichtig sei. Sie sollten bequem nebeneinandersitzen mit dem Laptop in ihrer Mitte, so dass beide einen guten Blick auf den Bildschirm hätten. Abschliessend wurde darauf hingewiesen, dass die Rollen regelmässig gewechselt werden sollten (Williams & Kessler, 2000; Werner et al., 2004; Williams et al., 2008). Als Zeitpunkt für den Platzwechsel wurde das Abschliessen einer Teilaufgabenstellung gewählt. Zhong et al. (2017) weisen darauf hin, dass Lernende mehr Freude am Pair Programming haben, wenn sie entsprechend ihrer Präferenzen, aber mindestens einmal pro Sitzung, wechseln dürfen. Dadurch reduziert sich die Frequenz der Rollenwechsel. Da in dieser Studie nur ein kurzer Zeitabschnitt gefilmt wurde, hätte bei einem freien Platzwechsel die Gefahr bestanden, dass die Paare selten oder gar nie die Rollen wechseln. Deshalb wurde vorgegeben, dass die Gruppen nach jeder Teilaufgabenstellung die Rollen wechseln müssen. Dabei handelte es sich um einen klar festgelegten Zeitpunkt, für welchen nicht auf die Uhr geschaut werden musste. Die Paare wechselten dadurch selbstständig, die Kursleitung musste sie nicht anleiten. Bei einem Rollenwechsel wurden die Plätze physisch getauscht. Diesbezüglich gibt es in der Forschungsliteratur unterschiedliche Empfehlungen. Während Williams et al. (2008) darauf hinweisen, dass bei einem Rollenwechsel nicht die Stühle gewechselt werden sollen, empfehlen Werner et al. (2004), dass die Lernenden aufstehen und die

Positionen wechseln. Da sich Werner et al. (2004) im Gegensatz zu Williams et al. (2008) auf Schülerinnen und Schüler beziehen, wurde die physische Variante des Rollenwechsels gewählt. Dies unterstützte auch den Entscheid, die Rollen Driver und Navigator entsprechend der Sitzposition zuzuteilen.

3.5.5. Evaluation der Lernumgebung

Vier Tage nach der letzten Durchführung der Lernumgebung erhielten alle Lehrpersonen, welche mit ihrer Klasse mindestens einen Halbttag gebucht hatten, eine Maileinladung, um an einer Evaluation des Kursangebotes teilzunehmen. Die Lehrpersonen hatten vom 04.11.2020 bis 23.11.2020 Zeit, die Evaluation auszufüllen. Sechs Lehrpersonen haben die Lernwerkstatt mehrmals mit unterschiedlichen Klassen besucht. Somit waren nicht 53, sondern nur 47 verschiedene Lehrpersonen in der Lernwerkstatt. Mehr als die Hälfte dieser Lehrpersonen (29 von 47) haben den Fragebogen ausgefüllt. Für jede Klassenstufe ausser der dritten Sekundarstufe hat mindestens eine Lehrperson geantwortet. Die Rückmeldungen waren sehr positiv. Beispielsweise wurde erwähnt, dass der Halbttag den Lernenden viel Spass gemacht habe, die Anleitungen vor Ort sehr klar waren, die Lernwerkstatt attraktiv gestaltet und abwechslungsreich war. Ausserdem wurde die Betreuung durch die Kursleitungen und Assistierenden positiv hervorgehoben. Die Vorinformationen zur Lernwerkstatt und zu den darin durchgeführten Studien wurden als ausführlich und klar beschrieben. Viele Lehrpersonen haben ihre Klasse mit Robotik- oder allgemein Programmierübungen auf die Lernumgebung vorbereitet. Dies spricht für die Vermutung, dass die Lernenden, welche die Lernwerkstatt besuchten, im Vergleich zu anderen Klassen überdurchschnittlich hohe Programmierkenntnisse hatten. Unabhängig von der eigenen Klassenstufe gaben alle Lehrpersonen an, dass die Lernumgebung stufengerecht war. Lehrpersonen der Begabungs- und Begabtenförderung erwähnten zusätzlich, dass die Aufgaben auch für ihre Lernenden der dritten und vierten Primarklasse lösbar waren. Die verschiedenen Schwierigkeitsgrade der Aufgaben wurden als gute Differenzierungsmöglichkeit innerhalb der Klasse erwähnt. Die Zusammenarbeit mit den Pair Programming Regeln wurde von mehreren Lehrpersonen als positiv hervorgehoben. Die klare Arbeitsteilung habe dazu geführt, dass alle Lernenden, unabhängig ihres Interesses an der Informatik, ihrer Leistungsstärke oder ihres Geschlechts, aktiv waren. Eine Lehrperson erwähnte, dass sich in der Nachbesprechung in der Schule gezeigt habe, dass die Herangehensweisen der geschlechtshomogenen Mädchen- und Jungengruppen völlig unterschiedlich, aber beide zielführend waren. Diese Lehrperson möchte in Zukunft häufiger

geschlechtsheterogene Gruppen machen, weil sie den Eindruck hatte, dass ihre Lernenden motivierter wirkten und die Mädchen besser mitmachten.

Kritisiert wurde vereinzelt, dass die Einführung der Lernumgebung zu lange oder zu theoretisch gestaltet wurde. In einem anderen Fall seien die Erklärungen zu schnell erfolgt. Mehrere Personen erwähnten aber, dass die Lernwerkstatt einen guten Mix aus Theorie und Praxis gefunden habe. Jemand gab als Optimierungsbedarf an, dass ein kurzer Input nach der Pause zusätzlich motiviert hätte. In einem Fall hätte sich die Lehrperson von den zwei Fachpersonen während der selbstständigen Arbeitsphase eine aktivere Unterstützung gewünscht. Andere Lehrpersonen hoben dagegen positiv hervor, dass die Betreuungspersonen nicht zu viel geholfen haben. Die Mehrheit der Lehrpersonen beschrieb die Betreuungspersonen der Lernwerkstatt als freundlich, sehr kompetent, hilfsbereit und wohlwollend. Eine Lehrperson erwähnte, dass für den Abschluss zu viel Zeit eingerechnet wurde. In einem anderen Fall sei die Reflexion zu kurz ausgefallen. Teilweise fehlte den Lehrpersonen der Bezug zur Zukunftsstadt. Andere empfanden den Praxisbezug als sehr gelungen. Diese Rückmeldungen fielen so unterschiedlich aus, weil es verschiedene Kursleitungen und Assistierende gab. Während viele Lehrpersonen angaben, dass der Kurs zu kurz war, gab es auch Stimmen, welche die Programmierphasen oder den ganzen Halbtage als zu lange empfanden. Für Lernende, welche kein grosses Interesse am Programmieren oder an der Robotik als Thema hatten, hätte sich eine Person ein Alternativprogramm gewünscht.

3.6. Erhebungsinstrumente

In diesem Kapitel wird zuerst der Beobachtungsauftrag für die Lehrpersonen und anschliessend das Codebuch, mit welchem die Videoaufnahmen und der Beobachtungsauftrag analysiert wurden, vorgestellt.

3.6.1. Beobachtungsauftrag

Zu Beginn des Kurses erhielt die Lehrperson einen Beobachtungsauftrag (siehe Anhang 2). Das Dokument wurde ausgedruckt auf einem Klemmbrett abgegeben und sollte bis zur Pause ausgefüllt werden. In einem Informationstext wurde die Lehrperson darüber informiert, welche ihrer Schülerinnen und Schüler wann gefilmt wurden, und wie sie sich selbst während der Aufnahmen verhalten sollte. Zunächst musste die Lehrperson kontrollieren, ob an den Kamerastandorten Lernende sassen, welche aufgrund fehlender Einwilligung der Erziehungsberechtigten nicht gefilmt werden durften. Danach wurde darauf hingewiesen, dass es bei der Beobachtung um die Zusammenarbeit der Lernenden ging. Die Lehrperson durfte bei Nachfragen der Schülerinnen und Schüler

aber nur sagen, dass der Umgang mit den Aufgabenstellungen der Lernumgebung untersucht wurde. Ebenfalls wurde erwähnt, dass die Lehrperson die Lernenden an den Kamerastandorten lediglich beobachten, und nur wenn unbedingt nötig eingreifen sollte.

Damit die Zuordnung des Beobachtungsauftrages zu den Videoaufnahmen möglich war, gab die Lehrperson das Datum, den Halbtage (Vor- oder Nachmittag) und die Klassenstufe (Begabungs- und Begabtenförderung, 5./6. Klasse, Oberstufe) an. Die Information der Klassenstufe war zusätzlich wichtig, da es sich dabei möglicherweise um einen Einflussfaktor auf das Pair Programming handelt. Studienergebnisse weisen beispielsweise darauf hin, dass jüngere Kinder eine weniger stereotypisch geprägte Vorstellung von Informatik haben (Barman, 1999). Jeder Kamerastandort wurde mit einer gut sichtbaren Nummer (1-4) ausgestattet, welche die Lehrperson dabei unterstützen sollte, die verschiedenen Gruppen auf dem Beobachtungsauftrag zu unterscheiden. Die Videoaufnahmen wurden mit der gleichen Nummer abgespeichert, damit sie beim Codieren ohne Probleme den Informationen der Lehrpersonen zugeordnet werden konnten. Damit die zwei Lernenden einer Gruppe unterschieden werden konnten, ohne dass ihre Namen genannt wurden, sollte die Lehrperson pro Person ein visuelles Erkennungszeichen notieren (beispielsweise die T-Shirt-Farbe). Dadurch wurden die Schülerinnen und Schüler zu Lernende oder Lernender 1 und Lernende oder Lernender 2.

Die Angabe des Geschlechts war wichtig für die Klassifizierung als geschlechtshomogene männliche, geschlechtsheterogene oder geschlechtshomogene weibliche Paare. Die Lehrperson teilte ihre Schülerinnen und Schüler im Vorfeld mithilfe vorgegebener Kriterien in Zweiergruppen ein (siehe Kapitel 3.3.3. Paareinteilung und -auswahl). Die Kriterien waren so formuliert, dass die Gruppenmitglieder in Bezug auf ihre Leistungsstärke, Programmierkenntnisse und ihr Alter homogen waren, und eine neutrale Beziehung zueinander hatten. In der Forschungsliteratur wird darauf hingewiesen, dass dies Faktoren sein könnten, welche sich auf die Zusammenarbeit auswirken (Zhong et al., 2016; Arisholm et al., 2007; Choi, 2015). In der Praxis ist es meist nicht möglich, bei allen Paarzuteilungen alle Kriterien zu erfüllen. Deshalb wurden diese möglichen Störvariablen zusätzlich mit dem Beobachtungsauftrag erhoben. Die Lehrperson schätzte das allgemeine schulische Leistungsvermögen auf einer vierstufigen Skala von leistungsschwach bis leistungsstark ein. Dabei handelte es sich lediglich um eine grobe Einschätzung der Lehrperson, welche nicht neutral war. Ebenfalls um eine eher ungenaue Angabe handelte es sich bei der Einschätzung der Lehrperson, ob die Lernenden Programmierkenntnisse hatten oder nicht. Der Begriff wurde nicht genauer definiert, weshalb er von den Lehrpersonen unterschiedlich

interpretiert werden konnte. Häufig wurde freiwillig ein Kommentar hinzugefügt, welcher spezifizierte, was die Lernenden bisher gemacht hatten, beispielsweise, dass die ganze Klasse zuvor drei Lektionen mit Scratch gearbeitet hatte. Die Lehrperson schätzte ausserdem die Beziehung der Lernenden auf einer fünfstufigen Skala (negativ, eher negativ, neutral, eher positiv, positiv) ein. Gewisse Lehrpersonen kannten ihre Schülerinnen und Schüler schon seit mehreren Jahren, während andere die Klasse erst vor ein paar Wochen übernommen hatten. Somit konnten nicht alle Lehrpersonen die Einschätzungen gleich präzise vornehmen. Manchmal liessen die Lehrpersonen deshalb Felder leer und schrieben dazu, dass sie die Klasse noch zu wenig gut kannten. Die im Beobachtungsauftrag gesammelten Informationen waren subjektiv geprägt, weshalb sie nur bedingt verlässlich sind.

Als zusätzliche Aufgabe sollte die Lehrperson die Zusammenarbeit der vier Gruppen, welche gefilmt wurden, beobachten. Im Auftrag wurde erwähnt, dass positives, negatives und neutrales Verhalten notiert werden sollte. Die Lehrpersonen kannten ihre Schülerinnen und Schüler und konnten deshalb auch ein allenfalls aussergewöhnliches Verhalten begründen. Hatten die Lernenden beispielsweise gleich vor dem Besuch der Lernumgebung gestritten, könnte dies einen grossen Einfluss auf ihre Zusammenarbeit haben. Vor der Durchführung der Lernwerkstatt wurde ein Testlauf des Beobachtungsauftrages mit mehreren Lehrpersonen gemacht, durch welchen Verbesserungsvorschläge eingebaut und die Machbarkeit der Beobachtung geprüft werden konnten.

3.6.2. Codebuch

Die Videoaufnahmen wurden mithilfe eines Codebuchs analysiert (siehe Anhang 3), welches neben Definitionen, Anweisungen und Kategorien auch Hinweise für das Vorgehen beim Codieren enthielt. Zuerst wurde mit drei Personen ein Testlauf des Codebuchs durchgeführt, welches dann den Rückmeldungen entsprechend angepasst wurde. Danach wurden zwei studentische Hilfskräfte im Rahmen einer einstündigen Sitzung in die Handhabung des Codebuchs eingeführt. Um Verzerrungen der Resultate zu verhindern, wurden die zwei Personen nicht über das genaue Forschungsziel oder die Hypothesen dieser Studie informiert. Am Anfang des Codebuchs wurde mit technischen und organisatorischen Tipps aufgezeigt, wie die Videos mithilfe des *VLC Media Players* analysiert werden konnten. Zur Eingabe der Daten wurde eine vorstrukturierte Excel-Datei abgegeben. Als weitere Hilfsmittel wurden ein Smartphone mit einer Multi-Timer-App, oder zwei Stoppuhren, und ein Notizblock benötigt. Falls auf dem Beobachtungsauftrag oder im Video ein Wert fehlte, oder dieser nicht eindeutig war, wurde im entsprechenden Excel-Feld nichts eingetragen. Mit einem Excel-Kommentar

sollte aber angegeben werden, weshalb dieses Feld nicht ausgefüllt werden konnte. Dadurch war es möglich, fehlende Werte bei der Datenbereinigung nachzuvollziehen, oder unklare Werte zu überprüfen. Beispielsweise sollte ein Feld leer gelassen werden, wenn die Lehrperson im Beobachtungsauftrag bei den Programmierkenntnissen ein Kreuz zwischen *Ja* und *Nein* gesetzt hatte. Fiel den studentischen Hilfskräften während der Codierung etwas auf, hielten sie dies mit einer Notiz fest. Dabei konnte es sich unter anderem um Probleme bei der Codierung, beispielsweise aufgrund einer defekten Tonspur, oder um Zusatzinformationen zur Zusammenarbeit der Lernenden handeln. In der Excel-Datei entsprach eine Zeile einer Videoaufnahme, und somit einer Zweiergruppe. Zuerst wurde die Aufnahmebezeichnung des Videos notiert, welche sich aus dem Datum, dem Halbtage (Vor- oder Nachmittag) und dem Kamerastandort (1-4) zusammensetzte. Danach wurden die Angaben der Lehrperson auf dem Beobachtungsauftrag ins Excel übertragen. Die Notizen der Lehrperson sollten wortwörtlich übernommen werden. Unlesbare Stellen wurden mit einer Klammer und drei Punkten gekennzeichnet.

Die selbstständige Arbeitsphase vor der Pause sollte planmässig 50 Minuten dauern. Da dieses Zeitfenster von der Dauer der Einführung abhing, hatten nicht alle Lernenden gleich viel Zeit, an den Aufgaben zu arbeiten. Je nach Kursleitung verlängerte oder verkürzte sich diese Arbeitsphase. Das wirkte sich auf die Resultate dieser Studie aus, da unter anderem in einem grösseren Zeitfenster auch mehr Aufgaben gelöst werden konnten. Mit dem Codebuch wurde deshalb das Zeitfenster erhoben, welches den Lernenden tatsächlich zur Verfügung stand. Dazu wurden die Zeitpunkte des Startes und des Abschlusses der selbstständigen Arbeitsphase notiert. Der Start wurde so definiert, dass die Einführungsaufgabe abgeschlossen wurde, und die Kursleitung viel Spass beim Programmieren gewünscht hat. Das Ende der selbstständigen Arbeitsphase konnte entweder dann eintreten, wenn die Kursleitung die Arbeit der Lernenden für die Pause unterbrach, oder wenn das Paar die Startaufgabe *Das autonome Fahrzeug* fertig gelöst hatte. Das Lösen der anderen Aufgabensets wurde nicht untersucht.

Damit die für die Beantwortung der ersten Hypothese nötige Lösegeschwindigkeit der Zweiergruppen berechnet werden konnte (siehe Kapitel 3.1. Forschungsfragen und Hypothesen), wurde zusätzlich zur Gesamtarbeitszeit auch die Anzahl der bearbeiteten Aufgaben benötigt. Die letzte bearbeitete Aufgabe konnte aus dem Video durch die Kommunikation der Lernenden und die gerade aufgeklappte Seite des Aufgabensets eruiert werden. Diese Angabe war nicht so genau, da es keine Rolle spielte, ob die Lernenden gerade mit dieser Aufgabe begannen, oder schon kurz vor der Lösung standen. Diese Variante wurde dennoch gewählt, weil es nicht möglich war,

einzuschätzen, wie lange die Lernenden noch für eine Aufgabe gebraucht hätten. In der Excel-Datei wurde die Nummer der zuletzt bearbeiteten Teilaufgabenstellung (1-4) notiert. Arbeitete die Gruppe gerade an der Hauptaufgabenstellung der Übersichtskarte, wurde eine Fünf eingetragen. Wurde das Aufgabenset ganz abgeschlossen, wurde dies mit der Nummer Sechs festgehalten. Auch wenn die Abstände zwischen den Aufgaben nicht gleich gross waren, da sich die Schwierigkeit der Aufträge unterschied, ergab sich eine logische Reihenfolge. Je höher die Zahl war, desto mehr Aufgaben hat die Gruppe in der selbstständigen Arbeitsphase bearbeitet.

Für die Beantwortung der zweiten Hypothese (siehe Kapitel 3.1. Forschungsfragen und Hypothesen) wurde gezählt, wie oft die Lernenden eine Hilfestellung, beispielsweise durch ein Handzeichen, anforderten und auch erhielten. Eine Hilfestellung konnte durch eine Betreuungsperson (Kursleitung, Assistenz, Lehrperson oder Begleitperson) oder andere Lernende gegeben werden. Hilfestellungen, welche nicht aktiv eingefordert wurden, zählten nicht dazu. Auch wenn die Lehrpersonen und die Betreuungspersonen darüber informiert wurden, dass sie während der Videoaufnahmen bei den gefilmten Paaren möglichst nicht eingreifen sollten, kam es dennoch vor, dass proaktiv geholfen wurde. Eine angeforderte, aber nicht erhaltene Hilfe wurde ebenfalls nicht gezählt, beispielsweise wenn sich das Team mit Handzeichen meldete, aber niemand zu Hilfe kam. Hilfestellungen bei technischen Problemen, welche die Lernenden nicht selbst ausgelöst hatten, wurden auch nicht gezählt. Dazu gehörten beispielsweise Verbindungsprobleme mit dem Roboter oder das Abstürzen der Software.

Damit für die fünfte Hypothese geprüft werden konnte, ob eine Geschlechterzusammensetzung häufiger die Rollen wechselte (siehe Kapitel 3.1. Forschungsfragen und Hypothesen), musste die Anzahl Rollen-, beziehungsweise Platztäusche erhoben werden. Da es für die dritte und sechste Hypothese entscheidend war, wie lange die Lernenden jeweils die Rollen Navigator und Driver übernahmen (siehe Kapitel 3.1. Forschungsfragen und Hypothesen), waren sowohl die Anzahl als auch die Zeitpunkte der Rollenwechsel relevant. Deshalb sollte der Zeitpunkt jedes Platztausches im Format hh:mm:ss notiert werden. Einzelne Kursleitungen initiierten einen ersten Platztausch direkt nach der Einführungsaufgabe. Dieser Platztausch wurde im Manual nicht vorgegeben und nur von wenigen Kursleitungen gemacht. Da der Platztausch vor dem Beginn der selbstständigen Arbeitsphase erfolgte, wurde er nicht gezählt. Als Platztausch-Zeitpunkt galt der Moment, in welchem sich die erste Person auf den anderen Stuhl setzte. Wenn sich die Lernenden aus Versehen auf den falschen Stuhl setzten und gleich zurücktauschten, zählte dies nicht als Platztausch.

Um die Hypothesen 3, 4, 6 und 7 überprüfen zu können (siehe Kapitel 3.1. Forschungsfragen und Hypothesen), musste gezählt werden, wie häufig und wie lange die Lernenden eingriffen, während sie die Navigatoren-Rolle innehatten. Damit eine Person überhaupt eingreifen konnte, musste sie während der selbstständigen Arbeitsphase mindestens einmal die Navigatoren-Rolle übernehmen. Je länger jemand diese Rolle innehatte, desto höher konnte die Anzahl Eingriffe werden. Deshalb war es entscheidend, wie lange die Lernenden diese Rolle übernommen hatten. Um dieses Zeitfenster erheben zu können war zunächst wichtig, welche Rolle die Lernenden beim Start einnahmen. Die Person, welche am Anfang rechts am Tisch am Computer sass, begann in der Driver-Rolle, die Person links beim Roboter in der Navigatoren-Rolle. Durch die Kombination der Startrolle mit den Platztauschzeitpunkten, sowie den Start- und Schlusszeiten der selbstständigen Arbeitsphase konnte später ausgerechnet werden, wie lange die Lernenden die Rollen Driver und Navigator innehatten.

Die Eingriffe in der Navigatoren-Rolle wurden für beide Lernenden einzeln gezählt. Als Eingriff galt entsprechend den Regeln des Pair Programmings (siehe Kapitel 2.2.2. Pair Programming Leitlinien), wenn die Person in der Navigatoren-Rolle die Computermaus oder -tastatur berührte. Eine Berührung des Computerbildschirmes galt nicht als Eingriff, weil die Person in der Navigatoren-Rolle nicht nur auf den Bildschirm zeigen durfte, sondern dies laut den Pair Programming Leitlinien sogar tun sollte. Dabei konnte aus Versehen der Bildschirm berührt werden. Mit einer Ausnahme wurden für die Datenerhebung an den Kamerastandorten keine Geräte mit Touchscreens platziert. Das irrtümlich falsch platzierte Gerät mit Touchscreen wurde nach ein paar Wochen ausgetauscht. Die Videos, welche damit gefilmt wurden, analysierten die studentischen Hilfskräfte mit dem Wissen, dass auch über den Touchscreen eingegriffen werden konnte. Ein Eingriff begann, sobald die Computermaus oder -tastatur berührt wurde und endete, wenn die Hände klar zurückgezogen wurden. Erfolgte gleich danach noch eine Berührung, zählte dies als neuer Eingriff. Die Dauer der Eingriffe der Lernenden wurde mit einem Multi-Timer-App gemessen. Nach dem Codieren des Videos wurde die Gesamtzeit aller Eingriffe der Lernenden notiert. Zusammen mit der Dauer der Übernahme der Rolle Navigator, konnte die Anzahl Eingriffe pro Navigatoren-Stunde berechnet werden.

3.7. Datenauswertung

In diesem Kapitel werden zuerst das Vorgehen bei der Datenbereinigung, sowie bei der Berechnung der Intercoder-Reliabilität beschrieben. Dann wird dargestellt, wie und weshalb der ursprüngliche Datensatz umstrukturiert wurde. Anschliessend werden die

gefilmten Gruppen mit deskriptiver Statistik beschrieben. Darauf folgen Ausführungen zur Variablenberechnung, sowie zur Prüfung des Einflusses möglicher Störvariablen.

3.7.1. Datenbereinigung

Zuerst wurden die unklaren oder fehlenden Daten betrachtet, welche von den studentischen Hilfskräften mit Excel-Kommentaren markiert wurden. Waren Zusatzinformationen der Lehrperson auf dem Beobachtungsauftrag vorhanden, beispielsweise, dass die ganze Klasse Scratch-Vorkenntnisse hatte, wurde dies dem Lehrpersonenkommentar hinzugefügt. Wenn die Lehrperson zwei Kreuze statt eines, oder ein Kreuz zwischen zwei Kästchen gesetzt hatte, wurde diese Angabe gelöscht. Der Grund für das Fehlen dieses Wertes wurde im Lehrpersonenkommentar ergänzt. Gab es zusätzlich zu einem angekreuzten Feld ein Pfeil in eine Richtung, wurde der Standort des Kreuzes übernommen. War neben einem richtigen Kreuz zusätzlich ein Kreuz eingeklammert, wurde das nicht eingeklammerte Kreuz gezählt. In beiden Fällen wurde dieser Entscheid im Lehrpersonenkommentar protokolliert. Wurde bei den Programmiervorkenntnissen statt eines Kreuzes notiert, welche Programmiersprachen oder Roboternmodellen die Lernenden bereits kannten, wurde dies als Vorhandensein von Programmiervorkenntnissen gezählt. Auch diese Anpassung wurde im Lehrpersonenkommentar festgehalten. In den Codier-Notizen hielten die studentischen Hilfskräfte fest, falls es Probleme bei der Codierung oder Zusatzinformationen zur Zusammenarbeit der Lernenden gab. Für die Schwierigkeiten, welche gehäuft vorkamen, wurden neue Variablen erstellt (siehe Tabelle 1). Bei diesen Variablen wurde eine Eins eingetragen, wenn diese Problematik auf das Video zutraf, und eine Null, wenn nicht. In der Tabelle 1 wird aufgeführt, bei wie vielen der gefilmten Gruppen diese Problematiken auftauchten.

Tabelle 1 Neue Variablen aus Codier-Notizen

Variablenname	Beschreibung	Häufigkeit
Eingriffe durch andere	Viele nicht angeforderte Hilfestellungen oder Eingriffe durch andere Paare, durch die Lehrperson oder Betreuungspersonen der Lernwerkstatt.	18
Anweisung Platztausch	Die Lernenden wurden von der Lehrperson oder Betreuungspersonen der Lernwerkstatt auf den Rollenwechsel oder andere Pair Programming Regeln aufmerksam gemacht.	8
Keine Zusammenarbeit	Eine Person der Gruppe hat gar nichts gemacht, beide arbeiteten nicht an den Aufgaben oder sie arbeiteten nicht zusammen.	3
Lange Abwesenheit	Eine Person oder beide Personen waren lange abwesend.	9
Codier-Schwierigkeiten	In den Codier-Notizen wurde erwähnt, dass das Video schwierig zu codieren war. Dies konnte verschiedene Gründe haben:	34

	<ul style="list-style-type: none"> • Technische Probleme (beispielsweise Tondefekte, Personen vor der Kamera, Eingriffe über den Touchscreen) • Unklarheiten beim Ablauf des Kurses (beispielsweise, wenn der Start- oder Endpunkt der selbstständigen Arbeitsphase nicht klar zu ermitteln waren) • Schwierig auszuwertende Zusammenarbeit (beispielsweise, wenn die Person rechts am Tisch fälschlicherweise die Navigatoren-Rolle übernahm) 	
--	---	--

Einige dieser Punkte hätten die Resultate dieser Studie verfälschen können. Beispielsweise wirkte sich die lange Abwesenheit einer Person oder beider Personen auf die Anzahl Eingriffe und die Dauer der Rollenübernahme aus. Deshalb wurde bei jedem Fall geprüft, ob gewisse Daten dieser Zweiergruppe gelöscht werden mussten, und somit nicht in die Auswertung miteinfließen sollten. Falls das Video doppelt codiert wurde, und die Codier-Schwierigkeit bei der zweiten Person nicht auftauchte, wurden keine Daten gelöscht. War die Ton- oder Videospur defekt, wurde überprüft, ob die Daten realistisch waren oder sogar mit einer doppelten Codierung gesichert werden konnten. Handelte es sich um eine kleine Verzerrung, beispielsweise wenn die Gruppe nur kurz abwesend war, wurden keine Daten gelöscht. Bei den 18 Videos (siehe Tabelle 1), in welchen die Lernenden viele nicht angeforderte Hilfestellungen durch die Lehrperson, Betreuungspersonen der Lernwerkstatt oder andere Gruppen erhielten, wurde der Wert der Variable *Letzte Aufgabe* gelöscht. In den acht Fällen, in welchen die Lernenden auf den Rollenwechsel oder andere Pair Programming Regeln aufmerksam gemacht wurden, erfolgte eine Löschung der Platztauschzeitpunkte, sowie der Anzahl und der Dauer der Eingriffe. In einem Fall übernahm die Person rechts am Tisch fälschlicherweise die Rolle des Navigators, und die Person links die des Drivers. Dort wurden die Anzahl und Dauer der Eingriffe, sowie die Platztauschzeitpunkte gelöscht. Insgesamt gab es neun Gruppen, für welche aufgrund dieser Löschungen keine Berechnung der Eingriffe in der Navigatoren-Rolle gemacht werden konnten. Dies betraf die Beantwortung der Hypothesen 3, 4, 6 und 7. Bei diesen neun Ausfällen handelte es sich um drei Mädchen-, vier Jungen- und zwei geschlechtsheterogene Paare. Somit traten solche Problematiken nicht nur bei einer Geschlechterzusammensetzung auf. Im nächsten Schritt wurden die Minimal- und Maximalwerte, sowie die fehlenden Werte der Variablen überprüft. Extreme Werte, beispielsweise ein unpassendes Geburtsdatum, wurden überprüft und allenfalls korrigiert.

3.7.2. Intercoder-Reliabilität

Um die Intercoder-Reliabilität, also das Ausmass der Übereinstimmungen der Codierungen der zwei studentischen Hilfskräfte, zu überprüfen, wurden 20% der Videos

doppelt codiert. Die 40 doppelt zu codierenden Videos wurden zufällig ausgewählt, wobei darauf geachtet wurde, dass von allen Klassenstufen und Geschlechterzusammensetzungen ungefähr gleich viele Videos dabei waren. Damit die Codierungen der studentischen Hilfskräfte unterschieden werden konnten, wurde eine Variable namens *Codiererin oder Codierer* hinzugefügt. Zuerst wurden die Werte aller Variablen ausser der abhängigen Variablen von Hand geprüft. Bei grösseren Unterschieden wurde das Video erneut codiert. Gab es bei den Werten des Beobachtungsauftrages Unterschiede, wurden die Codierungen mit dem Originaldokument verglichen.

Die Übereinstimmung der Werte, welche für die Hypothesen entscheidend waren, wurde mit der Intraklassenkorrelation (ICC) geprüft. Dabei handelt es sich um ein Zusammenhangsmass, welches zeigt, wie sehr sich zwei oder mehr Personen in ihren Urteilen einig sind. Somit schätzt die Intraklassenkorrelation die Reliabilität der studentischen Hilfskräfte ein (Wirtz & Caspar, 2002). Damit die Intraklassenkorrelation durchgeführt werden darf, müssen die abhängigen Variablen intervallskaliert, und die unabhängige Variable kategorial und deren Stufen unabhängig sein. Es sollen mindestens zehn Fälle von denselben Personen bewertet werden. Diese Voraussetzungen waren erfüllt. Zusätzlich müssen die abhängigen Variablen normalverteilt sein, Varianzhomogenität zwischen den Stufen der unabhängigen Variablen muss gegeben sein, und es darf keine Interaktion zwischen den Raterinnen und Ratern und bewerteten Personen vorliegen. Die dafür nötigen Tests wurden durchgeführt und werden nachfolgend beschrieben. Alle dazugehörigen Tabellen befinden sich im Anhang 4 dieser Arbeit.

Zuerst wurde geprüft, ob die abhängigen Variablen normalverteilt waren. Dazu wurde ein Shapiro-Wilk-Test verwendet (siehe Anhang 4.1). Da die Signifikanz bei allen abhängigen Variablen kleiner als .05 war, wurde davon ausgegangen, dass die Daten nicht normalverteilt waren. Weil die Stichprobe hinreichend gross war, konnte jedoch vom zentralen Grenzwertsatz und damit einer relativen Robustheit des Tests hinsichtlich der Verletzung dieser Voraussetzung ausgegangen werden (Wirtz & Caspar, 2002). Um zu überprüfen, ob Varianzhomogenität zwischen den Stufen der unabhängigen Variablen gegeben war, wurde ein Levene-Test basierend auf dem Median durchgeführt (siehe Anhang 4.2). Der Levene-Test wurde für keine der abhängigen Variablen signifikant ($\alpha=.25$, auf Empfehlung von Wirtz & Caspar, 2002, S. 180). Somit konnte von Varianzhomogenität ausgegangen werden. Mit dem Tukey-Additivitätstest wurde überprüft, ob eine Interaktion zwischen den Raterinnen und Ratern und bewerteten Personen vorlag (siehe Anhang 4.3). Mit Ausnahme der abhängigen Variablen *Letzte*

Aufgabe und *Anzahl Platztäusche* waren alle Werte unterhalb des von Wirtz und Caspar (2002, S. 210) empfohlenen Signifikanzwertes von $\alpha=.25$. Somit hatten signifikante Interaktionen zwischen den Raterinnen und Ratern und bewerteten Personen stattgefunden. Da eine Verletzung dieser Annahme vorlag, wurde geprüft, ob die Trennschärfen der Raterinnen und Ratern insgesamt ähnlich waren (siehe Anhang 4.4). Die Werte der zwei studentischen Hilfskräfte waren alle hoch ausgeprägt und immer gleich. Deshalb konnte davon ausgegangen werden, dass die Raterinnen und Rater insgesamt recht homogen urteilten. Laut Wirtz und Caspar (2002, S. 212) darf die Intraklassenkorrelation trotz der signifikanten Interaktionen zwischen den Raterinnen und Ratern und bewerteten Personen berechnet werden, wenn die Trennschärfen für alle Personen homogen sind. Dabei muss beachtet werden, dass in diesem Fall die Intraklassenkorrelation die wahre Reliabilität systematisch unterschätzt (Shrout & Fleiss, 1979; Asendorpf & Wallbott, 1979).

Die Prüfung der Voraussetzung zeigte, dass die Intraklassenkorrelation durchgeführt werden durfte. Es wurde das zweifaktorielle Modell der Intraklassenkorrelation gewählt, weil für alle 40 Videos Beurteilungen beider Personen vorhanden waren. Die Auswahl der studentischen Hilfskräfte erfolgte mit der Präferenz für Assistierende der Lernumgebung. Da es sich hierbei nicht um eine zufällige Auswahl handelte, wurde das zweifach gemischte Modell gewählt. Es wurde davon ausgegangen, dass es einen systematischen Fehler gab, weshalb die Absolute Übereinstimmung überprüft wurde. Berechnungsgrundlage sollten die einzelnen Rohwerte der Raterinnen und Rater, keine Mittelwerte, sein, weshalb *single measure* gewählt wurde. Entsprechend dieser Entscheidungen wurde der *ICC (3,1), two-way mixed, single measure absolute agreement* berechnet.

Der Intraklassenkorrelationskoeffizient (Spalte *Korrelation innerhalb der Klasse* in der Tabelle 2) fiel für die Variable *L2DauerEingriffe* (Dauer der Eingriffe der Lernenden oder des Lernenden 2) mit $ICC=.767$ und dem Konfidenzintervall $[.603,.869]$ am tiefsten, und für die Variable *L1DauerEingriffe* mit $ICC=.994$ und einem Konfidenzintervall von $[.988,.997]$ am höchsten aus. Die Anweisungen im Codebuch für die Codierung der zwei Variablen *L1DauerEingriffe* und *L2DauerEingriffe* waren dieselben. Ein Blick in die Daten zeigte, dass in zwei Fällen eine studentische Hilfskraft eine lange Dauer eintrug, während die andere Person gar keinen Eingriff zählte. In den Codier-Notizen wurde erwähnt, dass es bei einer Gruppe zwar einen physischen Platztausch gab, die Rollen aber nie gewechselt wurden. Dies erklärt, weshalb sich die studentischen Hilfskräfte uneinig waren. Im anderen Fall war eine Lernende oder ein Lernender kurz abwesend, und die andere Person bediente trotz dem Innehaben der Navigatoren-Rolle den

Computer. Da dieser Fall im Codebuch nicht explizit erwähnt wurde, wertete eine studentische Hilfskraft diese Situation als Eingriff, und die andere als Einzelarbeit. Solche unklaren Situationen gab es vermutlich auch bei den nicht doppelt codierten Videos. Diese Fälle konnten aber mit den Codier-Notizen ausfindig gemacht werden (siehe Kapitel 3.7.1. Datenbereinigung). In der Literatur wird davon ausgegangen, dass die Übereinstimmung von Raterinnen und Ratern ab einem Wert von 0.7 gut ist (Greve & Wentura, 1997; Greguras & Robie, 1998). Da alle Werte höher waren als dieser Richtwert, kann davon ausgegangen werden, dass die Übereinstimmung der studentischen Hilfskräfte zufriedenstellend ist. Der Intraklassenkorrelationskoeffizient wurde für alle abhängigen Variablen signifikant (siehe Spalte *Sig.* in der Tabelle 2).

Tabelle 2 Intraklassenkorrelationskoeffizient

Abhängige Variable	Korrelation innerhalb der Klasse	95%-Konfidenzintervall		F-Test mit wahrem Wert 0			
		Untergrenze	Obergrenze	Wert	df1	df2	Sig.
L1 Dauer Eingriffe	.994	.988	.997	405.883	39	39	.000
L1 Anzahl Eingriffe	.984	.967	.992	145.138	39	39	.000
Anzahl Platztäusche	.931	.873	.963	27.350	39	39	.000
Letzte Aufgabe	.919	.852	.956	23.373	39	39	.000
L2 Anzahl Eingriffe	.895	.810	.943	18.519	39	39	.000
Anzahl Hilfestellungen	.881	.785	.936	16.669	39	39	.000
L2 Dauer Eingriffe	.767	.603	.869	7.576	39	39	.000

Nach der Berechnung der Intercoder-Reliabilität wurde eine neue SPSS-Datei angelegt. Bei den doppelt codierten Videos wurde jeweils eine Codierung zufällig ausgewählt. Dabei wurde darauf geachtet, dass beide studentischen Hilfskräfte je die Hälfte der Videos codiert hatten.

3.7.3. Umstrukturierung des Datensatzes

Im Datensatz entsprach eine Zeile einer gefilmten Zweiergruppe. Es wurden aber auch Daten pro Schülerin oder Schüler gesammelt. Weil diese zwei Personen in einer Gruppe zusammenarbeiteten, beeinflussten sie sich gegenseitig und waren somit statistisch voneinander abhängig. In einer solchen sogenannten Dyade können die Personen laut Kenny et al. (2006) unterscheidbar oder nicht unterscheidbar sein. In geschlechtsheterogenen Gruppen konnten die Lernenden anhand ihres Geschlechts eindeutig unterschieden werden. Bei den geschlechtshomogenen Gruppen handelte es sich um Dyaden, welche nicht alle einheitlich anhand eines Merkmals unterschieden werden konnten. Die Lehrperson teilte den Schülerinnen und Schülern mit dem Ziel der Anonymisierung im Beobachtungsauftrag willkürlich die Bezeichnungen Lernende oder Lernender 1 (L1) und Lernende oder Lernender 2 (L2) zu. Während bei den geschlechtshomogenen Paaren L1 und L2 immer dasselbe Geschlecht hatten, konnte

bei geschlechtsheterogenen Paaren L1 männlich oder weiblich sein. Um die Auswertungen zu vereinfachen, wurde der Datensatz so umstrukturiert, dass bei den geschlechtsheterogenen Gruppen immer L1 männlich und L2 weiblich ist. Dazu wurden alle Angaben der Lernenden getauscht (Geburtsdaten, Anzahl Eingriffe, etc.).

3.7.4. Beschreibung der Gruppen

Die 55 verfügbaren Termine der Lernumgebung wurden innerhalb von zwei Wochen nach Erstausschreibung gebucht. Pandemiebedingt konnten zwei Schulklassen kurzfristig aufgrund eines Exkursionsverbots der Schule nicht teilnehmen. Über die Hälfte der Klassen waren 5.-6.-Klassen (29 von 53), etwas weniger als ein Drittel waren 7.-9.-Klassen (16 von 53). Bei den restlichen acht Klassen handelte es sich um Begabungs- und Begabtenförderungsprogramme der 3.-6.-Primarstufe (8 von 53). Erwartungsgemäss handelte es sich mehrheitlich um Luzerner Schulklassen (34 von 53 insgesamt, davon 9 Stadtschulen). Es waren aber auch Klassen aus allen Zentralschweizer Kantonen ausser Uri, sowie zwei Klassen aus dem Aargau dabei.

Bei 53 Klassen mit je vier Kamerastandorten wären theoretisch 212 Aufnahmen möglich gewesen. Aufgrund eines Medienbesuches, zu kleinen Klassengrössen, oder technischen Problemen, fielen neun Aufnahmen weg. Insgesamt konnten 203 Videos ausgewertet werden. Die Verteilung der gefilmten Gruppen auf die Klassenstufen lässt sich aus Tabelle 3 entnehmen. Die oder der jüngste Lernende auf den Videos war acht Jahre, die oder der älteste Lernende 15 Jahre alt. Die gefilmten Gruppen waren gleichmässig auf geschlechtshomogene weibliche, geschlechtsheterogene und geschlechtshomogene männliche Paare verteilt (siehe Tabelle 4). Bei den Programmiervorkenntnissen der Lernenden zeigte sich, dass etwa die Hälfte bereits programmiert hatte (190 von 406 Lernenden), während die andere Hälfte der Lernenden (202 von 406) noch kein Vorwissen hatte. In 14 Fällen fehlte diese Angabe. Wie im Kapitel 3.3.2. *Rekrutierung von Schulklassen* erwähnt, wurde vermutet, dass die Lernenden in der Lernwerkstatt im Vergleich zu anderen Klassen überdurchschnittlich viel Vorwissen hatten.

Tabelle 3 Verteilung der gefilmten Gruppen auf die Klassenstufen

Klassenstufe	Häufigkeit	Prozent
5.-6. Klasse	115	56.7%
7.-9. Klasse	60	29.5%
Begabungs- und Begabtenförderung (3.-6. Klasse)	28	13.8%
Gesamt	203	100%

Tabelle 4 Verteilung der gefilmten Gruppen auf die Geschlechterzusammensetzungen

Geschlechterzusammensetzung	Häufigkeit	Prozent
geschlechtshomogene weibliche Gruppe	68	33.5%
geschlechtsheterogene Gruppe	71	35.0%
geschlechtshomogene männliche Gruppe	64	31.5%
Gesamt	203	100%

3.7.5. Berechnung und Beschreibung der unabhängigen und abhängigen Variablen

In diesem Kapitel wird erläutert, wie die unabhängigen und abhängigen Variablen für die Hypothesenprüfung berechnet wurden.

Unabhängige Variable

Die unabhängige Variable Geschlechterzusammensetzung wurde aus dem von der Lehrperson auf dem Beobachtungsauftrag angegebenen biologischen Geschlecht der beiden Lernenden abgeleitet. Waren es zwei Mädchen, handelte es sich um ein geschlechtshomogenes weibliches Paar. Arbeitete ein Mädchen mit einem Jungen zusammen, war es ein geschlechtsheterogenes Paar. Waren es zwei Jungen, handelte es sich um eine geschlechtshomogene männliche Gruppe.

Gesamtarbeitszeit

Die Paare konnten nicht alle gleich lange an den Aufgaben arbeiten. Im Manual wurde vorgegeben, dass der gefilmte Block der selbstständigen Arbeitsphase 50 Minuten dauern sollte. Dieses Zeitfenster konnte sich verlängern oder verkürzen, je nachdem wie die Schülerinnen und Schüler arbeiteten. In einem ersten Schritt wurde deshalb die tatsächliche Gesamtarbeitszeit, welche der Gruppe zur Verfügung stand, mit der Start- und Endzeit berechnet. Im Durchschnitt konnten die Lernenden etwas mehr als 45 Minuten an den Aufgaben arbeiten ($M= 00:46:11$; $SD= 0:08:20$). Das grösste Zeitfenster der selbstständigen Arbeitsphase dauerte über eine Stunde (Maximum= 1:09:20). Die Arbeitsphase endete im Rahmen dieser Studie allerdings nicht nur, wenn die Kursleitung diese beendete, sondern auch, falls die Gruppe die Startaufgabe fertig gelöst hatte. So lässt sich auch der kleinste Wert der Gesamtarbeitszeit von weniger als 30 Minuten (Minimum= 0:26:35) erklären.

Lösegeschwindigkeit

Da die Gruppen nicht alle gleich viel Zeit zum Lösen der Aufgaben zur Verfügung hatten, musste die zuletzt bearbeitete Aufgabe mit der Gesamtarbeitszeit kombiniert werden. Die Lösegeschwindigkeit, also wie viele Aufgaben eine Gruppe durchschnittlich pro Stunde bearbeitet hatte, konnte durch diese Kombination berechnet werden. Dafür wurde die Angabe der letzten bearbeiteten Aufgabe durch die Gesamtarbeitszeit geteilt und dann mit 3600 multipliziert. In dieser Studie bearbeitete die langsamste Gruppe durchschnittlich 2.80 Aufgaben pro Stunde, und die schnellste 13.54. Der Mittelwert lag bei 5.93 bearbeiteten Aufgaben pro Stunde mit einer Standardabweichung von 1.43.

Angeforderte Hilfestellungen

Die Gesamtarbeitszeit hatte auch einen Einfluss auf die Anzahl angeforderten Hilfestellungen, und wurde deshalb gleich wie bei der Lösegeschwindigkeit miteingerechnet. Dabei wurde die Anzahl Hilfestellungen durch die Gesamtarbeitszeit geteilt und das Resultat mit 3600 multipliziert. Dadurch konnte eine Aussage dazu getroffen werden, wie viele Hilfestellungen eine Gruppe durchschnittlich pro Stunde angefordert hatte. Da es Gruppen gab, welche keine Hilfestellungen angefordert hatten, lag das Minimum bei null. Die Gruppe mit dem höchsten Wert forderte durchschnittlich 17.02 Hilfestellungen pro Stunde an. Der Mittelwert betrug 4.18 mit einer Standardabweichung von 3.33.

Navigator- und Driver-Zeit

Die Anzahl Eingriffe der Lernenden hing damit zusammen, ob und wie lange sie die Navigatoren-Rolle innehatten. Deshalb wurden mit den Platztauschzeitpunkten und der Startrolle der Lernenden die Navigator- und Driver-Zeiten der einzelnen Personen berechnet. Startete eine Schülerin beispielsweise in der Navigatoren-Rolle und die Gruppe tauschte zweimal die Rollen, wurde die Dauer zwischen dem Start der selbstständigen Arbeitsphase bis zum ersten Platztausch und zwischen dem zweiten Platztausch und dem Ende der selbstständigen Arbeitsphase summiert, um ihre Navigator-Zeit zu berechnen. Es gab sowohl Lernende, welche nie die Navigatoren-Rolle innehatten, als auch Personen, welche diese Rolle sehr lange übernahmen (Minimum= 0:00:00; Maximum=1:01:07). Der Mittelwert lag bei 0:23:03 mit einer Standardabweichung von 0:13:47. Somit hatten bei den meisten Gruppen während des geplanten Zeitrahmens der selbstständigen Arbeitsphase von 50 Minuten beide Lernenden etwa gleich lange die Navigatoren-Rolle inne. Wurden nur die geschlechtsheterogenen Gruppen ($n= 69$) betrachtet, zeigte sich, dass die Jungen ($M= 0:21:28$; $SD= 0:11:38$) und Mädchen ($M= 0:24:17$; $SD= 0:13:42$) ungefähr gleich lange

die Navigatoren-Rolle übernommen hatten. Bei fünf geschlechtsheterogenen Gruppen übernahm der Junge nie die Navigatoren-Rolle, in vier Fällen das Mädchen. Somit konnte bezüglich Navigator-Zeit kein grosser Unterschied zwischen Mädchen und Jungen in geschlechtsheterogenen Gruppen nachgewiesen werden.

Anzahl Eingriffe in der Navigatoren-Rolle

Die Anzahl Eingriffe wurde mit Einbezug der Navigator-Zeit pro Person berechnet, so dass eine Aussage darüber gemacht werden konnte, wie häufig pro Navigatoren-Stunde eingegriffen wurde. Fälle, in welchen eine Person nie die Rolle Navigator innehatte, flossen nicht in diese Variable mit ein. Nahm diese Variable den Wert Null an, bedeutete dies, dass diese Person zwar mindestens einmal die Navigatoren-Rolle übernahm, aber nie eingegriffen hatte. Dies war bei etwa einem Viertel der Lernenden der Fall (87 von 354). Die Person, welche die höchste Anzahl Eingriffe aufwies, war weiblich und tat dies 99.65-mal pro Navigatoren-Stunde. Allerdings hatte diese Person nur etwa fünf Minuten diese Rolle inne, und griff dabei acht Mal ein. Diese durch die Wahl des Datenauswertungsverfahrens entstandenen extremen Werte waren problematisch und werden im Kapitel 5.2. *Diskussion der Methode* genauer betrachtet. Durchschnittlich wurde 11.40-mal pro Navigatoren-Stunde eingegriffen ($SD= 15.30$).

Durchschnittliche Dauer eines Eingriffs in der Navigatoren-Rolle

Zur Berechnung der durchschnittlichen Dauer eines Eingriffs der Lernenden, wurde pro Person die Dauer der Eingriffe durch die Anzahl Eingriffe geteilt. Fälle, in welchen gar nicht eingegriffen wurde, flossen nicht in diese Variable mit ein. Die kürzeste durchschnittliche Dauer lag bei 0:00:01, die längste bei 0:02:30. Der Mittelwert betrug 0:00:11 mit einer Standardabweichung von 0:00:13.

Anzahl Rollenwechsel

Mit der Anzahl Platztäuschen wurde berechnet, wie häufig die Gruppen die Rollen gewechselt hatten. Teil der Instruktion für das Pair Programming war, dass die Plätze nach jeder Teilaufgabenstellung gewechselt werden sollten. Arbeitete eine Gruppe am Ende der selbstständigen Arbeitsphase gerade an der dritten Aufgabe, müssten die Lernenden zwei Platztäusche gemacht haben. Somit handelte es sich beim Sollwert der Rollenwechsel um die Zahl bei der letzten Aufgabe minus Eins. Diese Regel galt nur bei den Gruppen nicht, welche das Aufgabenset komplett gelöst hatten. Dort haben die studentischen Hilfskräfte eine sechs eingetragen. Der fünfte Platztausch hätte erst beim Beginn des nächsten Aufgabensets erfolgen müssen. Allerdings stoppte die Beobachtung nach dem Lösen der Übersichtskarte. Deshalb waren im Falle des

Abschlusses des untersuchten Aufgabensets nur vier Platztäusche und nicht fünf nötig. Die tatsächlich getätigten Platztäusche wurden von diesem Sollwert subtrahiert. Negative Werte dieser Variable gaben an, wie viele Rollenwechsel zu wenig, und positive Werte wie viele Rollenwechsel zu viel gemacht wurden. Eine Null zeigte an, dass die Anzahl Platztäusche dem Sollwert entsprach. Fast die Hälfte der Gruppen führte die korrekte Anzahl Rollenwechsel durch (77 von 176). Es gab Gruppen, welche vier Platztäusche zu wenig (Minimum) oder sechs zu viel (Maximum) machten. Der Mittelwert lag bei -.44 mit einer Standardabweichung von 1.84. Die meisten Gruppen hielten sich somit an diese Anweisung, tendenziell wurden aber weniger Platztäusche als gefordert gemacht.

Summe der Anzahl Eingriffe und Dauer der Eingriffe im Paar

Damit eingeschätzt werden konnte wie häufig die Lernenden einer Gruppe insgesamt eingegriffen hatten, wurden die Eingriffe der einzelnen Personen summiert. Fälle, in welchen eine Person nie die Navigatoren-Rolle innehatte, wurden nicht gewertet. Auch für die durchschnittliche Dauer eines Eingriffs in der Gruppe wurden die Einzelwerte der Lernenden summiert und Fälle, in welchen eine Person nie die Navigatoren-Rolle übernommen hatte, nicht gewertet. Bei 8.13% der Gruppen (13 von 160) griffen beide Lernenden nie ein. In der Gruppe mit den meisten Eingriffen wurde insgesamt 137.58-mal pro Navigatoren-Stunde eingegriffen. Der Mittelwert lag bei 21.77 mit einer Standardabweichung von 22.41. Die tiefste Summe der durchschnittlichen Dauer eines Eingriffs einer Gruppe lag bei 0:00:02, die höchste bei 0:02:39. Der Mittelwert betrug 0:00:23 mit einer Standardabweichung von 0:00:21.

3.7.6. Berechnung und Beschreibung weiterer Variablen

In diesem Kapitel wird gezeigt, wie die möglichen Störvariablen berechnet wurden. Die Begründung, weshalb vermutet wurde, dass diese Faktoren einen Einfluss auf die Resultate hatten, kann im Kapitel 3.3.3. *Paareinteilung und -auswahl* nachgelesen werden. Die Beschreibung der Variablen wird von Tabellen ergänzt, welche im Anhang 5 zu finden sind.

Altersunterschied

Der Altersunterschied der Lernenden wurde mit den von der Lehrperson auf dem Beobachtungsauftrag angegebenen Geburtsdaten berechnet. Danach wurden der kleinste und der grösste Altersunterschied, der Mittelwert, sowie die Häufigkeiten der Altersunterschiede ermittelt. Entsprechend dieser Daten wurden Kategorien des Altersunterschiedes gebildet, welche ungefähr gleich viele Fälle umfassten (siehe

Anhang 5.1). Die Vorgabe an die Lehrpersonen, dass die Lernenden einer Gruppe möglichst maximal ein Jahr Altersunterschied haben sollten, wurde fast immer umgesetzt (162 von 188; 86.17%). Der grösste Altersunterschied in einer Gruppe betrug 3.8 Jahre, der Mittelwert .55 Jahre mit einer Standardabweichung von .49. Bei einer Gruppe hatten die Lernenden den genau gleichen Geburtstag.

Leistungsunterschied

Aus den von der Lehrperson auf dem Beobachtungsauftrag angegebenen allgemeinen schulischen Leistungsniveaus der Lernenden einer Gruppe wurde ihr Leistungsunterschied berechnet und in Kategorien eingeteilt. Unterschieden sich die Einschätzungen nicht, wurde ein gleiches allgemeines schulisches Leistungsniveau angenommen. War das Kreuz um eine Position verschoben, galt dies als ähnliches allgemeines schulisches Leistungsniveau. Gab es eine Verschiebung um zwei Positionen, wurde dies als unterschiedliches allgemeines schulisches Leistungsvermögen interpretiert. Wurde der maximale Unterschied von drei Positionen erreicht, handelte es sich um ein stark unterschiedliches allgemeines schulisches Leistungsvermögen. Die Verteilung der Gruppen auf diese Kategorien ist im Anhang 5.3 ersichtlich. Die Vorgabe der Paareinteilung beinhaltete, dass Lernende mit einer ähnlichen Leistungsstärke zusammen eingeteilt werden sollten. In 92.02% der Fälle (173 von 188) hatten die Lernenden in einer Gruppe ein gleiches oder ähnliches schulisches Leistungsniveau. Somit setzten die Lehrpersonen diese Anforderung gut um.

Unterschied der Programmierkenntnisse

Die Lehrperson gab auf dem Beobachtungsauftrag an, ob die Lernenden einer Gruppe Programmierkenntnisse hatten oder nicht. Dadurch konnte ermittelt werden, ob beide Personen, nur eine oder keine Person in der Gruppe Vorkenntnisse aufwiesen. Die Häufigkeiten dieser Kategorien werden im Anhang 5.3 dargestellt. Während sich 39.90% der Gruppen aus Personen ohne Vorkenntnisse zusammensetzten, hatten in 36.90% der Gruppen beide Personen Programmierkenntnisse. In 19.20% hatte nur eine Person zuvor bereits programmiert. Die Vorgabe, dass Schülerinnen und Schüler mit Programmier-Vorkenntnissen zusammen eingeteilt werden sollten, wurde somit gut umgesetzt.

Beziehung der Lernenden

Die Lehrperson schätzte auf dem Beobachtungsauftrag die Beziehung der Lernenden auf einer Likert-Skala mit fünf Stufen (*negativ, eher negativ, neutral, eher positiv, positiv*) ein. Diese Zuteilung wurde direkt in die Auswertung übernommen, wobei *negativ* mit

einer Null, *eher negativ* mit einer Eins, *neutral* mit einer Zwei, *eher positiv* mit einer Drei und *positiv* mit einer Vier codiert wurden. Die Verteilung der Gruppen auf die fünf Stufen ist im Anhang 5.4 ersichtlich. In keiner der untersuchten Gruppen herrschte eine negative Beziehung vor. Die Vorgabe, dass möglichst Schülerinnen und Schüler miteinander eingeteilt werden sollten, die weder sehr gut befreundet, noch zerstritten waren, wurde gut umgesetzt. Die Lehrpersonen gaben bei 38.4% der Gruppen (78 von 198) an, dass die Lernenden eine neutrale Beziehung hatten. 24.6% (50 von 198) der Gruppen wiesen eine eher positive Beziehung auf. Der Mittelwert lag bei 2.90 mit einer Standardabweichung .90, wobei die Nummer drei der Bezeichnung *eher positiv* entsprach.

3.7.7. Prüfung des Einflusses möglicher Störvariablen

Die Forschungsliteratur weist darauf hin, dass es viele Faktoren gibt, welche sich auf die Zusammenarbeit beim Pair Programming auswirken können (siehe Kapitel 2.4. Einflussfaktoren auf das Pair Programming). Da zur Beantwortung der Forschungsfragen dieser Studie der Einfluss der Geschlechterzusammensetzung auf die Zusammenarbeit der Lernenden möglichst isoliert untersucht werden sollte, wurden die Lehrpersonen mit Kriterien geleitet, um bezüglich anderer Faktoren möglichst homogene Gruppen zusammenzustellen (siehe Kapitel 3.3.3. Paareinteilung und -auswahl). Da es in einer Klasse nicht möglich ist, alle Gruppen entsprechend dieser Kriterien perfekt einzuteilen, wurde davon ausgegangen, dass die Gruppen nicht bezüglich aller Faktoren homogen waren. Deshalb wurden die Klassenstufe, der Alters-, Leistungs- und Programmierkenntnis-Unterschied der Lernenden, sowie ihre Beziehung zusätzlich mithilfe des Beobachtungsauftrages erhoben (siehe Kapitel 3.6.1. Beobachtungsauftrag). Um zu überprüfen, ob diese Faktoren ohne Einbezug der Geschlechterzusammensetzung einen Effekt auf die abhängigen Variablen hatten, wurden die genannten möglichen Störvariablen als unabhängige Variablen betrachtet. Zuerst wurde mit einem Shapiro-Wilk-Test geprüft, ob die abhängigen Variablen mit den Faktoren Altersunterschied, Beziehung der Lernenden, Klassenstufe, Leistungsunterschied oder Unterschied der Programmierkenntnisse normalverteilt waren. Da nur vereinzelt Normalverteilungen vorhanden waren, wurden nicht parametrische Verfahren gewählt. Der Einfluss der möglichen Störvariablen wurde deshalb mithilfe eines Kruskal-Wallis-Tests überprüft. Die Variablen *Altersunterschied* und *Beziehung der Lernenden* zeigten keine statistisch signifikanten Einflüsse auf die abhängigen Variablen. Bei der Hypothesenprüfung wurden diese Variablen deshalb nicht miteinbezogen. Bei den Variablen *Klassenstufe*, *Leistungsunterschied* und *Unterschied Programmierkenntnisse* konnten statistisch signifikante Einflüsse auf die

abhängigen Variablen der Studie nachgewiesen werden. Diese Berechnungen und Resultate werden nachfolgend genauer vorgestellt.

Klassenstufe

Die Klassenstufe der Lernenden hatte keinen statistisch signifikanten Einfluss auf die Anzahl angeforderten Hilfestellungen pro Stunde, die Anzahl Eingriffe der einzelnen Lernenden, die durchschnittliche Dauer eines Eingriffs der Lernenden, sowie die Summe der durchschnittlichen Dauer in der Gruppe und die Anzahl Platztäusche. Statistisch signifikante Unterschiede zeigten sich bei der Lösegeschwindigkeit ($H= 9.72$, $p= .008$) und der Summe der Anzahl Eingriffe der Lernenden ($H= 8.342$, $p= .015$). Ein Dunn-Bonferroni-Test ergab, dass sich die Lösegeschwindigkeit der Gruppen *Begabtenförderung* und *Oberstufe* ($z= -2.59$, $p= .029$), sowie *5./6. Klasse* und *Oberstufe* ($z= -2.74$, $p= .018$) signifikant unterschied. Um die Bedeutsamkeit dieser Ergebnisse zu beurteilen, wurden die Effektstärken berechnet. Da sich die Gruppen hinsichtlich ihrer Grösse stark unterschieden, wurde die Effektstärke von Cohen mit gepoolter Standardabweichung (d_s) gewählt. Laut Cohen (1988) gilt ein Wert zwischen 0.2 und 0.5 als kleiner, zwischen 0.5 und 0.8 als mittlerer, und ab 0.8 als grosser Effekt. Beim Unterschied der Lösegeschwindigkeit zwischen der *Begabtenförderung* und der *Oberstufe*, handelte es sich laut Cohen (1988) um einen kleinen Effekt ($d_s= 0.28$). Ebenfalls klein war der Effekt bei der *5./6. Klasse* und *Oberstufe* ($d_s= 0.37$). Im folgenden Box-Plot (Abbildung 7) ist die unterschiedliche Lösegeschwindigkeit der Klassenstufen sichtbar. Lernende der *Oberstufe* bearbeiteten die Aufgaben schneller als diejenigen der *5./6. Klasse* oder der *Begabtenförderung*. Der Unterschied der Lösegeschwindigkeit zwischen den Lernenden der *5./6. Klasse* und denjenigen der *Begabtenförderung* war nicht statistisch signifikant. Um den Einfluss der Klassenstufe auf die Lösegeschwindigkeit zu berücksichtigen, wurde bei der Prüfung der ersten Hypothese zusätzlich nach Klassenstufe getrennt ausgewertet.

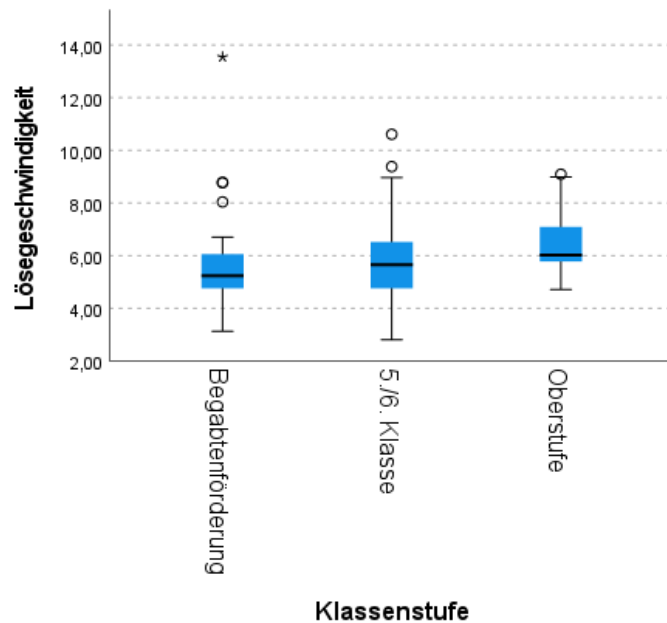


Abbildung 7 Box-Plot zur Lösegeschwindigkeit nach Klassenstufe (n=185)

Ein Dunn-Bonferroni-Test ergab, dass sich die Summe der Anzahl Eingriffe der Gruppen *Oberstufe* und *5./6. Klasse* signifikant unterschied ($z= 2.75, p= .018$). Laut Cohen (1988) handelte es sich um einen schwachen Effekt ($d_s=0.45$). Im folgenden Box-Plot (Abbildung 8) ist erkennbar, dass in Gruppen der Oberstufe weniger häufig eingegriffen wurde als in Gruppen der 5./6. Klasse. Die Unterschiede zu den Gruppen der Begabtenförderung waren nicht statistisch signifikant. Da die Klassenstufe einen Einfluss auf die Summe der Anzahl Eingriffe einer Gruppe hatte, wurde zur Beantwortung der sechsten Hypothese zusätzlich nach Klassenstufe getrennt ausgewertet.

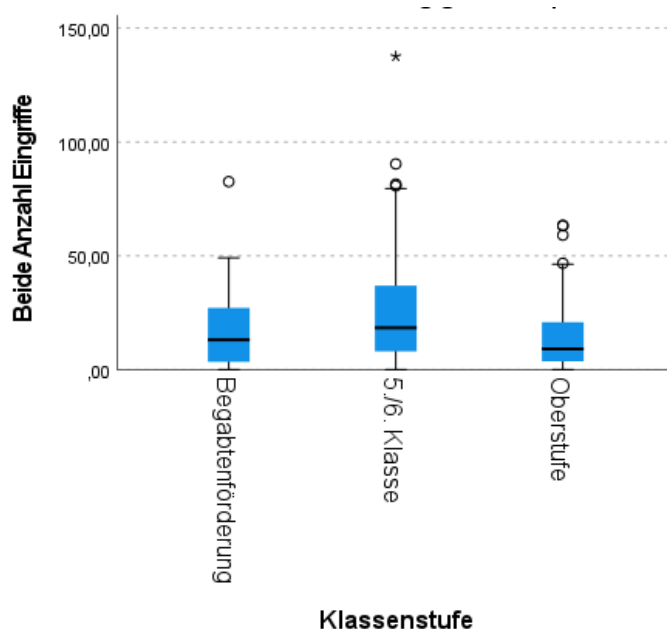


Abbildung 8 Box-Plot zur Summe der Anzahl Eingriffe der Lernenden nach Klassenstufe (n=160)

Leistungsunterschied

Der Leistungsunterschied der Lernenden hatte keinen statistisch signifikanten Einfluss auf die Lösegeschwindigkeit, die Anzahl Eingriffe, die Summe der Eingriffe, die durchschnittliche Dauer eines Eingriffs, sowie die Summe der durchschnittlichen Dauer, und die Anzahl Platztäusche. Einen statistisch signifikanten Einfluss hatte der Leistungsunterschied der Lernenden auf die Anzahl angeforderten Hilfestellungen ($H=8.69$, $p=.034$). Ein Dunn-Bonferroni-Test ergab, dass sich die Anzahl der angeforderten Hilfestellungen der Gruppen *ähnliches allgemeines schulisches Leistungsniveau* und *gleiches allgemeines schulisches Leistungsniveau* statistisch signifikant unterschied ($z=2.73$, $p=.038$). Die Effektstärke wurde entsprechend Cohen (1988) als tief bewertet ($d_s=0.37$). In der Abbildung 9 ist erkennbar, dass Gruppen, in welchen die Lernenden das gleiche allgemeine schulische Leistungsniveau hatten, mehr Hilfestellungen anforderten als Gruppen, welche ein ähnliches allgemeines schulisches Leistungsniveau hatten. Da sich dies auf die zweite Hypothese auswirken könnte, wurde dort zusätzlich nach Leistungsunterschied getrennt ausgewertet.

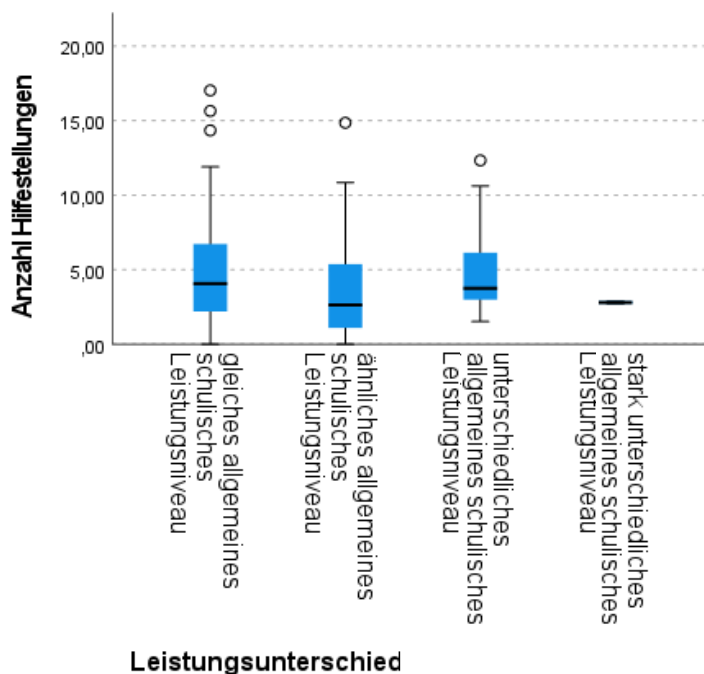


Abbildung 9 Box-Plot zur Anzahl angeforderte Hilfestellungen nach Leistungsunterschied der Lernenden ($n= 188$)

Unterschied Programmierkenntnisse

Der Unterschied der Programmierkenntnisse der Lernenden hatte keinen statistisch signifikanten Einfluss auf die Lösegeschwindigkeit, die Anzahl Eingriffe der Lernenden, die Summe der Eingriffe in der Gruppe, die durchschnittliche Dauer eines Eingriffs der Lernenden, sowie die Summe der durchschnittlichen Dauer in der Gruppe, und die

Anzahl Platztäusche. Der Unterschied in den Programmierkenntnissen der Lernenden einer Gruppe hatte einen statistisch signifikanten Einfluss auf die Anzahl angeforderter Hilfestellungen ($H= 14.21$, $p= .001$). Ein Dunn-Bonferroni-Test ergab, dass sich die Anzahl der angeforderten Hilfestellungen der Gruppen *beide Personen Vorkenntnisse* und *beide keine Vorkenntnisse* statistisch signifikant unterschied ($z= 3.67$, $p= .001$). Die Effektstärke wurde entsprechend Cohen (1988) als mittel eingestuft ($d_s= 0.59$). In der Abbildung 10 ist erkennbar, dass Gruppen, in welchen beide Lernenden keine Programmierkenntnisse hatten, mehr Hilfestellungen anforderten als Gruppen, in welchen beide Personen Vorkenntnisse hatten. Die Unterschiede zu den Gruppen, in welchen eine Person Vorkenntnisse hatte, waren nicht statistisch signifikant. Da sich dieser Unterschied der Programmierkenntnisse auf die zweite Hypothese auswirken könnte, wurde dort zusätzlich nach Programmierkenntnissen getrennt ausgewertet.

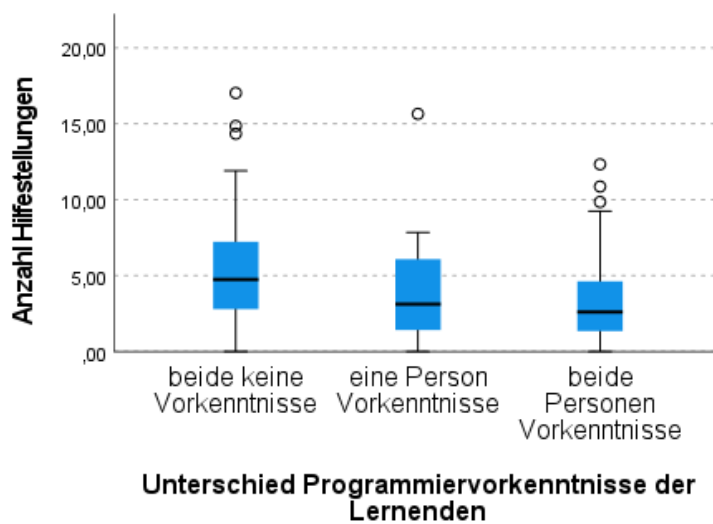


Abbildung 10 Box-Plot zur Anzahl Hilfestellungen nach Unterschied in den Programmierkenntnissen der Lernenden einer Gruppe ($n=195$)

4. Ergebnisse

In diesem Kapitel werden Berechnungen und Resultate zur Hypothesenprüfung vorgestellt. Zunächst wurde mit einem Shapiro-Wilk-Test geprüft, ob die abhängigen Variablen normalverteilt waren. Da im vorliegenden Datensatz nur vereinzelt Normalverteilungen vorhanden waren (siehe Anhang 6), wurden für die Hypothesenprüfung nicht parametrische Verfahren gewählt. In dieser Studie wurden sowohl die Zusammenarbeit der Geschlechterzusammensetzungen (Hypothesen 1, 2, 5, 6 und 7), als auch das Verhalten von Mädchen und Jungen in geschlechtsheterogenen Gruppen (Hypothesen 3 und 4) verglichen. Für die gruppenübergreifenden Hypothesen wurden Kruskal-Wallis-Tests durchgeführt, für die Gruppeninternen Wilcoxon-Tests. Der Datensatz, mit welchem die Hypothesenprüfung durchgeführt wurde, ist online downloadbar (siehe Link im Anhang 7).

4.1. Lösegeschwindigkeit nach Geschlechterzusammensetzung

H1: Geschlechtshomogene und geschlechtsheterogene Paare unterscheiden sich nicht in Bezug auf die Geschwindigkeit des Aufgabenlöses.

Ein Kruskal-Wallis-Test ergab, dass sich die Geschlechterzusammensetzungen bezüglich Lösegeschwindigkeit nicht statistisch signifikant unterscheiden ($H= 2.56$, $p= .279$). Dies ist auch im Box-Plot (Abbildung 11) erkennbar. Geschlechtshomogene männliche Paare lösten die Aufgaben tendenziell am schnellsten. Diese Geschlechterzusammensetzung wies die höchsten Minimal-, Maximal- und Durchschnittswerte, aber auch die grösste Standardabweichung auf (siehe Tabelle 5).

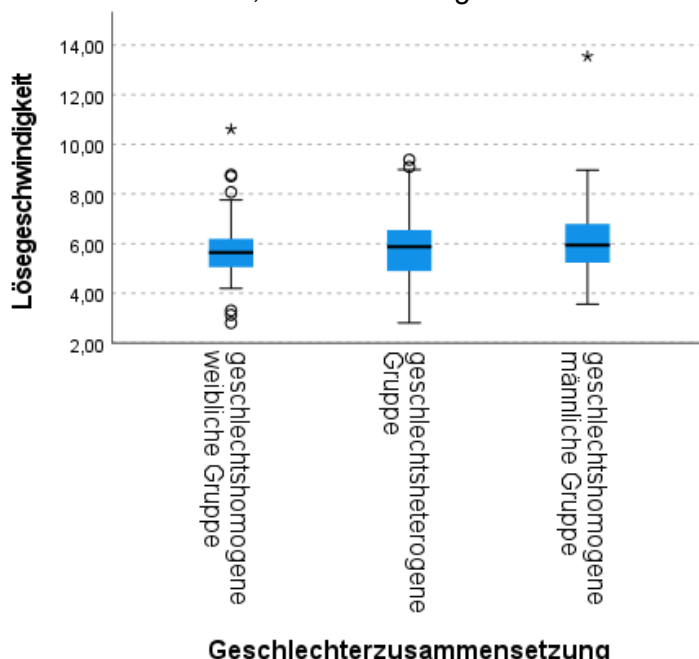


Abbildung 11 Box-Plot zur Lösegeschwindigkeit nach Geschlechterzusammensetzung ($n= 185$)

Tabelle 5 Deskriptive Statistik zur Lösegeschwindigkeit der Geschlechterzusammensetzungen (n= 185)

Geschlechterzusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Paare	64	2.80	10.61	5.79	1.34
geschlechtsheterogene Paare	63	2.81	9.38	5.86	1.39
geschlechtshomogene männliche Paare	58	3.56	13.54	6.17	1.56
Gesamt	185	2.80	13.54	5.93	1.43

Bei der Prüfung des Einflusses möglicher Störvariablen (siehe Kapitel 3.7.7.) zeigte sich, dass sich die Lösegeschwindigkeit der Klassenstufen statistisch signifikant voneinander unterschied. Deshalb wurde diese Hypothese zusätzlich getrennt nach Klassenstufe ausgewertet. Auf der Oberstufe ($H= 3.45$, $p= .179$), bei 5./6.-Klässlerinnen und -Klässlern ($H= 2.65$, $p= .266$) und in der Begabtenförderung ($H= .40$, $p= .818$) zeigten sich keine statistisch signifikanten Unterschiede zwischen den Geschlechterzusammensetzungen. Bei den Lernenden der Oberstufe fiel auf, dass die geschlechtsheterogenen Gruppen die grösste Variabilität in der Lösegeschwindigkeit, sowie den höchsten Mittelwert aufwiesen (siehe Abbildung 12). Dies zeigte sich auch in der deskriptiven Statistik. Geschlechtsheterogene Gruppen wiesen die grösste Standardabweichung auf (siehe Tabelle 6).

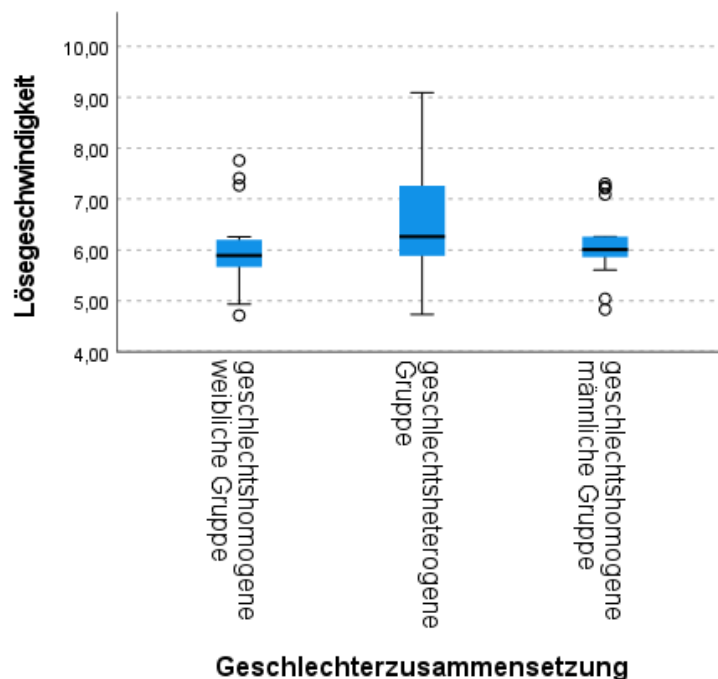


Abbildung 12 Box-Plot zur Lösegeschwindigkeit nach Geschlechterzusammensetzung auf der Oberstufe (n= 54)

Tabelle 6 Deskriptive Statistik zur Lösegeschwindigkeit der Geschlechterzusammensetzungen auf der Oberstufe (n=54)

Geschlechterzusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Paare	18	4.71	7.76	6.01	.79
geschlechtsheterogene Paare	19	4.73	9.09	6.60	1.16
geschlechtshomogene männliche Paare	17	4.83	7.30	6.14	.72
Gesamt	54	4.71	9.09	6.2571	.94

4.2. Anzahl angeforderte Hilfestellungen nach Geschlechterzusammensetzung

H2: Geschlechtshomogene Mädchenpaare fordern mehr Hilfestellungen an als geschlechtshomogene Jungenpaare und geschlechtsheterogene Paare.

Die Geschlechterzusammensetzungen unterschieden sich nicht statistisch signifikant in ihrer Anzahl angeforderter Hilfestellungen ($H= .18, p= .916$). Dieses Resultat ist auch auf der Abbildung 13 und in den deskriptiven Statistiken (siehe Tabelle 7) ersichtlich. Die geschlechtsheterogenen Paare wiesen aber den höchsten Maximal- und Mittelwert, sowie die grösste Standardabweichung der Geschlechterzusammensetzungen auf.

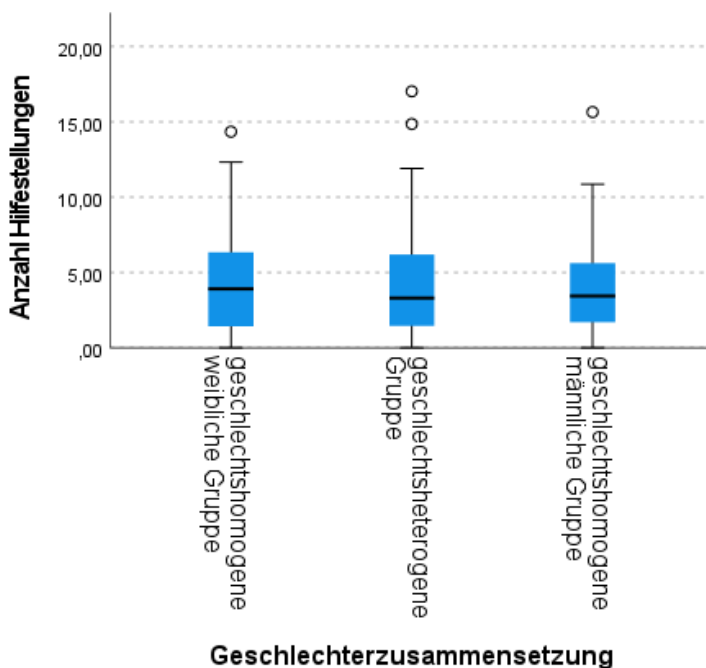


Abbildung 13 Box-Plot zur Anzahl Hilfestellungen nach Geschlechterzusammensetzung (n= 203)

Tabelle 7 Deskriptive Statistik zur Anzahl Hilfestellungen der Geschlechterzusammensetzungen (n=203)

Geschlechter-zusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Paare	68	.00	14.34	4.25	3.33
geschlechtsheterogene Paare	71	.00	17.02	4.27	3.48
geschlechtshomogene männliche Paare	64	.00	15.65	4.00	3.22
Gesamt	203	.00	17.02	4.18	3.33

Bei der Prüfung des Einflusses möglicher Störvariablen (siehe Kapitel 3.7.7.) zeigte sich, dass sich der Leistungs- und Programmierkenntnis-Unterschied der Lernenden statistisch signifikant auf die Anzahl angeforderter Hilfestellungen auswirkte. Deshalb wurden die Daten zusätzlich entlang dieser Störvariablen getrennt ausgewertet, um die zweite Hypothese zu überprüfen. Da sich auch nach der separaten Auswertung keine statistisch signifikanten Unterschiede oder Tendenzen zeigten, wird auf diesen Auswertungsschritt nicht näher eingegangen.

4.3. Anzahl Eingriffe in geschlechtsheterogenen Gruppen

H3: In geschlechtsheterogenen Paaren berühren die Jungen in der Navigatoren-Rolle entgegen dem Leitfaden des Pair Programmings häufiger die Computertastatur oder die Maus als die Mädchen in der Navigatoren-Rolle.

Mit einem Wilcoxon-Test wurde geprüft, ob die zentralen Tendenzen der Anzahl Eingriffe der Jungen und der Mädchen verschieden waren. Die Anzahl Eingriffe der Jungen ($Mdn= 6.0577$) unterschied sich nicht signifikant von der Anzahl der Eingriffe der Mädchen ($Mdn= 5.5872$; asymptotischer Wilcoxon-Test: $z= -.604$, $p= .546$, $n= 60$). Ein Box-Plot (siehe Abbildung 14) und deskriptive Statistiken (siehe Tabelle 8) liessen aber vermuten, dass die Jungen tendenziell häufiger eingriffen. Der Mittelwert der Mädchen lag bei 9.68 Eingriffen ($SD=$ von 12.53), derjenige der Jungen bei 11.96 ($SD= 13.96$). Die höchste Anzahl Eingriffe von 53.90 Eingriffen pro Navigatoren-Stunde konnte einem Jungen zugeschrieben werden.

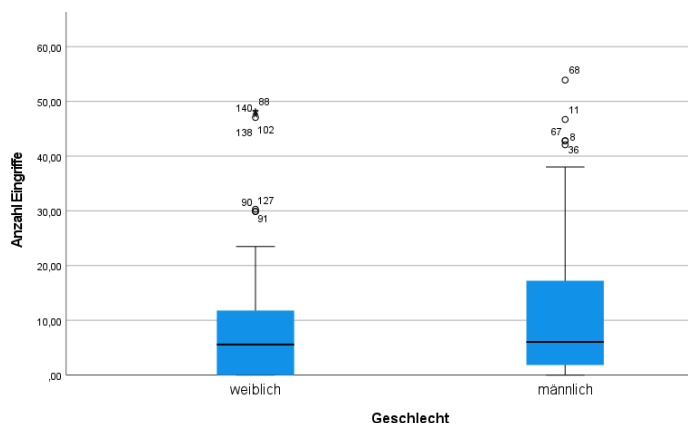


Abbildung 14 Box-Plot zur Anzahl Eingriffe nach Geschlecht in geschlechtsheterogenen Gruppen (n=129)

Tabelle 8 Deskriptive Statistik zur Anzahl Eingriffe der Jungen und Mädchen in geschlechtsheterogenen Gruppen (n=129)

Geschlecht der Lernenden	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
männlich	64	.00	53.90	11.96	13.96
weiblich	65	.00	48.26	9.68	12.53
Gesamt	129	.00	53.90	10.81	13.26

4.4. Dauer der Eingriffe in geschlechtsheterogenen Gruppen

H4: In geschlechtsheterogenen Paaren dauern die Eingriffe der Jungen in der Navigatoren-Rolle durchschnittlich länger als die der Mädchen in der Navigatoren-Rolle.

Ein Wilcoxon-Test ergab, dass sich die zentralen Tendenzen der durchschnittlichen Dauer eines Eingriffs von Jungen ($Mdn= 0:00:08$) und Mädchen ($Mdn= 0:00:07$) nicht signifikant voneinander unterschieden (asymptotischer Wilcoxon-Test: $z= -.72$, $p= .469$, $n= 35$). Dies ist auch auf dem Box-Plot (Abbildung 15) erkennbar. Deskriptive Statistiken (Tabelle 9) zeigten, dass die Mädchen einen höheren Mittelwert aufwiesen ($M= 0:00:13$). Die längste durchschnittliche Dauer von 2.5 Minuten war einem Mädchen zuzuschreiben und die Standardabweichung der Mädchen war höher ($SD= 0:00:22$).

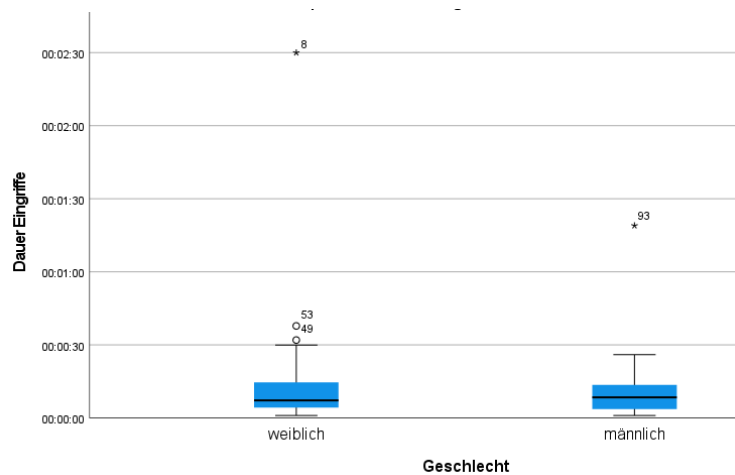


Abbildung 15 Box-Plot zur durchschnittlichen Dauer eines Eingriffs nach Geschlecht in geschlechtsheterogenen Gruppen (n= 97)

Tabelle 9 Deskriptive Statistik zur durchschnittlichen Dauer eines Eingriffs nach Geschlecht in geschlechtsheterogenen Gruppen (n=97)

Geschlecht der Lernenden	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
männlich	50	0:00:01	0:01:19	0:00:10	0:00:11
weiblich	47	0:00:01	0:02:30	0:00:13	0:00:22
Gesamt	97	0:00:01	0:02:30	0:00:11	0:00:17

4.5. Anzahl Rollenwechsel nach Geschlechterzusammensetzung

H5: In geschlechtshomogenen Paaren werden beim Pair Programming die Rollen häufiger gewechselt als in den geschlechtsheterogenen Paaren.

Ein Kruskal-Wallis-Test ergab, dass keine statistisch signifikanten Unterschiede der Anzahl Platztäusche zwischen den verschiedenen Geschlechterzusammensetzungen vorhanden waren ($H= .93$, $p= .628$). Ein Box-Plot (siehe Abbildung 16) und nach Geschlechtergruppen getrennte deskriptive Statistiken (siehe Tabelle 10) liessen hingegen Unterschiede vermuten. Der Mittelwert der geschlechtsheterogenen Gruppen war am wenigsten von der gewünschten Anzahl Rollenwechsel (was bei dieser Variablen einem Wert von Null entspricht) entfernt und hatte die kleinste Standardabweichung ($SD= 1.60$). Dies ist auch auf dem Box-Plot erkennbar. Geschlechtshomogene weibliche und männliche Gruppen hatten einen sehr ähnlichen Mittelwert, die Jungengruppen wiesen aber eine grössere Standardabweichung auf.

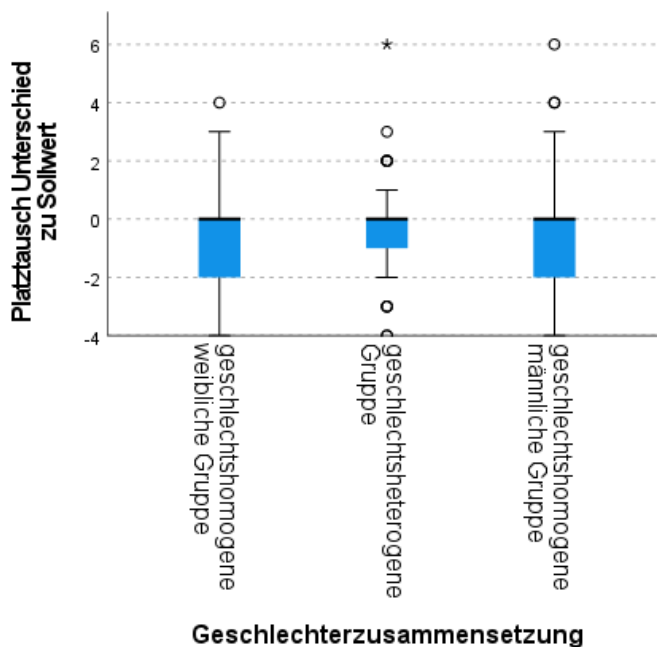


Abbildung 16 Boxplot zur Anzahl Platztäusche (Differenz zum Sollwert) nach Geschlechterzusammensetzung ($n= 176$)

Tabelle 10 Deskriptive Statistik zur Anzahl Platztäusche (Differenz zum Sollwert) nach Geschlechterzusammensetzung ($n= 176$)

Geschlechterzusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Gruppe	61	-4	4	-.52	1.78
geschlechtsheterogene Gruppe	61	-4	6	-.26	1.60
geschlechtshomogene männliche Gruppe	54	-4	6	-.54	2.17
Gesamt	176	-4	6	-.44	1.84

Die Betrachtung der Häufigkeiten der Werte (siehe Tabelle 11) liess ebenfalls Unterschiede vermuten. Sowohl bei den geschlechtshomogenen weiblichen (44.3%) als auch bei den geschlechtsheterogenen Gruppen (49.2%) führte die Mehrheit der Paare die korrekte Anzahl Platztäusche durch. Bei den geschlechtshomogenen männlichen Gruppen machte dagegen die Mehrheit der Paare zu wenig Platztäusche (42.6%). Den höchsten Prozentwert der richtigen Anzahl Platztäusche (49.2%), sowie den tiefsten Wert bei den fehlenden Platztäuschen (32.8%), hatten die geschlechtsheterogenen Paare. Im Vergleich zu den anderen Geschlechterzusammensetzungen machten geschlechtshomogene männliche Gruppen am häufigsten mehr Platztäusche als nötig (20.4%). Tendenziell hielten sich somit geschlechtsheterogene Paare am besten an die Anweisungen des Rollenwechsels, gefolgt von den geschlechtshomogenen weiblichen Gruppen. Die geschlechtshomogenen männlichen Paare hielten sich tendenziell am wenigsten gut an die Platztauschweisung.

Tabelle 11 Verteilung der Geschlechterzusammensetzungen auf zu wenig, genau richtig oder zu viele Platztäusche (n= 176)

Geschlechterzusammensetzung	Weniger Platztäusche als nötig	Richtige Anzahl Platztäusche	Mehr Platztäusche als nötig
geschlechtshomogene weibliche Gruppe	37.7%	44.3%	18%
geschlechtsheterogene Gruppe	32.8%	49.2%	18%
geschlechtshomogene männliche Gruppe	42.6%	37%	20.4%
Gesamt	37.5%	43.8%	18.7%

4.6. Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung

H6: In geschlechtsheterogenen Paaren berühren die Lernenden in der Navigatoren-Rolle entgegen dem Leitfaden des Pair Programmings häufiger die Computertastatur oder die Maus als die Lernenden in den geschlechtshomogenen Paaren.

Ein Kruskal-Wallis-Test ergab, dass keine statistisch signifikanten Unterschiede der Anzahl Eingriffe einer Gruppe zwischen den verschiedenen Geschlechterzusammensetzungen vorhanden waren ($H= 3.07, p= .215$). Ein Box-Plot (Abbildung 17) und deskriptive Statistiken (Tabelle 12) wiesen dagegen darauf hin, dass in geschlechtshomogenen männlichen Paaren tendenziell am häufigsten eingegriffen

wurde. Geschlechtshomogene männliche Gruppen wiesen den höchsten Mittelwert auf ($M= 25.55$).

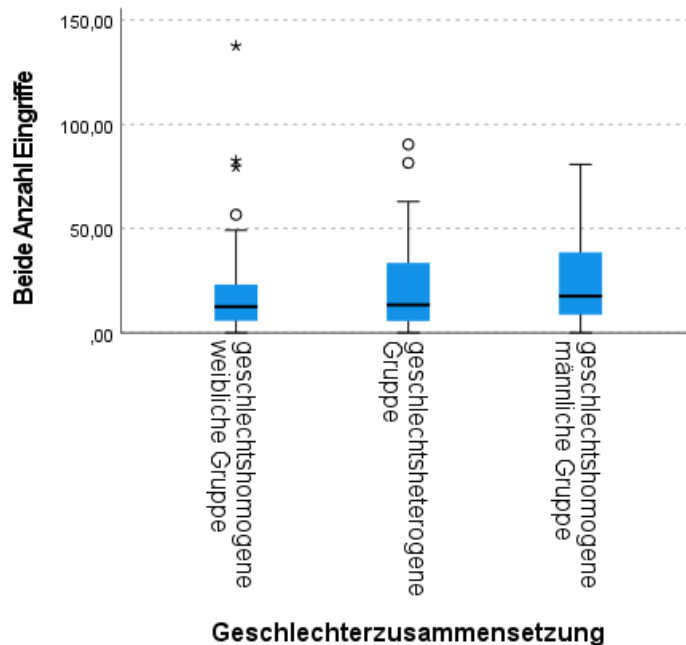


Abbildung 17 Box-Plot zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung ($n= 160$)

Tabelle 12 Deskriptive Statistik zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung ($n= 160$)

Geschlechter-zusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Paare	54	.00	137.58	19.23	24.10
geschlechtsheterogene Paare	60	.00	90.34	21.15	20.83
geschlechtshomogene männliche Paare	46	.00	80.72	25.55	22.34
Gesamt	160	.00	137.58	21.77	22.41

Bei der Prüfung des Einflusses möglicher Störvariablen (siehe Kapitel 3.7.7.) zeigte sich, dass die Klassenstufe einen statistisch signifikanten Einfluss auf die Summe der Anzahl Eingriffe der Lernenden einer Gruppe hatte. Lernende der Primarstufe griffen häufiger ein als diejenigen der Oberstufe. Deshalb wurden die Primar- und Sekundargruppen für die sechste Hypothese zusätzlich getrennt voneinander untersucht. Bei der Betrachtung der Lernenden der Primarstufe zeigte sich bezüglich der Anzahl Eingriffe kein statistisch signifikanter Unterschied der Geschlechterzusammensetzungen. Bei den Gruppen der Sekundarstufe ergab dagegen ein Kruskal-Wallis-Test, dass ein statistisch signifikanter Unterschied der Anzahl Eingriffe zwischen den Geschlechterzusammensetzungen vorhanden war ($H= 8.61$, $p= .014$). Ein Dunn-Bonferroni-Test ergab, dass sich die Summe der Anzahl Eingriffe zwischen den Gruppen *geschlechtshomogene weibliche Gruppe* und *geschlechtshomogene männliche Gruppe* ($z= -2.52$, $p= .035$), sowie

zwischen den Gruppen *geschlechtsheterogene Gruppe* und *geschlechtshomogene männliche Gruppe* signifikant unterschied ($z = -2.62, p = .027$). Sowohl die Effektstärke des Unterschiedes zwischen den geschlechtshomogenen weiblichen und geschlechtshomogenen männlichen Gruppen ($d_s = 1.069$), als auch diejenige zwischen geschlechtsheterogenen und geschlechtshomogenen männlichen Gruppen ($d_s = 1.062$) wurden als hoch eingestuft. Auf dem Box-Plot (Abbildung 18) ist erkennbar, dass auf der Oberstufe in geschlechtshomogenen männlichen Gruppen die Lernenden insgesamt häufiger eingriffen als in den anderen Geschlechterzusammensetzungen. Dieses Resultat ist auch in den deskriptiven Statistiken ersichtlich (siehe Tabelle 13). Geschlechtshomogene männliche Paare wiesen den höchsten Maximal- und Mittelwert, sowie die grösste Standardabweichung auf.

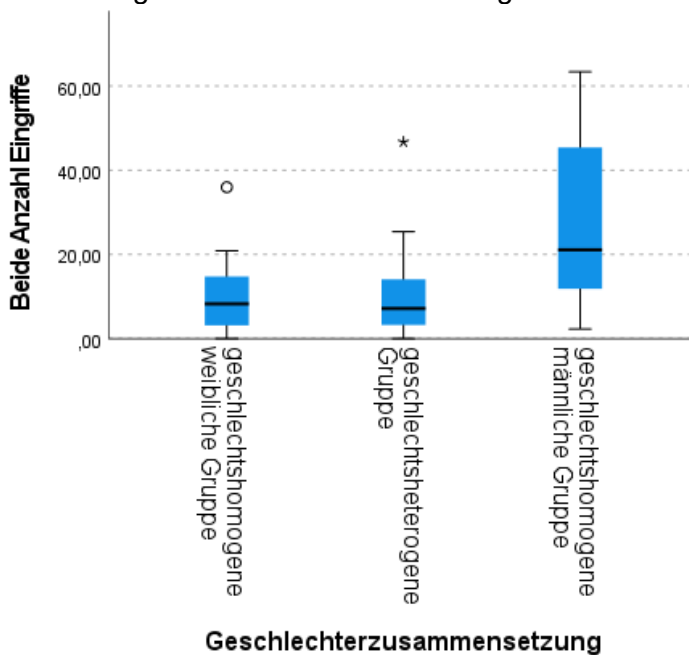


Abbildung 18 Box-Plot zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung auf der Oberstufe ($n=50$)

Tabelle 13 Deskriptive Statistik zur Summe der Anzahl Eingriffe nach Geschlechterzusammensetzung auf der Oberstufe ($n= 50$)

Geschlechter-zusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Paare	15	.00	35.96	10.11	9.79
geschlechtsheterogene Paare	20	.00	46.71	10.60	11.16
geschlechtshomogene männliche Paare	15	2.27	63.37	28.17	21.81
Gesamt	50	.00	63.37	15.73	16.71

Dieses Ergebnis warf die Frage auf, ob männliche Lernende auf der Sekundarstufe in geschlechtsheterogenen oder in geschlechtshomogenen Gruppen häufiger eingriffen. Ein Mann-Whitney-U-Test ergab, dass die Jungen auf der Sekundarstufe in geschlechtshomogenen Gruppen ($Mdn= 8.00$) tendenziell häufiger eingriffen als in Geschlechtsheterogenen ($Mdn= 3.10$). Dieser Unterschied war aber nicht signifikant ($z= -1.88$, $p= .061$). Ein Box-Plot (Abbildung 19) und deskriptive Statistiken (Tabelle 14) wiesen ebenfalls auf die Tendenz hin, dass Jungen der Sekundarstufe in geschlechtshomogenen Gruppen häufiger eingriffen als in Geschlechtsheterogenen.

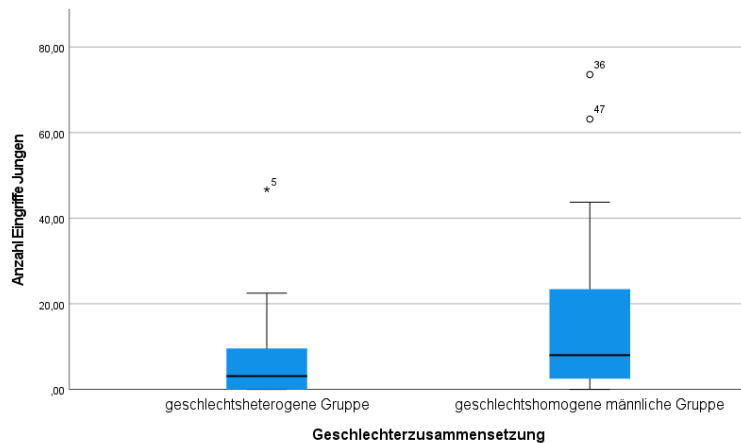


Abbildung 19 Box-Plot zur Anzahl Eingriffe der Jungen in geschlechtsheterogenen und -homogenen Gruppen auf der Oberstufe ($n=53$)

Tabelle 14 Deskriptive Statistik zur Anzahl Eingriffe der Jungen in geschlechtsheterogenen und -homogenen Gruppen auf der Oberstufe ($n= 53$)

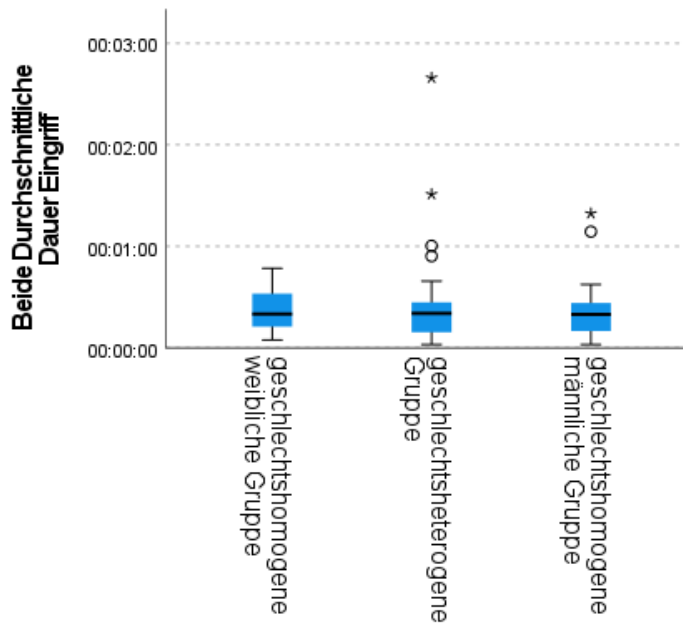
Geschlechter-zusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtsheterogen	21	.00	46.71	7.88	11.49
geschlechtshomogen	17	.00	73.56	14.56	19.55
Gesamt	53	.00	73.56	12.86	16.87

4.7. Summe der Dauer der Eingriffe nach Geschlechterzusammensetzung

H7: In geschlechtsheterogenen Paaren dauern die Eingriffe der Lernenden in der Navigatoren-Rolle durchschnittlich länger als die der Lernenden in geschlechtshomogenen Paaren.

Ein Kruskal-Wallis-Test ergab, dass sich die Geschlechterzusammensetzungen bezüglich Summe der durchschnittlichen Dauer eines Eingriffs nicht statistisch signifikant unterschieden ($H= .54$, $p= .765$). Ein Box-Plot (Abbildung 20) und deskriptive Statistiken (Tabelle 15) wiesen jedoch darauf hin, dass in den geschlechtsheterogenen Gruppen die grösste Streuung der durchschnittlichen Dauer eines Eingriffs vorhanden war. Geschlechtsheterogene Gruppen wiesen die höchste durchschnittliche Dauer der Eingriffe, den höchsten Mittelwert und die grösste Standardabweichung der

Geschlechterzusammensetzungen auf. Allerdings handelte es sich bei zwei Werten um Ausreisser.



Geschlechterzusammensetzung

Abbildung 20 Box-Plot zur Summe der durchschnittlichen Dauer eines Eingriffs nach Geschlechterzusammensetzung (n= 95)

Tabelle 15 Deskriptive Statistik zur Summe der durchschnittlichen Dauer eines Eingriffs nach Geschlechterzusammensetzung (n= 95)

Geschlechter-zusammensetzung	N	Minimum	Maximum	Mittelwert	Std.-Abweichung
geschlechtshomogene weibliche Gruppe	30	0:00:04	0:00:46	0:00:21	0:00:11
geschlechtsheterogene Gruppe	35	0:00:02	0:02:39	0:00:25	0:00:29
geschlechtshomogene männliche Gruppe	30	0:00:02	0:01:19	0:00:21	0:00:16
Gesamt	95	0:00:02	0:02:39	0:00:23	0:00:21

5. Diskussion und Schlussfolgerungen

Im Rahmen dieser Studie wurde untersucht, inwiefern sich die Zuteilung in geschlechtshomogene und -heterogene Paare auf die Zusammenarbeit der Lernenden auswirkt. Im nachfolgenden Kapitel werden zuerst die vier aus der Forschungsliteratur abgeleiteten Forschungsfragen und die sieben dazugehörigen Hypothesen diskutiert. Darauf folgt die Diskussion der Methode. Schlussfolgerungen und ein Forschungsausblick schliessen dieses Kapitel ab.

5.1. Diskussion der Forschungsfragen und Hypothesen

In diesem Kapitel werden die vier Forschungsfragen mit den sieben dazugehörigen Hypothesen in Unterkapiteln besprochen.

5.1.1. Lösegeschwindigkeit nach Geschlechterzusammensetzung

Mit der ersten Forschungsfrage soll geprüft werden, ob sich geschlechtshomogene und -heterogene Paare bezüglich ihrer Lösegeschwindigkeit unterscheiden (F1). Vermutet wird, dass sich die Geschlechterzusammensetzung nicht auf die Geschwindigkeit des Aufgabenlösens auswirkt (H1).

F1: Inwiefern unterscheiden sich geschlechtshomogene und geschlechtsheterogene Paare in Bezug auf die Geschwindigkeit des Aufgabenlösens?

H1: Geschlechtshomogene und geschlechtsheterogene Paare unterscheiden sich nicht in Bezug auf die Geschwindigkeit des Aufgabenlösens.

Die Resultate dieser Videostudie sprechen dafür, dass sich geschlechtshomogene und -heterogene Paare bezüglich Lösegeschwindigkeit nicht statistisch signifikant unterscheiden. Bei der Prüfung der möglichen Störvariablen hat sich gezeigt, dass die Klassenstufe einen signifikanten Einfluss auf die Lösegeschwindigkeit hat. Sekundarschülerinnen und -schüler lösen die Aufgaben schneller als Lernende im Primarschulalter. Deshalb wurde die Hypothese 1 zusätzlich getrennt nach Klassenstufe geprüft. Auch hier zeigt sich kein signifikanter Unterschied der Lösegeschwindigkeit zwischen geschlechtshomogenen und -heterogenen Paaren. Somit hat sich die erste Hypothese im Rahmen dieser Studie bewährt, geschlechtshomogene und -heterogene Paare unterscheiden sich nicht in Bezug auf die Geschwindigkeit des Aufgabenlösens.

Dieses Resultat entspricht den Ergebnissen bisheriger Studien, welche ebenfalls keinen signifikanten Unterschied der geschlechtshomogenen und -heterogenen Gruppen bezüglich Codeproduktivität feststellen können (Demir & Seferoglu, 2021a; Choi, 2015; Gómez et al., 2017; Akinola, 2016; Zhong et al., 2016). Das Ergebnis von Jarratt et al. (2019), dass geschlechtshomogene Männerpaare die höchste Produktivität zeigen, kann mit den Resultaten dieser Studie nicht bestätigt werden. Mit deskriptiven Statistiken kann

lediglich festgestellt werden, dass geschlechtshomogene männliche Paare die Aufgaben tendenziell etwas schneller lösen als die anderen Geschlechterzusammensetzungen. Dieser Unterschied ist aber nicht statistisch signifikant. Die von Gómez et al. (2017) angesprochene grösste Variabilität der Produktivität bei geschlechtsheterogenen Paaren zeigt sich auch bei Sekundarschülerinnen und -schülern dieser Studie. Dies könnte auf eine höhere Wahrscheinlichkeit von Kompatibilitätsproblemen bei geschlechtsheterogenen Paaren auf der Oberstufe hinweisen. Mehrere Forschende können zeigen, dass geschlechtsheterogene Paare weniger kompatibel sind als Geschlechtshomogene (Katira et al., 2005; Choi, 2015). Solche Unterschiede können bisher im schulischen Kontext nicht nachgewiesen werden (Zhong et al., 2016). Allerdings wiesen in einer Studie von Underwood et al. (2000) geschlechtsheterogene Paare auch im schulischen Kontext eine geringere verbale Interaktion auf als Geschlechtshomogene. Da eine gute Kommunikation entscheidend für ein effektives Pair Programming ist, könnte sich dies negativ auf die Lösegeschwindigkeit dieser Gruppen auswirken (Werner et al., 2004; Hanks et al., 2011; Denner et al., 2014; Rodríguez et al., 2017). Allerdings wiesen in dieser Studie die Lernenden in geschlechtsheterogenen Paaren auf der Sekundarstufe den höchsten Mittelwert der Lösegeschwindigkeit auf. Mit dem Hintergrundwissen, dass Jugendliche meist gleichgeschlechtliche Freundinnen und Freunde haben, wird vermutet, dass viele Lernende in geschlechtsheterogenen Gruppen nicht befreundet sind. Forschungsergebnisse zeigen, dass nicht befreundete Paare mehr leisten (Demir & Seferoglu, 2021a). Allerdings kommt es auch häufiger zu Situationen, in welchen gar nicht interagiert wird (Campe et al., 2020). Dementsprechend wird vermutet, dass gewisse geschlechtsheterogene Paare sehr gut, und andere gar nicht gut funktionieren. Dies könnte die grosse Variabilität der Lösegeschwindigkeit und den hohen Mittelwert erklären.

Die erste Forschungsfrage kann somit folgendermassen beantwortet werden: Geschlechtshomogene und -heterogene Paare unterscheiden sich bezüglich Geschwindigkeit des Aufgabenlösen nicht statistisch signifikant. Über alle Schulstufen betrachtet, lösen geschlechtshomogene Jungenpaare die Aufgaben tendenziell etwas schneller. Werden nur Sekundarschülerinnen und -schüler betrachtet, erreichen dagegen geschlechtsheterogene Paare den höchsten Mittelwert der Lösegeschwindigkeit. Gleichzeitig weisen diese Paare die grösste Variabilität der Geschwindigkeit des Aufgabenlösen auf. Dies könnte bedeuten, dass gewisse geschlechtsheterogene Paare auf der Oberstufe sehr gut, und andere gar nicht gut

funktionieren. Erklärungen hierfür könnten die Kommunikation im Paar oder Freundschaft der Lernenden sein.

5.1.2. Anzahl angeforderte Hilfestellungen nach Geschlechterzusammensetzung

Im Rahmen der zweiten Forschungsfrage (F2) soll geprüft werden, ob sich geschlechtshomogene und -heterogene Paare bezüglich der Anzahl angeforderter Hilfestellungen unterscheiden. Es wird angenommen, dass geschlechtshomogene Mädchenpaare mehr Hilfestellungen anfordern als die anderen Geschlechterzusammensetzungen (H2).

F2: Inwiefern unterscheiden sich geschlechtshomogene und geschlechtsheterogene Paare in Bezug auf die angeforderten Hilfestellungen?

H2: Geschlechtshomogene Mädchenpaare fordern mehr Hilfestellungen an als geschlechtshomogene Jungenpaare und geschlechtsheterogene Paare.

Die Resultate dieser Videostudie weisen darauf hin, dass sich die Geschlechterzusammensetzungen nicht statistisch signifikant in ihrer Anzahl angeforderter Hilfestellungen unterscheiden. In den deskriptiven Statistiken zeigte sich, dass der Maximalwert der angeforderten Hilfestellungen von einem geschlechtsheterogenen Paar erreicht wurde, und diese Geschlechterzusammensetzung auch den höchsten Mittelwert aufwies. Diese Unterschiede waren aber sehr klein. Bei der Prüfung der möglichen Störvariablen hat sich gezeigt, dass sich der Leistungsunterschied und der Unterschied in den Programmiervorkenntnissen der Lernenden auf die Anzahl angeforderten Hilfestellungen auswirken. Paare, in welchen die Lernenden das gleiche allgemeine schulische Leistungsniveau haben, fordern mehr Hilfestellungen an als Gruppen, welche ein ähnliches allgemeines schulisches Leistungsniveau haben. Paare, in welchen beide Personen keine Programmiervorkenntnisse haben, fordern mehr Hilfestellungen an als Gruppen, in welchen beide Vorkenntnisse haben. Deshalb wurde die zweite Hypothese zusätzlich entlang dieser Störvariablen getrennt ausgewertet. Auch hier zeigen sich keine statistisch signifikanten Unterschiede der Geschlechterzusammensetzungen bezüglich der Anzahl angeforderter Hilfestellungen. Deshalb wird die zweite Hypothese verworfen: Geschlechtshomogene Mädchengruppen fordern nicht mehr Hilfestellungen an als geschlechtshomogene Jungenpaare und geschlechtsheterogene Gruppen.

Forschende vermuten, dass sich das männlich geprägte stereotype Bild der Informatik negativ auf die Interessen und das Selbstvertrauen der Frauen auswirkt (Diekman et al., 2010; Diekman et al., 2011; Cheryan et al., 2013). Frauen haben eine negativere Einstellung zur Informatik als Männer (Başer, 2013; Chang et al., 2012; Jarratt et al.,

2019). Trotz gleicher Leistung in diesem Bereich schätzen sie sich als weniger kompetent ein (Beyer et al., 2003; Maguire et al., 2014; Campe et al., 2020; Fraunhofer IAIS, 2019; Jarratt et al., 2019; Coto & Mora, 2019). Frauen geben häufiger als Männer an, Programmierkonzepte nicht vollumfänglich zu verstehen. Ausserdem sind sie weniger selbstsicher in ihrem Produkt (Jarratt et al., 2019). Die Vermutung, dass sich diese Unsicherheit von Mädchen auf die Anzahl angeforderter Hilfestellungen auswirkt, kann jedoch nicht bestätigt werden. Eine mögliche Erklärung dafür ist der Einsatz von Pair Programming. Weibliche Lernende werden durch Pair Programming produktiver und haben mehr Selbstvertrauen, als wenn sie allein programmieren (Zhong et al., 2016). Mit Pair Programming kann weiblichen Studierenden ausserdem gezeigt werden, dass Programmieren auch eine kollaborative Aufgabe sein kann. Dadurch kann das stereotype Bild der Informatik als nicht soziales (Werner et al., 2004; Liebenberg et al., 2012; Ying et al., 2019) und kompetitives (Werner et al., 2004; Choi, 2015) Arbeitsumfeld etwas aufgebrochen werden. Somit könnte Pair Programming dabei helfen, mehr Frauen für die Informatik zu begeistern (Ying et al., 2019; Werner & Denning, 2009). Frauen erwähnen in Interviews zum Pair Programming, dass sie bei Unklarheiten lieber bei erfahrenen Gleichaltrigen als bei der Kursleitung nachfragen und es schätzen, allfällige Unsicherheiten mit jemandem teilen zu können (Ying et al., 2019; Werner & Denning, 2009). Hierbei könnte es sich um Gründe handeln, weshalb Mädchenpaare nicht häufiger bei der Kursleitung nachfragen als die anderen Geschlechterzusammensetzungen.

Die zweite Forschungsfrage kann dementsprechend folgendermassen beantwortet werden: Geschlechtshomogene und -heterogene Paare unterscheiden sich bezüglich der Anzahl angeforderter Hilfestellungen nicht statistisch signifikant. Auch mit Einbezug der Störvariablen Leistungsunterschied und unterschiedliche Programmier-vorkenntnisse, zeigt sich kein Einfluss der Geschlechterzusammensetzung auf die Anzahl der angeforderten Hilfestellungen. Die Vermutung, dass sich die durch das männlich geprägte Bild der Informatik ausgelösten Unsicherheiten der Frauen auf ihre benötigte Anzahl Hilfestellungen auswirkt, kann nicht bestätigt werden. Mögliche Erklärungen hierfür sind, dass der Einsatz von Pair Programming solche Unsicherheiten reduzieren kann, und dass Frauen bei Unklarheiten lieber bei Gleichaltrigen als bei der Kursleitung nachfragen.

5.1.3. Befolgen des Pair Programming Leitfadens in geschlechts-heterogenen Gruppen

Die dritte Forschungsfrage beschäftigt sich nur mit den geschlechtsheterogenen Paaren. Dabei soll geprüft werden, inwiefern sich Mädchen und Jungen beim Befolgen eines

zuvor kommunizierten Pair Programming Leitfadens unterscheiden. Es wird vermutet, dass Jungen in der Navigatoren-Rolle häufiger die Computertastatur oder die Maus berühren als Mädchen (H3), und deren Eingriffe länger dauern (H4).

F3: Inwiefern unterscheiden sich Mädchen und Jungen in geschlechtsheterogenen Paaren beim Befolgen eines zuvor kommunizierten Pair Programming Leitfadens?

H3: In geschlechtsheterogenen Paaren berühren die Jungen in der Navigatoren-Rolle entgegen dem Leitfaden des Pair Programmings häufiger die Computertastatur oder die Maus als die Mädchen in der Navigatoren-Rolle.

H4: In geschlechtsheterogenen Paaren dauern die Eingriffe der Jungen in der Navigatoren-Rolle durchschnittlich länger als die der Mädchen in der Navigatoren-Rolle.

Jungen greifen nicht statistisch signifikant häufiger ein als Mädchen. Dennoch lassen Box-Plot und deskriptive Statistiken vermuten, dass die Jungen tendenziell häufiger eingreifen. Da dieser Unterschied nicht statistisch signifikant ist, wird die dritte Hypothese, dass Jungen Computertastatur oder Maus häufiger berühren als die Mädchen, verworfen. Auch die Dauer eines Eingriffs unterscheidet sich in geschlechtsheterogenen Paaren nicht statistisch signifikant zwischen Jungen und Mädchen. Der längste Eingriff konnte einem Mädchen zugeschrieben werden und dauerte über zwei Minuten. Auch der Mittelwert und die Standardabweichung der Mädchen waren höher als diejenigen der Jungen. Die vierte Hypothese, dass die Eingriffe der Jungen in geschlechtsheterogenen Paaren länger dauern als diejenigen der Mädchen, wird deshalb verworfen. Mädchen und Jungen greifen in geschlechtsheterogenen Gruppen ungefähr gleich häufig und lange ein.

Diese Resultate widersprechen der Annahme, dass sich Mädchen während der Zusammenarbeit mit Jungen bei einer Robotik-Aufgabe in ihrer Geschlechtsidentität bedroht fühlen und sich deshalb zurückhalten (Flore & Wicherts, 2015). Eine mögliche Erklärung dafür ist, dass die Lernumgebung genug gendergerecht gestaltet war, so dass die Mädchen die Robotik-Aufgabe nicht mehr als männlich wahrgenommen haben. Es kann aber auch sein, dass die männlich geprägten Stereotype zur Informatik bei Kindern und Jugendlichen mittlerweile weniger präsent sind. Die Tendenz der Jungen, ihre Partnerin oder ihren Partner beim Pair Programming für Codefehler verantwortlich zu machen (Zhong et al., 2016), und die Tastatur zu dominieren (Underwood et al., 2000), spiegelt sich nur leicht darin wider, dass die Jungen etwas häufiger eingreifen. Dieser Unterschied ist aber nicht statistisch signifikant. Es ist auch möglich, dass sich die Jungen zurückgehalten haben, weil ihnen bewusst war, dass sie gefilmt werden.

Die dritte Forschungsfrage kann entsprechend den Resultaten dieser Videostudie folgendermassen beantwortet werden: Mädchen und Jungen unterscheiden sich in

geschlechtsheterogenen Paaren weder in der Anzahl noch der Dauer der Eingriffe während der Navigatoren-Rolle statistisch signifikant. Tendenziell greifen die Jungen etwas häufiger ein. Bezüglich Dauer eines Eingriffs unterscheiden sich die Jungen und Mädchen weder statistisch signifikant noch tendenziell in den deskriptiven Statistiken. Mögliche Erklärungen für die fehlenden Unterschiede sind, dass die Lernumgebung genug gendergerecht gestaltet war, Informatikstereotype bei Kindern und Jugendlichen weniger präsent sind als angenommen, oder dass aufgrund der Anwesenheit von Kameras ein anderes Verhalten gezeigt wurde.

5.1.4. Unterschiede der Geschlechterzusammensetzungen bezüglich Befolgens des Pair Programming Leitfadens

Mit der vierten Forschungsfrage soll geprüft werden, inwiefern sich geschlechtsheterogene und -homogene Paare beim Befolgen eines Pair Programming Leitfadens unterscheiden (F4). Es wird vermutet, dass die geschlechtshomogenen Paare die Rollen häufiger wechseln als die geschlechtsheterogenen Paare (H5). Ausserdem wird angenommen, dass die Summe der Anzahl Eingriffe (H6) und die Summe der durchschnittlichen Dauer eines Eingriffs (H7) der Lernenden eines Paares bei geschlechtsheterogenen Paaren höher ist als bei geschlechtshomogenen Paaren.

F4: Inwiefern unterscheiden sich geschlechtsheterogene und geschlechtshomogene Paare beim Befolgen eines zuvor kommunizierten Pair Programming Leitfadens?

H5: In geschlechtshomogenen Paaren werden beim Pair Programming die Rollen häufiger gewechselt als in den geschlechtsheterogenen Paaren.

H6: In geschlechtsheterogenen Paaren berühren die Lernenden in der Navigatoren-Rolle entgegen dem Leitfaden des Pair Programmings häufiger die Computertastatur oder die Maus als die Lernenden in den geschlechtshomogenen Paaren.

H7: In geschlechtsheterogenen Paaren dauern die Eingriffe der Lernenden in der Navigatoren-Rolle durchschnittlich länger als die der Lernenden in geschlechtshomogenen Paaren.

Die Geschlechterzusammensetzungen unterscheiden sich in ihrer Anzahl Rollenwechsel nicht statistisch signifikant. Deskriptive Statistiken lassen aber Unterschiede vermuten. Die geschlechtsheterogenen Paare wiesen verglichen mit den anderen Geschlechterzusammensetzungen die höchste Prozentzahl richtiger Platztäusche und die niedrigste bei den fehlenden Platztäuschen auf. Bei den geschlechtshomogenen weiblichen und geschlechtsheterogenen Gruppen wechselte die Mehrheit entsprechend dem Sollwert oder mehr als nötig. Bei den geschlechtshomogenen Männerpaaren wechselte dagegen die Mehrheit nicht genug häufig. Dies lässt vermuten, dass geschlechtshomogene männliche Paare sich am wenigsten an die Pair Programming Regel des häufigen Rollenwechsels halten.

Geschlechtsheterogene Paare haben dagegen die höchste Wahrscheinlichkeit, die richtige Anzahl Rollenwechsel zu vollziehen. Die geschlechtshomogenen weiblichen Paare halten sich fast so gut an die Rollenwechsel wie die geschlechtsheterogenen Gruppen. Allerdings sind die genannten Unterschiede nicht statistisch signifikant und widersprechen der fünften Hypothese, weshalb diese verworfen wird. In geschlechtshomogenen Paaren werden die Rollen nicht häufiger gewechselt als in geschlechtsheterogenen Gruppen. Das Gegenteil ist der Fall, die geschlechtsheterogenen Paare halten sich tendenziell am besten an die Rollenwechsel.

Bezüglich der Summe der Anzahl Eingriffe der Lernenden eines Paares in der Navigatoren-Rolle zeigen sich keine statistisch signifikanten Unterschiede der Geschlechterzusammensetzungen. Ein Box-Plot und deskriptive Statistiken weisen aber darauf hin, dass in geschlechtshomogenen männlichen Paaren tendenziell am häufigsten eingegriffen wird. Der Mittelwert war bei diesen Paaren am höchsten. Dieser Unterschied ist allerdings nicht statistisch signifikant und widerspricht der sechsten Hypothese, dass geschlechtsheterogene Paare am häufigsten eingreifen. Bei der Prüfung der möglichen Störvariablen hat sich gezeigt, dass die Klassenstufe einen statistisch signifikanten Einfluss auf die Summe der Anzahl Eingriffe der Lernenden eines Paares hat. Paare der Primarstufe greifen häufiger ein als Gruppen der Oberstufe. Deshalb wurde die sechste Hypothese zusätzlich getrennt nach Klassenstufe geprüft. Bei Lernenden der Oberstufe zeigt sich, dass in geschlechtshomogenen männlichen Paaren die Lernenden häufiger eingreifen als in den anderen Geschlechterzusammensetzungen. Dieser Unterschied ist statistisch signifikant, und die Effektstärke kann entsprechend Cohen (1988) als hoch eingestuft werden. Tendenzial scheinen die Jungen auf der Sekundarstufe in geschlechtshomogenen Paaren häufiger einzugreifen als in geschlechtsheterogenen. Dieser Unterschied ist aber nicht statistisch signifikant. Die sechste Hypothese, dass die Lernenden in den geschlechtsheterogenen Paaren häufiger Computertastatur oder Maus berühren als diejenigen in anderen Geschlechterzusammensetzungen, wird verworfen. Die Geschlechterzusammensetzungen unterscheiden sich ebenfalls nicht statistisch signifikant in ihrer Summe der durchschnittlichen Dauer eines Eingriffs der Lernenden eines Paares. Ein Box-Plot und deskriptive Statistiken weisen jedoch darauf hin, dass in den geschlechtsheterogenen Gruppen die stärkste Streuung der durchschnittlichen Dauer eines Eingriffs sowie der höchste Mittel- und Maximalwert vorhanden sind. Die siebte Hypothese, dass in geschlechtsheterogenen Paaren die Eingriffe länger dauern als in den anderen Geschlechterzusammensetzungen, wird ebenfalls verworfen.

Im Theorieteil dieser Arbeit (siehe Kapitel 2.5. Gleichberechtigung der Partizipation beim Pair Programming) wird hergeleitet weshalb vermutet wird, dass eine unausgeglichene Partizipation einer der Gründe für Kompatibilitätsprobleme in einem Paar sein könnte. Das Verbot, in der Navigatoren-Rolle einzugreifen, soll zu einem respektvollen Umgang untereinander und zu einer ausgeglichenen Driver- und Navigator-Zeit der Lernenden führen. Ein regelmässiger Rollenwechsel ermöglicht, dass beide gleichberechtigt partizipieren können. Die Ergebnisse dieser Videostudie weisen darauf hin, dass die Pair Programming Regeln in geschlechtsheterogenen Paaren am besten eingehalten werden, gefolgt von den geschlechtshomogenen weiblichen Paaren. Am wenigsten halten sich die geschlechtshomogenen männlichen Paare an die Regeln.

Dementsprechend könnte aus den Resultaten dieser Studie abgeleitet werden, dass geschlechtsheterogene Paare am besten kooperieren und dadurch die höchste Kompatibilität haben, und die geschlechtshomogenen männlichen Paare die tiefste. Dies widerspricht Studien, welche keine signifikanten Unterschiede bezüglich Kompatibilität zwischen geschlechtshomogenen Frauen- und Männerpaaren (Choi, 2015) oder zwischen geschlechtshomogenen und -heterogenen Paaren (Demir & Seferoglu, 2021a; Zhong et al., 2016) finden können. Ebenso stimmen diese Resultate nicht mit Studien überein, welche darauf hinweisen, dass geschlechtsheterogene Paare weniger kompatibel sind als geschlechtshomogene (Katira et al., 2005; Choi, 2015). Allerdings können die Resultate dieser Videostudie nicht direkt mit diesen Forschungsergebnissen verglichen werden, da die Kompatibilität nur indirekt gemessen und durch bisher nicht bewiesene Annahmen abgeleitet wird. So berichten beispielsweise Studentinnen in geschlechtsheterogenen Paaren von Schwierigkeiten und Konflikten, diejenigen in geschlechtshomogenen Gruppen von einer guten Kompatibilität (Choi, 2015). In dieser Videostudie wurde die Zufriedenheit der Lernenden dagegen nicht erhoben. Die Vermutung, dass Mädchen am meisten davon profitieren, mit Mädchen in der Gruppe zu sein, weil geschlechtshomogene weibliche Paare die engste Partnerschaft aufweisen und mehr kommunizieren und diskutieren (Zhong et al., 2016), kann in dieser Studie nicht bestätigt werden. Auch die Annahmen, dass in geschlechtshomogenen Gruppen die Geschlechtsidentität gegenüber dem Arbeitsselbst in den Hintergrund rückt (Faulstich-Wieland et al., 2004; Kessels, 2002; von Ow & Husfeldt, 2011), und sich Mädchen in geschlechtshomogenen Gruppen selbstbewusster zeigen können (Booth & Nolen, 2012; Kröll, 2010), werden durch die Resultate dieser Studie nicht gestützt. Kinder und Jugendliche haben meist gleichgeschlechtliche Freundinnen und Freunde. Lernende in geschlechtsheterogenen Paaren könnten weniger gut befreundet sein als diejenigen in geschlechtshomogenen. Bisherige Studienergebnisse weisen darauf hin,

dass nicht befreundete Paare mehr leisten (Demir & Seferoglu, 2021a). Ausserdem gibt es Lernende, welche lieber mit einer nicht vertrauten Person zusammenarbeiten, weil dann professioneller zusammengearbeitet wird (Demir & Seferoglu, 2021a). Ein professionelles Arbeiten bedeutet im Kontext von Pair Programming auch das Einhalten der Regeln dieser Praktik. Dies könnte erklären, weshalb geschlechtsheterogene Paare die Leitlinien am besten befolgen. Die enge Freundschaft gleichgeschlechtlicher Lernender könnte auch erklären, weshalb sich geschlechtshomogenen männliche Paare am wenigsten an die Regeln halten. Dazu kommt, dass Jungengruppen häufiger über nicht-aufgabenbezogene Themen kommunizieren als Mädchengruppen (Campe et al., 2020).

Die vierte Forschungsfrage kann dementsprechend folgendermassen beantwortet werden: Geschlechtsheterogene Paare setzen die Pair Programming Regel des regelmässigen Rollenwechsels am besten um, gefolgt von geschlechtshomogenen weiblichen Paaren. Geschlechtshomogene männliche Paare befolgen diese Regel am wenigsten gut. Die Anweisung, dass die Person in der Navigatoren-Rolle weder Computertastatur noch Maus berühren darf, wird von den geschlechtshomogenen männlichen Paaren am häufigsten nicht befolgt. Dies kann auf der Sekundarstufe mit einem statistisch signifikanten Ergebnis, und über alle Klassenstufen mit einer Tendenz gezeigt werden. Die Jungen auf der Sekundarstufe greifen in geschlechtshomogenen Paaren häufiger ein als in geschlechtsheterogenen, wobei dieser Unterschied nicht statistisch signifikant ist. Die Dauer der Eingriffe unterscheidet sich nicht. Diese Resultate widersprechen Studien, welche keine signifikanten Unterschiede bezüglich Kompatibilität der Geschlechterzusammensetzungen oder tiefere Werte bei geschlechtsheterogenen Paaren ausweisen. Allerdings wird in dieser Videostudie nicht direkt die Kompatibilität gemessen, weshalb diese Ergebnisse nicht direkt miteinander verglichen werden können.

5.2. Diskussion der Methode

Die Stichprobe ist nicht repräsentativ. Die Ergebnisse können nicht auf die Grundgesamtheit der Studie, Lernende der fünften Primar- bis dritten Sekundarklasse, sowie Begabungs- und Begabtenförderungsprogramme der dritten bis sechsten Primarklasse in der Zentralschweiz, generalisiert werden. Die Lehrperson meldete ihre Klasse für einen Besuch der Lernumgebung an. Auf diese Entscheidung konnten unter anderem die Dauer der Anreise und das Interesse der Lehrperson an Informatik einen Einfluss haben. Dementsprechend buchten insbesondere Klassen aus dem Kanton Luzern die Lernumgebung. Ausserdem scheinen Klassen, welche die Lernwerkstatt besuchten, ein überdurchschnittlich hohes Programmierwissen gehabt zu haben. In

der Forschungsliteratur zu Gender und Pair Programming im Schulkontext werden hauptsächlich freiwillige Kurse untersucht, bei welchen nur interessierte Lernende teilnehmen. Da Mädchen in freiwilligen Informatikkursen häufig unterrepräsentiert sind, fehlen meist genügend Daten von Frauenpaaren. Bei der vorliegenden Videostudie hatten die Schülerinnen und Schüler selbst keine Wahl, ob sie am Robotik-Kurs teilnehmen möchten oder nicht. Dies erweitert die Erkenntnisse auf eine neue Gruppe von Lernenden. Da die Lernumgebung eine lange Laufzeit hatte und viele Klassen die Lernumgebung besuchten, konnten genügend Daten gesammelt werden, auch von Frauenpaaren. Dies wirkte sich positiv auf die Validität der Ergebnisse aus. Allerdings muss davon ausgegangen werden, dass sich die Lernenden in einem ausserschulischen Kurs anders verhalten als sie es im Klassenzimmer tun würden. Da es sich um einen Robotik-Kurs handelt, mussten die Pair Programming Regeln erweitert werden. Die Person in der Navigatoren-Rolle bediente zusätzlich den Roboter. Dies führt dazu, dass die Resultate dieser Studie nicht ohne Vorbehalt mit Pair Programming Studien ohne Roboter verglichen werden können. Ausserdem könnte es sich bei der Interaktion mit dem Roboter um einen Störfaktor handeln. Andererseits macht es Sinn, einen Bildungsroboter einzubeziehen, weil im Schulkontext häufig mit solchen Modellen gearbeitet wird und sich bisher nur wenige Forschende mit Pair Programming in einem Robotik-Kurs beschäftigen.

Die Lehrpersonen, Eltern und Lernenden kannten das genaue Thema und Ziel der Studie nicht. Die Kursleitungen und Assistierenden wurden nur grob über die Studie informiert. Die Kommunikation über die Ziele der Studie wurde durch Anweisungen im Manual für die Kursleitungen und Assistierenden und durch den Beobachtungsauftrag für die Lehrpersonen standardisiert. Dies wirkte sich positiv auf die Objektivität der Resultate aus. Die Kursleitungen und Assistierenden erhielten klare Anweisungen, damit der Kurs möglichst konform durchgeführt wurde. Dadurch konnten trotz anderer Kursleitung ähnliche Voraussetzungen für alle Lernenden erreicht werden, was sich positiv auf die Reliabilität und Objektivität der Daten auswirkte. Zu den Anweisungen gehörte auch, dass während der Videoaufnahmen bei den Paaren an den Kamerastandorten möglichst nicht eingegriffen werden sollte. Da diese Massnahme nicht genauer konkretisiert wurde, kam es zu unterschiedlich strikten Umsetzungen. Teilweise wurde trotzdem zu viel eingegriffen, wobei diese Fälle durch die Codier-Notizen aussortiert werden konnten. In anderen Fällen ist auf den Videos sichtbar, dass die Paare an den Kamerastandorten sehr lange auf Hilfe warten mussten. Dies könnte zu Frust geführt und dadurch die Validität der Ergebnisse negativ beeinflusst haben.

Die Kameras wurden in der Lernumgebung gut sichtbar platziert, weshalb die Möglichkeit bestand, dass die Lernenden sich anders verhielten, als sie es sonst tun würden. Sie könnten es beispielsweise als sozial erwünscht wahrgenommen haben, sich an die Regeln des Pair Programmings zu halten. Dies könnte die Validität und Reliabilität der Resultate eingeschränkt haben. Die Platzierung der Kameras hatte ausserdem den Nachteil, dass weder der Laptopbildschirm noch die Arbeit auf den Bodenmatten sichtbar waren. Was dazu führte, dass alle Situationen und Konversationen, welche sich ausserhalb der Kamerasicht abspielten, nicht in die Ergebnisse einflossen. Dies wirkte sich negativ auf die Validität der Ergebnisse aus. Die Kameraplatzierung erfolgte aus organisatorischen, technischen und forschungsethischen Gründen, weshalb eine Alternative in diesem Kontext nicht möglich war. Die Paare wurden ausserdem nicht isoliert, das heisst, sie konnten auch mit anderen Paaren interagieren, was die Resultate der Studie beeinflussen konnte. Die Interaktion mit anderen Paaren wird aber in der Forschungsliteratur als positiv hervorgehoben, da dies auch in der akademischen, beruflichen und schulischen Praxis dazugehört und sich positiv auf das Pair Programming auswirken kann (Socha & Sutano, 2015).

Die Lernenden wurden ausschliesslich während der ersten selbstständigen Arbeitsphase direkt nach der Einführung gefilmt. Die Ergebnisse beziehen sich deshalb nur auf diese Anfangsphase. Eine Beobachtung der Zusammenarbeit zu einem späteren Zeitpunkt könnte zu anderen Ergebnissen führen. Beispielsweise könnte es sein, dass sich die Lernenden zu Beginn noch besser an die Anweisungen halten. Die Aufnahmen wurden immer genau zum gleichen Zeitpunkt gemacht, was sich positiv auf die Reliabilität der Daten auswirkte. Da in der Anfangsphase immer das Aufgabenset *Das autonome Fahrzeug* gelöst wurde, beziehen sich die Ergebnisse nur darauf. Das Untersuchen einer weiteren Aufgabe mit einer alternativen Ausrichtung, könnte andere Resultate liefern. Die Evaluation der Lernwerkstatt zeigte, dass die Aufgabe für alle Zielklassenstufen stufengerecht war, und die Lernenden damit weder über- noch unterfordert waren. Dies wirkt sich positiv auf das Pair Programming aus, da diese Technik besonders effektiv bei problemorientierten (Denner et al., 2014) und nicht zu einfachen (Salge & Berente, 2016) Aufgaben ist. Die im Aufgabenset beinhalteten Tipps und Lösungen wurden unterschiedlich intensiv genutzt, was die Lösegeschwindigkeit beeinflusste. Allerdings ermöglichte dieser Aufbau mit den Lösungen auch, dass die Lernenden selbstständig arbeiten konnten und selten auf Hilfe der Kursleitung angewiesen waren. Da die Kursleitungen und Assistierenden wechselten, erhielten nicht alle Lernenden dieselben Hilfestellungen. Die Aufgaben mit den Lösungen führten zu

ausgeglicheneren Bedingungen für die Lernenden und somit zu einer höheren Reliabilität.

Die Paareinteilung erfolgte nach Kriterien, welche aus der Forschungsliteratur abgeleitet wurden. Deshalb lassen sich die Ergebnisse nicht mit Studien vergleichen, in welchen die Lernenden ihre Partnerin oder ihren Partner selbst wählen konnten, sie zufällig, oder nach anderen Kriterien eingeteilt wurden. Die Bildung von homogenen Paaren anhand Leistungsstärke, Programmiervorkenntnissen, Freundschaftslevel und Alter der Lernenden, hat gut funktioniert. Dies zeigte sich in den positiven Rückmeldungen der Lehrpersonen und darin, dass die Störfaktoren kaum einen Einfluss auf die abhängigen Variablen hatten. Die vorhandenen Einflüsse der Störvariablen waren logisch erklärbar und sprachen deshalb für die Validität der abhängigen Variablen. Sekundarschülerinnen und -schüler lösen die Aufgaben schneller als Lernende der Primarschule, was sich durch das höhere Alter und mehr Schulerfahrung erklären lässt. In Paaren der Oberstufe greifen die Lernenden in der Navigatoren-Rolle weniger häufig ein als in Gruppen der Primarstufe. Jüngere Lernende haben häufig mehr Mühe damit, sich an Regeln zu halten. Paare, in welchen beide Lernenden keine Programmiervorkenntnisse haben, fordern mehr Hilfestellungen an als Paare mit programmiererfahrenen Lernenden. Der Unterschied, dass Paare, in welchen die Lernenden das gleiche allgemeine schulische Leistungsniveau haben, mehr Hilfestellungen anfordern als Paare, welche ein ähnliches allgemeines schulisches Leistungsniveau haben, ist dagegen schwierig zu erklären.

Die Störvariablen, welche einen Einfluss auf die abhängigen Variablen gezeigt haben, wurden bei der Hypothesenprüfung miteinbezogen. Allerdings wurde keine Kombination der Störvariablen getestet. Dazu kommt, dass die Einteilung der Paare und die Bewertung von Leistungsstärke, Programmiervorkenntnissen und Freundschaftslevel der Lernenden durch eine subjektive Einschätzung der Lehrperson erfolgte. Meistens kannte die Lehrperson ihre Lernenden sehr gut, allerdings kam es auch vor, dass eine Klasse erst gerade übernommen wurde. Die Einschätzungen erfolgten ausserdem sehr ungenau. Beispielsweise wurde das allgemeine schulische Leistungsvermögen der Lernenden ohne Konkretisierung auf einer vierstufigen Skala von leistungsschwach bis leistungsstark bewertet. Ebenfalls nicht genau war die Angabe, ob die Lernenden Programmiervorkenntnisse hatten oder nicht, weil der Begriff nicht definiert wurde, und deshalb von den Lehrpersonen unterschiedlich interpretiert werden konnte. Somit konnten keine gesicherten Aussagen der Einflüsse dieser Störvariablen getroffen werden. Sie dienten lediglich dazu, sehr starke Einflüsse ausfindig zu machen, um diese bei der Hypothesenprüfung miteinbeziehen zu können.

Die Lernenden wurden angewiesen, die Rollen nach jeder gelösten Teilaufgabenstellung zu wechseln. Dabei wurde der Rollenwechsel auch physisch mit einem Platztausch ausgeführt, was nicht bei allen bisherigen Studien der Fall ist. Die Wahl dieser Anweisung hatte einen Einfluss auf die Ergebnisse und führte dazu, dass die Resultate nicht mit allen Pair Programming Studien verglichen werden können. Dazu kommt, dass einzelne Kursleitungen direkt nach der Einführungsaufgabe oder zu einem anderen Zeitpunkt Platztäusche initiierten. Da dies im Manual nicht vorgegeben wurde, zählte ein erster initiiertes Platztausch nicht. Fälle, in welchen später auf den Rollenwechsel aufmerksam gemacht wurde, wurden ausgeschlossen. Dazu kommt, dass einzelne Kursleitende kommunizierten, dass auch häufiger als nach jeder Teilaufgabenstellung gewechselt werden konnte. Diese Massnahme entsprach nicht den Anweisungen des Manuals, und konnte sich auf die abhängige Variable des Rollenwechsels auswirken. Somit wurde die abhängige Variable Rollenwechsel nicht immer objektiv erfasst.

Der hohe Intraklassenkorrelationskoeffizient wies darauf hin, dass das Codebuch verständlich und klar formuliert war und die zwei studentischen Hilfskräfte sich gut darangehalten haben. Dies wirkte sich positiv auf die Reliabilität und Objektivität aus. Fälle, welche schwierig zu codieren waren, konnten durch die Codier-Notizen ausgeschlossen werden. Die Angabe der letzten bearbeiteten Aufgabe war nicht so genau, da es keine Rolle spielte, ob die Lernenden gerade mit dieser Aufgabe begonnen hatten oder schon kurz vor dem Abschluss standen. Ausserdem dauerte das Bearbeiten der verschiedenen Aufgaben nicht gleich lange und die Lernenden hatten bei der Schlussaufgabe keine Lösung zur Verfügung. Dies führte dazu, dass die Genauigkeit der abhängigen Variable Lösegeschwindigkeit in Frage gestellt werden musste. Bei der Berechnung der abhängigen Variablen fiel ausserdem auf, dass durch das gewählte Datenauswertungsverfahren extreme Werte entstehen konnten. Wenn beispielsweise eine Person nur kurz die Navigatoren-Rolle innehatte und in dieser Zeit sehr häufig eingriff, entstand eine sehr hohe Anzahl Eingriffe pro Navigatoren-Stunde. Ebenfalls sehr hoch fiel die durchschnittliche Dauer eines Eingriffs aus, wenn nur ein Eingriff erfolgte, dieser aber mehrere Minuten lang dauerte. Bei der Anzahl Eingriffe und der durchschnittlichen Dauer wurden die Werte der Lernenden summiert, was die Aussagekraft dieser Variable einschränkte. Da aber immer nur zwischen den Geschlechterzusammensetzungen verglichen wurde, und alle Daten gleich ausgewertet wurden, wirkten sich diese Problematiken nicht stark auf die Validität der Ergebnisse aus. Dennoch wird empfohlen, in zukünftigen Studien alternative Erhebungs- und Auswertungsverfahren zu verwenden.

5.3. Schlussfolgerungen und Forschungsausblick

Das Ziel dieser Studie war es, zu untersuchen, ob und inwiefern sich die Zuteilung in geschlechtshomogene und -heterogene Paare auf die Zusammenarbeit von Lernenden der Primar- und Sekundarstufe beim Pair Programming in einem Robotik-Kurs auswirkt. Gender ist ein kulturelles und soziales Konstrukt, welchem die Gesellschaft geschlechtstypische Verhalten und Charakteristika zuschreibt. Wenn das andere Geschlecht anwesend ist, entsteht der Druck die eigene Geschlechtsidentität zu bewahren, und sich deshalb geschlechterkonform zu verhalten (Maccoby, 1990; Brutsaert, 1999; Amon et al., 2012; Kröll, 2010; von Ow & Husfeldt, 2011; Wedl & Bartsch, 2015). Da das stereotype Bild von Informatik männlich geprägt ist, könnten sich Mädchen in einem Robotik-Kurs bei der Zusammenarbeit mit Jungen in ihrer Geschlechtsidentität bedroht fühlen. Deshalb wurde vermutet, dass sich die Geschlechterzusammensetzungen in ihrer Zusammenarbeit unterscheiden. Eine Technik, welche sich allgemein positiv auf die Kollaboration auswirkt, und für Frauen besonders förderlich sein soll, ist Pair Programming. Dabei programmieren zwei Personen gemeinsam, und übernehmen die Rollen Driver und Navigator. Die Geschlechterzusammensetzung der Paare wird als einer der möglichen Einflussfaktoren auf den Erfolg des Pair Programmings genannt. Forschende haben bereits viele Leitlinien formuliert, welche eingehalten werden müssen, damit Pair Programming erfolgreich ist. In dieser Videostudie wurde das Einhalten eines zuvor kommunizierten Pair Programming Leitfadens untersucht. Der Schwerpunkt wurde auf den regelmässigen Rollenwechsel und das Verbot in der Navigatoren-Rolle Maus oder Tastatur zu berühren, gelegt. Zusätzlich wurde untersucht, ob sich die Paare bezüglich Effizienz des Pair Programmings unterscheiden. Dazu wurden die Lösegeschwindigkeit und die Anzahl der angeforderten Hilfestellungen erhoben. Die Videostudie wurde in der Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken*, welche im Herbstsemester 2020 lief und von Schülerinnen und Schülern der 5. bis 9. Klasse besucht werden konnte, durchgeführt.

Die Ergebnisse dieser Studie weisen darauf hin, dass sich geschlechtshomogene und -heterogene Paare bezüglich Lösegeschwindigkeit nicht statistisch signifikant unterscheiden, was bisherigen Studien entspricht. Geschlechtsheterogene Paare auf der Sekundarstufe zeigen aber die grösste Variabilität der Lösegeschwindigkeit, was auf Kompatibilitätsprobleme hinweisen könnte. Auch bezüglich der Anzahl angeforderter Hilfestellungen unterscheiden sich die Geschlechterzusammensetzungen nicht statistisch signifikant. In geschlechtsheterogenen Paaren greifen die Jungen nicht signifikant häufiger oder länger ein als die Mädchen. Die Anzahl Rollenwechsel der

Geschlechterzusammensetzungen unterscheidet sich ebenfalls nicht statistisch signifikant. Geschlechtsheterogene Paare führen aber mit der höchsten Wahrscheinlichkeit die richtige Anzahl Rollenwechsel durch. Die geschlechtshomogenen männlichen Paare wechseln dagegen am wahrscheinlichsten zu wenig oft. Diese Unterschiede sind aber nicht statistisch signifikant. Die Summe der Anzahl und Dauer der Eingriffe der Lernenden eines Paares unterscheidet sich nicht statistisch signifikant zwischen den Geschlechterzusammensetzungen. Bei Lernenden der Sekundarstufe zeigt sich aber, dass in geschlechtshomogenen männlichen Paaren die Lernenden statistisch signifikant häufiger eingreifen als in den anderen Geschlechterzusammensetzungen. Tendenziell scheinen Jungen auf der Sekundarstufe in geschlechtshomogenen männlichen Paaren häufiger einzugreifen als in geschlechtsheterogenen. Dieser Unterschied ist aber nicht statistisch signifikant. Geschlechtsheterogene Gruppen weisen die grösste Variabilität der Summe der durchschnittlichen Dauer eines Eingriffs sowie den höchsten Maximal- und Mittelwert auf. Diese Unterschiede sind aber nicht statistisch signifikant.

Zusammengefasst kann gesagt werden, dass sich geschlechtsheterogene Paare tendenziell am besten, und geschlechtshomogene männliche Paare am schlechtesten an die Pair Programming Regeln des regelmässigen Rollenwechsels und des Verbots der Eingriffe in der Navigatoren-Rolle halten. Allerdings zeigen geschlechtsheterogene Paare auch die grösste Variabilität der Lösegeschwindigkeit und der Summe der durchschnittlichen Dauer eines Eingriffs, was auf Kompatibilitätsprobleme hinweisen könnte. Deshalb kann mit den Ergebnissen dieser Studie keine Empfehlung für oder gegen eine der Geschlechterzusammensetzungen gemacht werden. Die Daten weisen aber darauf hin, dass es für Jungen vorteilhafter ist, in geschlechtsheterogenen als in geschlechtshomogenen Paaren zu arbeiten, insbesondere auf der Sekundarstufe. Die geschlechtshomogenen weiblichen Paare scheinen eine sichere Wahl zu sein und sich nicht signifikant von den geschlechtsheterogenen Paaren zu unterscheiden. Die Zusammenarbeit in den geschlechtsheterogenen Paaren muss aber noch genauer untersucht werden, damit die grossen Unterschiede der Lösegeschwindigkeit und der Summe der durchschnittlichen Dauer eines Eingriffs erklärt werden können.

Diese Videostudie beschäftigte sich nur mit quantitativ messbaren Indikatoren der Kollaboration. Für eine vollständige Bewertung der Zusammenarbeit der Geschlechterzusammensetzungen beim Pair Programming wäre auch eine qualitative Betrachtung nötig. Die Forschung weist beispielsweise darauf hin, dass eine gute Kommunikation entscheidend für den Erfolg des Pair Programmings ist. Hierzu könnten die ebenfalls aufgezeichneten Gespräche der Lernenden untersucht werden. Bisher

wurde die Audiospur nur als Hilfestellung beim Codieren verwendet. Dabei wäre beispielsweise interessant, ob jemand das Gespräch dominiert, über was gesprochen wird, oder welche Paare die Empfehlungen der Forschung zur Kommunikation beim Pair Programming am besten einhalten. Auch interessant wäre, die durch die quantitative Betrachtung am besten bewerteten Gruppen qualitativ genauer zu betrachten. Dadurch könnten die quantitativen Bewertungen überprüft und Best Practice Beispiele gefunden werden. Ebenfalls könnten die Notizen der Lehrpersonen auf dem Beobachtungsauftrag analysiert werden. Diese wurden bisher nur bei der Datenbereinigung berücksichtigt. Beispielsweise wäre interessant, ob sich die Einschätzung der Lehrperson zur Zusammenarbeit der Lernenden mit den durchgeführten Auswertungen deckt oder nicht. Die Produktivität der Paare wird in dieser Studie nur mit der Lösegeschwindigkeit untersucht. Allerdings widerspiegelt diese Angabe nur bedingt die Programmierleistung. Deshalb würde auch hier ein qualitativer Ansatz die Erkenntnisse erweitern. Besonders spannend wäre auch eine Befragung der Lernenden, um herauszufinden, in welcher Geschlechterzusammensetzung sich die Schülerinnen und Schüler am wohlsten fühlen. Die Lernenden könnten auch, wie in anderen Studien, nach Kompatibilitätsproblemen gefragt werden.

Durch die Kameraposition konnten weder die Tätigkeiten auf dem Bildschirm noch diejenigen auf der Bodenmatte untersucht werden. Deshalb ist es mit den vorhandenen Videoaufnahmen nicht möglich, die Interaktion mit dem Roboter zu analysieren. Für ein zukünftiges Forschungsprojekt könnte dies interessant sein, weil es bisher wenige Studien gibt, welche das Pair Programming bei der Arbeit mit Robotern untersuchen. Beispielsweise könnte geprüft werden, ob jemand den Roboter häufiger berührt. Bei dieser Videostudie wurde nur das Lösen der Aufgabe *Das autonome Fahrzeug* und nur die Anfangsphase untersucht. Für künftige Forschungen wäre es auch interessant, die Zusammenarbeit der Lernenden bei anderen Aufgaben zu untersuchen, beispielsweise bei einer Programmier-, einer Konstruktions- und einer Gestaltungsaufgabe. Ein anderer Zeitpunkt der Aufnahme als der Start wäre ebenfalls interessant. Die Lernenden könnten zu Beginn noch motivierter sein oder sich mehr zurückhalten. Zusätzlich könnten einzelne Lernende betrachtet werden. Beispielsweise wäre interessant, ob eine Person häufiger eingreift, wenn diese bereits Programmierwissen hat.

6. Literaturverzeichnis

- Abd Gani, D., Mohammed, N., & Idrus, Z. (2010). Experimenting with pair programming in an introductory programming course. *Jurnal Teknologi Maklumat dan Multimedia*, 9, 71-82.
- Akinola, S. O. (2016). Computer programming skill and gender difference: An empirical study. *American Journal of Scientific and Industrial Research*, 7(1), 1–9.
- Akour, M., & Al-Radaideh, K., Alazzam, I., & Alsmadi, I. (2013). Effective pair programming practice: toward improving student learning in software engineering class. *International Journal of Teaching and Case Studies*, 4(4), 336-345. <https://doi.org/10.1504/IJTCS.2013.060635>
- Al-Ramahi, M., Alazzam, I., & Alsmadi, I. (2013). The impact of using pair programming: a case study. *International Journal of Teaching and Case Studies*, 4(4), 313-329. <https://doi.org/10.1504/IJTCS.2013.060633>
- Alshehri, S., & Benedicenti, L. (2014). Ranking and rules for selecting two persons in pair programming. *Journal of Software*, 9(9), 2467-2473.
- Amon, H., Bartosch, I., Lembens, A., & Wenzl, I. (2012). *Gender_Diversity-Kompetenz im naturwissenschaftlichen Unterricht. Fachdidaktische Anregungen für Lehrerinnen und Lehrer*. IMST Gender_Diversitäten Netzwerk. https://www.imst.ac.at/app/webroot/files/nawi_fertig_28.11.pdf
- Anjum, S., Batul, H., & Sirshar, M. (2015). A comparative study of ensuring quality in pair programming. *International Journal of Computer and Communication System Engineering (IJCCSE)*, 2(1), 160-171.
- Arisholm, E., Gallis, H., Dybå, T., & Sjøberg, D. (2007). Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering*, 33(2), 65-86. <https://doi.org/10.1109/TSE.2007.17>
- Asendorpf, J., & Wallbott, H. G. (1979). Masse der Beobachterübereinstimmung. Ein systematischer Vergleich. *Zeitschrift für Sozialpsychologie* 10(3), 243-252.
- Bailey, R., & Mentz, E. (2017). The value of pair programming in the IT classroom. *The Independent Journal of Teaching and Learning*, 12(1), 90-103.
- Barman, C. R. (1999). Students' views about scientists and school science: Engaging K-8 teachers in a national study. *Journal of Science Teacher Education*, 10(1), 43–54. <https://www.jstor.org/stable/43156207>
- Başer, M. (2013). Attitude, gender and achievement in computer programming. *Middle East Journal of Scientific Research*, 14(2), 248-255.
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77. <https://doi.org/10.1109/2.796139>
- Beck, K. (2003). *Extreme Programming: die revolutionäre Methode für Softwareentwicklung in kleinen Teams; [das Manifest]*. Pearson Deutschland GmbH.
- Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. (2003). Gender differences in computer science students. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, 35(1), 49-53. <https://doi.org/10.1145/611892.611930>

- Booth, A. L., & Nolen, P. (2012). Gender differences in risk behaviour: Does nurture matter? *The Economic Journal*, 122(558), F56–F78. <https://doi.org/10.1111/j.1468-0297.2011.02480.x>
- Bosinski, H. A. G. (2000). Determinanten der Geschlechtsidentität - Neue Befunde zu einem alten Streit. *Sexuologie* 7, 96-140.
- Bowman, N. A., Jarratt, L., Culver, K. C., & Segre, A. M. (2019). How prior programming experience affects students' pair programming experiences and outcomes. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 170-175. <https://doi.org/10.1145/3304221.3319781>
- Bowman, N. A., Jarratt, L., Culver, K. C., & Segre, A. M. (2020). Pair programming in perspective: effects on persistence, achievement, and equity in computer science. *Journal of Research on Educational Effectiveness*, 13(4), 731-758. <https://doi.org/10.1080/19345747.2020.1799464>
- Brought, G., Wahls, T., & Eby, L. M. (2011). The case for pair programming in the computer science classroom. *ACM Transactions on Computing Education*, 11(1), 1-21. <https://doi.org/10.1145/1921607.1921609>
- Brutsaert, H. (1999). Coeducation and gender identity formation: A comparative analysis of secondary schools in Belgium. *British Journal of Sociology of Education*, 20(3), 343-353. <https://doi.org/10.1080/01425699995308>
- Bundesamt für Statistik (2017a). *Berufliche Grundbildung*. Bern.
- Bundesamt für Statistik (2017b). *Studierende und Abschlüsse der Hochschulen in den MINT-Fächern*. Neuchâtel.
- Çal, H., & Can, G. (2020). The influence of pair programming on secondary school students' confidence and achievement in computer programming. *Trakya Eğitim Dergisi*, 10(1), 221-237. <https://doi.org/10.24315/tred.575098>
- Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: variations in interactions and behaviors. *Computer Science Education*, 30(1), 22-46. <https://doi.org/10.1080/08993408.2019.1648119>
- Carli, L. L. (1989). Gender differences in interaction style and influence. *Journal of Personality and Social Psychology*, 56(4), 565–576. <https://doi.org/10.1037/0022-3514.56.4.565>
- Celepölu, M., Galdo, A. C., Boyer, K. E., Wiebe, E. N., Mott, B. W., & Lester, J. C. (2020). Student reflections on pair programming in middle school: A thematic analysis. In M. Gresalfi & I. S. Horn (Eds.), *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020* (pp. 803-804).
- Chang, S., Shieh, R. S., Liu, E. Z., & Yu, P. (2012). Factors influencing women's attitudes towards computers in a computer literacy training program. *Turkish Online Journal of Educational Technology*, 11(4), 177-187.
- Cheryan, S., Plaut, V. C., Handron, C., & Hudson, L. (2013). The stereotypical computer scientist: gendered media representations as a barrier to inclusion for women. *Sex Roles* 69, 58-71. <https://doi.org/10.1007/s11199-013-0296-x>
- Cheryan, S., Siy, J., Vichayapai, M., Drury, B., & Kim, S. (2011). Do female and male role models who embody STEM stereotypes hinder women's anticipated success in STEM? *Social Psychological and Personality Science*, 2(6), 656–664. <https://doi.org/10.1177/1948550611405218>

- Choi, K. S. (2015). A comparative analysis of different gender pair combinations in pair programming. *Behaviour & Information Technology*, 34(8), 825–837. <https://doi.org/10.1080/0144929X.2014.937460>
- Cockburn, A., & Williams, L. (2000). The costs and benefits of pair programming. In M. Marchesi & G. Succi (Eds.), *Extreme programming and flexible processes in software engineering* (pp. 223–244).
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.).
- Coto, M., & Mora, S. (2019). Are there any gender differences in students' emotional reactions to programming learning activities? *Proceedings of the XX International Conference on Human Computer Interaction (Interacción 2019)*, 51, 1-8. <https://doi.org/10.1145/3335595.3335608>
- Creamer, E. G., Lee, S., & Meszaros, P. S. (2007). Predicting women's interest in and choice of a career in information technology: A statistical model. In C. J. Burger, E. G. Creamer & P. S. Meszaros (Eds.), *Reconfiguring the firewall: Recruiting women to information technology across cultures and continents* (pp. 15–38). AK Peters.
- Deaux, K. & LaFrance, M. (1998). Gender. In D.T Gilbert, S. Fiske & G. Lindzey (Eds.), *The handbook of social psychology* (4th ed., pp. 788-828). McGraw Hill.
- Demir, Ö., & Seferoglu, S. S. (2021a). The effect of determining pair programming groups according to various individual difference variables on group compatibility, flow, and coding performance. *Journal of Educational Computing Research*, 59(1), 41–70. <https://doi.org/10.1177/0735633120949787>
- Demir, Ö., & Seferoglu, S. S. (2021b). A comparison of solo and pair programming in terms of flow experience, coding quality, and coding achievement. *Journal of Educational Computing Research*, 58(8), 1448–1466. <https://doi.org/10.1177/0735633120949788>
- Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, 46(3), 277-296. <https://doi.org/10.1080/15391523.2014.888272>
- Diekman, A. B., Brown, E. R., Johnston, A. M., & Clark, E. K. (2010). Seeking congruity between goals and roles: A new look at why women opt out of STEM careers. *Psychological Science*, 21(8), 1051–1057. <https://doi.org/10.1177/0956797610377342>
- Diekman, A. B., Clark, E. K., Johnston, A. M., Brown, E. R., & Steinberg, M. (2011). Malleability in communal goals and beliefs influences attraction to STEM careers: Evidence for a goal congruity perspective. *Journal of Personality and Social Psychology*, 101(5), 902-918. <https://doi.org/10.1037/a0025199>
- Doctor, R.F. (1988). *Transvestites and transsexuals. Toward a theory of cross-gender behavior*. Plenum Press. <https://doi.org/10.1007/978-1-4613-0997-0>
- Eagly, A. H. (1987). *Sex differences in social behavior: A social-role interpretation*. Psychology Press. <https://doi.org/10.4324/9780203781906>
- Eagly, A. H., Wood, W., & Diekman, A. B. (2000). Social role theory of sex difference and similarities: A current appraisal. In T. Eckes & H. M. Trautner (Eds.), *The developmental social psychology of gender* (pp. 123-174). Lawrence Erlbaum Associates Publishers.

- Eckes, T. (2008). Geschlechterstereotype: Von Rollen, Identitäten und Vorurteilen. In R. Becker & B. Kortendiek (Eds.), *Handbuch Frauen- und Geschlechterforschung. Theorie, Methoden, Empirie* (3rd ed.) (pp. 171-182). VS Verlag. https://doi.org/10.1007/978-3-531-91972-0_20
- Elster, D. (2007). In welchen Kontexten sind naturwissenschaftliche Inhalte für Jugendliche interessant? *Plus Lucis*, 3, 2-8.
- Esch, M., & Herrmann, J. (Eds.) (2008). *Qualität durch Chancengleichheit. Gleichstellung als strategischer Faktor im Qualitätsmanagement technischer Universitäten*. Bertelsmann.
- Faja, S. (2011). Pair programming as a team-based learning activity: A review of research. *Issues in Information Systems*, 12(2), 207-216. https://doi.org/10.48009/2_iis_2011_207-216
- Faja, S. (2014). Evaluating effectiveness of pair programming as a teaching tool in programming courses. *Information Systems Education Journal (ISEDJ)*, 12(6), 36-45.
- Faulstich-Wieland, H., Weber, M., & Willems, K. (2004). *Doing Gender im heutigen Schulalltag. Empirische Studien zur sozialen Konstruktion von Geschlecht in schulischen Interaktionen*. Juventa.
- Felder, R., & Silverman, L. (1988). Learning and teaching styles in engineering education. *Engineering Education*, 78(7), 674-681.
- Flore, P. C., & Wicherts, J. (2015). Does stereotype threat influence performance of girls in stereotyped domains? A meta-analysis. *Journal of School Psychology*, 53(1), 25–44. <https://doi.org/10.1016/j.jsp.2014.10.002>
- Fraunhofer IAIS (Ed.) (2019). *Roberta-Grundlagenband* (3rd ed.). Fraunhofer Verlag.
- Funke, A., Berges, M., Mühling, A., & Hubwieser, P. (2015). Gender differences in programming: research results and teachers' perception. *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, 161-162. <https://doi.org/10.1145/2828959.2828982>
- Gómez, O. S., Solari, M., Calvache, C. J. P., & Ledezma-Carrizalez, A. C. (2017). A controlled experiment on productivity of pair programming gender combinations: Preliminary results. *Proceedings of the XX Ibero-American Conference on Software Engineering (ESELAW '17 – Experimental Software Engineering Track)*, 197-210.
- Greguras, G. J., & Robie, C. (1998). A new look at within-source interrater reliability of 360-degree feedback ratings. *Journal of Applied Psychology*, 83(6), 960-968. <https://doi.org/10.1037/0021-9010.83.6.960>
- Greve, W., & Wentura, D. (1997). *Wissenschaftliche Beobachtung: Eine Einführung* (2nd ed.). Beltz.
- Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: a literature review, *Computer Science Education*, 21(2), 135-173. <https://doi.org/10.1080/08993408.2011.579808>
- Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies*, 24, 2943–2952. <https://doi.org/10.1007/s10639-019-09905-3>

- Isong, B., Moemi, T., Dladlu, N., Motlhabane, N., Ifeoma, O., & Gasela, N. (2016). Empirical confirmation of pair programming effectiveness in the teaching of computer programming. *International Conference on Computational Science and Computational Intelligence (CSCI)*, 276-281. <https://doi.org/10.1109/CSCI.2016.0060>
- Jarratt, L., Bowman, N. A., Culver, K. C., & Segre, A. M. (2019). A large-scale experimental study of gender and pair composition in pair programming. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '19)*, 176–181. <https://doi.org/10.1145/3304221.3319782>
- Jeon, T., & Manuelli, K. (2013). Investigating the impact of pair-programming on entry level IT students. In M. Lopez & M. Verhaart (Eds.), *Proceedings of the 4th Annual Conference of Computing and Information Technology Education and Research in New Zealand (CITRENZ2013)* (pp. 178-179).
- Katira, N., Williams, L., & Osborne, J. (2005). Towards increasing the compatibility of student pair programmers. *Proceedings. 27th International Conference on Software Engineering (ICSE 2005)*, 625-626. <https://doi.org/10.1109/ICSE.2005.1553618>
- Kenny, D. A., Kashy, D. A., & Cook, W. L. (2006). *Dyadic data analysis*. Guildford.
- Kessels, U. (2002). *Undoing Gender in der Schule. Eine empirische Studie über Koedukation und Geschlechtsidentität im Physikunterricht*. Juventa.
- Kolb, D. A. (1999). *Learning style inventory, Version 3*. Hay Resources Direct.
- Kröll, D. (Ed.) (2010). "Gender und MINT". *Schlussfolgerungen für Unterricht, Beruf und Studium*. Tagungsband zum Fachtag am 15.02.2010, Universität Kassel und Max-Eyth-Schule Kassel. Kassel University Press. <http://www.uni-kassel.de/upress/online/frei/978-3-89958-974-0.volltext.frei.pdf>
- Lewis, C. M., & Shah, N. (2015). How Equity and Inequity Can Emerge in Pair Programming. *Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15)*, 41-50. <https://doi.org/10.1145/2787622.2787716>
- Li, Z., Plaue, C., & Kraemer, E. (2013). A spirit of camaraderie: The impact of pair programming on retention. *26th International Conference on Software Engineering Education and Training (CSEE&T)*, 209-218, <https://doi.org/10.1109/CSEET.2013.6595252>
- Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education*, 22(3), 219-236. <https://doi.org/10.1080/08993408.2012.713180>
- Lui, K. M., Barnes, K. A., & Chan, K. C. C. (2010). Pair programming: Issues and challenges. In T. Dingsøyr, T. Dybå & N. Moe (Eds.), *Agile Software Development* (pp. 143-163). Springer. https://doi.org/10.1007/978-3-642-12575-1_7
- Lünenborg, M., & Maier, T. (2013). *Gender media studies: Eine Einführung*. UVK.
- Maccoby, E. E. (1990). Gender and relationships: A developmental account. *American Psychologist*, 45(4), 513-520. <https://doi.org/10.1037/0003-066X.45.4.513>
- Maccoby, E. E. (1998). *The two sexes: growing up apart, coming together*. Belknap Press/Harvard University Press.

- Maguire, P., Maguire, R., Hyland, P., & Marshall, P. (2014). Enhancing collaborative learning using pair programming: Who benefits? *All Ireland Journal of Teaching and Learning in Higher Education (AISHE-J)*, 6(2), 1411-14125. <http://ojs.aishe.org/index.php/aishe-j/article/view/141>
- McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8), 90-95. <http://dx.doi.org/10.1145/1145287.1145293>
- Mercier, E. M., Barron, B., & O'Connor, K. M. (2006). Images of self and others as computer users: The role of gender and experience. *Journal of Computer Assisted Learning*, 22(5), 335–348. <https://doi.org/10.1111/j.1365-2729.2006.00182.x>
- Mohamed, S. F. P. (2017). Enhancing students' interest and achievement in programming 1 subject through pair programming. *Sintok International Conference on Social Science and Management (SICONSEM 2017)*, 7-9. <http://repo.uum.edu.my/id/eprint/24484>
- Mosley, P., & Kline, R. (2006). Engaging students: A framework using LEGO® robotics to teach problem solving. *Information Technology, Learning, and Performance Journal*, 24(1), 39–45.
- Myers, I. B., McCaulley, M. H., Quenk, N. L., Hammer, A. L. (1998). *MBTI manual: A guide to the development and use of the Myers Briggs Type Indicator* (3rd ed.). Consulting Psychologists Press.
- Nawahdah, M., & Taji, D. (2016). Investigating students' behavior and code quality when applying pair-programming as a teaching technique in a middle eastern society. *IEEE Global Engineering Education Conference (EDUCON)*, 32-39. <https://doi.org/10.1109/EDUCON.2016.7474527>
- Nosek, B. A., Smyth, F. L., Sriram, N., Lindner, N. M., Devos, T., Ayala, A., Bar-Anan, Y., Bergh, R., Cai, H., Gonsalkorale, K., Kesebir, S., Maliszewski, N., Neto, F., Olli, E., Park, J., Schnabel, K., Shiomura, K., Tulbure, B. T., Wiers, R. W., ... Greenwald, A. (2009). National differences in gender–science stereotypes predict national sex differences in science and math achievement. *Proceedings of the National Academy of Sciences*, 106(16), 10593–10597. <https://doi.org/10.1073/pnas.0809921106>
- O'Donnell, C., Buckley, J., Mahdi, A. E., Nelson, J., & English, M. (2015). Evaluating pair-programming for non-computer science major students. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*, 569-574. <https://doi.org/10.1145/2676723.2677289>
- Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers? A case study. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, 13(1), 1-16. <http://doi.org/10.4018/IJWLTT.2018010101>
- Plonka L., Segal J., Sharp H., van der Linden, J. (2011). Collaboration in pair programming: Driving and switching. In A. Sillitti, O. Hazzan, E. Bache & X. Albaladejo (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2011. Lecture Notes in Business Information Processing 77* (pp. 43-59). Springer. https://doi.org/10.1007/978-3-642-20677-1_4
- Plonka, L., Segal, J., Sharp, H., & van der Linden, J. (2012a). Investigating equity of participation in pair programming. *Agile India 2012*, 20-29. <https://doi.org/10.1109/AgileIndia.2012.16>

- Plonka, L., Sharp, H., & van der Linden, J. (2012b). Disengagement in pair programming: Does it matter? *Proceedings of the 34th International Conference on Software Engineering (ICSE '12)*, 496-506. <https://doi.org/10.1109/ICSE.2012.6227166>
- Poonam, R., & Yasser, C. M. (2018). An experimental study to investigate personality traits on pair programming efficiency in extreme programming. *Proceedings of the 5th International Conference on Industrial Engineering and Applications (ICIEA)*, 95-99. <https://doi.org/10.1109/IEA.2018.8387077>
- Pörksen, B. (Ed.) (2015). *Schlüsselwerke des Konstruktivismus* (2nd ed.). VS Verlag für Sozialwissenschaften. <https://doi.org/10.1007/978-3-531-19975-7>
- Pröpper, N. (2012). *Agile Techniken für klassisches Projektmanagement: Qualifizierung zum PMI-ACP*. MITP-Verlag.
- Radermacher, A., & Walia, G. (2011). Investigating student-instructor interactions when using pair programming: An empirical study. *Proceedings of the 24th IEEE-CS Conference on Software Engineering Education and Training (CSEET)*, 41-50. <https://doi.org/10.1109/CSEET.2011.5876117>
- Rodríguez, F. J., Price, K. M., & Boyer, K. E. (2017). Exploring the pair programming process: Characteristics of effective collaboration. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*, 507-512. <https://doi.org/10.1145/3017680.3017748>
- Salge, C., & Berente, N. (2016). Pair programming vs. solo programming: What do we know after 15 years of research? *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS)*, 5398-5406. <https://doi.org/10.1109/HICSS.2016.667>
- Salleh, N. (2008). A systematic review of pair programming research - Initial results. *Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRSC08)*, Christchurch.
- Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering*, 37(4), 509-525. <https://doi.org/10.1109/TSE.2010.59>
- Salleh, N., Mendes, E., & Grundy, J. (2014). Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments. *Empirical Software Engineering*, 19, 714-752. <https://doi.org/10.1007/s10664-012-9238-4>
- Schmidt, S. J. (1994). *Kognitive Autonomie und soziale Orientierung. Konstruktivistische Bemerkungen zum Zusammenhang von Kognition, Kommunikation, Medien und Kultur*. Suhrkamp. <https://doi.org/10.17192/ep1995.1.4366>
- Seo, Y., & Kim, J. (2016). Analyzing the effects of coding education through pair programming for the computational thinking and creativity of elementary school students. *Indian journal of science and technology*, 9(46). <https://doi.org/10.17485/ijst/2016/v9i46/107837>
- Sfetsos, P., Adamidis, P., Angelis, L., Stamelos, I., & Deligiannis, I. (2012). Investigating the impact of personality and temperament traits on pair programming: A controlled experiment replication. *Proceedings of the Eighth International Conference on the Quality of Information and Communications Technology*, 57-65. <https://doi.org/10.1109/QUATIC.2012.36>

- Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, *86*(2), 420- 428. <https://doi.org/10.1037//0033-2909.86.2.420>
- Singh, K., Allen, K. R., Scheckler, R., & Darlington, L. (2007). Women in computer-related majors: A critical synthesis of research and theory from 1994 to 2005. *Review of Educational Research*, *77*(4), 500–533. <https://doi.org/10.3102/0034654307309919>
- Smith, M. O., Giugliano, A., & DeOrio, A. (2017). Long term effects of pair programming. *IEEE Transactions on Education*, *61*(3), 187-194. <https://doi.org/10.1109/TE.2017.2773024>
- Socha, D., & Sutanto, K. (2015). The "Pair" as a problematic unit of analysis for pair programming. *IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, 64-70. <https://doi.org/10.1109/CHASE.2015.16>
- Ströhl, A. (2014). Der Konstruktivismus. In A. Ströhl, *Medientheorien kompakt* (pp. 122-126). UVK.
- Sutter, M., Bosman, R., Kocher, M.G., & van Winden, F. (2009). Gender pairing and bargaining—Beware the same sex! *Experimental Economics*, *12*(3), 318–331. <https://doi.org/10.1007/s10683-009-9217-9>
- Tsan, J., Boyer, K. E., & Lynch, C. F. (2016). How early does the CS gender gap emerge? A study of collaborative problem solving in 5th Grade Computer Science. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*, 388-393. <https://doi.org/10.1145/2839509.2844605>
- Tunga, Y., & Tokel, T. (2018). The use of pair programming in education: A systematic literature review. *Education Conference 2018*, 19-29.
- Umapathy, K., & Ritzhaupt, A. D. (2017). A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education*, *17*(4), 1-13. <https://doi.org/10.1145/2996201>
- Underwood, G., McCaffrey, M., & Underwood, J (1990). Gender differences in a cooperative computer-based language task. *Educational Research*, *32*(1), 44–49. <https://doi.org/10.1080/0013188900320105>
- Underwood, J., Underwood, G., & Wood, D. (2000). When does gender matter? Interactions during computer-based problem solving. *Learning and Instruction*, *10*(5), 447–462.
- von Ow, A., & Husfeldt, V. (2011). *Geschlechterdifferenzen und schulische Leistungen. Eine Übersicht zum Forschungsstand*. PH FHNW.
- Weber, S. (Ed.) (2010). *Theorien der Medien: Von der Kulturkritik bis zum Konstruktivismus* (2nd ed.). UVK.
- Wedl, J. & Bartsch, A. (2015). *Teaching Gender? Zum reflektierten Umgang mit Geschlecht im Schulunterricht und in der Lehramtsausbildung*. transcript Verlag. <https://doi.org/10.14361/9783839428221>
- Werner, L., Campe, S., & Denner, J. (2005). Middle school girls + games programming = information technology fluency. *Proceedings of the 6th conference on information technology education (SIGITE '05)*, 301–305. <https://doi.org/10.1145/1095714.1095784>

- Werner, L., Denner, J., & Bean, S. (2004). Pair Programming strategies for middle school girls. *Proceedings of the 7th IASTED International Conference on Computers and Advanced Technology in Education*, 161-166. <https://doi.org/10.13140/2.1.2533.2165>
- Werner, L., Denner, J., Campe, S., Ortiz, E., DeLay, D., Hartl, A. G. & Laursen, B. (2013). Pair programming for middle school students: Does friendship influence academic outcomes? *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*, 421-426. <https://doi.org/10.1145/2445196.2445322>
- Werner, L., & Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education* 42(1), 29-49. <https://doi.org/10.1080/15391523.2009.10782540>
- Werner, L., Hanks, B., & McDowell, C. (2004). Pair-programming helps female computer science students. *ACM Journal of Educational Resources in Computing*, 4(1), 1-8. <https://doi.org/10.1145/1060071.1060075>
- Williams, L., & Kessler, R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, 43(5), 108-114. <https://doi.org/10.1145/332833.332848>
- Williams, L., Layman, L., Osborne, J., & Katira, N. (2006). Examining the compatibility of student pair programmers. *Proceedings of the conference on AGILE 2006 (AGILE '06) Conference*, 411-420. <https://doi.org/10.1109/AGILE.2006.25>
- Williams, L., McCrickard, D. S., Layman, L., & Hussein, K. (2008). Eleven guidelines for implementing pair programming in the classroom. *Proceedings of the AGILE 2008 (AGILE '08) Conference*, 445-452. <https://doi.org/10.1109/Agile.2008.12>
- Wirtz, M., Caspar, F. (2002). *Beurteilerübereinstimmung und Beurteilerreliabilität. Methoden zur Bestimmung und Verbesserung der Zuverlässigkeit von Einschätzungen mittels Kategoriensystemen und Ratingskalen*. Hogrefe.
- Ying, K. M., Pezzullo, L.G., Ahmed, M., Crompton, K., Blanchard, J., & Boyer, K. E. (2019). In their own words: Gender differences in student perceptions of pair programming. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*, 1053-1059. <https://doi.org/10.1145/3287324.3287380>
- Yuen, T., Boecking, M., Stone, J. P., Tiger, E. P., Gomez, A., Guillen, A., & Arreguin, A. (2014). Group tasks, activities, dynamics, and interactions in collaborative robotics projects with elementary and middle school children. *Journal of STEM Education: Innovations and Research*, 15(1), 39-45. <https://www.learntechlib.org/p/152524/>
- Zarb, M., Hughes, J., & Richards, J. (2013). Industry-inspired guidelines improve students' pair programming communication. *Proceedings of the 18th Association for Computing Machinery (ACM) conference on Innovation and technology in computer science education (ITiCSE '13)*, 135-140. <https://doi.org/10.1145/2462476.2462504>
- Zarb, M., Hughes, J., & Richards, J. (2014). Evaluating industry-inspired pair programming communication guidelines with undergraduate students. *Proceedings of the 45th Association for Computing Machinery (ACM) technical symposium on computer science education (SIGCSE '14)*, 361-366. <https://doi.org/10.1145/2538862.2538980>

- Zarb, M., Hughes, J., & Richards, J. (2015). Further evaluations of industry-inspired pair programming communication guidelines with undergraduate students. *Proceedings of the 46th Association for Computing Machinery (ACM) Technical Symposium on Computer Science Education (SIGCSE '15)*, 314–319. <https://doi.org/10.1145/2676723.2677241>
- Zhong, B., & Wang, Y. (2021). Effects of roles assignment and learning styles on pair learning in robotics education. *International Journal of Technology and Design Education*, 31(1), 41–59. <https://doi.org/10.1007/s10798-019-09536-2>
- Zhong, B., Wang, Q., & Chen, J. (2016). The impact of social factors on pair programming in a primary school. *Computers in Human Behavior*, 64, 423-431. <https://doi.org/10.1016/j.chb.2016.07.017>
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2017). Investigating the period of switching roles in pair programming in a primary school. *Journal of Educational Technology & Society*, 20(3), 220-233. <https://www.jstor.org/stable/10.2307/26196132>

Anhangsverzeichnis

Anhang 1: Einwilligungserklärung	106
Anhang 2: Beobachtungsauftrag.....	107
Anhang 3: Codebuch.....	110
Anhang 4: Prüfung Voraussetzungen Intraklassenkorrelation	113
Anhang 4.1: Prüfung der Normalverteilung der abhängigen Variablen der Studie mit einem Shapiro-Wilk-Test	113
Anhang 4.2: Prüfung der Varianzhomogenität zwischen den Stufen der unabhängigen Variablen mit einem Levene-Test basierend auf dem Median	113
Anhang 4.3: Prüfung, ob eine Interaktion zwischen Raterinnen und Ratern und bewerteten Personen vorliegt mit einem Tukey-Additivitätstest	113
Anhang 4.4: Prüfung der Trennschärfen der Rater.....	113
Anhang 5 Beschreibung möglicher Störvariablen.....	114
Anhang 5.1: Kategorien Altersunterschied.....	114
Anhang 5.2: Kategorien Leistungsunterschied	114
Anhang 5.3: Kategorien Unterschied Programmierkenntnisse.....	114
Anhang 5.4: Kategorien Beziehung Lernende	114
Anhang 6 Prüfung auf Normalverteilung.....	115
Anhang 6.1: Abhängige Variablen ohne Faktor	115
Anhang 6.2: Abhängige Variablen mit Faktor Geschlechterzusammensetzung	115
Anhang 7 Zugang zum Datensatz.....	117

Anhang 1: Einwilligungserklärung

**Pädagogische
Hochschule Luzern**
Institut für Fachdidaktik
Natur, Mensch,
Gesellschaft (IF NMG)
Janine Küng
Sentimatt 1
CH-6003 Luzern

Einwilligungserklärung

Liebe Eltern

Im Herbst 2020 wird in der Lernwerkstatt der PH Luzern die Lernumgebung *Mit Roberta® die Stadt der Zukunft entdecken* durchgeführt. Ihr Kind wird mit der ganzen Klasse an diesem halbtägigen Workshop kostenlos teilnehmen und in die Welt der Robotik eintauchen können. Um nähere Informationen zur Lernwirksamkeit dieses Angebots zu erhalten, werden wir eine schriftliche Befragung sowie eine Videostudie durchführen.

Auch Ihr Kind kann auf den Aufnahmen erkennbar sein. Deshalb brauchen wir Ihre Einwilligung. Die Aufnahmen dienen ausschliesslich der Forschung und werden nicht an Dritte weitergegeben. Die persönlichen Daten der gefilmten Personen werden nicht bekannt gegeben. Für die Untersuchung und Darstellung der Ergebnisse werden die Daten anonymisiert.

Mit Ihrer Unterschrift erklären Sie sich damit einverstanden, dass Ihr Kind aufgenommen werden darf, und dass die Aufnahmen für Forschungszwecke eingesetzt werden dürfen. Gerne möchten wir Sie bitten, die Einwilligungserklärung zu unterschreiben und **der Klassenlehrperson** Ihres Kindes zurückzugeben.

Für Fragen stehen wir Ihnen gerne zur Verfügung.

Herzlichen Dank für Ihre Bereitschaft und freundliche Grösse

Janine Küng, Wissenschaftliche Mitarbeiterin

Kontaktadresse für inhaltliche Rückfragen: Janine Küng, PH Luzern, 041 203 00 39,
janine.kueng@phlu.ch

Einwilligungserklärung Videoaufnahmen zu Forschungszwecken

Name Ihres Kindes:

- Wir sind damit einverstanden, dass unser Kind während des Besuchs der Lernwerkstatt aufgenommen wird, und dass die Aufnahmen für Forschungszwecke verwendet werden.
- Wir sind damit nicht einverstanden und wünschen, dass keine Aufnahmen von unserem Kind gemacht und verwendet werden.

Ort, Datum:

Unterschrift Eltern (Erziehungsberechtigte):

Anhang 2: Beobachtungsauftrag

Für die Videostudie werden bis zur Pause vier Zweiergruppen Ihrer Klasse gefilmt. Die entsprechenden Arbeitsstationen sind mit Nummern markiert. Achten Sie bitte darauf, dass an diesen Standorten keine Lernenden sitzen, die nicht gefilmt werden dürfen. In der Studie geht es um die Zusammenarbeit der Lernenden. Um die Beobachtungen nicht zu verfälschen, sagen Sie bei Nachfragen der Schülerinnen und Schüler bitte nur, dass wir den Umgang mit den von uns entwickelten Aufgabenstellungen beobachten.

Zusätzlich zu den Video- und Audioaufnahmen werden weitere Angaben benötigt. Deshalb bitte ich Sie, diesen Beobachtungsauftrag bis zur Pause auszufüllen, und dann einer der Betreuungspersonen der Lernwerkstatt zurückzugeben. Falls Sie mehr Platz benötigen, können Sie die Rückseiten beschreiben. Kennzeichnen Sie dann bitte klar, zu welcher Gruppe die Notizen gehören.

Beobachten Sie nur, greifen Sie bei den vier Kameragruppen bis zur Pause möglichst nicht ein.

Herzlichen Dank für Ihre Unterstützung
Janine Küng

Besuch der Lernwerkstatt am: Datum: _____ Vormittag Nachmittag

Klasse: Begabtenförderung 5./6. Klasse Oberstufe

Kamerastandort 1

Lernende/r 1	Lernende/r 2
<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich	<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich
Geburtsdatum:	Geburtsdatum:
Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):	Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):
Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark	Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark
Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein	Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein
Beziehung der Lernenden zueinander <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> negativ eher negativ neutral eher positiv positiv	
Fällt Ihnen bei der <u>Zusammenarbeit</u> dieser Gruppen etwas auf (positiv, negativ und neutral)? Können Sie ein allenfalls aussergewöhnliches Verhalten begründen?	

Kamerastandort 2

Lernende/r 1	Lernende/r 2
<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich	<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich
Geburtsdatum:	Geburtsdatum:
Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):	Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):
Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark	Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark
Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein	Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein
Beziehung der Lernenden zueinander <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> negativ eher negativ neutral eher positiv positiv	
Fällt Ihnen bei der <u>Zusammenarbeit</u> dieser Gruppen etwas auf (positiv, negativ und neutral)? Können Sie ein allenfalls aussergewöhnliches Verhalten begründen? 	

Kamerastandort 3

Lernende/r 1	Lernende/r 2
<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich	<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich
Geburtsdatum:	Geburtsdatum:
Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):	Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):
Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark	Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark
Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein	Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein
Beziehung der Lernenden zueinander <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> negativ eher negativ neutral eher positiv positiv	

Fällt Ihnen bei der Zusammenarbeit dieser Gruppen etwas auf (positiv, negativ und neutral)?
Können Sie ein allenfalls aussergewöhnliches Verhalten begründen?

Kamerastandort 4

Lernende/r 1	Lernende/r 2
<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich	<input type="checkbox"/> Weiblich <input type="checkbox"/> Männlich
Geburtsdatum:	Geburtsdatum:
Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):	Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe):
Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark	Allgemeines schulisches Leistungsvermögen <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> leistungsschwach leistungsstark
Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein	Programmiervorkenntnisse <input type="checkbox"/> Ja <input type="checkbox"/> Nein
Beziehung der Lernenden zueinander <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> negativ eher negativ neutral eher positiv positiv	
Fällt Ihnen bei der <u>Zusammenarbeit</u> dieser Gruppen etwas auf (positiv, negativ und neutral)? Können Sie ein allenfalls aussergewöhnliches Verhalten begründen?	

Anhang 3: Codebuch

Vorbereitungen

- Festplatte mit Videoaufnahmen an den Computer anschliessen
 - o Zu codierende Videoaufnahme herausuchen und in VLC öffnen (Achtung, pro Gruppe können es bis zu drei Videodateien sein!)
 - VLC Tastenkürzel festlegen (Werkzeuge-Einstellungen-Tastenkürzel).
Vorschlag:
 - Abspielen/Pause: Space
 - Schneller: f
 - Langsamer: s
 - Normale Geschwindigkeit: d
 - Sprünge in der Zeitleiste um 10 Sekunden: Pfeiltasten links und rechts
- Ordner mit Beobachtungsaufträgen hervorheben
 - o Passenden Beobachtungsauftrag zur Aufnahme herausuchen
- Excel-Datei öffnen
 - o Zeile 1 fixieren: Ansicht - Fenster einfrieren (Fenster fixieren) - Oberste Zeile fixieren
- Smartphone mit Multi-Timer-App: zwei Timer öffnen, in „L1Eingriffe“ und „L2Eingriffe“ umbenennen und auf null zurücksetzen. Alternative: zwei beschriftete Stoppuhren
- Notizblatt für Strichliste (für die Anzahl Eingriffe und Hilfestellungen) bereitlegen

Ablauf Codierung

Fehlende und unklare Werte: Fehlt ein Wert, oder ist er unklar, wird in diesem Feld nichts eingetragen. Es braucht aber einen Kommentar, weshalb das Feld nicht ausgefüllt werden konnte: Rechtsklick auf das leere Feld, Neuer Kommentar. Beispiel: „Lehrperson hat das Kreuz zwischen Ja und Nein gesetzt“. Ausnahme: Nicht erfolgte Platztäusche werden ohne Kommentar leer gelassen.

1. Aufnahmebezeichnung des geöffneten Videos notieren

Datum (tt.mm.), Vor-/Nachmittag (VO/NA), Kamerastandort (1-4)	Bsp.: 01.09. NA 1
---	-------------------

2. Angaben vom Beobachtungsauftrag in Excel eintragen (Kamerastandort beachten!)

- Klassenstufe

Begabtenförderung	0
5./6. Klasse	1
Oberstufe	2

- Zuerst Daten von L1, dann L2 übertragen:

- Geschlecht

Weiblich	0
Männlich	1

- Geburtsdatum

tt.mm.jjjj	Bsp.: 02.03.2008
------------	------------------

- Visuelles Erkennungszeichen für die Videoauswertung (z.B. T-Shirt-Farbe)

Von Lehrperson notiertes Erkennungszeichen	Bsp.: schwarze Haare
--	----------------------

- Allgemeines schulisches Leistungsvermögen

leistungsschwach	0
Eher leistungsschwach	1
Eher leistungsstark	2
leistungsstark	3

- Programmiervorkenntnisse

Ja	0
Nein	1

- Beziehung der Lernenden zueinander

negativ	0
eher negativ	1
neutral	2
eher positiv	3
positiv	4

- Notizen Lehrperson zur Zusammenarbeit

Von LP notierte Notizen wortwörtlich übernehmen. Unlesbare Stellen folgendermassen kennzeichnen: (...)	Bsp.: L1 arbeitet konzentriert...
--	-----------------------------------

3. Video im VLC-Player öffnen

- In der Zeitachse mit der Maus durchspulen, um die zu codierenden Zeitpunkte grob zu finden. Dann mit der Leertaste Video stoppen, mit den Pfeiltasten kleine Sprünge nach vorne und zurück machen, um den Zeitpunkt genau zu finden. Zeitpunkt notieren und dann mit der Leertaste das Video zur Kontrolle an diesem Zeitpunkt starten.

- Start der selbstständigen Arbeitsphase¹

Stunden:Minuten:Sekunden (hh:mm:ss)	Bsp.: 00:42:30
-------------------------------------	----------------

- Rolle von L1 beim Start²

Driver	0
Navigator	1

- Rolle von L2 beim Start³

Driver	0
Navigator	1

- Ende der selbstständigen Arbeitsphase⁴

Stunden:Minuten:Sekunden (hh:mm:ss) im gerade geöffneten Videoteil <u>PLUS</u> die Dauer der vorherigen Videoteile	Bsp.: 00:19:00+00:53:00=00:72:00 → wird automatisch zu 01:12:00 umgewandelt
--	---

- Letzte bearbeitete Aufgabe⁵ notieren

Aufgabe 1: Roberta fährt vorwärts	1
Aufgabe 2: Roberta pausiert	2
Aufgabe 3: Roberta fährt Links- und Rechtskurven	3
Aufgabe 4: Roberta erkennt eine schwarze Linie	4
Übersichtskarte: Das autonome Fahrzeug	5
Aufgabenset ganz abgeschlossen	6

¹ Start der selbstständigen Arbeitsphase= Nach der Einführungsaufgabe, sobald die Kursleitung viel Spass beim Programmieren wünscht.

² Rolle von L1 beim Start= Sitzt L1 beim Start der selbstständigen Arbeitsphase rechts am Tisch am Computer (Driver) oder links am Tisch vor dem Roboter (Navigator)?

³ Rolle von L2 beim Start= Sitzt L2 beim Start der selbstständigen Arbeitsphase rechts am Tisch am Computer (Driver) oder links am Tisch vor dem Roboter (Navigator)?

⁴ Ende der Arbeitsphase= Bevor die Kursleitung für die Pause unterbricht, oder die Gruppe das autonome Fahrzeug fertig gelöst hat. Nur bis dann wird codiert.

⁵ Letzte bearbeitete Aufgabe= An welcher Aufgabe arbeitet das Team zuletzt vor der Pause? Es spielt keine Rolle, wie weit die Lernenden diese Aufgabenstellung gelöst haben. Ausser das Aufgabenset wurde komplett gelöst: Ein neues Aufgabenset wurde begonnen und/oder an der Kommunikation der Lernenden ist klar erkennbar, dass die Aufgabe auf der Übersichtskarte fertig gelöst wurde.

- Zeitpunkte aller PlatztÄusche⁶

Zeitpunkt jedes Platztausches (Reihenfolge beachten) notieren: Stunden:Minuten:Sekunden (hh:mm:ss) im gerade geöffnerten Videoteil <u>PLUS</u> die Dauer der vorherigen Videoteile	Bsp.: 00:19:00+00:32:00= 00:51:00
--	---

- Geschwindigkeit ändern (Wiedergabe – Geschwindigkeit - Schneller (5x), noch besser mit Tastenkürzeln)

- Anzahl und Dauer der Eingriffe als Navigator⁷, einzeln für L1 und für L2

Anzahl Eingriffe: Strichliste auf Notizblatt machen, Summe ohne Kommazahlen eingeben	Bsp.: 4
Dauer der Eingriffe: Zwei Timer auf Smartphone bereithalten. Bei jedem Eingriff entsprechenden Timer starten. Gesamtzeit aller Eingriffe von L1 und L2 notieren: Stunden:Minuten:Sekunden (hh:mm:ss)	Bsp.: 00:01:23

- Anzahl angeforderte und erhaltene Hilfestellungen⁸ durch Betreuungspersonen oder andere Lernende

Nicht gezählt wird:

- Hilfe bei technischen Problemen, welche die Schülerinnen und Schüler nicht selbst ausgelöst haben, beispielsweise Verbindungsprobleme des Roboterkabels
- Nicht angeforderte Hilfe
- Angeforderte, aber nicht erhaltene Hilfe

Anzahl angeforderte und erhaltene Hilfestellungen: Strichliste auf Notizblatt machen, Summe ohne Kommazahlen eingeben	Bsp.: 4
---	---------

- Falls etwas bei der Codierung aufgefallen ist, kann dies unter „Notizen zur Codierung“ eingegeben werden.

⁶ PlatztÄusche= Der manchmal von der Kursleitung initiierte erste Platztausch nach der Einführungsaufgabe zählt nicht. Falsches Hinsetzen, welches gleich zurückgetauscht wird, zählt nicht. Der Zeitpunkt eines Platztausches ist genau dann, wenn sich mindestens ein/e Lernende/r auf den anderen Stuhl setzt. Es werden acht Variablen für acht mögliche PlatztÄusche zur Verfügung gestellt (Platztausch1, Platztausch2, etc.). Die stattgefundenen PlatztÄusche werden der Reihe nach eingetragen (Zeitpunkt des ersten Platztausches bei der Variablen „Platztausch1“, Zeitpunkt des zweiten Platztausches bei der Variablen „Platztausch2“, etc.). Gibt es weniger als 8 PlatztÄusche, beispielsweise nur 3, bleiben die restlichen Variablen leer. Falls mehr als acht PlatztÄusche stattfinden: Kommentar bei „Notizen zur Codierung“ mit Zeitpunkten der zusätzlichen PlatztÄusche.

⁷ Eingriff als Navigator= Wenn ein/e Lernende/r in der Rolle des Navigators ist (am Tisch links vor dem Roboter sitzt), und trotzdem Computermaus oder -tastatur berührt. Eine Berührung des Computerbildschirmes gilt nicht als Eingriff. Der Start des Eingriffs ist die Berührung von Computermaus oder -tastatur. Dann wird der Timer gestartet. Der Timer wird gestoppt, sobald die/der Lernende die Hände klar von Computermaus oder -tastatur zurückzieht. Erfolgt gleich danach noch eine Berührung, zählt dies als neuer Eingriff.

⁸ Angeforderte und erhaltene Hilfestellungen= Wenn das Team eine Hilfestellung aktiv anfordert (z.B. durch Handzeichen) und diese durch eine Betreuungsperson (Kursleitung, Assistenz, Lehrperson oder Begleitpersonen) oder andere Lernende erhält. Hilfe bei technischen Problemen gilt nicht als Hilfestellung (z.B. Verbindungsprobleme mit dem Roboter). Wenn andere Lernende oder eine Betreuungsperson ohne aktive Nachfrage des Teams hilft, wird dies ebenfalls nicht als Hilfestellung gezählt. Eine angeforderte, aber nicht erhaltene Hilfe wird ebenfalls nicht gezählt, beispielsweise wenn sich das Team mit Handzeichen meldet, aber niemand zu Hilfe kommt.

Anhang 4: Prüfung Voraussetzungen Intraklassenkorrelation

Anhang 4.1: Prüfung der Normalverteilung der abhängigen Variablen der Studie mit einem Shapiro-Wilk-Test

Abhängige Variable	Shapiro-Wilk		
	Statistik	df	Signifikanz
Letzte Aufgabe	.858	70	.000
Anzahl Platztäusche	.901	70	.000
L1 Anzahl Eingriffe	.690	70	.000
L2 Anzahl Eingriffe	.781	70	.000
L1 Dauer Eingriffe	.562	70	.000
L2 Dauer Eingriffe	.627	70	.000
Anzahl Hilfestellungen	.880	70	.000

Anhang 4.2: Prüfung der Varianzhomogenität zwischen den Stufen der unabhängigen Variablen mit einem Levene-Test basierend auf dem Median

Abhängige Variable	Levene-Statistik	df1	df2	Sig.
Letzte Aufgabe	.298	1	72	.587
Anzahl Platztäusche	.458	1	74	.501
L1 Anzahl Eingriffe	.082	1	74	.776
L2 Anzahl Eingriffe	.529	1	74	.469
L1 Dauer Eingriffe	.037	1	74	.849
L2 Dauer Eingriffe	.294	1	74	.589
Anzahl Hilfestellungen	.207	1	78	.651

Anhang 4.3: Prüfung, ob eine Interaktion zwischen Raterinnen und Ratern und bewerteten Personen vorliegt mit einem Tukey-Additivitätstest

Abhängige Variable	Friedmans Chi-Quadrat	Sig.
Anzahl Hilfestellungen	7.140	.011
L1 Anzahl Eingriffe	2.662	.111
L1 Dauer Eingriffe	128.883	.000
L2 Anzahl Eingriffe	5.368	.026
L2 Dauer Eingriffe	2.100	.155
Letzte Aufgabe	1.155	.289
Anzahl Platztäusche	.514	.478

Anhang 4.4: Prüfung der Trennschärfen der Rater

Abhängige Variable	Korrigierte Item-Skala-Korrelation	
	Rater 1 (PB)	Rater 2
Anzahl Hilfestellungen	.902	.902
L1 Anzahl Eingriffe	.987	.987
L1 Dauer Eingriffe	.999	.999
L2 Anzahl Eingriffe	.909	.909
L2 Dauer Eingriffe	.775	.775
Letzte Aufgabe	.920	.920
Anzahl Platztäusche	.930	.930

Anhang 5 Beschreibung möglicher Störvariablen

Anhang 5.1: Kategorien Altersunterschied

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	¼ Jahr oder weniger	54	26.6	28.7	28.7
	zwischen ¼ Jahr und ½ Jahr	48	23.6	25.5	54.3
	zwischen ½ Jahr und 1 Jahr	60	29.6	31.9	86.2
	mehr als 1 Jahr	26	12.8	13.8	100.0
	Gesamt	188	92.6	100.0	
Fehlend	System	15	7.4		
Gesamt		203	100.0		

Anhang 5.2: Kategorien Leistungsunterschied

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	gleiches allgemeines schulisches Leistungsniveau	101	49.8	53.7	53.7
	ähnliches allgemeines schulisches Leistungsniveau	72	35.5	38.3	92.0
	unterschiedliches allgemeines schulisches Leistungsniveau	13	6.4	6.9	98.9
	stark unterschiedliches allgemeines schulisches Leistungsniveau	2	1.0	1.1	100.0
	Gesamt	188	92.6	100.0	
Fehlend	System	15	7.4		
Gesamt		203	100.0		

Anhang 5.3: Kategorien Unterschied Programmierkenntnisse

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	beide keine Vorkenntnisse	81	39.9	41.5	41.5
	eine Person Vorkenntnisse	39	19.2	20.0	61.5
	beide Personen Vorkenntnisse	75	36.9	38.5	100.0
	Gesamt	195	96.1	100.0	
Fehlend	System	8	3.9		
Gesamt		203	100.0		

Anhang 5.4: Kategorien Beziehung Lernende

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	eher negativ	4	2.0	2.0	2.0
	neutral	78	38.4	39.4	41.4
	eher positiv	50	24.6	25.3	66.7
	positiv	66	32.5	33.3	100.0
	Gesamt	198	97.5	100.0	
Fehlend	System	5	2.5		
Gesamt		203	100.0		

Anhang 6 Prüfung auf Normalverteilung

Anhang 6.1: Abhängige Variablen ohne Faktor

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistik	df	Signifikanz	Statistik	df	Signifikanz
Lösegeschwindigkeit	.133	84	.001	.876	84	.000
Hilfestellungen pro Stunde	.146	84	.000	.894	84	.000
L1 Anzahl Eingriffe	.213	84	.000	.733	84	.000
L2 Anzahl Eingriffe	.225	84	.000	.807	84	.000
L1 Durchschnittliche Dauer Eingriff	.193	84	.000	.710	84	.000
L2 Durchschnittliche Dauer Eingriff	.252	84	.000	.528	84	.000
Platztausch Unterschied zu Sollwert	.312	84	.000	.844	84	.000
Beide Anzahl Eingriffe	.158	84	.000	.844	84	.000
Beide Durchschnittliche Dauer Eingriff	.184	84	.000	.700	84	.000

a. Signifikanzkorrektur nach Lilliefors

Anhang 6.2: Abhängige Variablen mit Faktor Geschlechterzusammensetzung

	Geschlechterzusammensetzung	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Statistik	df	Sig.	Statistik	df	Sig.
Lösegeschwindigkeit	geschlechtshomogene weibliche Gruppe	.133	28	.200*	.933	28	.074
	geschlechtsheterogene Gruppe	.087	30	.200*	.976	30	.720
	geschlechtshomogene männliche Gruppe	.198	26	.010	.758	26	.000
Hilfestellungen pro Stunde	geschlechtshomogene weibliche Gruppe	.123	28	.200*	.927	28	.053
	geschlechtsheterogene Gruppe	.137	30	.156	.861	30	.001
	geschlechtshomogene männliche Gruppe	.214	26	.003	.879	26	.006
L1 Anzahl Eingriffe	geschlechtshomogene weibliche Gruppe	.266	28	.000	.616	28	.000
	geschlechtsheterogene Gruppe	.180	30	.015	.857	30	.001
	geschlechtshomogene männliche Gruppe	.233	26	.001	.760	26	.000
L2 Anzahl Eingriffe	geschlechtshomogene weibliche Gruppe	.214	28	.002	.767	28	.000
	geschlechtsheterogene Gruppe	.219	30	.001	.824	30	.000
	geschlechtshomogene männliche Gruppe	.281	26	.000	.823	26	.000
L1 Durchschnittliche Dauer Eingriff	geschlechtshomogene weibliche Gruppe	.196	28	.007	.865	28	.002
	geschlechtsheterogene Gruppe	.232	30	.000	.591	30	.000
	geschlechtshomogene männliche Gruppe	.181	26	.029	.876	26	.005
L2 Durchschnittliche Dauer Eingriff	geschlechtshomogene weibliche Gruppe	.147	28	.124	.928	28	.054

tlische Dauer Eingriff	geschlechtsheterogene Gruppe	.300	30	.000	.471	30	.000
	geschlechtshomogene männliche Gruppe	.234	26	.001	.655	26	.000
Platztausch Unterschied zu Sollwert	geschlechtshomogene weibliche Gruppe	.282	28	.000	.890	28	.007
	geschlechtsheterogene Gruppe	.329	30	.000	.781	30	.000
	geschlechtshomogene männliche Gruppe	.322	26	.000	.865	26	.003
Beide Anzahl Eingriffe	geschlechtshomogene weibliche Gruppe	.269	28	.000	.670	28	.000
	geschlechtsheterogene Gruppe	.114	30	.200*	.933	30	.059
	geschlechtshomogene männliche Gruppe	.182	26	.027	.901	26	.017
Beide Durchschnittliche Dauer Eingriff	geschlechtshomogene weibliche Gruppe	.123	28	.200*	.959	28	.324
	geschlechtsheterogene Gruppe	.265	30	.000	.666	30	.000
	geschlechtshomogene männliche Gruppe	.198	26	.010	.801	26	.000

*. Dies ist eine untere Grenze der echten Signifikanz.

a. Signifikanzkorrektur nach Lilliefors

Anhang 7 Zugang zum Datensatz

Der SPSS-Datensatz, mit welchem die Datenauswertung durchgeführt wurde, ist anonymisiert hinter einem Passwort auf *SWITCHdrive* abgelegt. Der Zugang ist über den untenstehenden Link oder QR-Code möglich. Das Passwort kann mit Angabe des geplanten Verwendungszweckes bei der Autorin dieser Masterarbeit per Mail angefordert werden (janine.kueng@phlu.ch).

<https://drive.switch.ch/index.php/s/3KJ0KvUYRVGkaOM>

