Freie Universität Berlin

Fallstudie über die Leistung und Leistungsfähigkeit bei Paarprogrammierung in der Lehre

Bachelorarbeit von Jennifer Möwert

29. Mai 2012

Gutachter Prof. Dr. Lutz Prechelt
Zweitgutachter Prof. Dr.Carsten Schulte
Betreuerin: Edna Rosen

Danksagung

Ich möchte mich bei Herrn Prof. Dr. Prechelt für die Vergabe des Themas bedanken. Mit seiner Hilfe bekam ich sehr schnell eine engagierte Betreuerin durch deren Ideen ich zu diesem Thema gekommen bin.
Ganz besonders möchte ich mich daher bei meiner Betreuerin Edna Rosen bedanken, die mir mit Rat und Tat zur Seite stand, durch konstruktive Kritik die Qualität meiner Ausarbeitungen stark verbesserte und mir interessante Anregungen zu diesem Thema gab.

Auch möchte ich mich bei Christophe Sokol bedanken, sowie dem Mitarbeiter der AG Software Engineering Stephan Salinger, als auch Klaus Zöller und Franz Zieris die mich freundlich aufnahmen und mir Tipps gaben.

Mein Kommilitone Stephan Lau gab mir einige Tipps zur Software Saros, dafür möchte ich mich bei ihm bedanken. Auch Marco Kunis unterstützte mich indem er meine entworfenen Fragebögen testete.

Weiterhin bedanke ich mich noch bei Frau Prof. Esponda für die Erlaubnis die Untersuchung in ihrem Kurs durchführen zu können und auch ihren Tutoren, die immer versuchten mir wichtige Informationen bereit zu stellen.
Einen besonderen Dank geht an die Teilnehmer des untersuchten Blockkurses "Programmieren in Java", die mir sehr interessante Ergebnisse lieferten.

Auch den freiwilligen Teilnehmern aus dem Kurs "Softwareprozesse" möchte ich danken, sowie Julia Schenk, die mich auf die Möglichkeit freiwillige Teilnehmer aus diesem Kurs bei der Ausübung von Paarprogrammierung zu untersuchen, aufmerksam machte.

Außerdem möchte ich mich bei meiner Familie und meinem Freund Mark Birker für ihre Unterstützung und Liebe bedanken. Des weiteren danke ich Michael Lohmann der mir mit Rat und Tat zur Seite stand.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben. Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Schönefeld, den 29. Mai 2012

Jennifer Möwert

Zusammenfassung

In dieser Arbeit geht es um die Leistung und Leistungsfähigkeit bei Paarprogrammierung, welche sich aus der Zeit, den Punkten und der Motivation zusammensetzt. Diese Art des Programmierens gehört zu den agilen Methoden der Softwareentwicklung. Die Besonderheit liegt darin, dass sich zwei Programmierer Maus und Tastatur teilen.
Es wurden hauptsächlich Studenten in der Blockveranstaltung "Programmieren in Java" an der Freien Universität Berlin beobachtet und analysiert. Es handelt sich hier um eine Fallstudie mit den Charakterzügen eines Experiments, da sowohl qualitative Beobachtungen, als auch quantitative Messungen gemacht wurden. Der Fokus wurde bei dieser Arbeit allerdings auf den qualitativen Teil gelegt. Zu den verwendeten Forschungsmethoden zählen Fragebögen, Interviews und Beobachtungen die mit Hilfe der Studenten des Blockkurses gemacht wurden.
Die Blockveranstaltung fand zwei Wochen lang statt und wurde größtenteils von Mathematik-Studenten besucht. Nach der Vorlesung am Vormittag bekamen die Studenten Programmieraufgaben die sie im Paar oder einzeln lösen sollten. Dazu versammelten sich alle in den Computerräumen des Fachbereichs Informatik an der Freien Universität Berlin und programmierten dort etwa vier Stunden am Tag, wobei ihnen ein Tutor zur Seite stand, dem sie jederzeit Fragen stellen konnten. Während dieser Zeit beobachtete ich die Studenten und führte in der zweiten Woche auch Interviews. Nach jedem Tag bekamen die Studenten außerdem einen Fragebogen, wo sie Auskunft über den Verlauf des Programmierens geben konnten.

Inhaltsverzeichnis

1 Einleitung **8**
1.1 Paarprogrammierung . . . 9
1.2 Verteilte Paarprogrammierung . . . 9
1.3 Vor- und Nachteile bei der Paarprogrammierung . . . 9
1.4 Saros . . . 10

2 Planungsphase **11**
2.1 Pilot Test . . . 12
2.1.1 Clean Room . . . 12
2.2 Fallstudie im Blockkurs . . . 12
2.2.1 Raymond-Lister-Test . . . 13
2.3 Gruppenzusammenstellung . . . 13

3 Forschungsmethoden **14**
3.1 Forschungsstrategien . . . 14
3.1.1 Vorüberlegung . . . 14
3.2 Datenerhebungsmethoden . . . 14
3.2.1 Fragebögen . . . 14
3.2.2 Beobachtungen . . . 15
3.2.3 Interviews . . . 15
3.3 Quantitative Erhebung . . . 16
3.4 Qualitative Erhebung . . . 17

4 Vorbereitende Überlegungen **17**
4.1 Zeit . . . 17
4.1.1 Vorbereitungszeit . . . 17
4.1.2 Bearbeitungszeit . . . 18
4.2 Punkte . . . 18
4.2.1 Punkte der Übungsaufgaben . . . 18
4.2.2 Punkte des Tests . . . 18
4.3 Motivation . . . 18
4.3.1 Konzentriertes Arbeiten und Ablenkung . . . 18
4.3.2 Stimmung . . . 19
4.3.3 Rollenverteilung . . . 19
4.3.4 Hilfestellung . . . 19
4.3.5 Kommunikation der Einzelarbeiter . . . 19
4.3.6 Subjektive Wahrnehmung der Studenten . . . 20
4.4 Zeit, Punkte und Motivation in der systematischen Beobachtung . . . 20
4.4.1 Übersicht der geplanten Beobachtungen . . . 20
4.4.2 Beobachtungsannahme: Vorbereitungszeit, Einzelarbeiter auf Kommunikation und Hilfestellung untersuchen . . . 20
4.4.3 Beobachtungsannahme: Stimmung, konzentrierte Arbeitszeit . . . 21
4.4.4 Beobachtungsannahme: Rollenverteilung, Hilfestellung . . . 21

5 Ergebnisse des Pilot Tests **22**
5.1 Ergebnisse des Fragebogens . . . 22
5.2 Ergebnisse der Beobachtung von Gruppe 1 . . . 24
5.3 Ergebnisse der Beobachtung von Gruppe 2 . . . 25
5.4 Vergleich der ersten und zweiten Gruppe . . . 26
5.5 Erkenntnisse aus dem Pilot Test . . . 27
5.6 Probleme und Besonderheiten . . . 28

6 Ergebnisse der Fallstudie (Blockkurs) **29**
6.1 Analyse des Fragebogens . . . 29
6.1.1 Beteiligung der Studenten an den Fragebögen . . . 29
6.1.2 Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts . . . 30
6.1.3 Ressourcen die von den Studenten zur Problemlösung verwendet wurden . . . 35
6.1.4 Angaben in den Fragebögen zu Eigenschaften von Paarprogrammierung . . . 40
6.1.5 Angaben in den Fragebögen zu Eigenschaften vom einzeln arbeiten . . . 45
6.1.6 Probleme bei der Auswertung der Fragebögen . . . 47
6.2 Analyse des Tests . . . 47
6.2.1 Beteiligung an den Tests . . . 48
6.2.2 Anzahl der Punkte des Tests . . . 48
6.2.3 Probleme bei der Auswertung der Tests . . . 50
6.3 Analyse der Zeit für die Bearbeitung der Aufgaben . . . 50
6.3.1 Probleme bei dem Notieren der benötigten Zeit für die Aufgaben . . . 52
6.4 Phänomene . . . 52
6.4.1 Bildung von dreier Gruppen . . . 53
6.4.2 Paarprogrammierer werden zu Einzelarbeitern . . . 54
6.4.3 Ablauf von Rollenwechsel . . . 59
6.4.4 Konzentriertes Arbeiten . . . 64
6.4.5 Auffällige Stimmungen . . . 66

7 Bewertung der getroffenen vorbereitenden Überlegungen **68**
7.1 Vorbereitende Überlegungen zur Vorbereitungszeit . . . 68
7.2 Vorbereitende Überlegungen zur Kommunikation von Einzelarbeitern und Hilfestellung 69
7.3 Vorbereitende Überlegungen zur Stimmung und konzentrierten Arbeitszeit . . . 69
7.4 Vorbereitende Überlegungen zur Rollenverteilung und Hilfestellung . . . 70
7.5 Weitere Probleme der Fallstudie . . . 70

8 Fazit **71**

9 Ausblick **72**

Literatur **75**

10 Anhang **81**
10.1 Fragebogen zum Kurs “Softwareprozesse . . . 81
10.2 Beobachtungsvorlage für die Teilnehmer des Kurses “Softwareprozesse” . . . 83
10.3 Übersicht über die Beobachteten Gruppen des Blockkurses “Programmieren in Java” . 87

10.4 Ausführliche Beschreibung der Beobachten die ersten Tage im Kurs “Programmieren in Java” . . . 89
10.4.1 Erster Tag . . . 89
10.4.2 Zweiter Tag . . . 92
10.5 Fragebogen zur Gruppeneinteilung des Blockkurses “Programmieren in Java” . . . 96
10.6 Fragebogen für Einzelarbeiter des Blockkurses “Programmieren in Java” . . . 97
10.7 Fragebogen für Paarprogrammierer des Blockkurses “Programmieren in Java” . . . 100
10.8 Beobachtungsvorlage für den ersten Tag . . . 103
10.9 Beobachtungsvorlage für den zweiten Tag . . . 103
10.10 Fragebogen des letzten Tages für die Paarprogrammierer . . . 105
10.11 Fragebogen des letzten Tages für die Einzelarbeiter . . . 108
10.12 Übersetzung des Raymond-Lister-Tests . . . 110
10.13 Test des Blockkurses “Programmieren in Java” . . . 119

1 Einleitung

In diesem Abschnitt werde ich grundlegende Begrifflichkeiten erläutern und eine kleine Einführung in die Thematik geben.

Zu Beginn stand das Thema meiner Bachelorarbeit noch nicht ganz fest. Durch meine Betreuerin Edna Rosen bekam ich einen Anstoß für ein weiteres Thema, welches ich mir mit ihrer Hilfe erarbeitete und Stück für Stück immer mehr eingrenzte. Die Planungsphase der Bachelorarbeit nahm daher einen großen Teil ein. Als feststand, dass ich Paarprogrammierung in der Lehre untersuchen wollte, legte ich mir eine Strategie zurecht, wie ich entsprechende Probanden für meine Untersuchung gewinnen konnte. Da wir uns mittlerweile mitten im Semester befanden, war es schwer die Veranstaltungen zu untersuchen, jedoch nicht unmöglich. Der Blockkurs "Programmieren in Java" bot die besten Voraussetzung für die Untersuchung, da ich dort die Studenten über einen Zeitraum von zwei Wochen durchgängig beobachten konnte und so deren Entwicklung erkennen konnte. Sehr praktisch war auch, dass die Studenten zusammen an der Universität arbeiteten. So gab es keine Terminschwierigkeiten. Ich holte mir das Einverständnis der Professorin ein die Studenten ihres Kurses untersuchen zu dürfen und besprach mit ihr alle weiteren Einzelheiten. Die Professorin schlug vor einen Test schreiben zu lassen, welcher zu Beginn des Kurses und gegen Ende des Kurses geschrieben werden sollte. Der erste und zweite Test bestand aus den gleichen Fragen, um einen Vergleich machen zu können. Die erste Idee war den Raymond-Lister-Test zu verwenden (siehe Übersetzung des Raymond-Lister-Tests im Anhang 10.12) [11] . Jedoch stellte sich heraus, dass das Konzept der Objektorientierung, welches Hauptbestandteil in dem Kurs war, nicht in dem Test abgefragt wurde. So erstellte die Professorin einen Test abgestimmt auf die Vorlesung, welcher nicht in die Benotung der Studenten mit einging und nur für Studienzwecke gedacht war.
Da ich mir zu Anfang nicht sicher sein konnte, ob die Professorin zusagen würde, überlegte ich mir auch bei einer Absage zu fragen, ob ich freiwillige Probanden in dem Kurs suchen dürfte. Außerdem wollte ich mich mit den Professoren der im Semester laufenden Veranstaltungen "Algorithmen und Programmieren III" (ALP III) und "Algorithmen und Programmieren V" (ALP V) zusammensetzen und diese bitten eine Aufgabe in eines der aktuellen Übungsblätter aufzunehmen. Diese sollte die Studenten dazu bewegen eine der Aufgaben mit Hilfe von Paarprogrammierung zu lösen, sodass ich sie dabei beobachten könnte. Ich hatte mir eigentlich vorgenommen einige Konzepte der Paarprogrammierung, wie beispielsweise den Wechsel zwischen der Driver-Rolle und der Observer-Rolle, genauer zu untersuchen. Es zeigte sich jedoch, dass diese Konzepte sich durch individuelle soziale Zusammenhänge weitaus komplexer gestalten als erwartet. Die Aufgabe hätten die Studenten in der Universität oder mit Hilfe von Saros lösen können. Hätte ich dadurch nicht genug Material zusammen bekommen, wäre mir noch die Möglichkeit geblieben mich an anderen Universitäten nach vergleichbaren Veranstaltungen umzusehen. Da ich mit der Blockveranstaltung "Programmieren in Java" sehr viele Beobachtungen zusammentragen konnte, konzentrierte ich mich fast ausschließlich auf diese Veranstaltung und ließ die Anderen außen vor.
Mitten in der Vorbereitungsphase ergab sich die Gelegenheit meine Fragebögen und Beobachtungsweisen, sowie Ziele an Studenten des Kurses "Softwareprozesse" zu testen. Die Tutorin machte mich darauf aufmerksam, dass das nächste Übungsblatt mit der Anwendung von Paarprogrammierung zu tun hatte, sodass ich in dem Kurs nach Freiwilligen fragte und auch zwei Paare fand die ich beim Programmieren beobachten konnte.

1.1 Paarprogrammierung

Paarprogrammierung, auch kurz PP (Pair Programming), zählt zu den agilen Softwareentwicklungsmethoden. Dabei sitzen zwei Partner gemeinsam an einem Rechner und versuchen die gleiche Aufgabe zu implementieren, wobei einer, der sogenannte Driver, die Eingabe übernimmt und beispielsweise die Idee des Observers umsetzt. Es können auch gemeinsam entwickelte Ideen oder auch die eigene Idee umgesetzt werden. Der Andere, in dem Fall der Observer, hat die Aufgabe in der Zeit Vorschläge zu geben und auf Fehler aufmerksam machen. In der Regel ist angedacht, dass sich beide Partner regelmäßig und gleichmäßig beim Implementieren abwechseln, wobei es dort wieder verschiedene Vorgehensweisen geben kann.

1.2 Verteilte Paarprogrammierung

Verteilte Paarprogrammierung, auch kurz DPP (Distributed Pair Programming), zeichnet sich im Gegensatz zu lokaler Paarprogrammierung darin aus, dass nun nicht mehr beide Partner an einem Rechner physisch nebeneinander sitzen, sondern an unterschiedlichen Orten gemeinsam an einem Problem arbeiten.

1.3 Vor- und Nachteile bei der Paarprogrammierung

Häufig wird in der freien Wirtschaft oder auch an der Universität Paarprogrammierung nicht oft eingesetzt. In der Arbeit von Eike Starkmann kann man auch erkennen, dass es schwierig ist in Open Source Projekten Paarprogrammierung oder verteilte Paarprogrammierung richtig zu etablieren [36] . Dies liegt häufig daran, dass gerade aus Firmensicht der Kosten-Nutzen-Faktor zu gering ist. Man würde zwei Gehälter für ein Artefakt zahlen, welches einer allein auch erstellen könnte. Paarprogrammierung hat einige Vorteile, die allerdings noch nicht alle eindeutig durch Studien belegt werden konnten.
Ich habe persönlich die Erfahrung gemacht, dass in der Universität Paarprogrammierung meist nicht zum Einsatz kommt, da die Studenten die Aufgaben der Übungsblätter oft untereinander aufteilen, anstatt sich zu treffen, um Zeit zu sparen.
Dabei ist einer der Vorteile des Paarprogrammierens die Zeitersparnis, wie beispielsweise die quantitativen Studien von John T. Nosek und William et al. gezeigt haben [17, 15] . Durch den Gedankenaustausch mit dem Partner können neue Ideen entstehen, auf die man allein nicht so schnell gekommen wäre. Ein Nachteil ist sicherlich, dass die anfängliche Zusammenarbeit erst einmal länger dauern kann, da man sehr viel kommunizieren muss und jeden Gedankengang teilt und diskutiert. Das Paarprogrammierung auch länger dauern kann und eben nicht wie bei Nosek beschrieben effizienter ist, zeigt das Experiment von Jerzy Nawrocki und Adam Wojciechowski [16, 15] . Auch kann es vorkommen, dass Paare mit unterschiedlichen Stärken bzw. Wissensstand zusammenarbeiten und unter Umständen der Wissensstand angeglichen werden muss. Durch Paarprogrammierung lernt man meiner Meinung nach den vermittelten Stoff in der Universität schneller und auch intensiver, da jeder jede Aufgabe bearbeitet und sich bei Problemen schneller untereinander geholfen werden kann, an Stellen, bei denen einer allein vielleicht weitaus länger nachgedacht hätte. Mit intensiver meine ich hier, dass jeder sich gleich gut mit allen Aufgaben des Übungsblatt auskennt und nicht nur mit der Hälfte der Aufgaben. Außerdem kann man den Stoff durch das Diskutieren und Besprechen mit dem Partner besser verinnerlichen. Die Wahrscheinlichkeit durch die Klausur zu fallen würde somit sinken. Natürlich gibt es auch Fälle bei denen sich der Partner durch den Anderen gebremst fühlt, wenn dieser nicht auf dem gleichen Wissenstand ist oder eine andere Idee hat, die er umsetzen will.

Es kann allerdings auch von Vorteil sein einen etwas Stärkeren und einen Schwächeren in einem Team zu haben, da der Stärkere dem Schwächeren helfen kann und dieser so viel lernt. Des weiteren verstehen auch beide den Gedankengang des jeweils Anderen. Man kann einschreiten, wenn der Partner auf dem falschen Weg ist, oder etwas nicht verstanden wurde und kleine Kniffe im Programmieren die der Eine beherrscht, können an den Partner vermittelt werden. Es wird auch oft berichtet, dass man sauberer programmiert, da der Partner auch gleichzeitig eine Codedurchsicht durchführt und durch das Wissen, dass einer einen Blick darauf hat, versucht ordentlicher zu programmieren. Auch James E. Tomayko gewann laut Matthias M. Müller, Frank Padberg und Walter F. Tichy in seiner Studie die Erkenntnis, dass Paarprogrammierung zu fehlerfreieren Programmen führt [15] . Es wird in dem Paper auch auf einige Studien verwiesen, die zeige, dass Paarprogrammierung Spaß macht [15] . Paarprogrammierung kann sehr motivierend sein und so die Leistung steigern. Es kommt vor, dass Paarprogrammierer länger ohne Pausen arbeiten und längere Zeit konzentriert arbeiten,wie wir später noch sehen werden. Natürlich kann man auch als Nachteil anführen, dass sich die Partner gegenseitig ablenken und demotivieren könnten. Des weiteren kann ein Vorteil sein, dass die Stärken gegenseitig unterstützt und die Schwächen gemeinsam beseitigt werden können. Wobei man hier auch wieder anführen kann, dass Menschen sich oft zu den Aufgaben hingezogen fühlen, die sie gut können und das im Paar auch der Fall sein könnte. So bewältigt derjenige die Aufgaben, die er schon gut beherrscht und überlässt dem Partner die Aufgaben, die er weniger gut kann.
Gerade in der Universität wäre Paarprogrammierung meiner Meinung nach eine Hilfe, kontinuierliches Arbeiten zu lernen und dem Termindruck durch Terminabsprachen mit dem Partner und frühzeitigem Beginnen der Aufgaben zu minimieren. Auch wird durch das Diskutieren über die Aufgabe das gelernte vertieft und verinnerlicht.

1.4 Saros

Saros ist ein Plug-In für Eclipse, welches in der Arbeitsgruppe Software Engineering der Freien Universität Berlin zur verteilten Paarprogrammierung entwickelt wurde. Das Plug-In erlaubt Programmierpartnern gemeinsam an unterschiedlichen Orten zu entwickeln und zeigt mit sogenannten Highlights an, wer gerade welchen Codeabschnitt bearbeitet. Außerdem ist es möglich dem Host, also demjenigen der die Sitzung startet und Teilnehmer zu der Sitzung eingeladen hat, zu folgen. Diese Funktion zeigt dem Teilnehmer alle Aktionen an die der Host in dem Moment ausführt. Dies hat den Vorteil, dass man Codeabschnitte leichter erklären kann und die Teilnehmer wissen wovon gesprochen wird. Es gibt immer eine Driver und eine Observer-Rolle (siehe Abschnitt 1.1). Es können auch mehr als zwei Teilnehmer zusammen entwickeln und den Code gleichzeitig editieren. Es kann sogar gleichzeitig an mehreren Dateien gearbeitet werden. Auch können Reviews durchgeführt werden. Das Plug-In für Eclipse ist mit anderen Plug-Ins für verschiedene Programmiersprachen kompatibel und unterstützt die Programmiersprachen Java, C / C++, PHP, Ruby, Python, Javascript, Perl. Man verbindet sich über einen XMPP/Jabber Server miteinander und kann auch über einen in Eclipse integrierten Chat kommunizieren. Allerdings ist momentan eine externe Software wie z.B Skype oder auch Teamspeak3 zur sprachlichen Kommunikation zu empfehlen, da die Kommunikation noch integriert wird. Des weiteren gibt es die Möglichkeit des Screensharing, also seinen Bildschirm mit anderen Teilnehmern zu teilen. Der große Vorteil an Saros ist, dass man zusammenarbeiten kann ohne örtlich gebunden zu sein.

2 Planungsphase

Durch das neue Thema war die Planungsphase insgesamt sehr lang. Ich erarbeitete mir mit Hilfe meiner Betreuerin ein neues Thema und vertiefte dies immer weiter. In der Planungsphase versuchte ich alles vorzubereiten. Ich erstellte und überarbeitete die Fragebögen für die Studenten. Außerdem war ein Interview mit den Tutoren und Professoren angedacht, sodass ich jeweils einen Interviewleitfaden erstellte. Es stellte sich jedoch heraus, dass die Fragen viel zu Allgemein waren und kaum Aussagen in Bezug auf meine Arbeit hatten, dessen Ausrichtung sich inzwischen wieder etwas geändert hatte. Die Fragen des Fragebogens (siehe Anhang 10.6 und 10.7) passte ich mit Hilfe der Goal Question Metric (Abschnitt 3.1.1) an. Des weiteren erstellte ich einen kleinen Vortrag, den ich für Studenten halten wollte, indem ich auch auf die Vorteile von Paarprogrammierung einging, sowie erklärte, was Paarprogrammierung ist und wie man es im besten Fall anwendet. Dazu entwickelte ich eine kleine Anleitung zur Installation von Saros. Dies geschah, als ich noch nicht wusste wo genau ich meine Probanden herbekam und daher plante in die Tutorien der im Semester laufenden Veranstaltungen ALP III und ALP V zu gehen und Freiwillige zu finden. Ich wollte den Studenten auch Saros näher bringen und so zur Paarprogrammierung motivieren. Da es vielen Studenten schwer fällt, gemeinsame Termine zu finden um sich in der Universität zu treffen, wollte ich Saros als bequemes Mittel zur Paarprogrammierung von zu Hause vorstellen. So handelten die Fragen des ersten Fragebogens auch von Saros und fragten kaum Dinge zur Motivation ab. Dies änderte sich im Verlauf der Arbeit, nachdem ich bereits einen Testlauf mit ein paar Freiwilligen aus der Veranstaltung "Softwareprozesse" machte. Auch entwickelte ich vorbereitende Überlegungen (siehe Kapitel 4) erst für den Kurs "Programmieren in Java". Mit Hilfe dieser Überlegungen konnte ich die Dinge die ich beobachten wollte klarer definieren und konnte den Fokus bei meinen Beobachtungen auf diese Dinge richten.
Zudem bereitete ich einen Erstgesprächsleitfaden für das Gespräch mit Frau Prof. Esponda vor, da ich bei diesem Gespräch unbedingt eine Zusage bekommen und vorbereitet sein wollte. Lange Zeit war unklar, was genau ich bei den Studenten, die Paarprogrammierung anwenden, beobachten wollte. Ich schaute mir auch Videos an, die in der Arbeitsgemeinschaft Software Engineering von Paarprogrammierern aufgenommen wurden, um mir eine Vorstellung von dem zu machen, was ich beobachten könnte. Als feststand, dass ich den Kurs "Programmieren in Java" beobachten konnte, kam mir die Idee die Studenten in zwei Gruppen einzuteilen, um mehr Aussagen über Paarprogrammierung machen zu können und so erstellte ich einen Fragebogen zur Gruppeneinteilung. Mit dessen Hilfe war mir vorher die Anzahl der Studenten die Paarprogrammierung machen wollten bekannt und ich hatte einige Informationen über die Programmiererfahrungen der Studenten. Außerdem holte ich mir so die Einverständniserklärung der Studenten ein, sie beobachten und über sie schreiben zu dürfen.
Anfänglich war geplant im Blockkurs "Programmieren in Java" die Teilnehmer so aufzuteilen, dass etwa 50% Paarprogrammierung ausprobieren und 50% der Teilnehmer einzeln arbeiten. Da mir klar war, dass bei der Umfrage höchstwahrscheinlich ein ungleichmäßiges Ergebnis heraus kommen würde und nicht genau 50% der Teilnehmer Paarprogrammierung ausprobieren wollen, hatte ich vor die Studenten einzuteilen. Ich wollte versuchen die Wünsche der Studenten zu berücksichtigen, allerdings sollte die Einteilung in Einzelarbeiter und Paarprogrammierer mir obliegen. Bei meinem Vortrag in der Arbeitsgemeinschaft Software Engineering wurde ich darauf aufmerksam gemacht, dass dies Schwierigkeiten machen könnte und mir wurde der Vorschlag unterbreitet die Einteilung nicht gleichverteilt zu machen. Die Studenten sollten sich aussuchen, ob sie einzeln arbeiten oder Paarprogrammierung ausprobieren und es sollte egal sein, wie viele Studenten sich dafür entscheiden. Nach einigen Überlegungen entschied ich mich dies so umzusetzen.

2.1 Pilot Test

Es ergab sich für mich die Möglichkeit Freiwillige in dem Kurs "Softwareprozesse" zu suchen. Als ich zwei freiwillige Gruppen fand, die sich von mir bei der Paarprogrammierung beobachten lassen wollten, erstellte ich für diese Gruppe einen Fragebogen und Beobachtungsvorlagen. Die Studenten hatten in dem Übungsblatt die Aufgabe den Clean Room Ansatz für das Lösen der Aufgabe anzuwenden und sollten eine Art Mastermind, unter Verwendung der Programmiersprache Java, erstellen. Außerdem durften sie das Programm während der Implementierungsphase nicht ausführen, dies sollte erst im nachfolgenden Tutorium geschehen.

2.1.1 Clean Room

Ziel der Clean Room Technik ist es Software zu erstellen, die eine sehr hohe Qualität hat und mit wenig Fehlern behaftet ist. Fehler sollen nicht korrigiert werden, sondern von Vornherein ausgeschlossen werden. Innerhalb des Teams werden Codedurchsichten durchgeführt, um so eine hohe Qualität der Software zu ermöglichen. Beim Clean Room Ansatz werden nicht die Fehler gesucht, sondern die Zuverlässigkeit der Software gezeigt [24, 22, **?**] .

2.2 Fallstudie im Blockkurs

Die Blockveranstaltung "Programmieren in Java" fand im Winter Semester vom 20.02.2012 bis 02.03.2012 statt. Jeden Tag von 10:00 Uhr bis 13:00 Uhr Uhr gab es eine Vorlesung und anschließend von 14:00 Uhr bis 18:00 Uhr fanden die Übungen an den Rechnern des Instituts statt. Dort waren die etwa 60 Studenten auf vier Räume aufgeteilt, für den jeweils immer ein Tutor zuständig war. Während der zwei Wochen blieben die Studenten in denselben Räumen und hatten immer den gleichen Tutor, der sie betreute, sodass sich viele in der Zeit etwas besser kennenlernten. In der ersten Vorlesung bekamen die Studenten von mir eine kurze Einführung zur Paarprogrammierung. Dort wurde geklärt was Paarprogrammierung überhaupt ist und wie man es im besten Fall anwendet. Außerdem stellte ich mich und meine Arbeit kurz vor. Da es sich zum größten Teil um Mathematik-Studenten in diesem Kurs handelte, hielt ich die Einführung für sinnvoll und notwendig. Anschließend bekamen die Studenten von mir einen Fragebogen in dem sie angaben, ob sie einzeln arbeiten oder Paarprogrammierung ausprobieren wollen. Dort konnten sie auch den Namen des Wunschpartners angeben. Außerdem gaben die Studenten an wie viele Zeilen Code sie schon programmiert haben, sodass ich mir einen Eindruck der Fähigkeiten der Studenten machen konnte. So wusste ich, was für Fähigkeiten die Studenten in der Gruppe haben und konnte sehen wie sich dies auf die Zusammenarbeit auswirkt. Es wird häufig über die Gruppenzusammenstellung diskutiert und welche Konstellationen inwiefern sinnvoll sind. Es ist interessant zu sehen, dass Paare, bei denen ein Programmiererfahrener und ein Unerfahrener sind, gut zusammenarbeiten, während ein Paar derselben Konstellation nicht gut miteinander zurechtkommt. Der Erfahrene fühlt sich vom Unerfahrenen gebremst und erklärt ihm kaum etwas, wobei der Unerfahrene auch selten Fragen stellt und sehr zurückgezogen wirkt (siehe dazu unter anderem Abschnitt 6.4.2). Ich beobachtete die Studenten während sie in den PC-Räumen ihre Aufgaben lösten und am Ende des Tages gab es jeweils für die Einzelarbeiter einen Fragebogen, sowie auch für die Paarprogrammierer. Dort wurden Dinge zur Motivation gefragt (siehe Fragebögen im Anhang 10.2 und 10.3). Der Fragebogen bestand aus etwa 11 Fragen die man größtenteils durch Ankreuzen beantworten konnte. Es gab allerdings auch einige Fragen die schriftlich beantwortet werden sollten. Außerdem war hinter jeder vorgegebenen Antwort eine Linie, sodass die Studenten noch einen Kommentar bzw. eine Begründung dazu abgeben konnten. Die Studenten machten in dem Fragebogen

Aussagen zur Atmosphäre in ihrer Gruppe während der Zusammenarbeit (oder auch Einzelarbeit), ob es ihnen schwer fiel mit ihrem Partner zu kommunizieren und Probleme oder Fragestellungen mit diesem zu klären. Auch wurde gefragt, ob sie das Übungsblatt schwer fanden und welche Art von Problemen sie hatten und wie sie diese gelöst haben. Weiterhin fragte ich sie nach den positiven und negativen Eigenschaften des Paarprogrammierens oder des Einzelarbeitens, die sie während des Arbeitens feststellen konnten und wie viel Spaß sie hatten. Eine sehr interessante Frage war auch, ob die Studenten den Eindruck hatten, dass der Partner und sie selbst gleich viel Arbeit in das Übungsblatt gesteckt haben. Zudem fragte ich die Einzelarbeiter und auch die Paarprogrammierer, ob sie das Gefühl hatten das Aufgabenblatt schneller gelöst zu haben und ob sie mit der Lösung der Aufgaben zufrieden waren. Die Frage nach der Vorbereitungszeit war offensichtlich zu Beginn nicht eindeutig gestellt, auch wenn ich den Fragebogen vorher von einer unbeteiligten Person ausfüllen ließ. Zudem war sie bei diesem Kurs auch fast überflüssig. Die meisten Studenten bereiteten sich kaum vor, da ihnen auch wenig Zeit dafür blieb. Selbst als ich die Frage korrigierte, schienen sehr wenig Leute die Veränderung wahrzunehmen. Dem letzten Fragebogen fügte ich noch zwei weitere Fragen hinzu. Dort wurde gefragt ob die Teilnehmer ihren Übungspartner bereits vorher kannten und ob sie den Eindruck hatten, dass die Stimmung in dem Raum in dem sie arbeiteten gut war. Auch hier wurde nach der Ursache dafür gefragt. Außerdem wurden die Teilnehmer gefragt, ob sie den Eindruck hatten, ob die Stimmung in ihrem Raum besser oder schlechter war.

2.2.1 Raymond-Lister-Test

Während meines Vortrags zur Vorstellung meines Themas bekam ich den Hinweis, dass es einen solchen Test gibt, schaute ihn mir an und übersetzte ihn um ihn für den Blockkurs "Programmieren in Java" verwenden zu können.
Der Test wurde entwickelt, um Studenten vergleichen zu können und besteht aus 12 Fragen (siehe hierzu Anhang 10.12). In einer Studie von Raymond Lister et al. konnten aufgrund des Tests weitere Erkenntnisse über die Paarprogrammierung gewonnen werden [11] .

2.3 Gruppenzusammenstellung

Zum Thema Gruppenzusammenstellung bei der Paarprogrammierung gibt es die unterschiedlichsten Meinungen. Es hat sowohl Vorteile, als auch Nachteile zwei unterschiedlich starke Partner in einer Gruppe zu vereinen. Es kann sich positiv auswirken, denn derjenige der einen größeren Wissensstand hat, kann dem Anderen das fehlende Wissen vermitteln. Allerdings kann es vorkommen, je nach Persönlichkeit, dass sich der etwas versiertere vom Partner gebremst oder genervt fühlt, wenn er ihm viel erklären muss. Auch kann es sein, dass wenn der nicht so versierte Partner eher schüchtern ist, er sich nicht traut den Partner zu fragen und so wenig lernt und die Lösung des Anderen nicht mehr nachvollziehen kann. Ich selber konnte sehr unterschiedliche und interessante Beobachtungen in der Blockveranstaltung "Programmieren in Java" machen, auf die ich später noch eingehen werde. Wenn man zwei etwas programmierschwächere Personen in einer Gruppe vorfindet, kann es sein, dass beide sehr viel lernen und nach anfänglichen Schwierigkeiten sehr gut zurecht kommen. Es kann aber auch vorkommen, dass beide überhaupt nicht voran kommen und die Probleme nicht überwinden können, weil ihnen Hilfe oder eine Idee fehlt. Ich hatte zu Beginn angedacht die Paare mit Hilfe des Persönlichkeitstests mit Myers-Briggs-Typindikator [21] zusammen zuführen. Dort werden Persönlichkeiten nach vier verschiedenen Merkmalen aufgeteilt, sodass sich insgesamt 16 verschiedene Persönlichkeitstypen ergeben. Mit Hilfe des nun bekannten Persönlichkeitstyps hätte man die Gruppen passend nach ihren Eigenschaften zusammenstellen können, um so Konflikte zu vermeiden. Dies erschien mir

bei der Größe des Kurses dann aber sehr aufwendig. Die Studenten hätten einige Zeit für diesen Test benötigt, um einen Persönlichkeitstyp heraus zu bekommen. Außerdem wollte ich die Studenten frei wählen lassen, damit eine natürliche Situation entsteht, wie sie im Alltag auch vorgekommen wäre.

3 Forschungsmethoden

In diesem Kapitel möchte ich auf die Datenerhebungsmethoden eingehen und diese erläutern.

3.1 Forschungsstrategien

Es handelt sich bei der Untersuchung um eine Fallstudie, die etwas von einem Experiment hat. Ein Experiment zeichnet sich dadurch aus, dass jeder zu jedem Zeitpunkt mit den gleichen Mitteln das Experiment wiederholen kann und die selben Ergebnisse bekommt. Dies ist hier nicht der Fall. Da allerdings einige quantitative Messungen gemacht wurden wie sie bei einem Experiment üblich sind und zudem die Teilnehmer in zwei Gruppen eingeteilt waren, kann man hier davon reden, dass die Fallstudie Charakterzüge eines Experiments hat. Bei einer Fallstudie werden nicht nur Hypothesen oder Überlegungen belegt oder widerlegt, sondern es wird sehr in die Tiefe gegangen. Die beobachteten Details sind von hoher Bedeutung und man versucht komplexe Beziehungen in ihren Zusammenhängen aufzudecken und zu untersuchen. Ich möchte hier also erklären, warum gewisse Dinge in speziellen Situationen geschehen und welche Ursachen sie haben können. Wichtig bei einer Fallstudie ist im Gegensatz zu einem Experiment auch, dass die Untersuchung in einem natürlichen Umfeld passiert. Während wir bei einem Experiment versuchen eine bestimmte Situation nachzustellen, die jederzeit wiederholbar ist, versuchen wir bei einer Fallstudie nichts nachzustellen, sondern die Teilnehmer bei ihrer natürlichen Arbeit in ihrem natürlichen Umfeld zu beobachten und daraus Erkenntnisse zu gewinnen.

3.1.1 Vorüberlegung

Die Goal Question Metric, auch kurz GQM genannt befasst sich damit gezielt zu messen. Hierzu muss man sich erst bewusst machen, was man messen möchte und was das eigentliche Ziel ist. Fragen auf einem Fragebogen beispielsweise sollen einem möglichst die Antworten geben, die man für seine Hypothesen oder Studie verwerten kann. Mit Hilfe von GQM soll man dieses Ziel erreichen, da man das Ziel bzw. die Hypothesen der Untersuchung nicht aus den Augen verliert. Die Metrik ist wie ein Baum aufgebaut. Ganz oben in der Wurzel steht das Ziel das man hat (das sogenannte Goal). Als Unterblätter folgen Fragen (Question), die dieses Ziel beschreiben und die Kinder ganz unten sind die Messgrößen (Metric), wie man diese Fragen beantworten bzw. messen kann. Zu Beginn hatte ich einige Probleme GQM richtig anzuwenden, da ich immer mehrere Ziele, die miteinander zusammen hingen, sah. Durch meine Betreuerin Edna konnte ich die Ziele besser auseinander halten und besser ordnen. Mit Hilfe von GQM passte ich die Fragen meines Fragebogens an die Studenten an um möglichst die Antworten zu bekommen, die ich brauchte.

3.2 Datenerhebungsmethoden

3.2.1 Fragebögen

Zu den Datenerhebungsmethoden gehören im Allgemeinen Frageböden, Beobachtungen, Interviews und Dokumente. Ich verwendete hier einmal die Frageböden, mit deren Hilfe ich Dinge zur Motivation

der Studenten abfragen konnte. Außerdem beobachtete ich die Studenten und führte in der zweiten Woche Interviews mit ihnen. Die Fragebögen, die etwa 11 Fragen beinhalten, passte ich mit Hilfe von GQM (Goal Question Metric, Abschnitt 3.1.1) an. Die meisten Fragen hatten eine Multiple Choice Option. Wobei nach dem Setzen des Kreuzes eine Linie folgte, die nach einer Begründung in schriftlicher Form fragte. Es gab sowohl offene als auch geschlossene Fragen, wobei zu den meisten Fragen eine Antwortmöglichkeit oder sogar mehrere vorgeben waren und anschließend nach einer Begründung gefragt wurde. Die Fragebögen verteilte ich in Papierform, da ich mir so eine höhere Teilnahme erhoffte, da die Studenten den Bogen so gleich nach der Übung ausfüllen konnten. Zu Beginn war die Teilnahme auch sehr gut. Später ließ diese allerdings stark nach was erstens damit zusammenhing, dass die Studenten nicht den Grund verstanden jeden Tag den gleichen Fragebogen auszufüllen, da sich ihrer Meinung nach kaum etwas an den Antworten ändern würde. Zwischendurch versuchte ich die Teilnehmer mit dem Verteilen von Schokoladenbonbons für das bisherige Ausfüllen der Fragebögen zu motivieren. Dies hatte leider nicht den erhofften Erfolg, weshalb ich mich entschied für zwei Tage die Fragebögen nur an Studenten auszuteilen die ich intensiv beobachtet habe und bei denen ich Dinge festgestellt hatte die ich hinterfragen wollte. Außerdem bat ich jeden Teilnehmer eindringlich den Fragebogen auszufüllen. Nach dem ersten Tag fragten einige Teilnehmer bereits ob sie heute keinen Bogen bekommen würden und schienen verwundert, auch wenn sie die Bögen sonst eher selten ausfüllten oder sehr ungern. Am letzten Tag füllten alle Anwesenden den Fragebogen aus, in den ich noch zwei weitere Fragen aufnahm (sie Fragebögen im Anhang 10.10 und 10.11).

3.2.2 Beobachtungen

Bei den Beobachtungen bemühte ich mich sehr unauffällig zu sein und mischte mich nicht ein. Es handelte sich hierbei um eine systematische Beobachtung, da ich mir im Vorfeld Tabellen anlegte und Tag für Tag plante, was ich beobachten wollte. Nach dem zweiten Tag nahmen die Beobachtungsvorlagen Gestalt an, da ich mittlerweile die Namen aller Teilnehmer kannte und daraufhin die Vorlagen anpassen konnte um nur noch das Beobachtete in den entsprechenden Feldern einzutragen. Dies ermöglichte mir mich intensiver auf die Beobachtungen zu konzentrieren und Zeit beim notieren der Daten zu sparen. Die Beobachtungen zeichnete ich nicht mit der Videokamera auf. Dies entschied ich bereits im Vorfeld, da bei anfänglich 95 angemeldeten Teilnehmern über zwei Wochen die Fülle an Material zu groß gewesen wäre um sich alles noch einmal genau anzuschauen. So fertigte ich Beobachtungsvorlagen an und notierte alles was mir auffiel und ich für wichtig hielt. Es war schon vorher durch Beobachtungsvorlagen (siehe Anhang 10.8 und 10.9) geplant, welchen Beobachtungsgegenstand (siehe Abschnitt 4.4) ich an welchem Tag beobachtet wollte und daran hielt ich mich und legte den Fokus darauf.

3.2.3 Interviews

Die Interviews waren im Gegenzug nicht vordefiniert, sondern eher von spontaner Art. Es handelte sich hierbei also um unstrukturierte Interviews. Ich fing an sie in der zweiten Woche zu führen, um zwischendurch Fragen zu den Ereignissen, die ich beobachtete stellen zu können. Ich sprach die Studenten entweder in der jeweiligen Situation darauf an oder versuchte mir einzelne Teilnehmer im Anschluss beiseite zu nehmen. Da viele Studenten Paarprogrammierung ausprobierten versuchte ich, um der Wahrheit entsprechende Antworten zu erhalten, die Teilnehmer bei den Interviews möglichst einzeln zu befragen. Ich empfand es als positiv die Interviews erst in der zweiten Woche durchzuführen, da ich mir in der Zeit ein Bild von den Teilnehmern machen konnte und diese mich besser kannten. So entwickelte sich eine Vertrauensbasis und die Teilnehmer waren teilweise offener mir gegenüber.

Bei den Interviews entschied ich mich dafür kein Aufzeichnungsgerät zu verwenden und schrieb alles im Anschluss des Interviews auf. Die Atmosphäre ohne das Gerät erschien mir persönlicher und ungezwungener. Das Interview ähnelte eher einem normalen Gespräch und nicht einer Befragung, sodass sich die Studenten nicht ausgefragt vorkamen.

3.3 Quantitative Erhebung

Quantitative Messungen macht man häufig bei einem Experiment. Auch ich verwendete einige quantitative Messungen, wie zum Beispiel das Messen der benötigten Zeit für die einzelnen Aufgaben oder die Zeit, die zur Vorbereitung gebraucht wurde, wobei sich dies eher als schwierig herausstellte. Es war geplant, dass die Tutoren der Blockveranstaltung die Zeit eintragen, die die Studenten zum Bearbeiten der Aufgaben benötigten. Dies bot sich an, da es zu Beginn hieß, dass die Tutoren nach jeder Aufgabe gerufen werden sollten, um die Aufgabe zu kontrollieren und ab zu haken. Allerdings war das so nicht immer der Fall. Hierzu fertigte ich im Vorfeld eine Liste für jeden Tutor mit den Namen der Studenten seines Tutoriums und der Anzahl der Aufgaben an, in dem jeweiligen Feld sollte dann die Uhrzeit eingetragen werden, wann die Aufgabe beendet wurde. Bei einigen Tutoren trugen die Studenten ihre Zeiten selber ein, was nach einigen Startschwierigkeiten größtenteils gut funktionierte, allerdings in der zweiten Woche etwas nachließ. In anderen Tutorien riefen die Teilnehmer die Tutoren, sobald sie mit einer Aufgabe fertig waren, dies funktionierte im Allgemeinen gut. Des weiteren gab es Tutorien, wo der Tutor selbstständig beim Herumgehen und Nachgucken die Zeiten eintrug. Dies führte leider oft dazu, dass Zeiten fehlten oder sehr ungenau waren. Die Vorbereitungszeit wollte ich eher qualitativ als quantitativ messen. Da ich zur Vorbereitungszeit nicht bei allen Teilnehmern anwesend sein konnte und diese nicht für alle gleichzeitig messen konnte, versuchte ich diese bei einzelnen Paaren oder Einzelarbeitern zu beobachten und zu messen. Dies stellte sich allerdings als äußerst schwierig heraus, da die Vorbereitungszeit nicht fest abzugrenzen war. Es gab Paare und auch Einzelarbeiter, die kurz diskutierten oder sich die Aufgabe durchlasen und dann versuchten zu programmieren. Gleich zu Beginn kamen sie allerdings nicht weiter und fragten einen Tutor oder lasen sich wieder etwas zur Aufgabe durch. Also wurde im Endeffekt die Vorbereitungszeit nur unterbrochen. Da allerdings immer im Wechsel programmiert und wieder gelesen wurde, waren die Messungen nicht eindeutig und nicht zu verwenden. Außerdem war es schwierig die Vorbereitungszeit zu messen, da an einigen Tagen, als ich diese messen wollte, nur eine Aufgabe auf dem Übungsblatt war und mir dies vorher nicht bekannt war. So konnte ich nur bei einem Paar die Vorbereitungszeit messen. Anfänglich war geplant an dem Tag, an dem die Vorbereitungszeit gemessen werden sollte die Paare immer zu verlassen, sobald jemand anderes den Tutor ruft, der die Zeit für die Aufgabe eintragen sollte. So hätte ich bei vielen Teilnehmern die Vorbereitungszeit messen können. Da allerdings nicht immer der Tutor gerufen wurde, verfälschte dies die Ergebnisse und ich konnte die Vorbereitungszeit nicht oft genug messen um eine klare Aussage darüber zu machen. Deshalb entschied ich mich mit meiner Betreuerin in der zweiten Woche diese nicht mehr zu messen. Auch eine quantitative Messung sind die Ergebnisse des Tests und die Punktezahl der Aufgabenblätter. Mit Hilfe des Tests kann verglichen werden wie viel die Teilnehmer gelernt haben und wer besser abschneidet. Bei den erreichten Punkten der Aufgabenblätter kann man Paarprogrammierer und Einzelarbeiter vergleichen. Im Test hingegen kann man zusätzlich noch vergleichen inwiefern ein Unterschied zwischen den Partnern einer Gruppe besteht. Dies kann man mit den Beobachtungen in Zusammenhang setzen.

3.4 Qualitative Erhebung

Die Beobachtungen sind eher von qualitativer Art und machen den größeren Teil dieser Arbeit aus. Es wurde Wert darauf gelegt Details zu beobachten und in die Tiefe zu gehen. Dinge, die die Motivation betreffen wurden beobachtet. Hierzu zählt beispielsweise die konzentrierte Arbeitszeit und auch inwiefern die Teilnehmer sich ablenkten. Zudem wurde die Rollenverteilung auch wieder qualitativ beobachtet, d.h. es wurde nicht gemessen, wer wie lange die Rolle des Driver einnahm und wer für welche Zeit die Rolle des Observers einnahm, sondern inwiefern die Rollenverteilung gerecht und gleichverteilt war und was der Driver und der Observer jeweils taten. Wirken sich die Rollen auf die Bearbeitung des Übungsblatts aus und in welcher Form wirken sie sich aus? Interessant war auch der Aspekt der Rollenverteilung im Zusammenhang mit der untersuchten geleisteten Hilfestellung. Auch wurde beobachtet wie die Stimmung unter den Teilnehmern ist und es wurde versucht zu erkennen, woran die Stimmung festzumachen ist bzw. was zum Beispiel gute Stimmung wirklich ausmacht. Die Einzelarbeiter wurden ebenso daraufhin untersucht wie die Paarprogrammierer. Allerdings wurden ausschließlich die Einzelarbeiter auf Kommunikation hin untersucht. Das Paarprogrammierer, die sich in einer Gruppe befinden unmittelbar dazu gezwungen sind mehr oder weniger miteinander zu kommunizieren ist selbstverständlich. Interessant dagegen ist, ob Einzelarbeiter auch kommunizieren und sich über die Aufgabe mit jemandem austauschen. Genau das wurde untersucht und beobachtet. Auch eine qualitative Messung stellt der ausgeteilte Fragebogen dar, den die Studenten ausfüllten, sodass ich ihre subjektive Wahrnehmung mitbekam.

4 Vorbereitende Überlegungen

In diesem Kapitel möchte ich die Annahmen vorstellen nach denen ich meine Beobachtungen anfänglich ausgerichtet habe und die Beobachtungsvorlagen angefertigt habe. Diese waren nützlich, da ich den Fokus meiner Beobachtungen nun auf bestimmte Dinge legen konnte. Es war mir wichtig zu wissen auf welchem Beobachtungsgegenstand ich jeden Tag den Fokus legen wollte, damit ich bei meinen Beobachtungen nichts übersehen konnte.

4.1 Zeit

4.1.1 Vorbereitungszeit

Wie schon zuvor erwähnt habe ich die Vorbereitungszeit und die Zeit, die die Teilnehmer für das Bearbeiten der Aufgaben auf dem Übungsblatt benötigen gemessen. Als Zeit, die zum Bearbeiten des Aufgabenblattes benötigt wurde, wurde die Zeit angesehen, wo die Teilnehmer anfingen zu tippen. Die Vorbereitungszeit war die Zeit die die Teilnehmer hatten, bis sie mit dem Tippen anfingen. Während die Vorbereitungszeit von mir beobachtet werden sollte, notierten die Tutoren für mich die Zeit, die die Teilnehmer zum Bearbeiten des Blatts benötigten. Mit Hilfe der Vorbereitungszeit wollte ich herausfinden inwiefern diese die Implementierungszeit beeinflusst und die Einzelarbeiter mit den Paarprogrammierern vergleichen. Es wäre interessant gewesen herauszufinden ob die Paarprogrammierer eine längere Vorbereitungszeit im Gegensatz zu den Einzelarbeitern benötigen, da sie sich die Aufgabe und die dazugehörigen Ideen erst erklären und ausdiskutieren müssen, dafür aber eine kürzere Implementierungsphase haben. Oder aber auch, dass der Einzelarbeiter eine längere Vorbereitungszeit benötigt, da er sich Dinge selbst erschließen muss und an manchen Stellen die Ideen und Hilfestellungen des Partners fehlen.

4.1.2 Bearbeitungszeit

Bei der Bearbeitungsdauer sollten auch wieder Paarprogrammierer mit Einzelarbeitern verglichen werden. Durch den Vorteil die Teilnehmer über den Zeitraum von zwei Wochen zu beobachten, war hier genauso wie bei der Vorbereitungszeit angedacht zu schauen, ob diese sich innerhalb der zwei Wochen beispielsweise verkürzt. Hätte sie sich verkürzt, wäre dies darauf zurück zuführen das sich die Partner bereits besser kennen gelernt haben und nicht mehr so viele Absprachen benötigen. Auch ein interessanter Aspekt war die tatsächliche Dauer der Bearbeitungszeit mit der Einschätzung des Professors zu vergleichen. Wenn der Großteil der Studenten für eine Aufgabe auffällig lang braucht wäre zu untersuchen woran dies liegt. Beispiele dafür wären, dass die Aufgabe zu schwer war, der Stoff in der Vorlesung nicht ausreichend behandelt oder vermittelt wurde, die Aufgabe schwer verständlich oder sehr viel Spielraum gelassen hat.

4.2 Punkte

4.2.1 Punkte der Übungsaufgaben

Es sollten die Punkte des Übungsblatts verglichen werden. Da sich die Tutoren für die Vergabe der Noten die Punkte für die Teilnehmer notierten, können die Listen einfach weiterverwendet werden. Es sollten wieder Paarprogrammierer mit Einzelarbeitern verglichen werden. Dies soll pro Blatt und verteilt über die zwei Wochen gemacht werden, sodass man eventuell eine Entwicklung der Teilnehmer sehen kann.

4.2.2 Punkte des Tests

Auch bei den Punkten des Tests sollten Einzelarbeiter mit Paarprogrammierern verglichen werden. Da es einen Test zu Beginn gab und einen gegen Ende der Vorlesung kann auch die Entwicklung über die zwei Wochen gemessen werden. Außerdem kann ersichtlich werden inwiefern sich Paarprogrammierung auf die Teilnehmer ausgewirkt hat. Schneidet zum Beispiel einer der Gruppe deutlich schlechter im Test ab als sein Partner kann man sich die Frage stellen woran das lag. War der etwas schlechtere Partner eher ruhiger und abwesend oder war er eher aufmerksam und fragte viel nach?

4.3 Motivation

Zur Motivation zähle ich das konzentrierte Arbeiten, sowie das Ablenken, die Stimmung, die Rollenverteilung und die Hilfestellung die untereinander gegeben wird. Auch dazu zählt die Kommunikation der Einzelarbeiter und die subjektive Wahrnehmung der Studenten, die sie in einem Interview oder mit Hilfe des Fragebogens äußern.

4.3.1 Konzentriertes Arbeiten und Ablenkung

Interessant zu beobachten ist, ob sich die Paare gegenseitig ablenken und wie sie sich ablenken. Wann lenken sie sich ab und vor allem, wie kommen sie wieder zurück ins konzentrierte Arbeiten? Treiben sich die Paare gegenseitig an, sodass sie über einen längeren Zeitraum konzentriert arbeiten anstatt sich abzulenken? Lenken sie sich häufiger ab als Einzelarbeiter, lenken Einzelarbeiter Paarprogrammierer ab, weil sie nach Hilfestellung fragen oder lenken sogar Paarprogrammierer Einzelarbeiter ab, um Hilfe zu bekommen? Wie wirkt sich die Ablenkung oder das konzentrierte Arbeiten auf die Stimmung der Teilnehmer aus, oder wirkt sich umgekehrt die Stimmung der Teilnehmer auf ihre

Arbeitsmoral aus? Ist Ablenkung immer schlecht? Ein kurzer witziger Kommentar kann beispielsweise die Stimmung anheben und so zu einer Leistungssteigerung und besseren Motivation führen. Zu konzentriertem Arbeiten zählt das Lesen der Aufgaben, Vorlesungsfolien oder weiterem Material, sowie sich Notizen oder Skizzen zu machen. Auch das Tippen, recherchieren und nachdenken gehören zu konzentriertem Arbeiten. Diskussionen über die Aufgabe oder dem entsprechenden Thema zählen ebenso dazu. Zur Ablenkung zählen unter anderem Witze machen, über die Freizeit plaudern, sich beispielsweise etwas zu Essen holen oder anderweitig beschäftigen.

4.3.2 Stimmung

Wie wirkt sich die Stimmung auf die Bearbeitung der Aufgabenblätter aus? Was macht gute Stimmung aus und hat gute Stimmung immer etwas mit Ablenkung zu tun? Dies soll hier qualitativ Beobachtet werden. Beispiele für gute Stimmung wären Lachen, häufiger Blickkontakt und eine rege Kommunikation. Währenddessen schlechte Stimmung durch weniger Kommunikation, wenig Blickkontakt oder wenn jemand eine negativ ausgerichtete Mimik hat, festzumachen sind.

4.3.3 Rollenverteilung

Es soll hier nicht quantitativ gemessen werden, wie lange jeder der Gruppe die Rolle des Drivers oder des Observers hatte. Allerdings soll beobachtet werden ob die Rollenverteilung in etwa gleichverteilt ist und wie sich diese auf das Verhalten der Studenten auswirkt. Ist derjenige der die Observer-Rolle hat eher ruhiger und wird aktiver wenn er die Driver-Rolle hat? Wie werden die Rollen gewechselt, liegt ein Schema zugrunde, sprechen sich die Paare ab oder geschieht es automatisch? Wie sieht der Vorgang aus, wenn die Rollen getauscht werden, reißt einer der Partner die Tastatur zu sich heran oder haben die Partner vorher Blickkontakt? Welche Vor- und Nachteile kann man erkennen. Gibt es Paare mit sehr großem unterschiedlichem Leistungsniveau die viel voneinander lernen oder kommt es häufiger vor, dass sich der Leistungsstärkere gebremst fühlt und dem eher ruhigerem Partner wenig erklärt.

4.3.4 Hilfestellung

Hilfestellung wird von jemandem gegeben, wenn derjenige etwas erklärt, zeigt oder vormacht. Auch eine Diskussion über das Thema kann als Hilfestellung verstanden werden, wenn sie zur Klärung beiträgt. Zu beobachten ist, ob die Hilfestellung problemlos gegeben wird wenn danach verlangt wird, oder auch wenn nicht direkt danach gefragt wird. Wie sieht die gegebene Hilfestellung aus? Kommt es beispielsweise vor, dass der Partner ständig übergangen wird und ihm die Hilfe verwehrt wird? Welche Hilfestellung wird von Paarprogrammierern und welche von Einzelarbeitern eingefordert?

4.3.5 Kommunikation der Einzelarbeiter

Zur Kommunikation zählen Unterhaltungen, Blickkontakt, sowie Körpersprache. Da Paarprogrammierer in der Gruppe zwangsläufig mehr oder weniger miteinander kommunizieren müssen, wird hier eher der Fokus auf die Einzelarbeiter gelegt. Es wird untersucht inwiefern die Einzelarbeiter mit ihrem Umfeld kommunizieren. Hierzu zählt sowohl das Freizeitgespräch, als auch die Diskussion über die Aufgabe oder gar der Lösung der Aufgabe. Ist es wirklich so, dass diejenigen die keine Paarprogrammierung ausprobieren, wirklich allein arbeiten?

4.3.6 Subjektive Wahrnehmung der Studenten

Die subjektive Wahrnehmung wird aus den Fragebögen und Interviews mit den Studenten ersichtlich. Durch meine Beobachtungen können diese verifiziert oder widerlegt werden. Kreuzt der Student zum Beispiel an, dass er das Gefühl hatte die Stimmung wäre gut und lustig gewesen und ich beobachte wie sich das Paar die meiste Zeit anschweigt, streitet und die Mimik einer Person sehr negativ ist, sollte dies näher untersucht und hinterfragt werden.

4.4 Zeit, Punkte und Motivation in der systematischen Beobachtung

4.4.1 Übersicht der geplanten Beobachtungen

Tabelle 1: Übersicht der Beobachtungsannahmen

Tag	Beobachtungsannahme
1.Tag (Montag)	Vorbereitungszeit, Einzelarbeiter auf Kommunikation und Hilfestellung untersuchen
2.Tag (Dienstag)	Stimmung, konzentrierte Arbeitszeit
3.Tag (Mittwoch)	Rollenverteilung, Hilfestellung
4.Tag (Donnerstag)	Vorbereitungszeit, Einzelarbeiter auf Kommunikation und Hilfestellung untersuchen
5.Tag (Freitag)	Stimmung, konzentrierte Arbeitszeit
6.Tag (Montag)	Stimmung, konzentrierte Arbeitszeit
7.Tag (Dienstag)	Rollenverteilung, Hilfestellung
8.Tag (Mittwoch)	Vorbereitungszeit, Einzelarbeiter auf Kommunikation und Hilfestellung untersuchen
9.Tag (Donnerstag)	Rollenverteilung, Hilfestellung
10.Tag (Freitag)	Stimmung, konzentrierte Arbeitszeit

Mit Hilfe dieser Vorüberlegung konnte ich für jeden Tag den Fokus auf einen Beobachtungsgegenstand setzen.

4.4.2 Beobachtungsannahme: Vorbereitungszeit, Einzelarbeiter auf Kommunikation und Hilfestellung untersuchen

Vorbereitende Überlegungen:

Paare benötigen eine längere Vorbereitungszeit als Studenten, die einzeln arbeiten. Diese verkürzt sich allerdings im Laufe der Zeit von zwei Wochen. Außerdem verkürzt sich bei den Studenten, die Paarprogrammierung anwenden die Zeit, die sie für das Implementieren der Aufgabe benötigen. Einzelarbeiter in einem Raum kommunizieren trotzdem und geben sich auch teilweise Hilfestellungen. Einzelarbeiter fragen öfter den Tutor um Rat, als Studenten die Paarprogrammierung anwenden.

Details:

Es werden beide Gruppen, die Gruppe die Paarprogrammierung anwendet und die, die einzeln arbeitet, in Bezug auf die Vorbereitungszeit untersucht. Immer wenn eine Gruppe fertig ist mit der

Aufgabe und den Tutor ruft, wird zu dieser Gruppe gewechselt und beobachtet, wie lange sie brauchen um mit dem Programmieren anzufangen. Dies wird von 14:30 - 15:30 Uhr und 16:00 - 17:30 Uhr gemacht. Die Einzelarbeiter werden auf Kommunikation und gegenseitige Hilfestellung untersucht. Ich werde in dem entsprechendem Raum anwesend sein und alles im Überblick behalten und bei Auffälligkeiten zu der Gruppe gehen und diese beobachten. Dies wird am 1.Tag von 14:00 - 14:30 Uhr und 15:30 - 16:00 Uhr, sowie 17:30 - 18:00 Uhr gemacht. Am 4.Tag werden zwei Paare im Vergleich für etwa zwei Aufgaben lang untersucht. So kann man sehen, ob sie für die zweite Aufgabe weniger Vorbereitungszeit benötigen. Dies wird von 14:00 - 15:30 oder etwa 16:00 Uhr und 16:00 oder 16:30 - 17:30 Uhr untersucht. Die Einzelarbeiter werden in Bezug auf Hilfestellung und Kommunikation untereinander von 15:30 oder 16:00 - 16:00 oder 16:30 Uhr und 17:30 - 18:00 Uhr beobachtet. Am 8.Tag werden von 14:00 - 16:00 Uhr zwei Paare im Vergleich beobachtet und von 16:30 - 17:00 Uhr und 17:30 - 18:00 Uhr die Vorbereitungszeit ganz normal untersucht. Die Einzelarbeiter werden von 16:00 - 16:30 Uhr und 17:00 - 17:30 Uhr untersucht. Mit Absicht wurden alle Uhrzeiten unterschiedlich gewählt und es wurde bewusst ein fester Zeitrahmen im Voraus gewählt. So sollte es beispielsweise nicht vorkommen, dass die Einzelarbeiter immer von 17:30 - 18:00 Uhr beobachtet werden, damit nicht immer der Fall eintreten kann, dass einige Personen schon weg sind, da sie die Aufgabe schon frühzeitig fertig hatten. Auch sollte es mir ermöglichen im Zuge der Beobachtungen ein klares Schema zu haben, woran ich mich halten kann, um nicht die Beobachtungen aus den Augen zu verlieren und mich auf das Richtige zu konzentrieren und zu fokussieren.

4.4.3 Beobachtungsannahme: Stimmung, konzentrierte Arbeitszeit

Vorbereitende Überlegungen:

Bei Paarprogrammierern wird nur eine kurzzeitige Ablenkung stattfinden und die meiste Zeit konzentriert gearbeitet. Wenn extrem gute oder schlechte Stimmung herrscht, sind die Teilnehmer abgelenkt und arbeiten nicht konzentriert.
Freitags wird weniger konzentriert gearbeitet und die Stimmung ist ausgelassener. Am Montag ist die Stimmung nicht so gut wie sonst und es wird weniger konzentriert gearbeitet.

Details:

Ich werde in einem Raum unauffällig anwesend sein und alle Teilnehmer grob im Überblick behalten. Bei Auffälligkeiten wie Lachen, lauten Unterhaltungen oder wenn ein Paar sehr wenig Blickkontakt hält, still ist oder einer eine negative Mimik hat, werde ich zu dem entsprechendem Paar oder Einzelarbeiter gehen und diesen genauer beobachten. Es soll auch beobachtet werden, inwiefern von einer bestimmten Stimmung wieder in das konzentrierte Arbeiten übergegangen wird.

4.4.4 Beobachtungsannahme: Rollenverteilung, Hilfestellung

Vorbereitende Überlegungen:

Derjenige, der das Lösen der Aufgabe vorantreibt und eine eher führende Rolle einnimmt, beantwortet die Fragen des Partners. Studenten die Paarprogrammierung anwenden fragen seltener den Tutor um Hilfe und helfen sich im Team. Meistens wird derjenige dem Tutor Fragen stellen, der die führende Rolle übernimmt, was nicht unbedingt bedeuten muss, dass derjenige die Driver-Rolle inne

hat.

Details:

Es wird bei Paarprogrammierern untersucht, ob die Rollenverteilung gleichverteilt ist und wie sie sich auf die Hilfestellung auswirkt. Gibt beispielsweise derjenige der die Driver-Rolle hat und das Lösen der Aufgabe vorantreibt eher dem Anderen Hilfestellung und stellt dem Tutor Fragen? Wie wird die Hilfestellung gegeben, was wird gefragt und wie wird auf Fragen reagiert? In Bezug auf die Rollenverteilung ist auch wichtig zu erkennen, wie der Prozess vonstattengeht, wenn die Rollen getauscht werden. Wirkt sich die Rolle die jemand hat auf dessen Verhalten aus?

5 Ergebnisse des Pilot Tests

Da ich die Gruppe des Kurses "Softwareprozesse" noch in der eigentlichen Planungsphase beobachtete, unterscheidet sich die Analyse und die Beobachtungsgegenstände stark von denen des Kurses "Programmieren in Java". Viele Messungen wurden eher quantitativ gemacht (siehe Beobachtungsvorlage im Anhang 10.2). Ich hatte durch die freiwilligen Teilnehmer aus dem Kurs "Softwareprozesse" die Möglichkeit schon einige Dinge zu testen und auf mögliche Probleme beim Beobachten von Paarprogrammierern zu stoßen. Außerdem ergab sich dadurch die Möglichkeit weitere Beobachtungsgegenstände, sowie vorbereitende Überlegungen anzustellen.
In dem Kurs "Softwareprozesse" wurde die Aufgabe in einem Übungsblatt gestellt das Spiel Mastermind gemeinsam zu implementieren und dabei den Cleanroom-Ansatz anzuwenden. Wie in Abschnitt 2.1.1 erläutert legt man dabei viel Wert auf eine hohe Qualität des Codes, wobei Fehler nicht korrigiert, sondern von vornherein vermieden werden sollen.

Zu erwähnen ist, dass beide Gruppen die Aufgabe nicht vollständig gelöst haben. Während es bei der ersten Gruppe am Zeitmangel lag, hatte die zweite Gruppe gar nicht erst den Anspruch die Aufgabe komplett fertig zu stellen, da ihnen die bisher erreichten Punkte des Übungsblatt ausreichten.

Bei den nachfolgenden Namen handelt es sich um frei erfundene Synonyme, welche keine Ähnlichkeit mit den realen Probanden aufweisen.

5.1 Ergebnisse des Fragebogens

Die Studenten Jan und Dirk schätzten ihre Programmierfähigkeiten als gut ein und hatten bereits Erfahrungen mit Paarprogrammierung sammeln können. Auch das andere Paar schätzte seine Programmierfähigkeiten als gut ein und sammelte bereits im Vorfeld Erfahrungen in Paarprogrammierung. In den folgenden Tabellen werden Angaben aus dem Fragebogen vorgelegt.

Tabelle 2: Positive Eigenschaften von Paarprogrammierung

Dirk	"Das direkt eine Durchsicht stattfindet."
Jan	"Ist motivierender und man kommt gemeinsam auf bessere Ansätze."
René	"Besseres Verständnis, da man erfährt, welche Überlegungen zu gewissen Entscheidungen geführt haben."
Daniel	"Fehler werden schnell entdeckt/behoben. Ich kann auch Vorgehens- oder codetechnisch immer noch was lernen, es macht Spaß weil man sich gegenseitig motiviert."

Speziell auf diese Sitzung bezogen wurde noch angegeben, dass eine schnelle Fehleraufdeckung stattfand und eine schnelle, sowie saubere Spezifikation erstellt wurde. Das Diskutieren von potenziell fehleranfälligen Codeabschnitten empfanden sie ebenfalls als positiv.

Tabelle 3: Negative Eigenschaften von Paarprogrammierung

Jan	"Ich meine es dauert länger."
René	"Bei trivialen Problemen langweilt sich der Observer."
Daniel	"Nach einer gewissen Zeit "nervt" man sich möglicherweise kurzzeitig etwas an, vielleicht weil man unterschiedlich angespannt/motiviert/aufmerksam ist. Manchmal tritt ein gewisses "Rumposen" vor dem Anderen auf."

Einer der Paare gab an, dass er am Ende eine Pause gebraucht hätte. Das er diese allerdings nicht machte zeigt, dass er durch seinen Partner zum konzentrierten Weiterarbeiten animiert wurde. Weiterhin fiel beiden Paaren das Lösen des Aufgabenblatts gemeinsam leichter.

Tabelle 4: Ursachen die das Lösen des Aufgabenblatts erleichterten

Dirk	"Das noch jemand auf den Code schaut."
Jan	"Wegen der besseren Motivation."
René	"Man erhält durch den Partner eine zweite Sichtweise."
Daniel	"Geordnetes, aber zügiges Vorgehen in Richtung eines fest stehenden Ziels, PP macht Spaß, "gute"/praxisnahe Übungsaufgabe."

Außerdem gaben Dirk und Jan an, dass sie Spaß hatten und Probleme, sowie Fragen mit ihrem Partner offen klären konnten. Auch fiel es ihnen nicht schwer mit dem Partner zu kommunizieren. Sie sind auch der Meinung durch das Anwenden von Paarprogrammierung mehr zu lernen. Bei René und Daniel gab einer der beiden an, dass er Dinge, die er nicht verstanden hat nicht so gut mit seinem Partner klären konnte. Dies ließ sich auch gut beobachten, da an einer Stelle Code vom Partner der nicht verstanden wurde nicht erklärt wurde. Ansonsten hatte sie ebenfalls wie Dirk und Jan Spaß beim Bearbeiten der Aufgabe und sind der Meinung durch das Anwenden von Paarprogrammierung mehr zu lernen.

Tabelle 5: Gründe den Inhalt der Vorlesung durch Paarprogrammierung besser zu verstehen

Dirk	"Da man sich gemeinsam damit beschäftigt und ggf. falsch verstandene Dinge bemerkt."
Jan	"Man Fragen gleich klären kann und dadurch was lernt."
René	"Ich ein Lerntyp bin, dem gemeinsames Erläutern sehr viel bringt."
Daniel	"Zwei Hirne mehr als doppelt so viel leisten als nur eines -> "Synergieeffekte"

Ebenso empfinden sie das Erlernen von Paarprogrammierung für das spätere Berufsleben sinnvoll, da Kommunikation wichtig ist und man in einem Unternehmen arbeiten könnte, welches Paarprogrammierung anwendet.
Jan und Dirk hatten den Eindruck, dass das Lösen des Aufgabenblatts zusammen länger dauerte. René und Daniel hatten den Eindruck, dass das Lösen zusammen schneller ging.
Einer der beiden findet das die Anwendung von Paarprogrammierung oder verteilter Paarprogrammierung für das Lösen des Aufgabenblatts sinnvoll ist, während der Andere dies nicht findet, da er lieber seinen eigenen Ansatz umgesetzt hätte. Auch René und Daniel finden Paarprogrammierung für das Lösen des Aufgabenblatts sinnvoll, da die Kommunikation das Verständnis für die Aufgabe und dessen Lösung verbessert.
René und Daniel denken, dass das Erlernen von Paarprogrammierung für das Berufsleben sinnvoll ist, da es eine angenehme Art des Programmierens ist und sie es für effektiv halten. Außerdem wird dadurch besserer Code erstellt.

Weiterhin wurde in dem Fragebogen (siehe Anhang 10.1) abgefragt, welche Voraussetzungen in der Universität geschaffen werden müssten, um Paarprogrammierung oder verteilte Paarprogrammierung sinnvoll einsetzten zu können. Es wurde unter anderem geantwortet, dass es nichts zu beanstanden gibt, da Saros bereits gut läuft. Auch wurde geantwortet, dass mehrere und kleinere Computerräume bzw. Übungsräume von Vorteil wären, da es wichtig ist, dass das Paar ungestört ist. Die Paare gaben auch an mit Saros zufrieden zu sein, allerdings bevorzugen sie ganz klar das persönliche Treffen mit dem Partner, da sie die direkte Kommunikation für effektiver halten.

5.2 Ergebnisse der Beobachtung von Gruppe 1

Die erste Gruppe brauchte für das Lösen des Aufgabenblatts etwa 2 Stunden und 30 Minuten. Sie haben sich das Aufgabenblatt vorher 5 Minuten angeschaut und sie arbeiteten bereits das ganze Semester über so zusammen. Sie benötigten etwa 32 Minuten an Vorbereitungszeit, in der sie Skizzen zur Aufgabe anfertigten, die einem Zustandsautomaten ähnlich waren. Anschließend implementierten sie 40 Minuten und verbrachten 54 Minuten damit die Skizze auf dem Blatt zu vervollständigen und über die Aufgabe zu diskutieren, wobei Jan kurzzeitig nicht aufmerksam war und abwesend schien, bis ihn Dirk zu einer Antwort drängt. Dann beschäftigten sie sich wieder 14 Minuten mit Java, bis die nächste Unterbrechung von Jan kam. Er entdeckte auf meiner Beobachtungsvorlage, dass ich mitzählte wie oft die Beiden lachten und scherzte, bis Dirk zum weiterarbeiten aufrief. Er tat das ohne Umschweife und sehr deutlich, jedoch nicht unfreundlich. Beide lachten und Dirk sagte: "So jetzt lass uns aber weiter machen". Anschließend implementierte Jan etwa 7 Minuten allein, da Dirk sich etwas zu Essen holte. Dirk schob Jan einfach den Laptop zu, stand auf und ging, wobei er beim Aufstehen erklärte, er würde sich etwas zu Essen holen. Sie erkannten beide, dass sie die Aufgabe in

der ihnen verblieben Zeit nicht schaffen würden und versuchten einen Termin auszumachen um von zu Hause aus, mittels Saros daran weiter zu arbeiten.

Beide hatte Spaß bei der Anwendung von Paarprogrammierung, dies wird auch daran deutlich, dass sie des öfteren gelacht haben und die Atmosphäre sehr gut und freundschaftlich war. Beide waren ein eingespieltes Team, die Atmosphäre war von Anfang bis Ende locker aber konzentriert und nach vorne strebend. Fragen konnten offen geklärt werden und die Kommunikation war sehr gut und angeregt. Es wurde viel über die Aufgabe diskutiert, wobei Fragen nebenbei geklärt wurden und nicht als solche auffielen. Meistens erklärte Dirk Jan etwas und wendete einen Trick in Eclipse an, den Jan noch nicht kannte. Daraufhin fragte Jan, ob er ihm diesen noch einmal zeigen könnte. Dirk erklärte viel und sagte immer an, was er gerade machte. Sein Partner brachte sich allerdings auch ein und gab Tipps. Während der Implementationsphase wurde es etwas ruhiger. Beide motivierten sich gegenseitig und so kam es, dass 2 Std. 22 Min. von 2 Std. 30 Min. konzentriert gearbeitet wurde. Lediglich drei Minuten hat sich das Paar abgelenkt. Ihnen fiel das Lösen des Übungsblatts leichter, da sie Fragen offen klären und sich gegenseitig erklären konnten. Die Rollen waren größtenteils gleichverteilt, gerade zu Beginn erklärten sie sich gegenseitig viel und es herrschte ein fairer und respektvoller Umgang. Dirk versuchte die Driver-Rolle an seinen Partner abzugeben, bekam sie allerdings nach kurzer Zeit wieder, ohne dass Jan etwas am Code veränderte. Dirk schob den Laptop Jan zu und dieser schob ihn Dirk wieder zurück. Absprachen wurden immer gemeinsam getroffen. Allerdings programmierte Dirk mehr und implementierte auch vieles automatisiert von allein ohne nachzufragen. Auch fand er Fehler sehr schnell allein und schaute den Code noch einmal durch. Phasenweise leitete Dirk das Lösen des Aufgabenblatts etwas, was Jan unzufrieden machte, da er lieber seinen eigenen Ansatz implementieren wollte, wie ich dem Fragebogen entnehmen konnte. Allerdings hatte er das zu keiner Zeit angesprochen und es war nicht zu bemerken. Er stimmte seinem Partner, bei Absprachen über das Umsetzen der Aufgabe, auch oft zu. Gegen Ende holte Jan einen Stift heraus, validiert die Aussage des Partners und fand einen Fehler. Er übernahm kurzzeitig die Driver-Rolle. Außerdem versuchte Dirk seinen Partner einzubinden, indem er nach Vorschlägen für Variablennamen fragte. Einmal konnte ich jedoch beobachten, wie Dirk Jan fragte, ob er die Lösung für sinnvoll hält, jedoch die Antwort nicht abwartete und weiter implementierte. Beim Implementieren hatten beide etwas weniger Blickkontakt, da dort der Blick meist auf den Monitor gerichtet war. Ansonsten hatten sie einen regen Blickkontakt. Beide guckten sich oft an, insbesondere wenn sie eine Lösung besprachen oder über deren Möglichkeiten diskutierten. Zudem beobachtete ich die Körperhaltung, um Interesse oder auch Desinteresse ableiten zu können. Dirk und Jan waren oft nach vorn gebeugt, was zeigt, dass sie bei der Sache waren und großes Interesse hatten. Jan lehnte sich ab und an nach hinten oder stützte seinen Kopf auf dem auf den Tisch liegenden Arm ab. Sie verwendeten 74,6 % der Zeit mit dem Implementieren der Aufgabe und nur 25,4 % mit der Vorbereitung.

5.3 Ergebnisse der Beobachtung von Gruppe 2

Die zweite Gruppe benötigte für das Lösen der Aufgabe etwa 2 Stunden und 3 Minuten. Vor der Sitzung schauten sich beide das Blatt einmal kurz an. Ursprünglich wollten sie einige Begriffe schon vorab im Internet nachlesen und etwas für die Sitzung vorbereiten. Das ganze Semester über lösten sie die Aufgaben schon im Team. Sie benötigen 1 Stunde und 6 Minuten zur Vorbereitung. Acht Minuten lang lasen beide, jeder für sich, die Aufgabestellung und versuchten diese zu verstehen. Dabei war einer der Beiden früher fertig und wartete auf den etwas langsameren. 58 Minuten lang schrieben sie dann Pseudocode auf ein Blatt. Der Pseudocode hatte schon einige Ähnlichkeit mit

Java-Code, sodass der Observer später beim Implementieren fast nur noch diktieren musste. Für das Implementieren brauchten sie etwa 46 Minuten. Fünf Minuten verwendeten sie für ein gemeinsames Review.
Sowohl René als auch Daniel machte es Spaß Paarprogrammierung anzuwenden und sie hatten das Gefühl das Blatt schneller gelöst zu haben. Auch empfanden sie das Lösen des Aufgabenblatts zusammen als leichter. Kommunikation fand ausreichend statt. Obwohl sie nicht ganz so lebhaft war, sondern eher konzentriert, wurde auch des öfteren gelacht und die Arbeitsatmosphäre war die ganze Zeit über kollegial, freundlich und fokussiert. Absprachen waren immer vorhanden und es gab auch ruhigere Pausen zwischen drin, wo jeder für sich nachdachte. Der Blickkontakt fehlte größtenteils, bzw. war nur selten da. Oft kam es vor, dass einer den Anderen beim Sprechen anguckte, der Partner aber selten vom Blatt aufschaute. Die Rollenverteilung war immer gerecht und gleichverteilt. Jeder hatte für einen gewissen Part der Aufgabe die Driver-Rolle und implementierte ein Stück Code. Auch setzte der Driver immer die Vorschläge des Observers um und der Observer sagte dem Driver was er tun sollte. Hilfestellung war meistens vorhanden und wurde gegeben, mit einer sehr wichtigen Ausnahme. Als beide mit dem Lösen der Aufgabe nicht weiterkamen, versuchten beide eine Lösung auf einem Blatt anzufertigen. Anschließend erklärten sie sich gegenseitig ihren Code und diskutierten darüber welchen Code sie verwenden wollten. Daniel verstand den Code von René nicht ganz und fragte nach einer genaueren Erklärung. René gab ihm diese allerdings nicht und schien die Frage zu ignorieren bzw. zu übergehen. Das Übernehmen der Driver-Rolle kündigte sich meist schon im Gespräch an, indem er beispielsweise ankündigte, das er das Implementieren dieses Teils übernehmen würde, da er darin Experte ist und dies gut kann. Anschließend wurde der Laptop zu ihm geschoben. Auch kam es vor, dass beide diskutierten und der Eine das Blatt beispielsweise etwas zu sich heranzog und somit die Driver-Rolle schleichend übernahm, da ihm eine Idee gekommen war, die er versuchte umzusetzen. Es war fast immer so, dass einer die Idee des Anderen versuchte umzusetzen, auch wenn er eine andere Lösung vorgeschlagen hätte. Auch tippte der Driver oft und der Observer sagte dem Driver an, was er tippen sollte und machte ihn auf Fehler oder Code-Konventionen, die eingehalten werden sollten, aufmerksam. Bei diesem Paar war die Rollenverteilung absolut gleichverteilt und das Konzept der Paarprogrammierung wurde vorbildlich umgesetzt, wobei man den Eindruck hatte, dass das Paar immer so arbeiten würde. Der fehlende oder selten vorkommende Blickkontakt führte nicht dazu, dass die Teilnehmer weniger oder schlechter miteinander kommuniziert haben. Beide fanden Fehler sehr schnell und René gab Daniel einige Tipps und Vorschläge. Auch zeigte er ihm einige Tricks in Eclipse. Sie waren fast immer nach vorn gebeugt, was auf eine sehr hohe Aufmerksamkeit und Konzentration schließen lässt. Gegen Ende wirkt der Observer etwas Müde, streckt sich und lehnt sich ab und zu zurück, ist aber sofort wieder mit voller Aufmerksamkeit dabei, wenn der Driver Schwierigkeiten hat oder es zu einer Diskussion kommt. Es fiel ihnen schwer mit der Aufgabe zum Ende zu kommen, da sie nicht Kompilieren durften. Nach einer Codedurchsicht haben sie sich beide auf ein Ende geeinigt. Von 2 Std. 3 Min. wurde 1 Std. 59 Min. konzentriert gearbeitet und sie haben sich nur 4 Minuten abgelenkt. Zu den Ablenkungen gehörte ein kurzer Scherz über die Aufgabe und ein privates Gespräch. Außerdem ist auffällig, dass sie mehr Zeit mit der Vorbereitung verbracht haben als mit der Implementierung. 56,4% der Zeit wurde mit der Vorbereitung verbracht.

5.4 Vergleich der ersten und zweiten Gruppe

Obwohl die erste Gruppe nur knapp 25 Minuten länger für die gleiche Aufgabe brauchte und auch fast genau soviel schaffte, kam ihr das Bearbeiten des Blattes sehr lang vor und sie gaben im Fragebogen an, sie hätten den Eindruck gemeinsam länger für das Lösen der Aufgabe benötigt zu haben, als

wenn sie allein gearbeitet hätten. Auffällig ist auch die unterschiedliche Vorbereitungszeit und Art. Während Gruppe 1 eine Art Zustandsdiagramm machte, tippte in der Implementationsphase Gruppe 2 mehr oder weniger vom Blatt ab, da sie in der Vorbereitungszeit mit Pseudocode anfingen und auch viel länger für die Vorbereitung brauchten. Innerhalb beider Gruppen wurde sich Hilfestellung und gegenseitige Motivation gegeben. Auch kommunizierten sie viel miteinander und Sprachen sich ab. Die meiste Zeit haben beide Gruppen konzentriert miteinander gearbeitet und sich nur 3 bis 4 Minuten gegenseitig abgelenkt. Bei der zweiten Gruppe fällt durch den Fragebogen auf, dass einer der beiden ohne den Partner eher eine Pause gemacht hätte. Durch den Partner machte er diese allerdings nicht, was bestätigt, dass sich das Team gegenseitig motiviert, anspornt und zur Konzentration verhilft. Obwohl die Rollenverteilung bei Dirk und Jan nicht ungleichmäßig erschien, da auch Jan sich oft einbrachte und Lösungsansätze vorschlug, war es doch so, dass Dirk am Laptop eher die Driver-Rolle übernahm. Im Gegensatz dazu steht das vorbildliche Umsetzten des Konzepts Paarprogrammierung von René und Daniel. Die Rolle des Drivers wurde jedoch oft von demjenigen eingenommen. der für den entsprechenden Code-Abschnitt, den größeren Wissenstand aufwies. Außerdem ist zu erkennen, dass eine längere Vorbereitungszeit dazu führt, dass sich die Zeit für das Implementieren verkürzt.

5.5 Erkenntnisse aus dem Pilot Test

Da ich zum Zeitpunkt der Beobachtungen der Teilnehmer aus dem Kurs "Softwareprozesse" noch keine vorbereitenden Überlegungen gemacht hatte, möchte ich einige Überlegungen aus der Fallstudie vom Blockkurs verwenden um die Teilnehmer daraufhin untersuchen.

Die Überlegung, dass bei Paarprogrammierern nur eine kurzzeitige Ablenkung stattfindet und sie die meiste Zeit konzentriert arbeiten, hat sich im Pilot Test bestätigt. Beide Gruppen hatten im Gegensatz zur konzentrierten Arbeitszeit eine sehr kurze Zeit, in der sie sich ablenkten. Sie lenkten sich meistens durch Witze über die Aufgabe oder allgemeine Scherze ab, die die Stimmung hoben. Extrem schlechte Stimmung kam bei keinem der Paare vor, allerdings gab es Stimmungshochs durch Witze und lustige Bemerkungen, die dazu führten, dass die Gruppe abgelenkt war und für kurze Zeit nicht mehr konzentriert arbeitete. Somit bestätigt sich auch die Überlegung, dass bei extrem guter oder schlechter Stimmung sich die Teilnehmer gegenseitig ablenken und nicht konzentriert arbeiten.

Außerdem bestätigt sich die Überlegung, dass derjenige der das Lösen der Aufgabe vorantreibt und eine eher führende Rolle einnimmt, die Fragen des Partners beantwortet. Ich konnte dies teilweise bei Dirk und Jan aus Gruppe 1 beobachten. Wenn Dirk die Driver-Rolle hatte und seine Idee implementierte, stellte Jan fragen, wenn ihm etwas unklar war, die Dirk dann beantwortete. Auch erklärte Dirk beim Implementieren seine Idee. Da bei der zweiten Gruppe der Rollenwechsel sehr ausgeglichen war, fragten sich beide gegenseitig. Allerdings war auch ab und an zu beobachten, dass der Observer nachfragte, was der Driver eingab oder änderte.

Durch den Vorversuch wurde deutlich, dass ich viel zu sehr darauf bedacht war alles statistisch zu erfassen. Mit Hilfe meiner Betreuerin gelang es mir den Fokus mehr auf die qualitative Beobachtung zu legen anstatt alles eher quantitativ zu messen, sodass die Untersuchung eher einem Experiment glich. Auch fiel mir auf, dass meine Beobachtungsvorlage noch nicht richtig durchdacht war. Den Fragebogen für die Gruppe aus dem Kurs "Softwareprozesse" erstellte ich ausschließlich für diese Gruppe und mir war schon vorher bewusst, dass der Fragebogen für den Kurs "Programmieren in Java" sich sehr unterscheiden würde. Außerdem wurde mir durch diesen Vorversuch bewusster was

ich beobachten wollte und vor allem, welche Dinge ich nicht oder eher schwer beobachten konnte. Bei der Auswertung fielen mir auch einige Beobachtungsgegenstände auf die nicht oder nur teilweise auswertbar waren. Einige wurden dann für die Fallstudie vom Blockkurs weggelassen.

5.6 Probleme und Besonderheiten

Ich empfand es als sehr schwer die Paare unauffällig zu beobachten und sie nicht zu beeinflussen. Mit beeinflussen ist nicht gemeint sich bei den Gesprächen oder dem Handeln der Paare einzumischen, sondern ihr Handeln durch meine Anwesenheit und dem Bewusstsein, dass ich sie beobachtete, nicht zu beeinflussen und damit zu ändern. Außerdem saß ich sehr nah an den Teilnehmern. Es war schwierig den richtigen Platz zum Beobachten zu finden. Hätte ich den Paaren gegenüber gesessen, hätte ich weder den Monitor gesehen, der für meine Beobachtungen nicht so sehr ausschlaggebend war, noch in manchen Situationen die Gesichter der Teilnehmer. Beispielsweise wenn sie die Köpfe nahe zum Bildschirm bewegten wenn sie nach einem Fehler suchten, oder weit zurück gelehnt waren. Ansonsten hätte ich eine gute Sicht auf den Blickkontakt und die Mimik der Paare gehabt. Meine Position war neben dem Paar, das hatte zur Folge, dass ich Sicht auf die Gestiken der Teilnehmer hatte und auf den Monitor. Allerdings war diese manchmal gestört. Bei einem Paar konnte ich zum Beispiel sehr gut auf den Monitor gucken und bei dem Anderen nicht, da der Partner meist den Driver hatte, der am weitesten von mir weg saß. Größtenteils konnte ich die Gesichter der Paare gut erkennen, jedoch kam es vor, dass ich von dem Partner der direkt neben mir saß das Gesicht nicht sehen konnte, wenn dieser sich zu seinem Partner wandte und mit ihm Sprach. Auch kam es vor, dass ich mich möglichst unauffällig nach vorn beugen musste, um die Gesichter zu erkennen, da derjenige der direkt neben mir saß mir dem Blick auf seinen Partner versperrte. Außerdem waren die Paare versucht auf meine Beobachtungsvorlagen zu schauen, da sie Interesse an dem Thema hatten und die Beobachtungsvorlage immer in Sichtweite lag. Dies beeinflusste die Paare allerdings weniger als ich annahm. Ich denke im Nachhinein wäre es besser gewesen, wenn ich den Paaren gegenüber gesessen hätte, da ich dann einen besseren Blick auf die Mimik der Paare gehabt hätte. Jedoch hätte man beobachten müssen, ob sich die Paare durch das direkte Beobachten gestört fühlen, da ich direkt vor ihnen gesessen hätte und man sich so beobachtet vorkommen kann. Als ich neben den Paaren saß, ihnen leicht zugewandt, bemerkte ich bei einem Teilnehmer sehr stark, dass er öfter in Richtung seines Partners schaute und von meiner Richtung eher fern hielt. Ich denke dies half demjenigen meine Anwesenheit zu vergessen und sich auf die Aufgabe zu konzentrieren.

Besondere Phänomene sind mir bei den beiden Paaren nicht aufgefallen, ich denke dazu hätte man sie noch einmal beobachten müssen. Bei Dirk und Jan fiel mir auf, dass diese die Tastatur und Maus nicht gleichverteilt wechselten und einer mehr die Driver-Rolle hatte. René und Daniel wechselten sich sehr gleichmäßig ab und schienen dabei sehr routiniert, auch wenn einer der beiden oft den Blickkontakt des Anderen nicht erwiderte und auf das Blatt schaute an dem er etwas erklärte.

6 Ergebnisse der Fallstudie (Blockkurs)

Bei den nachfolgenden Namen handelt es sich um frei erfundene Synonyme, welche keine Ähnlichkeit mit den realen Probanden aufweisen..

6.1 Analyse des Fragebogens

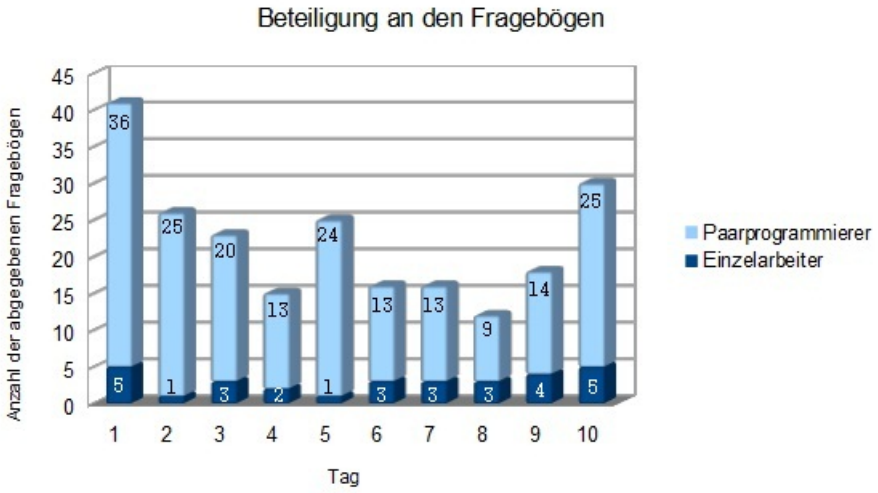
Ich möchte hier auf einige Fragestellungen aus dem Fragebogen näher eingehen und einen kleinen Einblick geben.

6.1.1 Beteiligung der Studenten an den Fragebögen

Die Beteiligung an dem Fragebogen war zu Beginn sehr groß und ließ dann sehr stark nach. Am vierten Tag versuchte ich die Studenten mit Hilfe von Schokoladenbonbons und einer kleinen Ansprache zum Ausfüllen des Fragebogens zu bewegen. Die Studenten schienen motiviert, was sich durch die Mimik und Gestik zeigte. Jedoch hatte dies nicht die erhoffte Wirkung. Anfang der letzten Woche ließ ich die Fragebögen an zwei Tagen weg, woraufhin sich die Studenten verwundert zeigten. Ich verteilte an diesen Tagen die Fragebögen nur an Studenten die ich intensiv beobachtet hatte und bat sie noch einmal eindringlich den Fragebogen auszufüllen. Am letzten Tag verteilte ich dann wieder an alle die Fragebögen und alle Anwesenden füllten diesen aus, obwohl er diesmal durch einige neu hinzugefügte Fragen länger war. Die Beteiligung wird in der nachfolgenden Abbildung (Abb.1) präsentiert.

Durch die Einführung und Ansagen die ich am ersten Tag in der Vorlesung machte füllten viele Studenten den Fragebogen aus ohne ihn zu hinterfragen. Die meisten Studenten gingen davon aus, dass sie den Fragebogen nur am ersten Tag ausfüllen müssen und fanden es unsinnig die gleichen Fragen jeden Tag zu beantworten. Dies fand ich durch Interviews und Gespräche mit den Teilnehmern in der zweiten Woche heraus. Auch konnte man dieses durch schnaufen und lauten Beschwerden an den Partner erkennen. Dies erklärt die insgesamt stark abfallende Kurve der Beteiligung (siehe Abb.1). Ich denke, wenn die Fragebögen Pflichtteil der Veranstaltung gewesen wären, hätten die Studenten eine höhere Motivation gehabt diese auszufüllen. Außerdem hing das Ausfüllen des Fragebogens stark von der Stimmung der Studenten ab, wie an den Freitagen, also Tag 5 und Tag 10 als die Stimmung sehr gut war, zu sehen ist. Gerade in der zweiten Woche nahm die Beteiligung stark ab, da die Studenten ihr Aufgabenblatt erst in letzter Minute abgaben oder länger blieben um es fertig zu stellen. Wenige wollten dann noch länger bleiben um den Fragebogen auszufüllen.

Abbildung 1: Überblick über die Beteiligung an den Fragebögen

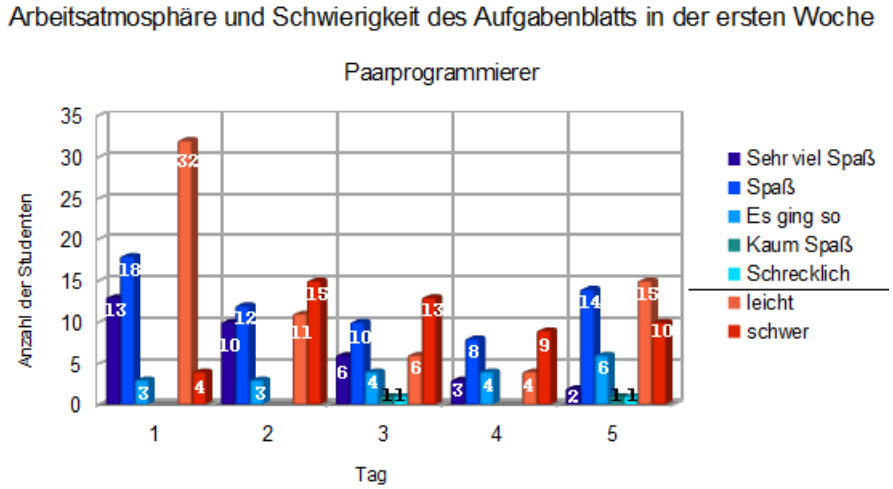


Wie man deutlich sieht nimmt die Anzahl der abgegebenen Fragebögen von Tag zu Tag stark ab. Am fünften Tag kann man einen Anstieg erkennen, dies hängt damit zusammen, dass an diesem Freitag gute Stimmung herrschte und es vorher von meiner Seite einen kleinen Motivationsschub in Form von Schokoladenbonbons gab. Auch sprach ich viele Studenten an, um sie nochmal aktiv zu bitten die Fragebögen auszufüllen. Auch am letzten Tag gibt es wieder einen deutlichen Anstieg, welcher mit der äußerst guten Stimmung zusammen hängt. Auch hier ging ich noch einmal herum und bat jeden Studenten mir den Fragebogen auszufüllen und abzugeben. Am siebten und achten Tag ist ein sehr starker Fall der Kurve zu beobachten, was daran liegt, dass ich mich kurzerhand auf Anraten meiner Betreuerin dazu entschieden habe die Fragebögen nur noch an Personen auszuteilen, die ich intensiv beobachtet und dabei Auffälligkeiten entdeckt habe. Auch dort bat ich jeden noch einmal eindringlich mir den Fragebogen auszufüllen. Am ersten Tag gaben 36 Paarprogrammierer und fünf Einzelarbeiter einen Fragebogen ab. Am letzten Tag hingegen gaben 25 Paarprogrammierer und fünf Einzelarbeiter den Fragebogen ab.

6.1.2 Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts

In den folgenden Abbildungen wurde die Arbeitsatmosphäre mit blauen Farben angegeben und die Schwierigkeit des Aufgabenblatts durch rote Farben dargestellt.

Abbildung 2: Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der ersten Woche bei den Paarprogrammierern

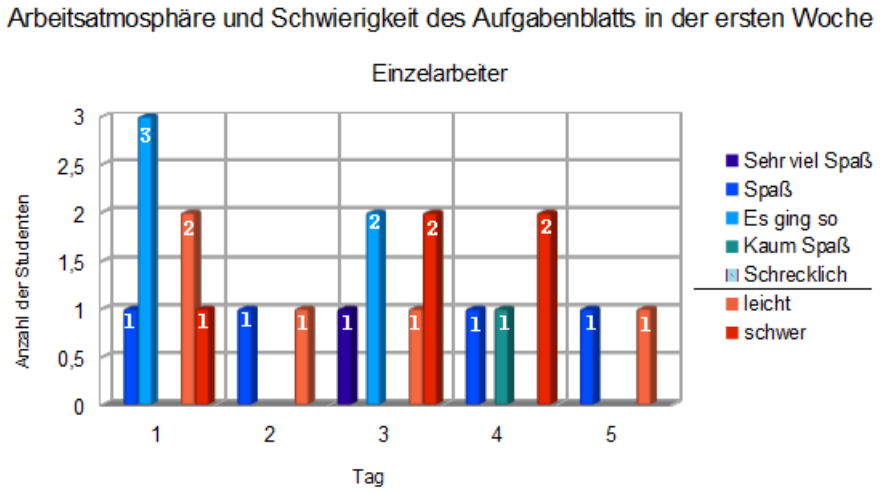


Hier in Abbildung 2 sehen wir ein Diagramm, was die Schwierigkeit des Aufgabenblatts und die Arbeitsatmosphäre in Zusammenhang stellt. Die Angaben sind von den Studenten im Fragebogen nach ihrem persönlichen Empfinden gemacht worden. In dem Diagramm sind der Übersicht wegen die ersten fünf Tage angeführt und es stellt nur die Sicht der Paarprogrammierer dar. Während 32 Paarprogrammierer das Übungsblatt am ersten Tag leicht fanden und 13 Studenten viel Spaß, sowie 18 Paarprogrammierer sehr viel Spaß an dem Lösen des Aufgabenblatts mit dem Partner hatten, fiel etwas mehr als der Hälfte der Paarprogrammierer am zweiten Tag das Lösen des Übungsblatts nicht mehr so leicht. Es nahmen nicht mehr so viele Studenten wie am ersten Tag teil, deshalb fallen die Blöcke im Allgemeinen stark ab. Allerdings blieb die Stimmung am zweiten Tag und auch am dritten und vierten Tag weiterhin gut bis sehr gut. Am dritten Tag fiel 13 Paarprogrammierern das Lösen des Übungsblatts schwer und sechs Paarprogrammierern fiel das Lösen des Aufgabenblatts leicht. Zum ersten Mal wurde jeweils einmal angekreuzt, dass es den Paaren kaum Spaß machte oder sie es sogar schrecklich fanden das Übungsblatt mit dem Partner zu lösen. Trotzdem machte es 10 Paarprogrammierern immer noch Spaß und sechs Paarprogrammierern sehr viel Spaß das Übungsblatt zu lösen. Am vierten Tag fiel es den meisten Studenten wieder schwer das Übungsblatt zu lösen, aber es gab immer noch die Mehrheit an, Spaß am Lösen des Blatts zu haben. Niemand gab an, dass er es schrecklich fand oder kaum Spaß beim Lösen des Aufgabenblatts hatte. Am fünften Tag fiel wieder mehreren Studenten das Lösen leichter und weniger gaben an sehr viel Spaß dabei gehabt zu haben. Jedoch gab ein Großteil von 14 Studenten an, dass sie Spaß dabei hatten.

Es ist zu erkennen, dass die Studenten mehr Spaß beim Lösen des Übungsblatts haben, wenn ihnen das Aufgabenblatt leichter fällt. Allerdings herrschte bei Paarprogrammierern auch bei schwierigen Aufgabenblättern immer noch überwiegend gute Stimmung. Dies kann damit zusammenhängen, dass sie sich auch untereinander viel absprechen konnten und Ideen austauschen konnten. Außerdem konnten sie den Tutor fragen. In Abbildung 3 ist zu erkennen, dass Einzelarbeiter weniger Spaß beim Lösen des Aufgabenblatts hatten. Ich denke der Unterschied wäre noch besser zu sehen gewesen, wenn weder die Paarprogrammierer noch die Einzelarbeiter einen Tutor zum Fragen gehabt hätten.

Da außerdem der Fragebogen immer im Anschluss an die Übung ausgefüllt wurde, kann man erwarten, dass die Studenten die Stimmung im Nachhinein als besser beurteilen als sie während des Lösens war. Sobald die Studenten die Aufgaben erfolgreich gelöst hatten waren sie deutlich zufriedener. Dies teilten sie mir mit oder war durch lauten Jubel im Raum vernehmbar. Meines Erachtens wirkt sich dies stark auf die Beantwortung der Fragen aus und es wäre besser gewesen vereinzelt Studenten immer wieder zwischen den einzelnen Aufgaben nach ihrer Stimmung zu fragen. Alle Paarprogrammierer lösten die Aufgabenblätter in den ersten drei Tagen komplett und bekamen von den Tutoren volle Punktzahl. Am vierten Tag fehlten bei einigen Studenten Aufgabenteile und sie bekamen nicht die volle Punktzahl. Am fünften Tag wurden wieder alle Aufgabenblätter vollständig gelöst. Somit ist erkennbar, dass die Studenten das Blatt zwar oft als schwierig empfanden, sie es dennoch lösen konnten. Auch ist daran zu erkennen, dass die Studenten das Aufgabenblatt nicht wegen der dort vorherrschenden Stimmung als leicht empfanden, was die These untermauert, dass die Studenten trotz der für sie schweren Aufgabenblätter eine gute Arbeitsatmosphäre hatten.

Abbildung 3: Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der ersten Woche bei den Einzelarbeitern



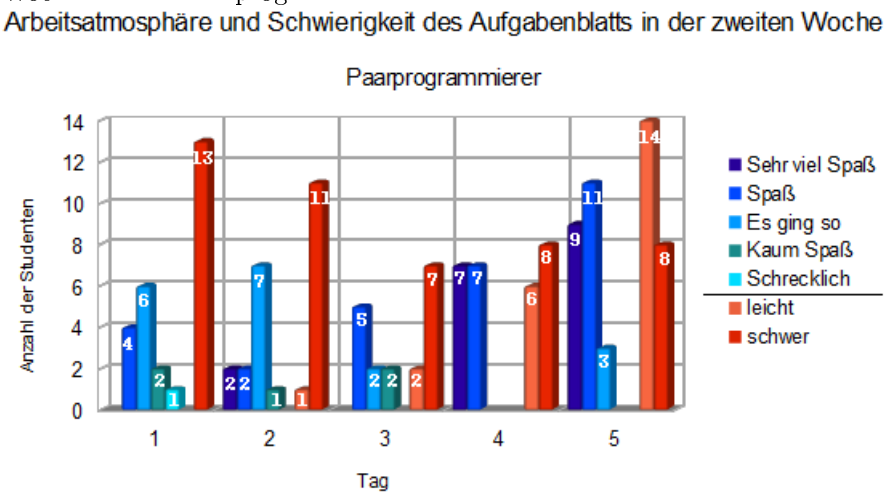
Im Gegensatz zu den Paarprogrammierern (siehe Abb. 2) hatten am ersten Tag deutlich weniger Studenten sehr viel Spaß und Spaß beim Lösen des Aufgabenblatts. Drei Einzelarbeiter gaben an, dass es gerade so ging, also sie weder Spaß noch keinen Spaß hatten. Wie die Paarprogrammierer fanden zwei Einzelarbeiter das Aufgabenblatt am ersten Tag eher leicht. Am zweiten Tag wurden nur Angaben gemacht, die aussagten, dass das Übungsblatt leicht sei und der Einzelarbeiter Spaß hatte. Auch hier nahm die Teilnahme an den Fragebögen deutlich ab. Am dritten Tag gab zum ersten Mal ein Student an, dass er sehr viel Spaß beim Lösen des Aufgabenblatts hatte. Das Übungsblatt fiel zwei Studenten schwer und sie hatten weder Spaß noch keinen Spaß. Am vierten Tag fanden alle Einzelarbeiter das Übungsblatt schwer und gaben sogar an, dass sie kaum Spaß hatten. Am fünften Tag gab ein Einzelarbeiter an, dass er das Übungsblatt leicht fand und Spaß dabei hatte.

In Abbildung 3 sieht man, dass den Studenten das Lösen des Übungsblatts mehr Spaß macht, wenn sie das Aufgabenblatt leichter finden. Allerdings wurde im Gegensatz zu den Paarprogrammie-

rern nur einmal angegeben, dass ein Student sehr viel Spaß beim Lösen des Aufgabenblatts hatte. Am Häufigsten wurde angegeben, dass sie weder Spaß noch keinen Spaß gehabt hatten. Gerade in den ersten drei Tagen als diese Angaben gemacht wurden waren viele Einzelarbeiter noch nicht in die Gruppe ihres Raumes integriert und blieben sehr für sich. Ich denke dies trägt zu der nicht so guten Arbeitsatmosphäre bei. Alle außer ein Einzelarbeiter gaben in den Fragebögen auch an, dass zum Lösen der Probleme ein Partner geholfen hätte. Auch dies unterstützt die Vermutung, dass die Einzelarbeiter weniger Spaß als die Paarprogrammierer hatten, da ihnen die Rücksprache und der Ideenaustausch mit einem Partner fehlte. Auch bei den Einzelarbeitern wurden alle Aufgaben der Aufgabenblätter gelöst bis auf das Aufgabenblatt vom vierten Tag, wo nicht alle Aufgabenteile von einem Einzelarbeiter gelöst wurden. Auch dies zeigt, dass den Studenten das Lösen des Aufgabenblatts tatsächlich leicht fiel und sie es nicht durch die gute Stimmung als leicht empfanden. Am ersten Tag sieht man, dass die Arbeitsatmosphäre nur einigermaßen gut war, obwohl die Mehrheit das Aufgabenblatt leicht fand und nur einer das Aufgabenblatt schwer fand. Als am vierten Tag beide Einzelarbeiter angaben das Aufgabenblatt schwer zu finden, hatte ein Einzelarbeiter Spaß und der Andere im Gegensatz kaum Spaß. Der Einzelarbeiter der Angab keinen Spaß gehabt zu haben hatte große Schwierigkeiten mit dem Aufgabenblatt und wollte von Anfang an nicht allein arbeiten. Er war deshalb sehr unzufrieden was sich durch den Fragebogen und einem Gespräch bestätigen lässt. Der Einzelarbeiter der angab, dass er Spaß hatte, arbeitete in einem Raum in dem immer eine sehr lustige Arbeitsatmosphäre war und alle in die Gruppe integriert waren. In diesem Raum halfen sich alle untereinander und kommunizierten sehr viel miteinander. Es handelte sich hierbei oft um Privatgespräche und Scherze, jedoch auch um Diskussionen um die Aufgaben.

Im Nachfolgenden sehen wir Abbildung 4. Zu beachten ist, dass das Diagramm einen anderen Maßstab hat, denn die Antworten waren in der zweiten Woche sehr verteilt. So sagten am letzten Tag 14 Studenten, dass sie das Aufgabenblatt als leicht empfunden haben und 8 Studenten fanden das Übungsblatt schwer. Am ersten Tag aus Abbildung 2 oben geht hervor, dass 32 Studenten die Paarprogrammierung ausprobierten das Übungsblatt leicht fanden und etwa 4 Studenten das Aufgabenblatt schwer fanden. Die Angaben des dritten und vierten Tags der zweiten Woche sind nicht Aussagekräftig, da ich an diesen Tagen die Fragebögen nur an Studenten austeilte die ich intensiv beobachtet hatte.

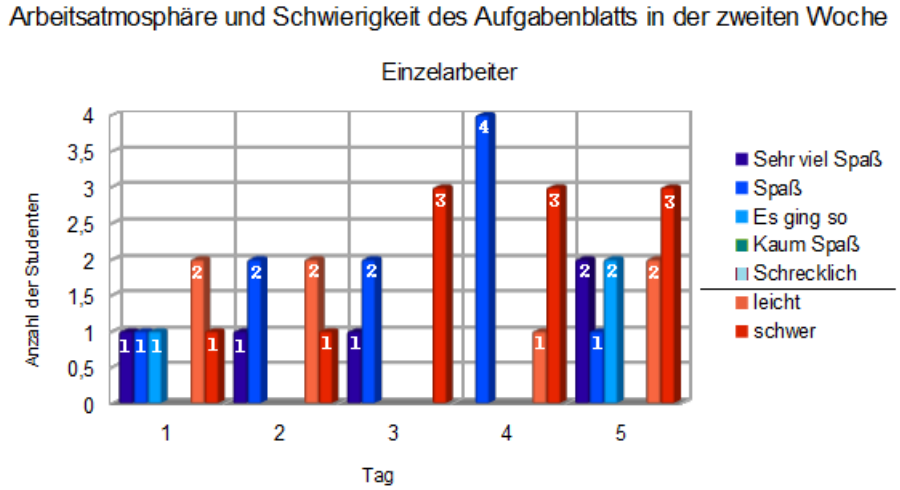
Abbildung 4: Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der zweiten Woche bei den Paarprogrammierern



Am Montag, dem ersten Tag der zweiten Woche, war die Stimmung bei den Paarprogrammierern nicht sehr gut. Sechs Studenten gaben an, dass es ihnen weder Spaß noch keinen Spaß gemacht hat und alle fanden das Übungsblatt schwer. Auch wurden mehr Angaben gemacht die besagen, dass die Teilnehmer kaum Spaß hatten und es teilweise schrecklich fanden. Am zweiten Tag gibt es wieder einige Meldungen die besagen, dass es den Studenten Spaß bis sehr viel Spaß gemacht hat, jedoch überwiegt die Aussage, dass es ihnen weder Spaß noch keinen Spaß gemacht hat. Weiterhin fanden 13 Paarprogrammierer das Übungsblatt schwer, was sich erst gegen Ende der Woche ändert. Am vorletzten Tag gibt zwar immer noch die Mehrheit von 8 Paarprogrammierern an das Aufgabenblatt sei schwer, jedoch geben sechs Paarprogrammierer an das Aufgabenblatt leicht zu finden. Am letzten Tag fand die Mehrheit von 14 Paarprogrammierern das Übungsblatt leicht. Während auch am dritten Tag die Stimmung weniger gut ist, steigt die Stimmung am vierten Tag und am letzten Tag stark an. Am vierten Tag geben jeweils sieben Studenten an, dass sie ausschließlich Spaß bis sehr viel Spaß hatten. Am Freitag steigt die Kurve durch die rege Teilnahme stark an und auch hier haben neun Studenten Spaß, sowie 11 sehr viel Spaß.

Es ist auch in der zweiten Woche zu erkennen, dass die Studenten mehr Spaß hatten, wenn sie den Eindruck hatten, dass Übungsblatt sei leicht. Allerdings wird in dieser Woche nur einmal die Angabe gemacht, dass die Studenten das Programmieren mit ihrem Partner als schrecklich empfanden. Am ersten und zweiten Tag der zweiten Woche empfanden jeweils 13 und 11 Paarprogrammierer das Aufgabenblatt als schwer. Dies bestätigten auch die Punkte der Aufgabenblätter, da einige Studenten nicht alle Aufgaben lösen konnten. An diesen Tagen gab es auch jeweils sechs und sieben Studenten die ankreuzten weder Spaß noch keinen Spaß gehabt zu haben. Ich denke wenn die Paare einzeln gearbeitet hätten, wäre die Arbeitsatmosphäre noch schlechter gewesen, da sie sich nicht mit ihrem Partner austauschen hätten können.

Abbildung 5: Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der zweiten Woche bei den Einzelarbeitern



Bei den Einzelarbeitern ist die Stimmung am ersten Tag der zweiten Woche ziemlich ausgeglichen und im Gegensatz zu den Paarprogrammierern findet die Mehrzahl, also zwei der Einzelarbeiter das Übungsblatt leicht. Dies ändert sich erst am dritten Tag, dort finden alle Einzelarbeiter das Übungsblatt schwer. Auch in den nachfolgenden Tagen findet die Mehrzahl, also drei Einzelarbeiter, das Aufgabenblatt schwer. Auch sagen alle aus, dass sie Spaß beim Lösen des Übungsblatts hatten. Am letzten Tag sind die Aussagen wieder gemischt und es gibt jeweils zwei Studenten die sehr viel Spaß hatten und weder Spaß noch keinen Spaß hatten.

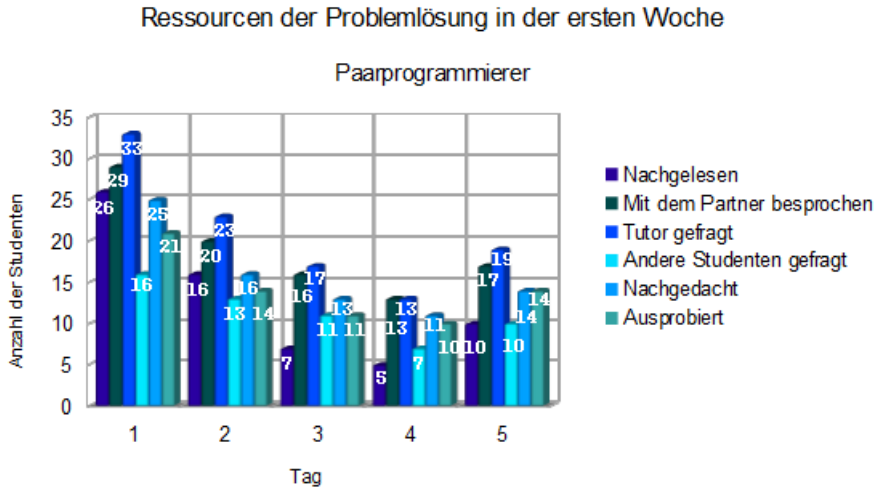
Erstaunlicherweise scheint hier die Wahrnehmung der Studenten mit den Punkten des Aufgabenblatts nicht übereinzustimmen. Während am ersten und zweiten Tag der Woche zwei von drei Einzelarbeitern angeben das Übungsblatt leicht zu finden zeigen die Punkte der Aufgabenblätter, dass es mehr als zwei Einzelarbeitern nicht gelang das komplette Aufgabenblatt zu lösen. Einerseits kann hier die Wahrnehmung der Studenten das Ergebnis verfälschen. Hauptsächlich kommt dieses verfälschte Ergebnis allerdings dadurch zu Stande, dass nicht alle fünf Einzelarbeiter den Fragebogen abgegeben haben. Diejenigen die den Fragebogen abgaben und dort angaben, dass ihnen das Lösen des Blatts leicht fiel, waren diejenigen die alle Aufgaben lösten. Aus diesem Grund stellt sich in Abbildung 5 auch die Arbeitsatmosphäre in den ersten beiden Tagen so positiv dar. Wir sehen hier also, dass durch die geringe Teilnahme an Einzelarbeitern dieses Ergebnis stark verfälscht wurde. Hätten mehr Studenten einzeln gearbeitet würden wir hier sicherlich einen besseren Mittelwert und damit Überblick bekommen, so wie es bei den Paarprogrammierern der Fall ist.

6.1.3 Ressourcen die von den Studenten zur Problemlösung verwendet wurden

Ebenso in dem Fragebogen abgefragt wurden die Ressourcen die die Studenten einsetzten um ein vorhandenes Problem zu lösen. Die Studenten konnten die vorgegebenen Antwortmöglichkeiten (nachgelesen, mit dem Partner besprochen, Tutor gefragt, andere Studenten gefragt, nachgedacht und ausprobiert) ankreuzen und in einer Zeile, die leer war, einen Vorschlag abgeben, bzw. etwas ange-

ben, was dort nicht aufgeführt war. Sie konnten dort mehrere Antwortmöglichkeiten ankreuzen. Die Diagramme habe ich wieder zur Übersicht in zwei Wochen unterteilt. Wir sehen zuerst die Angaben der Studenten aus der ersten Woche und anschließend die Angaben aus der zweiten Woche. Im Allgemeinen wurde als häufigst verwendete Ressource der Tutor sowohl von Paarprogrammierern, als auch von Einzelarbeitern genannt. Dies ist kaum verwunderlich, da der Kurs daraufhin ausgelegt war, das der Tutor aktiv während der Übungsstunde hilft. Pro Tutor gab es jeweils eine relativ kleine Gruppe von Studenten, die dieser betreute. Manche Tutoren kamen auf Zuruf, jedoch gab es einige, die stetig herumgingen und die Studenten fragten, ob sie Hilfe brauchen. Dadurch nahmen die Studenten häufig die Hilfe des Tutors in Anspruch. Interessant war zu beobachten, dass wenn der Tutor keine Zeit hatte um den Studenten zu helfen, sie sich entweder untereinander im Team halfen oder auch andere Studenten in ihrer Nähe fragten.

Abbildung 6: Verwendete Ressourcen der Paarprogrammierer zur Problemlösung in der ersten Woche

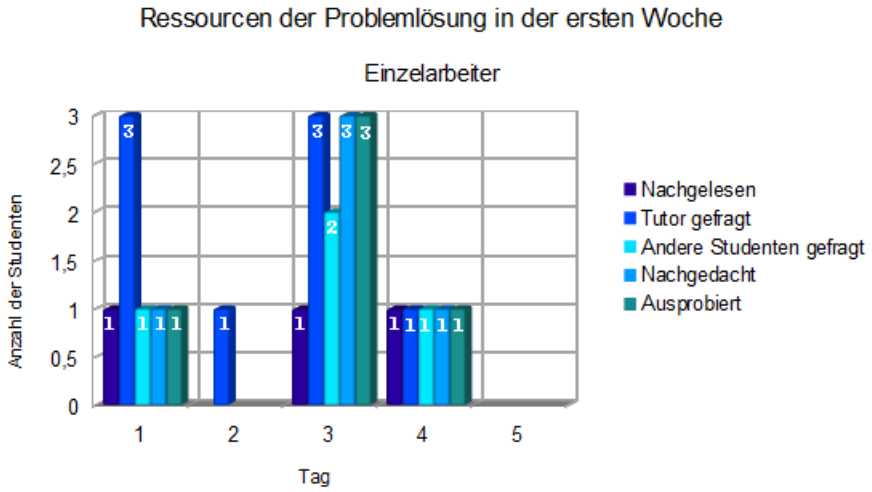


Man sieht, dass zu Anfang noch viel nachgelesen wurde und dies im Laufe der Woche nachließ. Am häufigsten wurde die ganze Woche über die Hilfe des Tutors in Anspruch genommen. Außerdem wurde nachgedacht und auch ausprobiert. Am ersten und zweiten Tag wurde im Vergleich zu den anderen Ressourcen am seltensten die Hilfe anderer Studenten in Anspruch genommen. Auffällig ist auch, dass sich die Meisten mit dem Partner besprochen haben.

Da beispielsweise auf dem ersten Aufgabenblatt einige Aufgaben waren die die Studenten an Java heranführen sollten, sowie Werte berechnet werden und entsprechende Typen ohne Verwendung eines Rechners zugeordnet werden sollten, lasen viele Studenten in den Folien noch einmal genauer nach. Auch sollten Programme aus der Vorlesung verändert werden oder aus den Folien kopiert und bearbeitet werden. Da für einen Großteil der Studenten die Programmiersprache Java neu war, mussten sie sich zu Beginn belesen. Dies ließ dann nach als sie etwas geübter waren, fremden Code zu lesen und selbst zu programmieren. Der Tutor wurde sehr häufig nach Hilfe gefragt, da die Tutoren immer vor Ort waren und teilweise die Studenten fragten, ob sie Hilfe brauchten. Da es kein Frontalunterricht war und die Tutoren sehr freundschaftlich mit den Studenten umgingen, fiel es den

Teilnehmern leichter die Tutoren um Hilfe zu bitten. Außerdem waren die Übungsblätter zu Anfang eher darauf ausgerichtet, dass die Studenten etwas ausprobierten. Es wurde sich weitaus häufiger mit dem Partner besprochen als andere Studenten zu fragen. Gerade gegen Ende der ersten Woche nimmt die Kurve stark ab und die Paarprogrammierer fragten immer seltener andere Studenten. Es liegt weitaus näher den eigenen Partner etwas zu fragen als auf fremde Hilfe zurück zu greifen, da der Partner den Code kennt und sich nicht in die Problemlösung einarbeiten muss. Ich denke, solange der Partner helfen konnte fragten die Studenten ihre Nachbarn selten. Leider kann in dieser Arbeit keine Aussage dazu getroffen werden, dass Paarprogrammierer den Tutor seltener nach Hilfe fragen, da beide Gruppen den Tutor ähnlich häufig fragten und es sehr auf die verschiedenen Persönlichkeiten, sowie stärken und Schwächen der Paare und Einzelarbeiter ankommt.

Abbildung 7: Verwendete Ressourcen der Einzelarbeiter zur Problemlösung in der ersten Woche

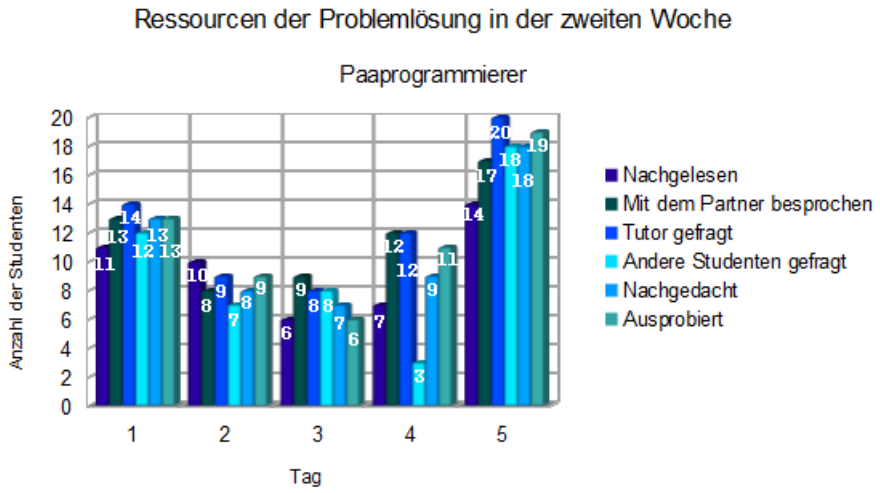


Auch die Einzelarbeiter fragten am häufigsten den Tutor um Hilfe. Während die Verwendung aller anderen Ressourcen die Woche über sehr gleichverteilt scheint, ist am dritten Tag zu sehen, dass eher nachgedacht und ausprobiert wurde, als andere Studenten zu fragen. Am fünften Tag gab nur ein Einzelarbeiter seinen Fragebogen ab und machte zu dieser Frage keine Angaben. Am zweiten Tag sehen wir, dass der Einzelarbeiter angab ausschließlich den Tutor gefragt zu haben.

Hier kann man nur sehr wenig Rückschlüsse ziehen da meiner Meinung nach zu wenige Einzelarbeiter ihren Fragebogen ausgefüllt haben. Ich denke es wäre von Vorteil gewesen meine anfängliche Idee umzusetzen und aus den Teilnehmern zwei gleich große Gruppen zu bilden. So hätte ich ein breiteres Spektrum an Meinungen von Einzelarbeitern bekommen können.

Die Angaben des dritten und vierten Tags der zweiten Woche sind nicht Aussagekräftig, da ich an diesen Tagen die Fragebögen nur an Studenten austeilte die ich intensiv beobachtet hatte.

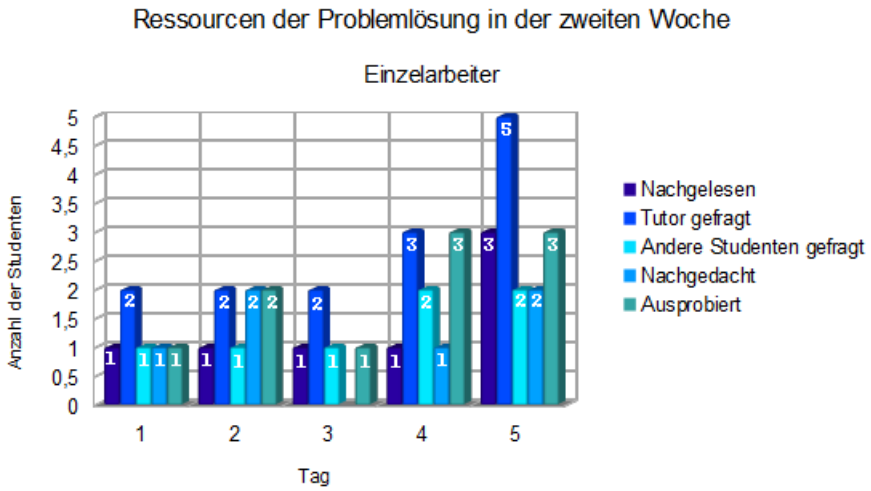
Abbildung 8: Verwendete Ressourcen der Paarprogrammierer zur Problemlösung in der zweiten Woche



Auch hier sehen wir in der zweiten Woche, dass die Teilnahme an dem Fragebogen deutlich zurückgegangen ist. Am ersten Tag der zweiten Woche wurde wieder am häufigsten der Tutor gefragt. Stark angestiegen sind aber auch das nachdenken und ausprobieren. Die Paarprogrammierer fragten eher ihren Partner, als andere Studenten. Am zweiten Tag lasen die Studenten mehr nach, als das sie den Tutor fragten. Wieder fragten sie eher ihren Partner als andere Studenten, wobei das ausprobieren einen genauso hohen Stellenwert hat wie den Tutor um Hilfe zu bitten. Am dritten Tag besprachen sich die Paare eher untereinander, als das sie den Tutor fragten oder andere Studenten um Hilfe baten. Am vierten Tag besprachen sich die Paare untereinander und fragten aber auch genauso häufig den Tutor. Das Ausprobieren stieg am vierten Tag auch wieder an. Am letzten Tag fragten die Studenten den Tutor oft und probierten viel aus.

Dass, das Nachdenken und Ausprobieren stark anstieg könnte gerade am ersten Tag der zweiten Woche mit der Schwierigkeit des Aufgabenblatts zusammenhängen. Im Allgemeinen wurden die Aufgabenblätter in der zweiten Woche schwieriger und auch durch Interviews wurde mir bestätigt, dass die Studenten viel eher etwas ausprobierten wenn sie nicht mehr weiterkamen. Ich denke, dass das Ausprobieren am letzten Tag anstieg resultiert aus der Aufgabenstellung, da am letzten Tag Threads behandelt wurden. Dort traten verschiedene Probleme auf, die teilweise durch Ausprobieren beseitigt werden und teilweise durch den Tutor gelöst werden konnten.

Abbildung 9: Verwendete Ressourcen der Einzelarbeiter zur Problemlösung in der zweiten Woche



Die Einzelarbeiter am ersten Tag der zweiten Woche fragten den Tutor sehr häufig. Am zweiten Tag probierten sie Dinge aus, dachten viel nach und fragten ebenso den Tutor. An den folgenden Tagen überwiegt wieder eindeutig das Fragen des Tutors. Allerdings ist auch zu erkennen, dass das Ausprobieren stark angestiegen ist.

Da auch die Einzelarbeiter oft nicht weiter wussten und mit den Aufgaben einige Schwierigkeiten hatten, probierten sie viel aus. Weiterhin fragten sie eher den Tutor als mit anderen Studenten über die Aufgaben zu diskutieren. Dies könnte damit zusammenhängen, dass die Studenten im Allgemeinen in der zweiten Woche viel zu tun hatten und nicht schnell voran kamen, sodass es effektiver war den Tutor zu fragen. Durch die Komplexität der Aufgabenstellungen wurden die Lösungen immer individueller und konnte durch verschiedene Ansätze nun nicht mehr so leicht übernommen oder übertragen werden.

6.1.4 Angaben in den Fragebögen zu Eigenschaften von Paarprogrammierung

Am ersten Tag schrieben sehr viele Paarprogrammierer die von Ihnen empfundenen positiven und negativen Eigenschaften von Paarprogrammierung auf. Die Paarprogrammierer fanden 13 positive Dinge und acht negative Dinge, die sie notierten.

Tabelle 6: Angaben von Paarprogrammierern zu positiven Eigenschaften von Paarprogrammierung am ersten Tag

Niklas	"Durch die Zusammenarbeit bin ich schneller auf die die Lösung gekommen, hab aber nicht das Gefühl teilweise die Lösungen meines Partners nur zu übernehmen, sondern es auch zu verstehen."
Lisa	"Alleine programmieren ist total anstrengend und nervenaufreibend, weil man seine Fehler nicht sieht."
Finn	"Schnellere Arbeit, mehr Spaß, einfacher."
Mylos	"Schneller, mehr Ideen."
Alesandro	"Ich bekomme Hilfe."
Martin	"Partner findet Fehler."
Angélique	"Habs verstanden."
Emma	"Fachlicher Austausch."

Tabelle 7: Angaben von Paarprogrammierern zu negativen Eigenschaften von Paarprogrammierung am ersten Tag

Kai	“Wissensunterschied.”
Emma	“Meinungskonflikte -> kostet Zeit.”
Martin	“Nicht so schnell.”
Alesandro	“Partner ist zu schnell.”
Richard	“Kann zum Nichtstun verleitet werden.”
Nina	“Man schreibt nicht alles selbst.”
Sophie	“Wenn der Andere tippt, verinnerlicht man die Eingaben nicht so sehr.”

Oft wurde als positive Eigenschaft von Paarprogrammierung der Spaß genannt. Als negativ wurde zwar genauso angeführt, dass es Meinungsverschiedenheiten geben kann, diese schienen allerdings nicht zu überwiegen. Bei meinen Beobachtungen konnte ich einige Meinungskonflikte feststellen, trotzdem gaben die Studenten an diesem Tag im Fragebogen jedoch an, dass sie Spaß beim Bearbeiten der Aufgabenblätter hatten. Auch wurde der Ideenaustausch erwähnt der sich positiv auf das Bearbeiten der Aufgabenblätter auswirkt und auch auf den Lerneffekt. Ist allerdings der Wissensunterschied zu groß kann sich dies negativ auf die Zusammenarbeit auswirken. Außerdem kann man sehen, dass als positives Argument die Schnelligkeit von Paarprogrammierung angeführt wird und von anderen Studenten diese wiederum bemängelt wird. Auch wurde als negativ angeführt, dass der Partner zu schnell sei. Hier geht die subjektive Wahrnehmung der Studenten sehr auseinander.

Ansonsten gaben die Studenten die Paarprogrammierung ausprobiert haben auch an, dass sie mehr geschafft hätten als zuvor, sie konstruktiv zusammenarbeiten würden und sich ergänzen. Sie halfen sich außerdem gegenseitig, hatten Lernerfolge und Spaß.

Tabelle 8: Angaben von Paarprogrammierern zu positiven Eigenschaften von Paarprogrammierung zwischen dem ersten und dem letzten Tag

Martin	"Lösungen auf die man nicht allein gekommen wäre."
Henrik	"Wenn beide etwa gleich viel Vorkenntnisse haben, ist es ziemlich gut (schneller etc.)."
Saskia	"Wenn man eine Aufgabe erfolgreich gemeinsam gelöst hat ist die Freude umso größer, wenn man sie teilen kann. Man kann sich gegenseitig helfen."
Nicole	"Teamwork ist super."
Richard	"Mehr Spaß."
Finn	"Sehr angenehme Atmosphäre. Lösen geht ziemlich schnell."
Sophie	"Wir ergänzen uns gut."
Noah	"Man kann sich gut bequatschen."
Mylos	"Es geht schneller!"
Angélique	"Es ging schneller und verstand Sachen besser."
Sven	"Syntaxfehler fallen wesentlich schneller auf."
Nina	"Fehlerkorrektur."
Malte	"Falsches Wissen wurde korrigiert, schnelleres Lösen der Aufgaben und neue Ideen."
Nina und Mia	"Man kann zwischendurch mal Pause machen, während der Partner arbeitet."

Tabelle 9: Angaben von Paarprogrammierern zu negativen Eigenschaften von Paarprogrammierung zwischen dem ersten und dem letzten Tag

Martin	"Gegenseitige Behinderung."
Lara	"Ermüdung."
Henrik	"Unterschiedliche Vorkenntnisse führen zu langsamerem Lösen der Übungen."
Sophie	"Er denkt schneller als ich und schreibt. Dadurch kann ich super Text checken, aber nicht selbst schreiben."
Mylos	"Bei manchen Themen wird gestritten."
Angélique	"Uneinigkeiten."
Paul	"Manchmal Unstimmigkeiten.""
Emma	"Je mehr Aufgaben und je schwieriger, desto höher ist das Konfliktpotenzial."
Max	"Manchmal löst der Partner ein Problem, ohne das man selbst sich lang genug selbst damit beschäftigt hat."
Nina	"Man schreibt nicht jedes Programm selbst."
Lara	"Man hat verschiedene Lösungsansätze, aber auch verschiedene Verständnisprobleme."
Patrick	"Man relaxed manchmal zu viel."

Im Laufe der Zeit von zwei Wochen haben sich die Vor- und Nachteile die die Studenten bei der Anwendung von Paarprogrammierung empfanden mehr herauskristallisiert. Weitaus öfter wurde nun als positiv angegeben, dass Paarprogrammierung schneller ist. Paare die einen großen Wissensunterschied hatten bemängelten teilweise, dass das Lösen der Aufgabenblätter im Paar länger dauert. Diese Angaben machte meist derjenige vom Paar der mehr Erfahrungen im Programmieren hatte. Ich denke oft entstand der Eindruck, dass es länger dauerte, weil derjenige seinem Partner viel erklären musste. In einem Interview erfuhr ich, dass jemand der Meinung war, dass allein Programmieren schneller ginge, da man dort nichts erklären müsste. Dies kann jedoch ein Trugschluss sein, da das Erklären zur Lösungsfindung beiträgt und der Partner auch Ideen einbringen kann, auf die man allein eventuell nicht so schnell gekommen wäre. Die Zeit die für das Bearbeiten der Aufgabenblätter gemessen wurde zeigt, dass es keine erheblichen Zeitunterschiede zwischen Paarprogrammierern und Einzelarbeitern gibt. Auch hier wäre es sicherlich hilfreich gewesen mehrere Einzelarbeiter untersuchen zu können. Der Großteil der Paarprogrammierer hat allerdings das Lösen des Übungsblatt im Paar als

schneller empfunden. Durch gute Stimmung und kleinere Ablenkungen zwischendurch erschien ihnen die benötigte Zeit kürzer. Außerdem wurde ebenso häufig angeführt, dass Paarprogrammierung Spaß macht. Dieser Spaß fördert auch die Leistung der Studenten, da sie motivierter sind. Von einer Paarprogrammiererin hörte ich in einem Interview, dass sie ohne ihren Partner allein nicht so weit gekommen wäre und früher aufgegeben hätte. Dies zeigt, dass sie die Paare auch gegenseitig motivieren und durch Ideen weiter voran bringen. Auch das schnelle Finden von Fehlern und sogar das Korrigieren von falschem Wissen durch den Partner wurde von den Studenten als positiv erkannt. Auch gaben zwei Paarprogrammiererinnen an, dass sie es gut finden ab und zu Pause machen zu können, während der Partner programmiert. Dies kann einerseits als positiv aufgefasst werden, geht allerdings meiner Meinung nach eher ins Negative, da der Partner nicht ganz bei der Sache ist und somit der Lerneffekt geringer wird. Dies sahen auch andere Paarprogrammierer ähnlich, wie man an der Negativliste erkennen kann. Einerseits ist der Ideenaustausch sehr von Vorteil. Beim Lösen der Übungsblätter hingegen kann sich dies negativ auswirken. Oft muss man selber einige Zeit über ein Problem nachgedacht haben um auf eine Lösung zu kommen. Wenn man diese vom Partner vorgesagt bekommt und dies vielleicht öfter passiert, wird die Lösungsfindung nicht geschult. Man kann dies in der Informatik mit einem mathematischem Beweis vergleichen. Wenn man die Lösung des Induktionsbeweis sieht versteht man sie, allerdings braucht man einiges mehr um auf diese zu kommen. Als negativ wurde oft das vorhandene Konfliktpotential gesehen. Man könnte hier näher untersuchen ob dieses tatsächlich so negativ ist und die Qualität des Codes sogar verbessert. Außerdem fördert es den Umgang mit anderen Menschen, da man auch oft nicht ein Programm allein schreiben wird, müssen Absprachen getroffen werden. Dies wird hierdurch meiner Meinung nach gefördert.

Oft waren die Studenten, die Paarprogrammierung ausprobierten mit ihren Lösungen zufrieden, wenn sie die Aufgaben richtig und vollständig gelöst hatten. Wurden sie hingegen nicht fertig, waren sie meist mit der Lösung der Aufgaben unzufrieden.

Tabelle 10: Angaben von Paarprogrammierern zu positiven Eigenschaften von Paarprogrammierung am letzten Tag

Mylos	"Schneller, effizienter, besser."
Nina	"Hohe Motivation/Spaß."
Lara	"Man kann selbst seinen Partner fragen, wenn man es nicht verstanden hat und kann sich auch selbst testen beim Erklären."

Tabelle 11: Angaben von Paarprogrammierern zu negativen Eigenschaften von Paarprogrammierung am letzten Tag

Mylos	"Man streitet auch mal."
Nina	"Zum Teil geringeres Arbeitstempo. Ablenkung. Ich denke, dass es selten der Fall ist, dass beide Partner gleich viel lernen, ich denke einer wird immer die aktivere Kraft sein."

Es wurde beispielsweise nicht einmal angeführt, dass die Paarprogrammierer den Eindruck hatten die Codequalität hätte sich gesteigert. Dies hängt meiner Meinung nach damit zusammen, dass es sich hier um Mathematik-Studenten handelt die das Programmierern größtenteils in diesem Kurs lernten. Ansonsten ergibt sich aus den beiden Tabellen 10 und 11 des letzten Tages dieselbe Schlussfolgerung wie zu den Tabellen 8 und 9. Hauptsächlich nannten die Studenten den Spaß, die hohe Motivation und der gegenseitige Ideenaustausch als positive Eigenschaft die sie bei der Paarprogrammierung erfahren haben. Am letzten Tag wird zusätzlich noch bemängelt, dass beide Partner nicht gleich viel lernen und durch Ablenkung das Arbeitstempo geringer werden kann. Ich denke, dass die Rollenverteilung an den unterschiedlichen Persönlichkeiten liegt. Auch dies könnte in einer weiteren Arbeit genauer untersucht werden.

6.1.5 Angaben in den Fragebögen zu Eigenschaften vom einzeln arbeiten

Am ersten Tag wurden bei den Einzelarbeitern nur negative Eigenschaften vom einzeln arbeiten angegeben, obwohl einer bei der Frage warum er sich dazu entschieden hat Paarprogrammierung nicht auszuprobieren angab, dass er so eine Zeitersparnis hätte.

Tabelle 12: Angaben von Einzelarbeitern zu negativen Eigenschaften vom einzeln arbeiten am ersten Tag

Paul	"Häufig nicht weiter kommen."
René	"Wenn man bei einer Aufgabe nicht weiter weiß -> Stagnation."

Hier spiegeln die negativen Eindrücke der Einzelarbeiter die aus dem Fragebogen angegebene Antwort wieder, dass ein Partner beim Lösen der Probleme des Aufgabenblatts geholfen hätte. Trotz allem hat sich zumindest einer freiwillig gegen das Konzept Paarprogrammierung entschieden. Dies ist verwunderlich, da diese Person noch keinerlei Erfahrungen im Programmieren zuvor sammeln konnte und auch später noch starke Schwierigkeiten beim Lösen des Aufgabenblatts hatte. Auch später als die Person einen Partner hatte, arbeiteten beide nicht zusammen. Dies war hauptsächlich darauf zurück zu führen, dass sich René zurückzog und versuchte die Aufgabe allein zu Lösen. Ich denke, dass es ihm unangenehm war mit einer anderen Person zusammen zu arbeiten, da er Wissenslücken und häufig Probleme beim Lösen der Aufgaben hatte. Weitere Eindrücke zu René sehen wir in Abschnitt 6.5.4.

Da Finn und Sophie im Laufe der zwei Wochen zu Einzelarbeitern wurden sind in Tabelle 13 auch Angaben von ihnen zu finden.

Tabelle 13: Angaben von Einzelarbeitern zu positiven Eigenschaften vom einzeln arbeiten zwischen dem ersten und letzten Tag

Henrik	“ Schneller.”
Sophie	“ Ich habe so viel gelernt wie noch nie, konnte schnell verstehen und umsetzen. Mein Tempo war anfangs langsam, wurde aber immer schneller.”
Sophie	“Ich bin zwar wesentlich langsamer, verstehe aber jeden Schritt.”
Finn	“Eigene Ideen gut umsetzbar. Konzentrierte Arbeit.”
Jack	“Tempo.”

Tabelle 14: Angaben von Einzelarbeitern zu negativen Eigenschaften vom einzeln arbeiten zwischen dem ersten und letzten Tag

Henrik	“Alles allein.”
Finn	“Bei Fehlersuche wäre ein Partner vielleicht hilfreich.”
Paul	“Lange gedauert.”

Es wird von den Einzelarbeitern die Schnelligkeit beim einzeln arbeiten als positiv angeführt. Diese Aussagen machten Einzelarbeiter die schon zuvor Programmiererfahrungen sammeln konnten. Es wäre interessant gewesen diese beiden für einen Tag zusammenarbeiten zu lassen und dann diese Eigenschaft von ihnen noch einmal zu hinterfragen. Es wurden auch Angaben gemacht, dass das einzeln arbeiten langsamer wäre und ein Partner bei der Fehlersuche geholfen hätte. Da Sophie bei der Paarprogrammierung die Driver-Rolle nicht übernehmen konnte und ihr Partner ein für sie sehr schnelles Lerntempo vorlegte, gab sie nun an mehr gelernt zu haben. Jedoch merkt sie auch, dass sie allein weitaus langsamer ist. Dies verbesserte sich in der Zeit ein wenig, dennoch löste sie die Aufgaben im Allgemeinen immer noch langsamer als zuvor mit Paarprogrammierung. Die Aussage von Henrik geht für mich eher in die Richtung der Stimmung und Arbeitsatmosphäre. “Alles allein.” klingt sehr demotiviert. Henrik hatte sonst einen Partner, dieser war allerdings an diesem Tag krank, sodass er allein arbeiten musste.

Tabelle 15: Angaben von Einzelarbeitern zu positiven Eigenschaften vom einzeln arbeiten am letzten Tag

Finn	"sehr zielstrebige Arbeit, man wird kreativ!"

Am letzten Tag wurde nur eine positive Eigenschaft vom einzeln arbeiten gemacht. Ich denke Finn empfand seine Arbeit als zielstrebiger, da er nun nicht mehr mit seiner Partnerin diskutieren und ihr nichts erklären musste. Im Allgemeinen wurde kein einziges Mal erwähnt, dass die Einzelarbeiter Spaß beim Programmieren hatten. Auch in den Fragebögen wurde oft "es ging so" angekreuzt. Der Spaßfaktor bei den Einzelarbeitern stand also nicht im Vordergrund. Ich denke, dass sie weniger Spaß hatten, sowie demotivierter waren wenn sie vor einem Problem standen und allein darüber grübelten. Es wurde fast immer im Fragebogen angegeben, dass ihnen ein Partner beim Lösen der Probleme geholfen hätte.

6.1.6 Probleme bei der Auswertung der Fragebögen

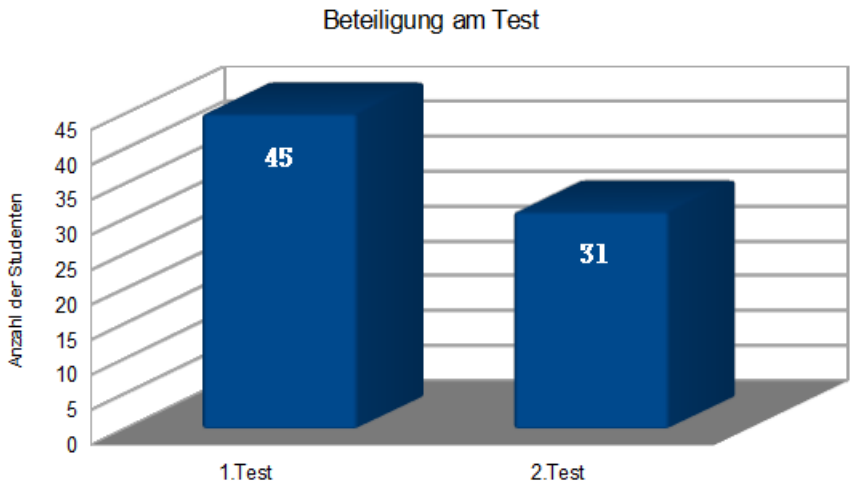
Zu Beginn ließ ich den Fragebogen von einer unbeteiligten Person testen. Trotzdem kam es dazu, dass die Studenten eine Frage missverstanden. Die Frage die die Vorbereitungszeit abfragte wurde falsch verstanden und deswegen kann die Vorbereitungszeit unter anderem nicht genügend ausgewertet werden. Es wurde gefragt: " Wie hast du dich auf den Übungsblatt vorbereitet und wie lange hast du dafür ungefähr gebraucht?". Viele Studenten gaben an, dass sie sich nicht vorbereitet haben und etwa vier Stunden gebraucht haben. Daraus lässt sich schließen, dass sie die Frage so verstanden, dass sie die benötigte Zeit für das Lösen des Übungsblatts angaben und nicht wie gefragt war, die Zeit die sie zur Vorbereitung des Übungsblatts benötigten. Da ich die Fragebögen schon vorgedruckt hatte und mir dies auch erst am dritten Tag auffiel, änderte ich die Frage sofort. Allerdings erwies sich dies als zu spät. Die Studenten schienen die Fragen nicht mehr richtig zu lesen und deshalb fiel vielen die Veränderung nicht auf und sie füllten den Fragebogen aus wie bisher. Außerdem ließ der Kurs eine intensive Vorbereitung kaum zu, da dieser täglich mehrere Stunden stattfand. Einige Studenten blieben sogar länger als bis 18:00 Uhr, um das Übungsblatt fertig zu bekommen. Die wenigen Studenten die Abends versuchten sich vorzubereiten, lasen sich noch einmal die Folien des letzten Tages durch. Auch konnte man sich schwer auf das Aufgabenblatt vorbereiten, da dies erst um 14:00 Uhr erschien. Da die Studenten nur bis 18:00 Uhr Zeit hatten die Aufgaben zu Lösen fingen sie meistens sofort an und schauten sich nebenbei die Folien an.

6.2 Analyse des Tests

Es wurde am Anfang des Kurses, sowie am Ende des Kurses derselbe Test geschrieben. Die Studenten hatten etwa 40 Minuten Zeit und sollten diesen ohne fremde Hilfe und Hilfsmittel lösen. Das die Studenten für die Bearbeitung des Tests 40 Minuten Zeit hatten entschied die Professorin spontan. In dem Test (siehe Anhang 10.13) wurden Grundkonzepte der Objektorientierung abgefragt. Ziel war es daraus abzulesen, ob sich die Studenten verbessert haben bzw. ob Paarprogrammierer sich stärker verbessert haben als Einzelarbeiter oder umgekehrt. Außerdem sollte untersucht werden, ob es beispielsweise bei einem Paar, wo mir ein Partner sehr still und unbeteiligt erscheint, Unterschiede in der Leistung im Test gibt.

6.2.1 Beteiligung an den Tests

Abbildung 10: Beteiligung der Studenten an den Tests

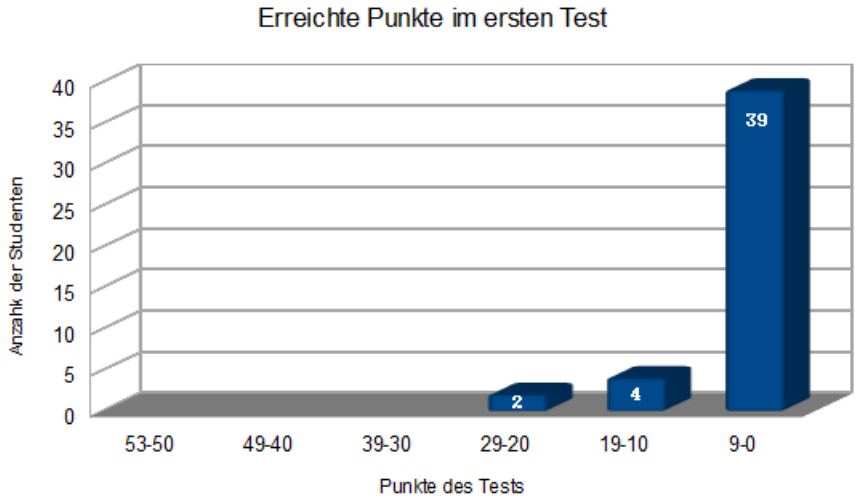


Die Beteiligung am Test sank deutlich, wie in Abbildung 10 zu erkennen, von 45 auf 31 teilnehmende Studenten. Am ersten Tag, als der Test zum ersten Mal geschrieben wurde, besuchten viele Studenten die Vorlesung um sich über organisatorische Dinge zu informieren. Am letzten Tag hingegen fehlten trotz der Ankündigung des Test einige Studenten. Einige erschienen ab der zweiten Vorlesung nicht mehr und kamen nur noch zu den Übungen. Auch fand gegen Ende der zweiten Woche ein Kurs für die Mathematik-Studenten statt. Einige Studenten besuchten am letzten Tag den Kurs gar nicht mehr, da feststand, dass diese Übung hauptsächlich für Studenten ist, die eine sehr gute Note haben wollten und in der Vorlesung wurde nichts besprochen was relevant für das Übungsblatt war, da dies bereits den Tag zuvor besprochen wurde. So kam es, dass in der Vorlesung weitaus weniger als 45 Studenten anwesend waren. Außerdem gab es einige Studenten die bei der ersten Vorlesung anwesend waren, dann aber den Kurs im Laufe der Wochen verließen. Einige Studenten schrieben den Test in den Übungen nach, jedoch schafften dies aus Zeitmangel nicht alle, da die Übung an diesem Tag frühzeitig von der Professorin und den Tutoren beendet wurde.

6.2.2 Anzahl der Punkte des Tests

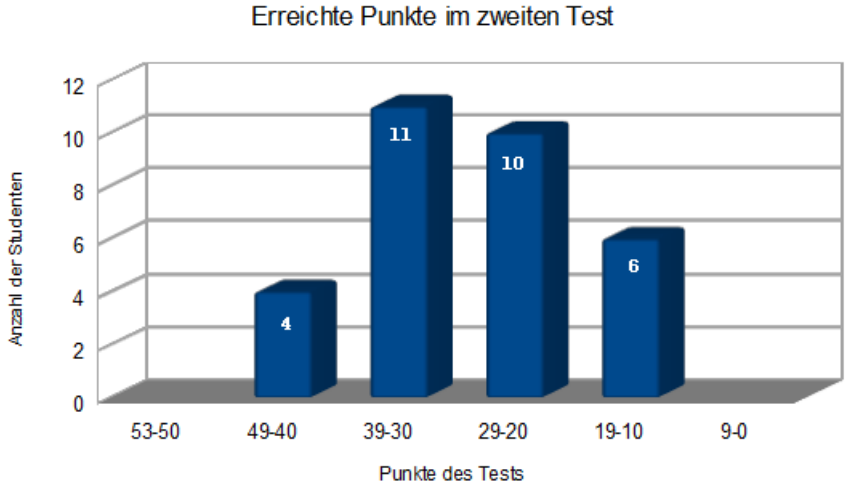
In der nachfolgenden Abbildung 11 sehen wir die Ergebnisse des ersten Tests. Es waren insgesamt 53 Punkte zu erreichen, wobei dies bei der Konzipierung des Tests noch nicht feststand und im Nachhinein von mir festgelegt wurde. Es gab eine Programmieraufgabe, sowie eine Multiple-Choice-Aufgabe. Bei den restlichen Aufgaben sollten die Studenten Werte von Methoden und zugehörige Datentypen angeben (siehe hierzu Anhang 10.13).

Abbildung 11: Ergebnis des ersten Tests



Den ersten Test (siehe Anhang 10.13) füllten 45 Studenten aus und es war auffällig, dass viele der Studenten die meisten Punkte in der ersten Aufgabe bekamen, da diese noch keine Programmierkenntnisse voraussetzte und mit Grundkenntnissen über Programmiersprachen lösbar war. Die höchste Punktzahl die erreicht wurde war 25,5 Punkte und 19 Studenten erreichten Null Punkte. Da dieser Kurs die Grundlagen der Objektorientierung in Java vermitteln sollte war abzusehen, dass der erste Test sehr schlecht ausfallen würde, da die Studenten wenig Programmiererfahrungen hatten und, wenn sie bereits in MatLab programmiert hatten, das Konzept Objektorientierung noch nicht kannten.

Abbildung 12: Ergebnisse des zweiten Tests



Im zweiten Test sieht man eine deutliche Verbesserung im Vergleich zum ersten Test (siehe Abb. 11). Die Studenten bewegten sich alle im Mittelfeld und niemand erreichte mehr als 47,5 Punkte. Allerdings erreichte auch keiner Null Punkte. Insgesamt nahmen 31 Studenten an dem Test teil. Die meisten Punkte wurden wieder in den ersten zwei Aufgaben erzielt. Jedoch lösten einige Studenten die vierte Aufgabe diesmal, welche eine Programmieraufgabe war. Am wenigsten Punkte wurden mit der letzten Aufgabe erzielt und die Meisten mit der ersten Aufgabe.

6.2.3 Probleme bei der Auswertung der Tests

Normalerweise sollte in den oben aufgezeigten Grafiken zwischen Paarprogrammierern und Einzelarbeitern unterschieden werden, sodass die Leistung der beiden Gruppen sichtbar wird. Dies konnte ich allerdings nicht umsetzen, denn das Diagramm wäre nicht Aussagekräftig geworden, was unter anderem auch daran liegt, dass zu Beginn fünf Studenten einzeln gearbeitet haben, dies zwischenzeitlich wechselte und am Ende etwa vier, wobei einer ab und an auch einzeln arbeitete. Da mir von vielen die Testergebnisse fehlen bzw. die Namen auf den Tests, kann das Diagramm nicht sinnvoll erstellt werden. Ich habe lediglich zwei zuzuordnende Namen von Einzelarbeitern zu den Tests, wobei die Personen im ersten und im zweiten Test nicht einmal identisch sind.
In der ersten Vorlesung als der Test nach meiner Einführung in Paarprogrammierung ausgeteilt und geschrieben wurde, wurde angesagt, dass die Studenten keine Namen auf die Tests schreiben brauchen, da dieser nur Forschungszwecken dient und die Studenten nur verglichen werden sollten. Nachdem die Ansage revidiert wurde, wurde zusätzlich ohne Rücksprache mit mir den Studenten vorgeschlagen Pseudonyme zu verwenden. Es gab eine große Unruhe, sodass es sein kann, das manche Studenten die Ansage, die Namen doch auf den Test zu schreiben, nicht hörten oder einfach den Namen nicht auf den Test schreiben wollten. So kam es, dass weit mehr als 50% der Tests nicht mit Namen versehen waren. Einige wenige hatten Pseudonyme verwendet und andere hatten den Namen aufgeschrieben. Bei der Korrektur gab ich den Tests ohne Namen ein Pseudonym mit der Hoffnung, dass die Studenten bei der Rückgabe ihren Test wiedererkennen würden und das Pseudonym für den zweiten Test verwenden. Bei der Rückgabe des Tests fanden allerdings viele Studenten ihren Test nicht wieder oder waren sich nicht sicher, da beim ersten Test kaum etwas ausgefüllt wurde. Beim zweiten Test bat ich die Studenten darum die Namen auf den Test zu schreiben oder die ihnen zugewiesenen Pseudonyme zu verwenden und auf den Tests zu notieren ob sie Paarprogrammierung ausprobiert haben oder einzeln gearbeitet haben. Es kam dazu, dass einige Studenten sich neue Pseudonyme ausdachten und nicht angaben, ob sie Paarprogrammierung gemacht oder einzeln gearbeitet haben. Allerdings war festzustellen, dass deutlich mehr Studenten ihren Namen auf die Test schrieben. Durch die fehlenden Angaben der Namen und der Auskunft, ob die Studenten einzeln gearbeitet haben oder Paarprogrammierung gemacht haben, kommt es allerdings dazu, dass der Test nicht in dem Maße ausgewertet werden kann, wie es ursprünglich geplant war. Auch ist zu erwähnen, dass die Konzipierung des Tests von einer anderen Person vorgenommen wurde als die Punktevergabe. Bei der Konzipierung des Tests wurde keine Punktevergabe berücksichtigt.

6.3 Analyse der Zeit für die Bearbeitung der Aufgaben

Man kann im Allgemeinen trotz der Ungenauigkeit der Messergebnisse sagen, dass weder Paarprogrammierer eindeutig schneller als Einzelarbeiter sind noch umgekehrt. Da sich in diesem Kurs sehr wenig Studenten für die Einzelarbeit entschieden haben kann man hier keine genaue Aussage treffen. Es gab sowohl Paarprogrammierer die sehr schnell mit dem Lösen des Aufgabenblatts fertig waren als auch Einzelarbeiter. Auch gab es Paarprogrammierer die länger blieben um das Übungsblatt zu lösen.

Dies war auch bei Einzelarbeitern der Fall. Wenn man davon ausgeht, dass die Paarprogrammierer doppelt so schnell beim Bearbeiten der Aufgabenblätter sein müssten, da wir auch die doppelte Arbeitskraft mit zwei Personen haben, waren die Paarprogrammierer langsamer als die Einzelarbeiter. Ich denke, dass man dies in diesem Kurs nicht so vergleichen kann wie es in der Wirtschaft getan wird, da es sich hier um Studenten handelt die die Programmiersprache neu erlernten und keine Informatik-Studenten waren. Man merkte durch Aussagen der Studenten stark, dass viele diesen Kurs besuchten, da er zu einer Pflichtveranstaltung im Mathematik-Studiengang wurde. Hier liegt somit eine andere Motivation vor und auch die Fähigkeiten sind etwas anders ausgeprägt als es bei Studenten der Fall gewesen wäre die sich für das Informatik-Studium interessieren.
Sophie und Finn (siehe Übersicht über die Beobachteten Gruppen des Blockkurses Programmieren in Java im Anhang 10.3) probierten in der ersten Woche Paarprogrammierung aus und arbeiteten in der zweiten Woche getrennt. Während die Bearbeitungsdauer der Blätter bei Finn fast unverändert blieb, brauchte Sophie länger und schaffte nicht immer alle Aufgaben komplett. Auch hier kann man also keinen eindeutigen Vergleich anstellen, da beide zusammen das Aufgabenblatt schnell bearbeiteten. Während Sophie sich sozusagen als Einzelarbeiterin zeitlich verschlechterte, blieb Finn fast gleich bzw. fielen seine Zeiten nur leicht ab, was auch mit der zunehmenden Schwierigkeit der Aufgabenblätter zusammenhängen könnte.
Vor allem in der zweiten Woche wurden starke Leistungsunterschiede deutlich sichtbar. Es gab Paare und Einzelarbeiter die weiterhin sehr früh mit dem Lösen des Aufgabenblatts fertig waren und es gab auch Paare und Einzelarbeiter die länger blieben um das Aufgabenblatt lösen zu können. Dies lag an der zunehmenden Schwierigkeit der Aufgabenblätter.

Die Tutoren gaben mir außerdem noch für jedes Aufgabenblatt und dessen Aufgaben eine Einschätzung der benötigten Bearbeitungsdauer. Wenn man diese mit der von den Studenten tatsächlich benötigten Zeit vergleicht, kommt man zu dem Schluss, dass die Einschätzung der Tutoren in den ersten Tagen mit der tatsächlich benötigten Zeit der Studenten größtenteils übereinstimmt. Am vierten Tag gaben die Tutoren an, dass sie denken die Studenten würden die Aufgabe insgesamt in 6 Stunden schaffen. Die Studenten hatten allerdings nur vier Stunden Zeit zum Bearbeiten der Aufgabe. Das die Aufgabe für die meisten Studenten in dieser Zeit nicht lösbar war kann man auch an den Zeiten sowie der Punktevergabe sehen, da viele Studenten es nicht schafften das Aufgabenblatt komplett zu lösen. Auch am fünften Tag verschätzten sich die Tutoren. Sie nahmen an, dass die Studenten in einer Stunde mit dem Übungsblatt fertig wären. Dies war nur selten der Fall, die meisten Studenten brauchten etwa zwei Stunden zum Lösen der Aufgabe. Auch an dieser Stelle ist verwunderlich, dass die Studenten vier Stunden Zeit hatten die Aufgabe zu bearbeiten aber diese weitaus schneller gelöst werden konnte. Wir haben hier also einen Tag, an dem die Studenten mehr Arbeit bekamen als sie lösen konnten, während die Studenten am nächsten Tag zu wenig zu bearbeiten hatten. An dieser Stelle kann man sich die Frage stellen, ob eine genauere Planung und Konzipierung der Aufgabenblätter vorteilhafter wäre. Auch in der zweiten Woche verschätzten sich die Tutoren mit der benötigen Zeit zum Bearbeiten der Blätter. Ich denke dies hängt mir der zunehmenden Schwierigkeit und Komplexität der Aufgaben zusammen. Außerdem bekamen die Studenten oft teilweise fertige Klassen zur Verfügung, die sie erweitern mussten. Hierbei kann man sich oft verschätzen bei der Zeit, die die Studenten zum verstehen des vorgegebenen Programms benötigen. Bei einigen Studenten dauerte die erste Teilaufgabe meist weitaus länger als die restlichen Aufgaben. Ich denke dies hängt damit zusammen, dass sie den vorgegebenen Code zuerst verstehen mussten.
Leider fehlt mir die Einschätzung der Professorin, sodass ich mich bei den Ergebnissen nur auf die Einschätzung der Tutoren beziehen kann.

6.3.1 Probleme bei dem Notieren der benötigten Zeit für die Aufgaben

Mit diesen quantitativen Messergebnissen gab es einige Probleme. Nach der Absprache mit der Professorin hatte ich den Eindruck, dass diese Daten für mich die am leichtesten zu bekommenden Daten waren. Diese Annahme bestätigte sich jedoch nicht. Es war im Vorfeld abgeklärt, dass die Tutoren für mich die Zeiten in eine von mir vorgefertigte Tabelle eintragen würden, sobald ein Student mit der Aufgabe fertig war und sich bei ihm meldete. Die Idee kam zu Stande, da von der Professorin geplant war, dass die Tutoren die Aufgaben der Studenten kontrollierten, sobald diese mit einer Aufgabe fertig waren. Am ersten Tag war dies sehr schwierig, da es viele organisatorische Dinge zu erledigen gab und jeder Teilnehmer sich an die neue Struktur des Kurses gewöhnen musste. So wurde oft vergessen die Zeit einzutragen. Auch vergaßen die Studenten nach der Ansage, dass sie dem Tutor Bescheid sagen sollten wenn sie eine Aufgabe gelöst hatten, sich beim Tutor zu melden.
Außerdem waren die Aufgaben in den ersten zwei Tagen sehr leicht gestellt und wurden schnell gelöst. So kam es auch am zweiten Tag noch dazu, dass die Zeiten ungenau waren oder vergessen wurden einzutragen. Auch war die Liste für die Tutoren am ersten Tag noch sehr unübersichtlich, da ich jedem eine komplette Liste mit allen Namen der angemeldeten 90 Teilnehmer übergab. Ab dem zweiten Tag bekam jeder Tutor eine Liste die nur die Teilnehmer aus ihrem Raum beinhaltete und nach Paaren zusammengefasst war.
Teilweise wurden für einige Zeit die Zeiten sehr zuverlässig von den Tutoren durch Mithilfe der Studenten erfasst. Es gab Tutoren die stets durch den Raum liefen und jeden Studenten fragten ob er mit der Aufgabe fertig sei und die Zeit dann notierten. Diese Methode stellte sich als die unzuverlässigste Methode heraus. Die Zeiten waren sehr ungenau und fehlten teilweise bei einigen Paaren, wo der Tutor nicht so oft war. Mit ungenau ist an dieser Stelle gemeint, dass die Studenten die Zeiten vergaßen einzutragen und am Ende der Übung von ihnen die benötigte Zeit geschätzt wurde. Ein anderer Tutor wiederum ließ die Studenten die Zeiten in die Liste selbständig eintragen. Dies funktionierte in der ersten Woche erstaunlich gut, ließ allerdings in der zweiten Woche stark nach. Die anderen beiden Tutoren führten die Liste so wie anfänglich besprochen. Die Studenten sollten sie rufen, sobald sie mit einer Aufgabe fertig waren. Dies funktionierte leider nur teilweise. Manchmal vergaßen sich die Studenten zu melden, manchmal war der Tutor verhindert oder vergaß ebenfalls eine Zeit einzutragen.
Außerdem kontrollierten die Tutoren die Aufgaben der Studenten wenn sie mit dem kompletten Aufgabenblatt fertig waren. Dies erwies sich für sie als praktischer, da sie die Zeit nutzen konnten um anderen Studenten bei Lösungsansätzen zu helfen.
Es kam auch vor, dass die Aufgabenblätter aus einer Aufgabe mit einigen Teilaufgaben bestanden. Da die Übungsblätter vor dem Kurs für mich nicht einsehbar waren fehlte mir diese Information und ich konnte nur verspätet darauf reagieren. So kam es, dass einige Tutoren nur die Zeit für eine Aufgabe eintrugen und die Zeit für die Teilaufgaben nicht eintrugen. Die Messergebnisse sind sehr ungenau und unzuverlässig und werden deshalb nicht weiter in diese Arbeit mit einfließen.

6.4 Phänomene

In diesem Abschnitt möchte ich einige besondere Erscheinungen von Verhalten, auch Phänomene genannt, näher beschreiben und erläutern. In Anhang 10.3 sehen sie eine detaillierte Tabelle die die einzelnen Personen näher beschreibt und ihnen Gruppennamen zuordnet.

6.4.1 Bildung von dreier Gruppen

"Die Drei":

Bei Linda, Nele und Pawel (siehe Anhang 10.3) machte es gerade in der ersten Woche den Eindruck, dass sie zu dritt arbeiteten. Oft fiel mir auf, dass Pawel den beiden Frauen Fragen stellte und sie ihm halfen. An einem Tag stimmten sogar die Zeiten, wie lang sie für eine Aufgabe gebraucht hatten exakt überein. Gerade in der ersten Woche machte Pawel den Eindruck die Frauen mit Fragen über die Aufgabe abzulenken. Allerdings gab er ihnen auch Tipps zur Aufgabe. Oft saßen sie nebeneinander und man konnte sie auf Grund der Sitzordnung für eine Dreiergruppe halten. Die Person die mittig saß kümmerte sich oft um ihre beiden Nachbarn und schaute auf beide Monitore. Als Linda einen Tag ohne ihre Partnerin die Aufgaben lösen musste, besprach sie sich oft mit Pawel. Sie bekam Hilfe von Pawel und sie riefen beide gemeinsam den Tutor zu sich. Während der Tutor Linda half, schaute Pawel auf Linda's Monitor und fragte ebenfalls interessiert nach. Als der Tutor nach einiger Zeit immer noch nicht bei Pawel war, fragte er Linda ob sie ihm helfen könne und mal bei ihm drüber schauen würde. Sie half ihm und erklärte ihm etwas zur Aufgabe. Selbst als die Drei nicht nebeneinander saßen, konnte ich beobachten wie Pawel sich öfter in die Richtung der Frauen umdrehte, oder auch zu anderen Paaren schaute. Er ging auch zu Linda und Nele um sie etwas zu fragen oder ihnen auch mal einen Tipp zu geben. Als es eine Aufgabe über zwei Tage verteilt gab, waren die Frauen bereits am ersten Tag fertig und Pawel saß am zweiten Tag weiterhin an der Aufgabe. Er fragte einiges bei dem sonst so ruhigen und zurückgezogenen Karl nach und sie unterhielten sich. Ansonsten schien Pawel etwas konzentrierter an diesem Tag zu sein.

"Die Dreiergruppe":

Paul verfügte über wenig Programmiererfahrung, am ersten Tag erschien er verspätet, gegen Ende der Übung. Er verpasste auch die erste Vorlesung teilweise und hatte Schwierigkeiten den Stoff aufzuholen und das Übungsblatt selbstständig zu lösen. Deshalb bekam er am ersten Tag viel Hilfe vom Tutor. Er sagte mir in einem Gespräch, dass er gern einen Partner hätte. Da bereits alle Teilnehmer die Paarprogrammierung ausprobieren wollten einen Partner hatten, war Paul gezwungen einzeln zu arbeiten. Es war auffällig, dass er die erste Woche sehr häufig den Tutor und andere Studenten aus dem Raum um Hilfe bat. Er ging auch mit seinem Laptop herum und setzte sich zu Paaren, die die Aufgabe erfolgreich gelöst hatten. Richard und Kai arbeiteten von Anfang an zusammen. Als Kai an einem Tag fehlte fragte Richard mich, ob es in Ordnung sei wenn er heute mit Paul zusammen arbeiten würde. Ich stimmte zu. Noch am selben Tag kam Kai wieder und setzte sich zu den Beiden. Von diesem Tag an arbeiteten alle zusammen, und Paul setzte sich zu Richard und Kai in den Raum. Obwohl die Bildung von Dreiergruppen in dem Kurs nicht gestattet war, ließ der Tutor und die Professorin sie zusammen arbeiten. Als ich Paul fragte ob sie die Übungsaufgaben nun als feste Dreiergruppe abgaben, bejahte er dies und gab an, dass ihm das allein arbeiten nicht gefiel und er die Gruppenarbeit effizienter fand. In der zweiten Woche fragte Paul nicht mehr soviel und führte das Lösen des Aufgabenblatts deutlich voran, während er in der ersten Woche als er allein arbeitete sehr unsicher wirkte und sehr viel Hilfe benötigte. Alle drei arbeiteten an einem Laptop und einem Institutsrechner. Paul, dem der Laptop gehörte saß oft vor dem Laptop und der andere Rechner wurde häufig dazu verwendet in Vorlesungsfolien nachzuschauen oder auch parallel zu programmieren. Paul und Richard schienen etwas besser zusammen zu arbeiten, da Kai sich öfter zurück zog und auch abwesend wirkte indem er auf den Boden schaute oder weit entfernt vom Monitor abgewandt saß. In der zweiten Woche spielte Kai oft mit Kommilitonen desselben Raums Tischtennis, während

Paul und Richard weiter arbeiteten. Auch war Kai sehr ruhig und hatte teilweise eine schlechte Sicht auf den Monitor oder Laptop. Es kam auch vor, dass Kai später kam und seine Gruppenmitglieder bereits früher nach Hause gegangen waren ohne, dass sie sich abgesprochen hatten. Auch kam Kai einmal vom Tischtennisspielen und entdeckte, dass Paul und Richard ohne ihm Bescheid zu sagen, gegangen waren. Trotzdem setzten sie sich auch am nächsten Tag wieder als Gruppe zusammen und schienen keine Probleme miteinander zu haben. Dies stellte sich auch bei einem Interview heraus.

"Die Drei" schienen sich schon vorher zu kennen und befreundet zu sein und halfen sich aus diesem Grund. Auch Paul gab an seine Partner schon zuvor gekannt zu haben. Ich denke dadurch, dass Kai fehlte und Paul mit Richard zusammen arbeitete ergab sich diese Dreierkonstellation automatisch. Paul wollte von Anfang an auch nicht allein arbeiten und eher Paarprogrammierung ausprobieren. Er fand allerdings keinen Partner, da alle Studenten die Paarprogrammierung ausprobieren wollten bereits einen Partner gefunden hatten als er verspätet in den Kurs kam. Da bereits zuvor ersichtlich wurde, dass es Paul schwer fiel allein zu arbeiten, schien die Bildung einer Dreiergruppe die logische Konsequenz zu sein. In Hinsicht auf das Konzept Paarprogrammierung sind Dreiergruppen allerdings nicht zu empfehlen. Man konnte an der Dreiergruppe von Paul sehen, dass eine Person mehr ausgeschlossen war und weniger zum Lösen der Aufgaben beitragen konnte. Allein physikalisch ist eine solche Konstellation unvorteilhaft. Dies konnte man auch hier beobachten, da die dritte Person immer einen schlechten Blick auf den Monitor hatte und sich dadurch immer weniger aktiv an der Lösung der Aufgabe beteiligte. Bei "den Drei" konnte man gegen Ende des Kurses beobachten, dass die beiden Frauen immer enger zusammen arbeiteten und Pawel selbstständiger wurde. Ab und zu halfen sie sich zwar gegenseitig aber es arbeitete immer mehr jeder für sich und nicht mehr alle Drei so stark zusammen, dass man sie für eine Dreiergruppe halten könnte. Beide Frauen hatten ein ähnlich gutes Ergebnis bei dem Test und Pawel war etwas besser. Das die Frauen die Rollen regelmäßig tauschten und ihre Ideen sehr gleichberechtigt umsetzten zeigt auch das Testergebnis.

6.4.2 Paarprogrammierer werden zu Einzelarbeitern

"Die Stillen":

Es gab ein Paar, Alesandro und Henrik (siehe Anhang 10.3), die vom Wissensstand nicht sehr stark voneinander entfernt waren. Alesandro hatte noch keine Programmierkenntnisse vor dem Kurs sammeln können und Henrik hatte etwa 100 Zeilen Code geschrieben. Henrik hatte die ersten Tage die Driver-Rolle, während Alesandro sehr still neben ihm saß und angestrengt auf den Monitor schaute. Alesandro gab in den Fragebögen außerdem an, dass er der Meinung war, dass sein Partner mehr Arbeit in das Übungsblatt reingesteckt hatte, da er selbst zu langsam war. Ihm ging das Lösen des Aufgabenblatts auch zu schnell und er war oft mit der Lösung nicht zufrieden, da sie zwar alles gelöst hatten, er aber nicht alles verstanden hatte. Er bemängelte auch, dass sein Partner ihm wenig erklärte, jedoch schien er laut seiner Antworten im Fragebogen die Schuld ausschließlich bei sich zu suchen. Sein Partner bemerkte laut des Fragebogens ebenfalls, dass Alesandro leistungsschwacher als er war und empfand die Zusammenarbeit als bremsend, da Alesandro ihm bei Problemen nicht weiterhelfen konnte. Obwohl die Arbeitsatmosphäre für mich als Beobachter sehr angespannt wirkte, gaben beide an, dass diese normal sei und zielstrebig. Einmal gab Alesandro sogar an, er hätte Spaß beim Lösen des Aufgabenblatts mit seinem Partner gehabt. Anfänglich sagte Alesandro seinem Partner nicht, dass er ihm zu schnell war und der Partner programmierte eher vor sich hin. Er erklärte ihm

kaum etwas und zeigte kein Interesse daran es dem Partner verständlich zu machen. Sein Ziel war die Aufgaben möglichst schnell fertig zu bekommen, wie ich in einem Interview erfuhr. Beide waren der Meinung, dass die Zusammenarbeit nicht so gut funktionierte, weil Alesandro zu wenig Vorkenntnisse mitbrachte. Andere Paare die ebenfalls keine Programmiererfahrungen mitbrachten oder teilweise sogar einen Partner hatten, der über reichlich Erfahrungen verfügte zeigten allerdings, dass die Zusammenarbeit sehr gut funktionieren konnte. Ich denke beide haben zu wenig kommuniziert und sind zu wenig aufeinander eingegangen. Am dritten Tag saßen beide noch nebeneinander arbeiteten aber offensichtlich getrennt, da Alesandro an seinem Laptop arbeitete, während Henrik am Rechner weiter programmierte. Sehr erstaunlich war, dass beide nun besser miteinander kommunizierten wie zuvor und Henrik Alesandro sogar ab und zu Hilfestellung gab, ohne dazu direkt aufgefordert worden zu sein. Ich konnte beobachten wie auch Henrik von Alesandro Hilfe bekam, da dieser ihm kleine Tipps gab. Es war sehr erstaunlich die Beiden dabei zu beobachten wie sie viel mehr kommunizierten und als Paar fungierten als zuvor, obwohl sie einzeln arbeiten wollten. Beide arbeiteten jeweils allein an den Aufgaben, wobei sie vor dem Tutor noch als Paar fungierten und Henriks Aufgaben vorzeigten. Während Henrik das Blatt komplett löste, hatte Alesandro große Schwierigkeiten und schaffte nur eine Aufgabe in der gesamten Zeit. Anfang der zweiten Woche fehlte Alesandro und ich ging davon aus, dass er krank sei. Als ich Henrik fragte, wusste er nicht was mit seinem Partner war, ging aber davon aus, dass dieser den Kurs abgebrochen hatte. Er sagte: "Weiß nicht, ewig nicht gesehen.", wobei Alesandro nur am Vortag gefehlt hatte. Henrik gab an, dass es für ihn keinen Unterschied machen würde im Paar oder einzeln zu arbeiten, es würde schon gehen. Alesandro kam auch in den nächsten Tagen nicht wieder und Henrik arbeitete von nun an einzeln. Er gab zwar an gerne wieder einen Partner haben zu wollen, allerdings nur, wenn dieser über die gleichen Vorkenntnisse wie er verfügte oder besser wäre. Henrik gab in einem Interview auch an, dass Alesandro ihm mitgeteilt hatte, dass er ihm zu schnell war und er wenig verstand. Trotz dieser Aussage änderte Henrik sein Verhalten nicht und ging nicht auf die Probleme seines Partners ein. Für ihn stand im Vordergrund die Übungsblätter in der vorgegebenen Zeit zu lösen und zu schaffen. Er wollte nicht wegen seinem Partner länger an dem Aufgabenblatt sitzen, wenn er es schneller fertig bekommen konnte. Alesandro versuchte mit dem allein arbeiten zu bewirken, das nicht verstandene aufzuholen, was ihm allerdings nicht gelang, da der Schwierigkeitsgrad der Übungsblätter stark anstieg.

Ich denke hier wäre es wichtig gewesen einzuschreiten und das Konzept Paarprogrammierung noch einmal zu erklären. Ich tat dies jedoch nicht, da in den ersten Tagen einige Paare etwas stiller waren und sie sich erst einmal kennenlernen mussten, wie beispielsweise Dirk und Jasmin die sich zuvor auch noch nicht kannten. Außerdem hatte ich mir in der ersten Woche vorgenommen die Beobachtungen nicht zu beeinflussen und die Studenten in ihrem natürlichen Verhalten zu beobachten. Da die Studenten in der ersten Vorlesung das Konzept der Paarprogrammierung von mir vorgestellt bekamen war es auch interessant zu sehen wie sie dieses umsetzten. Hier kann man auch wieder sehen, dass die Gruppenzusammenstellung eine essentielle Frage bei der Paarprogrammierung ist und diese noch näher untersucht werden sollte. Es wird meistens so sein, dass man nicht zwei gleich starke Partner hat. Man müsste bei der Gruppeneinteilung meiner Meinung nach darauf achten, dass die Leistungsunterschiede der Partner nicht zu groß sind, da sonst der Leistungstärkere zu sehr gebremst wird und der Leistungsschwächere Schwierigkeiten haben kann mit dem Tempo mit zu halten.

“Die Einzelarbeiter” :

Auch Sophie und Finn,(siehe Anhang 10.3) arbeiteten in der erste Woche zusammen. Während Sophie noch keine Erfahrungen im Programmieren hatte, schrieb Finn bereits zuvor etwa 500 Zeilen Code. Am ersten Tag war Sophie etwas ruhiger und Finn hatte die Driver-Rolle. Beide kommunizierten ansonsten gut miteinander, sie diskutierten über die Aufgaben und Sophie brachte sich ein. Die darauf folgenden Tage versuchte Sophie ihre Ideen immer mehr einzubringen und sie diskutierten über die Lösungen der Aufgaben. Von außen hatte man den Eindruck, dass sich beide gut verstanden und zusammenarbeiteten. Auch wenn beide anfänglich sehr angespannt wirkten, was sich durch die Mimik ausdrückte, schienen sie ein angenehmes Arbeitsverhältnis zu haben und es wurde auch ab und zu gelacht. In den Fragebögen gaben beide oft an, sie hätten Spaß daran gehabt zusammen zu arbeiten. Auch konnte man den Fragebögen entnehmen, dass sie sich beide sympathisch fanden. Wenn Sophie nachfragte, erklärte Finn ihr bereitwillig was sie nicht verstand. Auch als Finn bereits frühzeitig mit dem Lösen der Aufgaben begann und Sophie nachkam, nahm er sich Zeit und erklärte ihr den Programmcode und was die einzelnen Funktionen machten. Am dritten Tag bemerkte Sophie gegenüber einem Nachbarn, dass sie vorhatte am nächsten Tag allein zu arbeiten, da sie das Gefühl hatte sie komme nicht richtig mit und kann ihre Ideen allein nicht umsetzen. Allerdings arbeitet sie die nächsten Tage weiterhin mit Finn zusammen und von außen betrachtet waren sie ein gutes Team. Gegen Ende der ersten Woche konnte ich den Fragebögen von Sophie entnehmen, dass sie sehr unzufrieden mit der derzeitigen Situation war. Sie mochte Finn und er war ihrer Meinung nach sehr freundlich und Hilfsbereit. Sie merkte aber, dass er weitaus mehr Erfahrungen im Programmieren hatte als sie und sie mit seinem Tempo nicht Schritt halten konnte. Auch nachdem sie Finn fragte, überließ er ihr nur ungern die Tastatur. Wenn sie diese kurz hatte und etwas tippen wollte, nahm Finn die Tastatur nach kurzer Zeit wieder an sich. Auch gab sie in dem Fragebogen an, dass sie lieber allein arbeiten würde wollen, da sie das Gefühl hatte dadurch mehr zu lernen, es aber Finn nicht sagen wollte, da sie ihn sehr nett fand. Sie wollte mit ihm weiterarbeiten, obwohl das allein arbeiten ihre Schwäche theoretische Dinge in die Praxis umzusetzen gemindert hätte. Anfang der zweiten Woche war ich erstaunt als ich sah, dass beide getrennt an zwei Rechnern saßen und dies obwohl Sophie im Fragebogen am Freitag ausdrücklich angab mit ihm weiter zusammen zu arbeiten um ihn nicht zu kränken. Ich fragte Finn ob sie heute allein arbeiteten und dieser gab an, dass sie lediglich die Aufgaben aufgeteilt hätten und noch im Team arbeiten würden. Finn ging bis kurz vor Ende des Kurses davon aus, dass sie weiterhin zusammen arbeiten würden und sich die Aufgaben aufteilten, obwohl Sophie ihm gesagt hatte, dass sie allein arbeiten wollte. Zudem machten beide dieselben Aufgaben und teilten diese nicht auf. Dies konnte man dem Fragebogen entnehmen, in dem er angab, dass sie schneller seien, da simultanes Arbeiten bei sinnvoller Arbeitsteilung schneller wäre. Als er von mir einen Fragebogen für Einzelarbeiter bekam, schien er kurzzeitig verwundert zu sein, da sich die Fragen auf diesem Bogen von den Fragen für Paarprogrammierer leicht unterschieden. Er schaute sich kurz Sophies Fragebogen an, um sicher zustellen, dass er den richtigen Fragebogen bekommen hatte und füllte ihn aus. Er gab auch hier an, dass ein Partner nicht beim Lösen der Aufgaben geholfen hätte, da parallel arbeiten mit dem Partner schneller sei. Auch konnte ich bemerken, dass Sophie und Finn nicht mehr miteinander redeten seitdem sich Sophie für das allein Arbeiten entschieden hatte. Sie gab auch in einem Interview an, dass er sich seither ihr gegenüber anders verhielt und nicht mehr mit ihr redete. Dies verwunderte mich, da er zu diesem Zeitpunkt immer noch behauptete mit Sophie zusammen zu arbeiten.

Man sieht an diesem Beispiel, dass durch das Beobachten allein keine Aussagen getroffen werden können und sich die zusätzliche Abfrage von persönlichen Eindrücken und der Motivation in den Fragebögen als nützlich und wichtig erwiesen hat. Ohne die Fragebögen hätte man nicht ahnen können, dass Sophie mit der Zusammenarbeit unzufrieden war, auch der Partner wusste davon zuerst nichts. Da ich ihre Eindrücke bereits im Fragebogen gelesen hatte, hätte eventuell in der zweiten Woche ein Vermittlung zwischen Beiden geholfen, dies erübrigte sich allerdings, da Sophie bereits einzeln arbeitete und dies mit ihrem Partner geklärt hatte. Jedoch schien die Kommunikation zwischen ihnen nicht sehr gut zu funktionieren, da der Partner erst nach weiteren zwei Tagen und einem Gespräch verstand, dass sie beide nicht mehr zusammenarbeiten. Obwohl Finn ihr die Aufgaben erklärte war er doch zu schnell für sie. Außerdem überließ er ihr nicht die Driver-Rolle. Ich denke dies war ausschlaggebend. Hätten die Beiden die Rollen regelmäßig getauscht wäre der Lerneffekt bei Sophie größer gewesen. Vorausgesetzt Finn hätte Geduld gehabt und sie nicht gedrängt, sobald sie die Driver-Rolle gehabt hätte. Außerdem sieht man an diesem Paar, dass zu große Wissensunterschiede keine gute Voraussetzung für das Zusammenarbeiten sind.

"Die Einzelgänger":

Jan hatte im Gegensatz zu Luka noch keine Programmiererfahrungen. Luka war einer der wenigen die über 1000 Zeilen Code zuvor geschrieben hatten. Am ersten Tag beobachtete ich beide dabei wie sie getrennt an zwei Rechnern saßen aber sehr viel kommunizierten. Da es zu Anfang Probleme mit dem Aufstellen von Namensschildern gab und ich noch nicht alle kannte, fragte ich die Beiden nach ihrem Namen und warum sie nicht Paarprogrammierung ausprobieren wollten. Daraufhin gaben sie an, dass sie Paarprogrammierer seien. Ich erklärte ihnen noch einmal das Konzept Paarprogrammierung und beide arbeiteten von diesem Zeitpunkt an an einem Rechner. Im Laufe der ersten Woche schon konnte ich beobachten, dass ausschließlich Luka die Driver-Rolle übernahm. Zudem kommunizierten beide kaum miteinander. Jan fragte Luka was er gerade programmierte und wollte versuchen es zu verstehen. Luka gab ihm allerdings nur kurze Antworten und Jan schien überfordert zu sein und dem Tempo nicht stand zu halten. In der zweiten Woche kommunizierten beide kaum noch miteinander, da Jan es aufgegeben hatte Luka nach einer Erklärung zu seinem Programmcode zu fragen. In einem Interview mit Jan erfuhr ich, dass beide auch vorher schon zusammengearbeitet hatten und Luka schon sehr viel programmiert hatte. Außerdem gab er offen zu trotz des regelmäßigen Besuchs der Vorlesung bei den Übungseinheiten nicht mitzukommen. Luka besuchte die Vorlesung nicht und kam nur zu den Übungseinheiten, wo er sich ohne Rücksicht auf Jan hinsetzte und anfing zu programmieren.
Mitte der ersten Woche kam Jan etwas zu spät und Luka saß bereits am Rechner und programmierte. Als Jan ankam und sich neben Luka setzten wollte war es sehr auffällig, dass Luka Jan kaum beachtete und ihm sehr spät erst etwas Platz machte. Als Jan ihn ansprach, nickte Luka kaum merklich, ohne auch nur aufzusehen und tippte weiter. Als der Tutor im Raum etwas laut für alle erklärte, stand Jan ohne etwas zu sagen auf und setzte sich weiter nach vorn in Richtung der Tafel. Er hörte dem Tutor zu und schaltete den Rechner vor ihm ein. Währenddessen tippte Luka weite. Anschließend holte Jan seine Jacke von Luka's Platz. Beide schauten sich weder an, noch sagten sie etwas und Jan verließ den Raum. Später saßen sie wieder nebeneinander und kommunizierten sehr wenig. Jan gab auch in einem Interview mit mir an, dass er zwar zwischendurch versucht hatte einige Ideen einzubringen, Luka diese aber völlig ignorierte und nicht umsetzte. Dies konnte ich auch beobachten, als beide über eine Aufgabe diskutierten und sich die Vorlesungsfolien anschauten. Während Jan

noch beim nachlesen war, scrollte Luka schnell durch die Folien, ohne Rücksicht auf ihn zu nehmen. Jan beschwerte sich allerdings auch nicht, was wahrscheinlich daran lag, dass Luka die meiste Arbeit zum Lösen des Aufgabenblatts übernahm. Auch wenn Jan ohne etwas anzudeuten oder zu sagen den Raum verließ, schaute Luka nicht einmal auf und programmierte weiter wie zuvor. Er gab jedoch in einem Interview an, dass er die Arbeitsteilung so okay fände denn, wenn er alles erklären müsste, dann würde es länger dauern. Auch Jan war bewusst, dass die Arbeitsteilung nicht gleichverteilt war und Luka im Prinzip allein programmierte. Er war darüber nicht sehr glücklich, war allerdings auch nicht Fähig an der Situation etwas zu ändern, da er mit dem Vorlesungsstoff Schwierigkeiten hatte und ihm Lukas kurze Erklärungen oder bzw. das Zeigen von Code nicht halfen die Defizite aufzuholen.

Auch dieses Paar ist ein Beispiel dafür, dass zu große Wissensunterschiede für Paarprogrammierung nicht von Vorteil sind und sich auf die Zusammenarbeit schlecht auswirken. Auch denke ich spielen die verschiedenen Persönlichkeiten eine starke Rolle. Dies erkennt man, da es Luka völlig egal war was sein Partner machte und ob er anwesend war. Dies erfuhr ich auch von ihm in einem Interview. Luka war es egal, solange er die Aufgabe lösen konnte und von Jan hörte ich, dass Luka seine Ideen nicht umsetzte und darauf selten einging. Außerdem war Jan sehr zurückhaltend und konnte sich gegenüber Luka nicht durchsetzen. Ich denke dies hängt unter anderem mit seiner Persönlichkeit zusammen und auch mit der Tatsache, dass Luka für das Entstehen der Lösungen allein verantwortlich war, da Jan nicht so viele Vorkenntnisse hatte wie er und es ihm schwerer fiel. Man sollte die Gruppenzusammenstellung noch weitaus stärker untersuchen.

"Die Abwesenden":

Tim und Hannes probierten Paarprogrammierung ebenfalls aus und arbeiteten auch bis zum Schluss zusammen. In der ersten Woche fiel mir auch noch nichts sehr besonderes auf außer, dass Hannes immer die Driver-Rolle übernahm. Hannes erklärte Tim was er programmierte und sich dabei dachte und sie diskutierten. In der zweiten Woche konnte ich jedoch beobachten, dass beide wenig miteinander redeten und Tim zunehmend abwesend wirkte. Dies drückte sich dadurch aus, dass Tim sich im Raum umschaute während Hannes programmierte. Auch kramte er im Rucksack herum oder spielte mit seiner Wasserflasche. Teilweise saß Tim abgewandt von Hannes und dem Monitor da und schaute in die Luft oder auf den Boden. Er drehte sich sogar einmal mit dem Stuhl um die eigene Achse und tat dies einige Minuten ohne, dass Hannes dies wahrzunehmen schien. Als ich ein Interview mit dem Paar führte, fragte ich Tim ob er sich langweile, was er verneinte. Beide fragte ich auch, ob sie der Meinung seien eine gerechte Rollenaufteilung zu haben und gut miteinander zu arbeiten. Hannes gab zu, dass er meistens die Driver-Rolle hatte, da er die Vorlesung regelmäßig besuchte, was Tim nicht tat und deshalb eher wusste, was sie zu tun hatten. Außerdem hatte er ursprünglich vorgehabt allein zu arbeiten, bis Tim ihn fragte. Ansonsten waren beide der Meinung, dass sie gut zusammen arbeiteten und sich die Arbeit teilten. Tim war überzeugt davon, dass er die Aufgaben verstand und daraus lernte. Nachdem ich noch einmal nachdrücklich nachfragte, gab Hannes an, dass Tim sich immer einbrachte, auch Tipps gab und gut mitarbeitete. Oft konnte ich auch beobachten, dass während Hannes Tim etwas erklärte und dabei auch versuchte Blickkontakt aufzubauen, Tim ihn nicht anschaute und teilweise sogar abwesend wirkend wegschaute oder auf den Tisch guckte. Ich fragte Hannes ob er der Meinung ist, dass beide gleich viel Arbeit in das Blatt steckten und er bejahte dies sehr überzeugt. Dazu muss man sagen, wenn Hannes Tim ansprach und etwas fragte war dieser sofort wieder bei der Sache und in der ersten Woche wirkte er auch nicht so extrem gelangweilt und abgelenkt.

Ich konnte bei einigen Paaren beobachten, dass sie beispielsweise mit anderen Paaren kommunizierten und sich bezüglich der Aufgabe austauschten, wenn der eigene Partner an diesem Tag krank war. Es war interessant zu sehen, dass Studenten die sich ansonsten kaum untereinander absprachen oder Ideen ausdiskutierten, mehr miteinander kommunizierten wenn der eigene Partner nicht da war.

Anfänglich brachte Tim sich sehr gut ein. Jedoch setzten sie von Anfang an das Konzept Paarprogrammierung nicht wie gewünscht um, da Hannes immer die Driver-Rolle hatte. Zum Schluss wirkte Tim immer abwesender und reagierte nur auf Nachfragen seines Partners. Dieser wiederum schien so sehr mit dem Programmieren beschäftigt zu sein, dass es ihm kaum auffiel, wenn Tim desinteressiert oder abwesend wirkte. Ich denke wir haben hier ein Beispiel dafür, das das regelmäßige tauschen der Rollen essentiell für die Paarprogrammierung ist. Ist eine Person zu lange ausschließlich Observer wird er immer inaktiver und beteiligt sich weniger an der eigentlichen Lösung. Dies konnte ich auch bei anderen Paaren beobachten, wo sich der Observer anfänglich noch ab und an einbrachte und sich zum Schluss immer weniger bis gar nicht mehr einbrachte. Bei Tim und Hannes konnte auch durch den zweiten Test bestätigt werden, dass Hannes der immer die Driver-Rolle hatte weitaus besser abschnitt als Tim. Dieser bekam auch beim Ausfüllen des Tests von Hannes etwas Hilfe. Dies zeigt, dass es wichtig ist sich mit der Rollenverteilung abzuwechseln da man vor allem durch die Praxis lernt und nicht nur vom zuschauen. Dies Hilft zwar viele Dinge theoretisch zu verstehen, jedoch fehlt es dann oft an der Umsetzung. Dies bestätigte mir auch Sophie in einem Interview, da sie oft Schwierigkeiten hatte ihre Ideen in Programmcode umzusetzen.

6.4.3 Ablauf von Rollenwechsel

In diesem Abschnitt möchte ich beschreiben, wie sich der Rollenwechsel abgespielt hat.

"Die Spaßvögel":

Bei Patrick und Florian (siehe Anhang 10.3) konnte ich beobachten wie Patrick die Tastatur förmlich an sich riss. Dies hatte allerdings keine negativen Auswirkungen auf das Verhältnis der Beiden zueinander, da sie schon vorher befreundet waren, ihre Zeit auch in den Pausen gemeinsam verbrachten und sich viel absprachen. Bei einem Interview kam heraus, dass sie den Rollenwechsel nicht voraus planten, sondern sich jeder die Tastatur nahm, wenn ihm eine Idee kam oder er etwas zur Lösung beitragen konnte.

Auch "Die Kämpfer" Angélique und Mylos hatten einen ähnlichen Rollenwechsel. Beide diskutierten über die Aufgabe und einer nahm die Tastatur zu sich und fing an zu tippen. Derjenige der eine Idee hatte wollte diese umsetzten und nahm sich die Tastatur.

"Die Freundinnen":

Lisa und Lara (siehe Anhang 10.3) wechselten sich die erste Woche mit der Driver-Rolle einigermaßen regelmäßig ab und kommunizierten sehr viel miteinander. Sie gaben später an sich meist täglich mit der Driver-Rolle abgewechselt zu haben. Dies war sehr außergewöhnlich und konnte bei keinem anderen Paar beobachtet werden. Normalerweise tauschten die Paare mehrmals in einer Stunde die Rollen

und wechselt diese nicht pro Tag. Allerdings besprachen sich beide sehr viel miteinander, sodass die Driver-Rolle nicht immer der führenden Rolle entsprach. Der Observer diktierte dem Driver was er tippen sollte und der Driver setzte auch von allein die Ideen des Observers um. In der zweiten Woche fiel mir mehr und mehr auf, dass Lara nur noch die Driver-Rolle hatte. Lisa brachte sich dennoch ein und hatte einige Ideen. Als mir beide ein Interview gaben stellte sich heraus, das beide sich im Laufe der Zeit darauf geeinigt hatten, das Lara die Driver-Rolle übernehmen sollte, da sie schneller tippen konnte und sich besser auskannte.

Auch bei Lara und Lisa war durch den Test zu erkennen, dass diejenige der öfter die Driver-Rolle hatte im Test besser abschnitt, in diesem Fall sogar fast doppelt so gut. Dies liegt denke ich daran, dass beide die Rollen zum Schluss gar nicht mehr getauscht haben. Ich denke, dass der halbstündliche Rollentausch einen weitaus positiveren Effekt hat und man so mehr lernt. Gerade der Tausch innerhalb einer Aufgabe hat einen hohen Lerneffekt. Dies hat den Vorteil, dass der Observer immer aufmerksam sein muss und es auch ist, da er durch das regelmäßige tauschen der Rollen nicht zu lang in der Beobachter-Rolle ist. Außerdem werden so die Ideen des jeweils anderen öfter umgesetzt. Selbst wenn der Driver also seine eigene Idee umsetzt und die Rollen anschließend getauscht werden, setzt der Oberserver, der nun zum Driver geworden ist die Idee des Anderen um. Dies ist meiner Meinung nach sehr sinnvoll und lehrreich. Die Rollen täglich zu tauschen macht meines Erachtens weniger Sinn, da in den Übungen der Schwierigkeitsgrad steigt und die Themen sich von Tag zu Tag ändern. Ich denke, das diese Methode das Lösen des Aufgabenblatts stark erschwert und dies dazu führt, dass die Paare ab einem gewissen Punkt nicht mehr mitkommen. Wenn zum Beispiel am ersten Tag das Blatt sehr leicht ist und es am zweiten Tag schon schwieriger ist, hat derjenige der am ersten Tag das Blatt löste am dritten Tag Schwierigkeiten das Blatt nun zu lösen, da er den Anstieg des Schwierigkeitsgrades in Hinsicht auf die Umsetzung verpasst hat. Auch kann diese Methode dazu führen, dass der Observer im Laufe des Tages immer inaktiver wird. Wenn man von vorne hinein weiß, dass man an diesem Tag dem Partner beim programmieren zugucken wird, geht man auch mit einer anderen Einstellung an das Lösen des Aufgabenblatts heran.

"Die Respektvollen"

Saskia und Sven sind ein Beispiel dafür, wie ein noch unerfahrener Programmierer von einem erfahrenen Programmierer lernen kann. Während Saskia noch keinerlei Programmiererfahrungen mitbrachte, hatte Sven bereits zuvor etwa 500 Zeilen Code geschrieben. Bei beiden herrschte eine gute Arbeitsatmosphäre und sie gingen sehr respektvoll miteinander um. Sven erklärte Saskia alles und war immer darauf bedacht, dass sie alles verstand. Selbst wenn Sven schon lange die Lösung wusste, gab er Saskia wenn sie wieder den Driver hatte nur Tipps und Denkanstöße. Er versuchte immer sie nur leicht zu lenken, sodass sie von allein auf die Lösung kam. Anhand der Fragebögen war zu sehen, dass Sven in der zweiten Woche etwas gelangweilt war, da er über Probleme mit Saskia nicht diskutieren konnte. Trotzdem arbeiteten sie weiterhin gut zusammen und sie teilten sich die Driver-Rolle gut. Oft bewegte Sven Saskia dazu die Tastatur zu übernehmen und etwas zu programmieren. Saskia gab allerdings auch gute Tipps und machte Sven auf Fehler aufmerksam. " Du hast völlig recht, ich bin total bescheuert", sagte Sven und Saskia lachte. Die Stimmung war immer gut und locker. Ab und zu wurde gescherzt und gelacht.

Beide wechselten sich regelmäßig mit der Driver-Rolle ab, jedoch lenkte Sven das Lösen des Aufgabenblatts etwas mehr, was sich auch anhand der Testergebnisse erkennen lässt. Es wurden oft

Sven's Ideen umgesetzt und Saskia fiel es öfters schwer Ideen allein umzusetzen. Ich denke, dass auch hier die Motivation der Studenten eine Rolle spielt. Während sich Sven für das Programmieren interessieren zu schien, besuchte Saskia den Kurs, weil dieser Pflicht war.

"Die Gutgelaunten":

Nina und Mia wechselten sich anfänglich nicht regelmäßig mit der Driver-Rolle ab. Dies gaben sie auch in einem Gespräch zu und ließ sich anhand der Fragebögen ableiten. Allerdings schafften sie es gegen Ende der ersten Woche die Driver-Rolle, wie sie fanden, gerecht aufzuteilen. Während einer die Maus hatte, bediente der andere die Tastatur. Am Anfang des Tages hatte Nina oft die Driver-Rolle, während gegen Ende des Tages Mia die Driver-Rolle übernahm. Beide gaben in den Fragebögen auch an, dass ein Nachteil von Paarprogrammierung sei, dass man nicht alles selber macht. Dies bestärkte auch den Eindruck, dass Nina oft die Driver-Rolle übernahm und schwer abgeben wollte. Allerdings arbeiteten beide daran die Rollen gerechter aufzuteilen, was ihnen auch teilweise gelang. Da Nina immer die erste Hälfte des Tages die Driver-Rolle hatte, wie sie beschrieb, übernahm Mia gegen Ende des Tages die Driver-Rolle, wenn Nina sich müde fühlte, da ihr vier Stunden programmieren zu viel war. Der Observer übernahm in der Zeit die Maus und lenkte durch Foliensätze oder den Programmcode.

Es war interessant zu beobachten, dass beide Frauen die Driver-Rolle tatsächlich noch einmal aufteilen, indem eine die Maus und die Andere die Tastatur bediente. Ich denke dies hat den Vorteil, dass der Observer, der die Maus bedient weitaus aufmerksamer bleibt. Außerdem sind dadurch beide gezwungen stärker miteinander zu kommunizieren, da wenn der Driver eine Idee hat dieser den Observer lenken muss um an eine Codestelle zu gelangen und umgekehrt der Observer dem Driver Bescheid sagen muss wenn er etwas entdeckt hat und den Cursor in die Richtung bewegen muss. Allerdings verlangsamt es die Arbeitsweise. Oft geschehen Tippfehler, zu dessen Position der Driver vielleicht schnell mit dem Cursor gehen möchte. Auch Copy & Paste Aktionen, sowie das nachlesen von Codestellen werden durch diese Handhabung erschwert.

"Die Perfekten":

Malte und Max hatten beide sehr viel Programmiererfahrungen, wobei Malte etwas mehr hatte und schon zuvor über 1000 Zeilen Code geschrieben hatte (siehe Anhang 10.3). Sie wendeten das Konzept Paarprogrammierung sehr gut an und kommunizierten intensiv miteinander. Beide erklärten sich regelmäßig ihre Ideen oder wie sie vorgegangen sind, während der Andere noch nicht da war. Beispielsweise kam Max einmal etwas später und Malte hatte bereits mit dem Programmieren angefangen. Sobald Max sich hingesetzt hatte fing Malte an ihm die Aufgabe zu erklären und ihm zu zeigen wie er bisher vorgegangen ist. Es kam vor, dass beide über die Lösung der Aufgabe diskutierten und beispielsweise Malte Max fragend ansah, der daraufhin die Driver-Rolle übernahm. Oft kam es vor, dass der Rollentausch automatisch passierte. Mitten im Gespräch griff einer nach der Tastatur und setzte seine Idee um. Ich konnte sie dabei beobachten wie sie regelmäßig die Rollen tauschten. Derjenige der programmierte erklärte währenddessen immer was er gerade tat und warum, dies machten die anderen Paare sehr selten. Es gab eine Situation in der Malte seinem Partner eine Idee erklärte, beide schauten sich dabei die ganze Zeit über an. Als Malte mit seinen Erklärungen fertig war griff Max zur Tastatur und versuchte seine Idee umzusetzen. Als Max nicht mehr weiter kam nahm Malte Maus und Tastatur an sich und zeigte Max etwas. Nach weiterer Absprache über-

nahm dann Max wieder die Tastatur und programmierte weiter. Außerdem beobachtete ich wie Max die Tastatur Malte zuschob als er nicht mehr weiter kam und Malte eine Idee hatte. Auch kam es vor, dass beide den Rollenwechsel gleichzeitig vollzogen. Während Max in Richtung Tastatur griff, schob Malte sie ihm beim diskutieren zu.

Malte hatte ein leicht besseres Test-Ergebnis als Max. Da es sich allerdings nur um ein paar Punkte handelt und beide einen sehr guten Schnitt hatten, wird dies hier nicht weiter erörtert. Außerdem ist ein Test auch immer eine Abfrage von Wissen, bei dem einige sich mehr merken können und Andere weniger. Dazu gibt es auch unterschiedliche Lerntypen. Ich denke aber dies würde zu weit führen und ist nicht Teil des Themas.

"Die Unbekannten":

Dirk und Jasmin wurden von mir als Paar zusammengeführt und kannten sich vorher nicht. Sie waren das einzige Paar was sich zuvor noch nicht kannte. Dirk hatte etwas mehr Programmiererfahrungen als Jasmin. (siehe Anhang 10.3) Die ersten drei Tage waren beide sehr still. Vor allem Jasmin verhielt sich sehr zurückhaltend. Dirk hatte über die zwei Wochen hin die ganze Zeit die Driver-Rolle und beide schienen sehr zufrieden damit zu sein. Jasmin versuchte während meiner Beobachtungen die Driver-Rolle auch nicht zu übernehmen. Auch als Dirk den Raum verließ um ein Privatgespräch mit einem Kommilitonen zu führen, tippte sie nicht weiter sondern saß da und wartete. Jasmin legte ihre Schüchternheit allerdings gegen Mitte der ersten Woche ab und man konnte deutlich erkennen, dass sie sich einbrachte. Ihre Körperhaltung verriet mir, dass sie nun weitaus lockerer auf ihrem Stuhl saß und sich wohler fühlte. Sie zog ein Bein auf den Stuhl und umklammerte ihr Knie worauf sie den Kopf ablegte und auf den Monitor schaute. Beide lachten nun auch weitaus mehr miteinander und die Stimmung schien nicht mehr so angespannt zu sein wie zu Beginn. Als ich mit Beiden ein Interview führte stellte sich heraus, dass Jasmin sich zwar einbrachte, es aber für besser hielt Dirk die Driver-Rolle zu überlassen, da sie dachte das theoretische nicht gut in Programmcode umsetzen zu können wie Dirk. Sie war allerdings der Meinung trotzdem viel zu lernen und war sich bewusst ohne Dirk nicht immer zu wissen was sie tippen sollte. Auch Dirk gab zu ein Problem zu haben die Verantwortung bzw. Driver-Rolle abzugeben. Die Rollenaufteilung hatte sich bei ihnen so ergeben, da Dirk auch etwas mehr Erfahrung hatte, schneller war und gleich wusste was er tippen sollte, wenn er eine Aufgabe las. Beide waren mit ihrer Zusammenarbeit sehr zufrieden und man konnte bemerken, dass sich beide während der zwei Wochen stark verbesserten und schneller wurden. Ab und Zu blieb Dirk ohne seine Partnerin länger, da er noch einige Zusatzaufgaben machen wollte und mit der Lösung noch nicht ganz zufrieden war. Ihm machte es nichts aus, dass Jasmin nicht da war. Da sie dir Driver-Rolle nie wechselten setzten sie das Konzept der Paarprogrammierung nicht ideal um. Allerdings arbeiteten beide im Team und Jasmin machte gute Vorschläge und deckte Fehler auf. Auch diskutierten und besprachen sie sich zur Lösung der Aufgabe. Einmal übernahm Jasmin kurzzeitig die Driver-Rolle. Jedoch zog Dirk die Tastatur wieder zu sich und gab die Driver-Rolle nicht mehr ab. Jasmin verhielt sich sehr still und brachte keine Einwände entgegen. Auch sonst trieb Dirk das Lösen der Aufgaben voran, da er immer wieder Fragen stellte wie:" Machen wir dann gleich noch die 2?". Während er programmierte fragte er Jasmin auch immer wieder ob der Lösungsansatz so gut sei oder welche Ideen sie noch hatte, worauf sie antwortete und ihre Ideen einbrachte. An einem Tag bekam ich mit, wie Dirk wieder einmal zu Ralf und Jenny ging. Er sagt:" Ihr müsst mich mal motivieren, ich komm gerade nicht weiter.". Sie unterhielten sich und Dirk ging wieder an seinen Platz zurück. Anscheinend fühlte er sich von seiner Partnerin hinsichtlich des Lösens der Aufgabe

nicht motiviert genug und wollte sich mit seinen Freunden ablenken.

"Die Freunde":

Jakob und Nicole kannten sich bereits vorher (siehe Anhang 10.3). In den ersten Tagen gaben beide im Fragebogen an, dass sie sich gegenseitig ablenkten, nicht bei der Sache waren und eine lustige Arbeitsatmosphäre hatten. Trotzdem gaben sie weiterhin an, dass sie konzentriert und zielstrebig waren. Dies verwunderte mich und ich beobachtete die Beiden intensiver. Tatsächlich sah man, dass sie eine sehr lustige Arbeitsatmosphäre hatten. Es gab Phasen wo die Beiden sehr viel herumalberten und auch Phasen in denen sie sich gegenseitig zur Konzentration verhalfen und aufriefen. Mit der Driver-Rolle wechselten sie sich ab. Einmal konnte ich beobachten wie sie über eine Aufgabe diskutierten und Nicole die Tastatur behalten wollte. Während Jakob den Driver hatte diskutierten sie und fragten den Tutor um Hilfe. Nicole nahm in der Zeit die Maus in die Hand und zeigte etwas auf dem Monitor. Auch die Tastatur zog sie leicht zu sich heran. Während beide weiter diskutierten wollte Jakob die Tastatur wieder übernehmen und Nicole behielt sehr lange die Hände an der Tastatur. Letztendlich übernahm Jakob doch wieder kurz die Driver-Rolle und tippte. Als er ihr etwas mit der Maus zeigte übernahm sie die Driver-Rolle. Während also der Eine die Tastatur bediente, übernahm der Andere die Maus und zeigte damit auf Codestellen und gab Tipps.

"Die Vorbildlichen":

Trotz dessen, dass Emma noch keinerlei Erfahrungen im Programmieren hatte und Noah bereits etwa 500 Zeilen Code schrieb, setzten beide Paarprogrammierung effektiv ein. Emma hatte viele Ideen. Anfänglich konnte man bemerken, dass Emma etwas Erfahrung fehlte, dies holte sie allerdings sehr schnell auf und beide programmierten gleichberechtigt. Sie wechselten sich regelmäßig mit der Driver-Rolle ab, wobei einer die Tastatur bei oder nach einer Diskussion über die Lösung der Aufgabe automatisch übernahm. Sie hatten sehr häufig Blickkontakt, was bei vielen anderen Paaren nicht der Fall war. Auch wenn Emma mit dem Tippen etwas langsamer war, da sie nur mit zwei Fingern tippte, war Noah nicht genervt oder ungeduldig. Im Gegenteil, man hatte das Gefühl, dass es keinem der Beiden etwas ausmachte und gar nicht auffiel. Derjenige der die Observer-Rolle einnahm erklärte auch viel und zeigte dem Driver immer die entsprechenden Stellen auf dem Monitor. Dies konnte ich bei sehr vielen Paaren beobachten. In einem Interview gaben beide an, dass der Rollenwechsel spontan geschah. Wenn beide hingen und einer eine Idee bekam, übernahm derjenige die Driver-Rolle und tippte.

Trotz dieser guten Umsetzung des Konzepts der Paarprogrammierung hatte Emma ein besseres Testergebnis als Noah, der auch mehr Erfahrungen im Programmieren hatte. Dies ist erstaunlich und zeigt, dass auch unerfahrene Programmierer viel lernen können, wenn sie Paarprogrammierung anwenden.

Es ist klar zu erkennen, dass der Rollenwechsel nicht geplant wird. Dieser passiert häufig automatisch und eher unbewusst. Außerdem haben wir gesehen, dass es sehr unterschiedliche Methoden zum Wechsel der Rollen gibt. Der regelmäßige Rollenwechsel ist allerdings ein wichtiger Bestandteil der Paarprogrammierung, da dieser das Erlernen des Programmierens stark beeinflusst. Jemand der in regelmäßigen Abständen die Driver-Rolle hat, lernt seine Ideen besser in Programmcode umzusetzen,

als jemand der stets die Observer-Rolle hat. Dies konnte ich auch bei einigen Paaren beobachten.

6.4.4 Konzentriertes Arbeiten

Ich möchte hier näher darauf eingehen, wie die Teilnehmer von einer besonders guten oder schlechten Stimmung wieder in das konzentrierte Arbeiten zurückgekehrt sind und auch inwiefern sich die Teilnehmer abgelenkt haben und danach wieder konzentriert an den Aufgaben gearbeitet haben.

"Die Gutgelaunten":

Nina schrieb vor dem Kurs etwa 100 Zeilen Code und Mia hatte keine Erfahrungen im Programmieren. Jack, "Der Hilfsbereite" hatte bereits mehr als 1000 Zeilen Code geschrieben. Er war am ersten Tag sehr ruhig und arbeitete für sich allein. Am zweiten Tag bereits wurde er öfter von Nina und Mia abgelenkt, die ihm Fragen zur Aufgabe stellen. Auch andere Teilnehmer fragten ihn um Rat wenn der Tutor nicht da war oder noch bei einem anderen Paar saß. Oft blieb er länger, obwohl er mit dem Lösen des Aufgabenblatts bereits fertig war und als Grund dafür, dass er Paarprogrammierung nicht ausprobieren wollte angab, dass es allein schneller geht. Bei Nina und Mia war die Stimmung meistens sehr gut und sie lachten viel. Auch kommunizierten sie sehr gut miteinander. In der zweiten Woche fiel mir stark auf, dass die Beiden eher Jack um Hilfe baten als den Tutor zu rufen. Teilweise setzte sich Jack zu den Frauen und half ihnen. Er versuchte dabei meistens den Frauen nicht alles vorzusagen sondern ihnen lediglich Anregungen und Tipps zu geben, sodass sie ihr Problem selber überwanden. In der ersten Woche fragten sie Jack nicht so schnell nach Hilfe und versuchten die Aufgabe eher allein zu lösen, während sie ihn in der zweiten Woche viel eher ansprachen und um Hilfe baten. Auch die anderen Studenten im Raum baten Jack in der zweiten Woche oft um Hilfe. Jack kam zu jedem herum und gab Tipps. Interessant zu beobachten war, dass Nina Jack einen Platz neben ihnen freihielt, als durch den Ausfall eines Tutors die Gruppe den Raum zu einem anderen Tutor wechseln musste. Als sich jemand neben sie setzen wollte, machte Nina darauf aufmerksam, dass dieser Platz für Jack freigehalten wurde. Als dieser leicht verspätet auftauchte, wurde er auch sofort von Nina und Mia gefragt und erklärte ihnen die Aufgabe und gab Hilfestellung. Nachdem Jack die Fragen der Frauen beantwortet hatte oder ihnen geholfen hatte, war er sofort wieder auf seine Aufgabe konzentriert und arbeitete weiter. Auch wurde das Paar von einem Kommilitonen abgelenkt, der den Kurs nicht besuchte, dies kam allerdings sehr selten vor. Auch fiel mir auf, dass als die Gruppe den Raum wechseln musste sie in dem anderen Raum die gute Stimmung auf andere Studenten übertrugen und das in dem sonst so ruhigen Raum eine lustige Atmosphäre herrschte. Dies erreichten sie durch kleine Scherze und Kommentare, die sie laut von sich gaben und den anderen mitteilten. Jedoch blieb die Arbeitsatmosphäre trotzdem noch konzentriert.

"Der Fragensteller":

René verfügte genauso wie "Die Lustigen", Jenny und Ralf über keinerlei Programmiererfahrungen. Dies war die erste Woche sehr stark zu bemerken. Sie benötigten sehr viel Hilfe vom Tutor und fragten ihn oft. Auch drückten sie ihre schlechte Stimmung aus, wenn sie nicht weiterkamen. Genauso wurde allerdings auch laut gejubelt, wenn sie Erfolg hatten. Am ersten Tag hatte man den Eindruck, dass alle drei zusammenarbeiteten. René war allerdings Einzelarbeiter. Es war auffällig, dass er sehr oft den Tutor um Hilfe bat und wenn dieser keine Zeit hatte seine Nachbarn fragte. Auch Jack half ihm einige Male und kam dazu um den Tisch gelaufen, da er gegenüber von René saß. Auch lenkte

René die Teilnehmer durch Privatgespräche und Kommentare des Öfteren ab. Als ich ihn beobachtete, bemerkte ich, dass er sich nicht lange mit einer Sache durchgängig beschäftigen konnte. Immer wieder unterbrach er sich selbst bei der Arbeit oder dem Lesen und fragte etwas oder redete mit seinen Nachbarn. Außerdem lachte er sehr häufig, wobei er manchmal aus Verlegenheit lachte und nicht weil er etwas sehr lustig fand. Wenn er auf etwas angesprochen wurde und mitteilte, dass er beim Lösen der Aufgabe nicht weiterkam und so wie er sich ausdrückte nichts verstand, fing er auch an zu lachen. Am dritten Tag bekam er einen Partner, dieser war allerdings nicht regelmäßig da, besuchte nicht die Vorlesung und bereitete sich auch sonst nicht auf die Übungen vor. Er besuchte diesen Kurs, da er ein wenig Praxiserfahrungen sammeln wollte. Auffällig war, dass René nicht mit seinem Partner zusammen arbeitete, obwohl er sich diesen freiwillig ausgesucht hatte. Jeder arbeitete für sich und oft konnte ich beobachten, wie Renés Partner die Aufgaben allein löste, während René mit anderen Studenten Privatgespräche führte oder nicht anwesend war. René schien der Kurs nicht wichtig genug zu sein und ich denke er war sehr überfordert. In der zweiten Woche besorgte er sich ein Buch über die Programmiersprache Java welches er zu lesen begann und löste das Aufgabenblatt nur teilweise. René benötigte bis zum Ende des Kurses die Hilfe der Studenten und des Tutor und forderte diese auch immer wieder ein. Oft lenkte er seine Mitstudenten mit Privatgesprächen ab. Es kam äußerst selten vor, dass sich René ruhig allein beschäftigte. Selbst als er in dem Buch über die Programmiersprache Java las, hörte er bei Gesprächen seiner Nachbarn mit und führte mit ihnen Privatgespräche. René wurde trotzdem als sehr lustig und nett empfunden und so halfen ihm seine Kommilitonen bis zum Schluss. Die Überforderung von René konnte man auch am vierten Tag beobachten. Oft lenkte er wieder seine Nachbarn mit Scherzen und Gesprächen ab oder lief durch den Raum zu anderen Paaren. Er forderte viel Hilfe vom Tutor ein und saß ansonsten vor seinem Rechner und kam nicht weiter. Dies wird durch die Aussage: "Oh Gott, jetzt sitz ich schon ne halbe Stunde davor und hab nichts geschafft." klar. Nach einer Erklärung zur Aufgabe über Datentypen sagte René laut: "Ich glaub ich ess erst mal was, ich hab zwei Brötchen dabei.". Auch an diesem Tag konnte man beobachten, dass René die Studenten im selben Raum ablenkte und diese um Hilfe bat, obwohl er sich diesmal in einem anderen Raum als sonst befand. Sein Partner, der anfangs der Woche dazu kam, half einer anderen Gruppe und saß an seinem Laptop, während René an einem anderen Rechner saß und andere Kommilitonen um Hilfe bat. René ging an diesem Tag auch früher und sein Partner ließ seine eigenen von ihm gelösten Aufgaben vom Tutor kontrollieren und rechnete sie für René an. Wo René hinging konnte mir sein Partner bei einem Gespräch nicht genau sagen, er wusste nur, dass er öfter zur Bibliothek oder zum Sport ging.

Es kam auch oft vor, dass die Studenten durch Nachfragen des Tutors oder der Professorin abgelenkt wurden. Nachdem dann beispielsweise der Tutor weg war, arbeiteten die Studenten dann wieder konzentriert an ihrer Aufgabe weiter.

Man kann hier sehen, dass sich nicht nur Paarprogrammierer untereinander ablenken. Außerdem sind kurzzeitige Ablenkungen für die Arbeitsatmosphäre förderlich und stören das Bearbeiten der Aufgabenblätter nicht. Außerdem können Einzelarbeiter sich auch ablenken, nur ist dies nicht so leicht zu beobachten wie die Ablenkung die stattfindet, wenn sich Paarprogrammierer untereinander ablenken. Die Ablenkung der Einzelarbeiter könnte in einer späteren Arbeit noch einmal genauer untersucht werden. Einzelarbeiter könnten sich in diesem Fall auch mit anderen Studenten von den Aufgaben ablenken lassen. Es könnte allerdings weitaus öfter vorkommen, dass Einzelarbeiter sich durch malen auf dem Blatt, surfen im Internet oder mit ihrem Handy ablenken. Auch könnte es sein, dass jemand einfach nur da sitzt und gedanklich woanders ist. Dies wäre unter anderem sehr schwer

zu beobachten.

6.4.5 Auffällige Stimmungen

Hier erläutere ich einige prägnante Fälle, bei denen es zu Streitigkeiten gekommen ist oder sehr gute Stimmung geherrscht hat.

"Die Kämpfer":

Angélique und Mylos hatten beide noch keine Programmiererfahrungen bevor sie den Kurs besuchten. In der ersten Woche konnte ich beobachten, dass es ihnen sichtlich schwer fiel das Aufgabenblatt zu lösen. Dies drückten sie durch Mimik und Gestik aus und sagten dies auch offen. Auch der Tutor bemerkte die anfänglichen Schwierigkeiten und half. Beide wechselten sich mit der Driver-Rolle ab, wobei Mylos die letzte Woche diese etwas häufiger hatte. Außerdem war zu beobachten, dass sich beide erstaunlich verbesserten. Mylos gab in einem Interview an: " Wir sind beides Sturköpfe." und dies bemerkte man auch bei der Zusammenarbeit. In einer Situation gestikulierten beide stark mit den Armen und Angélique sprach mit erhobener Stimme. Angélique war sehr unzufrieden mit der Lösung der Aufgabe, die Mylos vorschlug. Beide diskutierten und Mylos versuchte zu schlichten und zeigte ihr beim diskutieren eine Stelle auf dem Monitor. Schließlich rief Mylos den Tutor um nachzufragen und der kleine Streit legte sich. Eine Weile lang waren beide sehr angespannt und ruhig, was ich wieder aufgrund ihrer Mimik beobachten konnte. Sie hatten einige Errors im Programmcode und das Programm lief nicht. Nachdem sie die Aufgabe erfolgreich gelöst hatten waren beide sehr entspannt und glücklich über den Erfolg.

"Die Drei":

Bei Linda und Nele kippte einmal die Stimmung ins Negative. Die Situation entstand schnell und war plötzlich da. Nele war Observer und diskutierte mit Driver Linda. Die Diskussion wurde immer lauter und angeregter. Nele hatte eine negative Mimik, da sie wollte, dass Linda das Programm testete, was sie nicht tat. Auch Pawel war bei der Diskussion beteiligt, allerdings nur als stiller Beobachter. Beide Frauen sahen sich angenervt an und fingen auf einmal an zu lachen. Dies entschärfte die vorher negative Stimmung und die Beiden arbeiteten wie gewohnt weiter und lächelten über weitere Probleme. Zwischenzeitlich fing Linda an zu lachen, da ihr auffiel, dass sie einiges aus dem Programm gelöscht hatte, woraufhin Nele und Pawel ebenfalls anfingen zu lachen.

"Die Lustigen":

Jenny und Ralf (siehe Anhang 10.3) verbreiteten oft gute Stimmung und es wurde viel gelacht. Auch die Art und Weise wie der Tutor mit den Studenten umging verhalf zu einem fortwährendem Stimmungshoch. Er scherzte oft und lenkte die Teilnehmer dadurch ab. Allerdings brachte er diese auch wieder dazu konzentriert weiter zu arbeiten. Dies machte er indem er sie direkt dazu aufforderte sich nun wieder mit den Aufgaben zu beschäftigen. Jenny und Ralf drückten ihre Stimmung direkt aus, sodass es jeder mitbekommen konnte. Hatten sie Probleme beim Lösen der Aufgaben, machte sich Unmut bei ihnen breit, sie meckerten und fragten vermehrt den Tutor. Auch wurde geschnauft, angestrengt geguckt, wobei der Ellenbogen auf dem Tisch abgestützt wurde und der Kopf auf der

Hand abgestützt wurde. Lösten sie eine Aufgabe erfolgreich, jubelten sie laut und ließen alle wissen, dass sie die Aufgabe gelöst hatten.
Als sie beide den Raum aufgrund des Ausfalls eines Tutors wechseln mussten, konnte man deutlich bemerken, dass es in dem anderen Raum weitaus lauter wurde und laut gelacht wurde. Die Stimmung war sehr lustig und ausgelassen und sie übertrugen ihre gute Stimmung teilweise auf die anderen Studenten. Auch dieser Tag bestand für die noch unerfahrene Gruppe aus Höhen und Tiefen. Beispielsweise sagte Ralf laut: "Ich kann nicht, sag du was." und Jenny übernahm die Tastatur und probierte etwas aus bis sie von einem Kommilitonen abgelenkt wurde. Später bekam Ralf sehr viel Hilfe von Renés Partner und Jenny half René. Als sie die Aufgaben wieder erfolgreich lösen konnten, brachen beide in lautem Jubel aus und die Studenten im Raum freuten sich für die Beiden mit. Auch dies hob die Stimmung stark an.

In dem Raum in dem sich die Lustigen befanden herrschte oft eine sehr ausgelassene und gute Stimmung. Dies lag nicht nur an ihnen, auch der Tutor machte viele Scherze oder erzählte kleine Anekdoten aus seinem Leben. Er lenkte die Studenten zwar ab, rief aber auch zur Arbeit auf. Dies tat er direkt in dem er sie darauf hinwies, zum weiterarbeiten anregte und das Gespräch wieder auf das eigentliche Thema lenkte. Auch die anderen Kommilitonen aus dem Raum hoben die Stimmung mit Witzen und entsprechenden Kommentaren. Entweder verebbte die Heiterkeit dann langsam und die Studenten fingen automatisch wieder an konzentrierter zu Arbeiten oder jemand lenkte das Thema wieder auf die aktuelle Aufgabe. Trotz dessen, dass die Stimmung immer sehr gut und heiter war, kann man nicht sagen, dass die Studenten für das Bearbeiten der Aufgabenblätter länger gebraucht haben. Meist gingen die Studenten pünktlich um 18:00 Uhr nach Hause, während die Studenten im Nebenraum, wo eine sehr konzentrierte Stimmung herrschte, teilweise länger blieben und das Aufgabenblatt noch nicht komplett gelöst hatten. Die Studenten in dem Raum sprachen oft so, dass alle im Raum es mitbekamen und einen Kommentar dazu geben konnten. Trotzdem gab es Phasen, in denen alle sehr konzentriert arbeiteten und sich davon nicht ablenken ließen.

Am Freitag der ersten Woche konnte man tatsächlich ein Stimmungshoch ausmachen. Dies lag daran, dass sich die Studenten darüber unterhielten was sie am Wochenende machen wollten. Außerdem war ein großer Teil froh ein paar Tage nicht programmieren zu müssen. Da es sich hier um einen Kurs handelte der für Mathematikstudenten zur Pflicht geworden ist, hatten wir hier viele Studenten, die sich nicht für das Programmieren interessierten und denen es keinen Spaß machte. Trotz dessen, das alle lachten und sich mäßig laut über die Freizeit und das Wochenendprogramm unterhielten arbeiteten zwei Paare noch weiter an den Aufgaben. Während sie versuchten die Aufgaben zu lösen lachten sie mit den Studenten die sich bereits zum gehen aufmachten ab und zu mit und redeten über Aktivitäten des Wochenendes. Mir fiel auch auf, dass selbst die Tutoren am Freitag eine lockere Stimmung hatten und einige Privatgespräche führten. Jeder schien sich auf das Wochenende zu freuen und drückte dies aus. Selbst ich ließ mich einmal kurz mitreißen als ich in dem Raum war, in dem die Lustigen Scherze mit ihren Kommilitonen und dem Tutor machten. Am letzten Freitag, als der Kurs endete, war die Stimmung äußerst ausgelassen. Die Aufgaben wurde von den Meisten sehr sorgfältig bearbeitet, aber der Spaß stand im Vordergrund. Es wurde überall sehr viel gelacht und sich mit Studenten aus anderen Räumen ausgetauscht. Die Freude darüber den Kurs endlich abgeschlossen zu haben war sehr groß und die Arbeitsatmosphäre war sehr locker. Der Druck der die letzten Tage auf den Studenten lastete durch zunehmend schwerer werdende Aufgaben, war weg.

Am Montag war zwar kein direktes Stimmungstief zu bemerken, allerdings schienen alle weniger

konzentriert und vor allem effektiv zu arbeiten. Oft konnte man bei den Teilnehmern eine leichte Unlust und Trägheit feststellen, was auch die Professorin bemerkte.

Die ganze Gruppe im Raum "der Lustigen" schien sich in der zweiten Woche viel besser kennen gelernt zu haben und ging weitaus lockerer und freundschaftlicher miteinander um. Auch Personen die sonst ruhiger waren, waren gut in die Gruppe integriert und erzählten Lustiges. Umso näher das Ende des Kurses heranrückte, umso ausgelassener wurde die Stimmung.

Alle Teilnehmer gaben im Fragebogen des letzten Tages an, dass sie die Stimmung in ihrem Raum für gut hielten. Im Allgemeinen gab es auch keine besonders schlechte Stimmung, jedoch ließ sich in jedem Raum ein Unterschied feststellen. Im Raum "der Lustigen" war die Stimmung sehr heiter und es wurde fast die ganze Zeit über gelacht. In einem anderen Raum waren die Studenten sehr konzentriert und verhielten sich ruhig. Dies war im Raum "der Lustigen" nicht zu bemerken, dort wurde oft so gesprochen, dass es jeder im Raum mitbekam und einen Kommentar dazu abgeben konnte. In einem anderen Raum war es in der ersten Woche sehr laut, da sich Studenten dort aufhielten die nicht zu diesem Kurs gehörten und sehr viel miteinander diskutierten. Außerdem gab es in diesem Raum die größte Anzahl an Teilnehmern, was den Lautstärkepegel ansteigen ließ. In der zweiten Woche wurde es dort mit dem Fehlen der fremden Studenten etwas ruhiger, jedoch hatte ich teilweise immer noch Probleme zu verstehen, was die Teilnehmer untereinander besprachen. In einem weiteren Raum in dem sehr wenig Teilnehmer waren, war es auch sehr ruhig und konzentriert. Ich bekam allerdings das Gefühl, dass dort meist jeder für sich arbeitete bzw. jedes Paar für sich arbeitete.
Ich denke gute Stimmung erhöht die Leistungsfähigkeit der Studenten, solange es jemanden gibt der wieder auf das Weiterarbeiten aufmerksam macht. Es sollte ein gutes Mittelmaß gefunden werden. Dies hängt auch stark von den Persönlichkeiten der einzelnen Personen im Raum ab.

7 Bewertung der getroffenen vorbereitenden Überlegungen

7.1 Vorbereitende Überlegungen zur Vorbereitungszeit

Paare benötigen eine längere Vorbereitungszeit als Studenten, die einzeln arbeiten. Diese verkürzt sich allerdings im Laufe der Zeit von zwei Wochen. Außerdem verkürzt sich bei den Studenten, die Paarprogrammierung anwenden die Zeit, die sie für das Implementieren der Aufgabe benötigen.

Das Paare eine längere Vorbereitungszeit als Einzelarbeiter benötigen, konnte aufgrund der Probleme bei den Messungen siehe Abschnitt (Abschnitt 7.5) weder belegt noch widerlegt werden. Außerdem bereiteten sich die Studenten oft nur dann vor wenn sie merkten, dass sie bei der Aufgabe nicht mehr weiter wussten und kamen. Dann erst schauten sie in den Folien nach oder fragten Tutoren. Auch hier kann also nicht klar belegt werden, dass Paare eine längere Vorbereitungszeit benötigen und dafür aber schneller beim Lösen der Aufgaben sind. Es gab beispielsweise auch einen Einzelarbeiter, der eine sehr kurze Vorbereitungszeit hatte und seine Aufgabe aber auch sehr schnell löste. Auch gab es einen Einzelarbeiter der für beides sehr lang im Vergleich brauchte. Als häufigstes Mittel zur Vorbereitung wurden die Folien herangezogen. Auch kann nicht eindeutig belegt werden, dass sich die Vorbereitungszeit der Paare im Laufe der Zeit von etwa zwei Wochen verkürzt. Allerdings gaben viele Paare im Fragebogen an, dass sie nun sehr gut miteinander auskommen und kommunizieren, da sie nun schon seid zwei Wochen intensiv zusammenarbeiten. Auch die Arbeitsaufteilung gelang den Studenten immer besser. Dies teilten mir einige Teilnehmer mit und ich konnte es auch häufig

beobachten. Während beispielsweise sich ein Paar zu Anfang gar nicht mit der Bedienung von Maus und Tastatur abwechselte, konnte man in der zweiten Woche beobachten, dass sie sich regelmäßig abwechselten und die Rollen tauschten.

7.2 Vorbereitende Überlegungen zur Kommunikation von Einzelarbeitern und Hilfestellung

Einzelarbeiter in einem Raum kommunizieren trotzdem und geben sich auch teilweise Hilfestellungen. Einzelarbeiter fragen öfter den Tutor um Rat, als Studenten die Paarprogrammierung anwenden (Abschnitt 4.4.2).

Das Einzelarbeiter öfter den Tutor um Hilfe bitten als Paarprogrammierer, traf in diesem Kurs nicht zu und kann weder belegt noch widerlegt werden. Der Großteil der Studenten fragte den Tutor um Rat, wobei Paarprogrammierer sich auch untereinander halfen. Ich denke das hängt auch stark mit dem Konzept des Kurses zusammen. Dieser ist sehr praxisorientiert und in jedem Raum war stets ein Tutor zur Verfügung. In einigen Räumen kamen die Tutoren auf Anfrage zu den Studenten, es gab aber auch Räume, wo die Tutoren herumgingen, wenn sie nicht gebraucht wurden und die Studenten fragten. In diesem Kurs war die Hemmschwelle den Tutor zu fragen sehr gering, vor allem als sich alle Beteiligten im Laufe der Zeit besser kannten. Es gab sehr wenige Studenten die sich für das Einzelarbeiten entschieden haben und diese fragten den Tutor genauso häufig wie die Paarprogrammierer. Am ersten Tag arbeitete ein Einzelarbeiter zwar sehr stark mit einem Tutor zusammen, dies lag aber daran, dass der Student eine Stunde vor Schluss kam und einen Teil der Vorlesung durch eine nachzuholende Klausur verpasst hatte. Ein anderer Einzelarbeiter hingegen fragte den Tutor kaum und kam zügig und gut voran. Es gab hingegen auch Paare, die oft den Tutor riefen und Paare, die kaum Hilfe brauchten. Somit kann die These weder belegt noch widerlegt werden. Eindeutig belegt werden kann die vorbereitende Überlegung, dass Einzelarbeiter trotzdem kommunizieren. Es gab Einzelarbeiter die sehr ruhig waren und niemanden außer den Tutor fragten. Es gab aber auch Einzelarbeiter die Paarprogrammierer um Hilfe baten und sich mit ihren Sitznachbarn unterhielten. Es kam auch vor, dass Studenten die Paarprogrammierung nicht ausprobierten den Paarprogrammierern halfen. Belegt werden kann die These, da die Hilfestellung des Tutors ebenso zur Kommunikation gehört. Auch wenn ein Einzelarbeiter beispielsweise wenig kommunizierte, kam es doch ab und zu vor, dass er dem Tutor eine Frage stellte. All das konnte ich bei meinen Beobachtungen feststellen. Aus den Fragebögen ergab sich, dass Paarprogrammierer sich häufiger mit ihrem Partner besprachen als andere Studenten zu fragen.

7.3 Vorbereitende Überlegungen zur Stimmung und konzentrierten Arbeitszeit

Bei Paarprogrammierern wird nur eine kurzzeitige Ablenkung stattfinden und die meiste Zeit konzentriert gearbeitet. Wenn extrem gute oder schlechte Stimmung herrscht, sind die Teilnehmer abgelenkt und arbeiten nicht konzentriert.
Freitags wird weniger konzentriert gearbeitet und die Stimmung ist ausgelassener. Am Montag ist

die Stimmung nicht so gut wie sonst und es wird weniger konzentriert gearbeitet (Abschnitt 4.4.3).

Es gab einen Fall, da schwankte die Stimmung kurzzeitig von sehr gut und locker zu schlecht. Dies passierte plötzlich in einer Diskussion über die Aufgabe. Der Observer wollte, dass der Driver die Aufgabe noch einmal testet und hatte eine negative Mimik. Genauso schnell wie die Situation entstand, löste sie sich allerdings auch. Beide schauten sich genervt an und fingen an zu lachen. Anschließend war nichts mehr von der vorherigen Situation zu bemerken und sie lösten weiterhin ihre Aufgaben gutgelaunt gemeinsam. Auch wenn bei Beiden immer nur eine kurze Ablenkung stattfand, ist zu bemerken, dass sie sich nicht gegenseitig ablenkten, sondern von einem Einzelarbeiter und dem Tutor abgelenkt wurden.

7.4 Vorbereitende Überlegungen zur Rollenverteilung und Hilfestellung

Derjenige der das Lösen der Aufgabe vorantreibt und eine eher führende Rolle einnimmt, beantwortet die Fragen des Partners. Studenten die Paarprogrammierung anwenden fragen seltener den Tutor um Hilfe und helfen sich im Team. Meistens wird derjenige den Tutor fragen stellen, der die führende Rolle übernimmt, was nicht unbedingt bedeuten muss, dass derjenige den Driver hat (Abschnitt 4.4.4).

Oft ist mir aufgefallen, dass diese Überlegung zutraf. Wobei man hier auch klar unterscheiden muss, dass es nicht unbedingt derjenige der die Driver-Rolle hat, die führende Rolle einnimmt und das Lösen des Aufgabenblatts vorantreiben. Bei Paaren die sehr ausgeglichen waren fragten oft beide. Ich konnte auch beobachten, wie sie sich absprachen um dann den Tutor zu fragen. Oft stellte einer der Beiden dem Tutor die Frage und der Partner ergänzte, während es bei ungleichen Paaren, wo einer mehr die Führung übernahm oft so war, dass derjenige einfach dem Tutor die Frage stellte und diesen heran rief, ohne dies vorher mit dem Partner zu besprechen.

7.5 Weitere Probleme der Fallstudie

Am ersten Tag ergaben sich schon einige Probleme, die die Beobachtung der Studenten erschwerten. Da es für alle der Beteiligten der erste Tag war, gab es noch viel organisatorisches zu klären. Einige Studenten sprachen mich in der Pause an und stellten noch Fragen zur Paarprogrammierung und meiner Arbeit und wie die Studenten in Räume und Gruppen eingeteilt werden. Das allergrößte Problem für mich war die Zuordnung von Namen und Gesichtern. Ich bat die Tutoren im Vorfeld, leider nur auf einen Zettel notiert, da ich ihnen gegen Ende der Vorlesung die Tests wiederbrachte, die Studenten darum zu bitten ein Namensschild aufzustellen. Dies würde meine Beobachtungen erleichtern und ich müsste nicht jeden Studenten nach seinen Namen fragen, was ich als störend und ablenkend empfand. Ich wollte bei meinen Beobachtungen versuchen erst einmal so unauffällig wie möglich zu sein, um den Studenten nicht das Gefühl zu geben sie würden wie in einem Labor beobachtet werden. Bei meinen Beobachtungen stellte ich häufig fest, dass die Studenten selten Namensschilder aufgestellt hatten. Ich fragte die Studenten also häufig nach ihren Namen und versuchte sie mir einzuprägen. Auch fragte ich die Tutoren, ob sie noch einmal eine Ansage machen könnten und die Studenten auffordern könnten sich ein Namensschild anzufertigen. Da ich zu Beginn nicht in jedem Raum gleichzeitig sein konnte, weiß ich nicht, ob manche Tutoren vergaßen dies anzusagen oder die Studenten einfach kein Namensschild aufstellen wollten. Das Zusammenfinden der Paare funktionierte auf Anhieb sehr gut und selbstständig. Ich ging in jeden Raum und fragte, ob jeder einen

Partner hat oder noch jemanden suchen würde. Ich fand in jeweils zwei verschiedenen Räumen zwei Teilnehmer die noch einen Partner suchten und fragte sie ob sie sich vorstellen könnten zusammen zuarbeiten. Sie willigten sofort ein und setzten sich an einen Computer. Dieses Paar arbeitete auch am Schluss noch weiterhin ohne Probleme zusammen. Für den ersten Tag fertigte ich eine komplette Liste mit allen Namen der Teilnehmer für jeden Tutor an, in die er die Zeiten für die Aufgaben für jeden Studenten notieren konnte. Wir einigten uns vorher mit Frau Prof. Esponda auf diese Methode. Zuerst wurde überlegt, ob man Listen anfertigt und diese in jedem Raum aushängt, sodass jeder Student selbstständig die Zeit einträgt, die er zum Bearbeiten der Aufgabe benötigt hat. Da am ersten Tag noch nicht klar war, welcher Tutor welche Studenten bekommt und wer Paarprogrammierung ausprobiert, wurde die Liste für die Tutoren sehr lang und unübersichtlich. Dies führte dazu, dass es die Tutoren schwer hatten die Zeiten einzutragen. Auch fiel mir gleich am ersten Tag auf, dass ich meinen Beobachtungsbogen dringend anpassen musste. Dieser war ebenfalls viel zu unübersichtlich und ich verschwendete viel Zeit damit die Namen zu notieren. Dies legte sich glücklicherweise sehr schnell, da ich mir einen Sitzplan anlegte und die Namen der Teilnehmer schnell lernte. So konnte ich meine Beobachtungsvorlage nach Räumen sortiert anfertigen. Pro Raum hatte ich nun ein Blatt auf dem die Namen der Studenten angegeben waren in einem Feld und so wusste ich auch sofort wer mit wem zusammen arbeitet. Durch die vielen organisatorischen Dinge, die zuerst für alle etwas neu waren, verbrauchte ich viel Zeit, in der ich nicht beobachten konnte. Das erste Aufgabenblatt war sehr schnell zu lösen, weshalb die meisten Studenten früher nach Hause gingen und sich meine Beobachtungszeit verkürzte. Zu Anfang gab es auch teilweise Schwierigkeiten zu erkennen, wer mit wem zusammenarbeitete, da sich ab und zu Dreierkonstellationen ergaben und man nicht wusste, wer der Einzelarbeiter ist. Dies ergab sich dann aber im Laufe des Tages, spätestens durch das direkte Nachfragen. Auch kam es vor, dass zwei Paarprogrammierer trotz meiner Erläuterung in der Vorlesung an zwei Rechnern saßen. Ich machte sie darauf aufmerksam und erklärte ihnen noch einmal was man unter Paarprogrammierung versteht und sie setzten sich ab diesem Zeitpunkt zusammen an einen Rechner.

Beim Messen der Vorbereitungszeit ergaben sich die größten Probleme. Deshalb entschieden meine Betreuerin und ich am Anfang der zweiten Woche das Messen der Vorbereitungszeit wegzulassen, da die Ergebnisse nicht verwendbar und ausschlaggebend waren. Mir war von Anfang an klar, dass das Messen der Vorbereitungszeit für eine einzelne Person bei so vielen Teilnehmern unmöglich werden würden. Deshalb wollte ich diese nicht quantitativ für alle messen sondern für einzelne von mir beobachtete Paare. Da ich davon ausging, dass die Studenten sich beim Tutor meldeten wenn sie eine Aufgabe fertig hatten, plante ich zu dem entsprechenden Paar oder dem Einzelarbeiter zu gehen und dann die Vorbereitungszeit für die nächste Aufgabe zu messen. Ich bekam jedoch nicht ausreichend Messergebnisse, da sich die Studenten nicht meldeten oder an Tagen wo die Vorbereitungszeit untersucht werden sollte das Aufgabenblatt nur aus einer Aufgabe bestand.

8 Fazit

Diese Arbeit sollte die Leistung und Leistungsfähigkeit bei Paarprogrammierung in der Lehre untersuchen. Durch die Fallstudie wurden interessante Phänomene beobachtet und beschrieben. Diese waren weitaus komplexer als ursprünglich angenommen und beispielsweise beim Wechseln der Driver-Rolle und Observer-Rolle ergaben sich verschiedenste Facetten des Wechsels. Mittels der verschiedenen Datenerhebungsmethoden konnte ich verschiedene Phänomene aufdecken und näher untersuchen. Obwohl es den Studenten freigestellt war sich für oder gegen das Ausprobieren von Paarprogrammie-

rung zu entscheiden, wählten von 41 Studenten die den Fragebogen am ersten Tag ausgefüllt haben 36 Paarprogrammierung und 5 entschieden sich für das einzeln Arbeiten. Die Zahl der Einzelarbeiter sank zwischenzeitlich und lag gegen Ende des Kurses wieder bei 5. Dies kam zu Stande, da die Studenten zu jeder Zeit selbst entscheiden durften, ob sie weiterhin Paarprogrammierung ausprobieren wollten oder lieber einzeln arbeiten wollten. Da sich die Mehrheit freiwillig für das Konzept der Paarprogrammierung entschieden hat, kann man davon ausgehen, dass sich Paarprogrammierung positiv auf die Arbeitsatmosphäre auswirkt. Wenn dem nicht so wäre, hätten sich viele Studenten im Laufe der Zeit dagegen entschieden. Auch zeigten dies die Fragebögen der Teilnehmer auf. Alle Teilnehmer die in dem Kurs bis zum Schluss anwesend waren, haben den Kurs bestanden. Es gab einige Beispiele die zeigten, dass Paarprogrammierung in der Lehre durchaus sinnvoll ist. Es gab Paare die miteinander lernten und Paare die voneinander lernten. Es gab Paare die sehr konzentriert miteinander arbeiteten und Paare die Phasen hatten, in denen sie sich mit etwas anderem beschäftigten. Beispielsweise wurden Witze gemacht, Privatgespräche geführt oder auch Tischtennis gespielt. Dies alles trug allerdings dazu bei, dass die Arbeitsatmosphäre gut war und anschließend weiter konzentriert gearbeitet werden konnte. Man kann auch nicht behaupten, dass Paarprogrammierer länger zum Bearbeiten eines Aufgabenblatts brauchen würde. Jedoch kann dies auch nicht eindeutig widerlegt werden. Ich denke das Wichtigste ist der Eindruck den der Paarprogrammierer selbst hatte, denn wenn dieser ein gutes Gefühl durch eine gute Arbeitsatmosphäre hatte, ist er auch Leistungsfähiger. Durch das miteinander Kommunizieren fand immer ein Ideenaustausch statt. Es wurde oft angegeben, dass die Studenten allein an einer Stelle nicht so schnell weitergekommen wären und sie bei dem Lösen des Übungsblatts weniger Spaß gehabt hätten. Außerdem gaben die Einzelarbeiter oft an, dass ein Partner ihnen beim Lösen der Probleme weitergeholfen hätte. Auch in dem Pilot Test wurde angegeben, dass ein Student eher eine Pause gemacht hätte, wäre sein Partner nicht da gewesen. In einigen Fällen konnte ich beobachten, dass die Paare im Laufe der Zeit von zwei Wochen immer besser miteinander arbeiteten und schneller waren. Dies wurde mir auch durch Interviews bestätigt.

9 Ausblick

Ich hatte mir erhofft durch diese Arbeit die Vorteile die Paarprogrammierung hat aufzeigen und näher untersuchen zu können um Paarprogrammierung in der Lehre zu etablieren. Allerdings ergaben sich einige unerwartete Probleme (siehe Abschnitte 6.1.6, 6.2.3, 6.3.1 und 7.5) bei der Durchführung der Fallstudie, die dies erschwerten. Für spätere Fallstudien in diese Richtung wäre es gut, wenn der Professor das Ziel hätte Paarprogrammierung in seinem Kurs langfristig einzuführen, da durch die Autorität des Professors der Erfolg, Paarprogrammierung erfolgreich einzuführen und umzusetzen steigt. Wenige Studenten sind motiviert genug um sich für die Forschung freiwillig beobachten oder testen zu lassen, weshalb die Vorgabe durch den Professor hilfreich sein kann. Dies konnte ich in dem Pilot Test feststellen, bei dem ich mit Fragen und Bitten letztendlich zwei Paare fand, die sich von mir bei der Ausführung von Paarprogrammierung beobachten lassen wollten. Diese meldeten sich sehr zögerlich als mir die Tutorin mit ein paar motivierenden Sätzen zur Seite stand. Auch in dem Hauptexperiment des Blockkurses konnte man erkennen, dass die Studenten sich häufig beobachten ließen, weil dies in der Vorlesung angesagt wurde. Anfänglich beschwerten sich einige bei ihren Partnern. Auch der Fragebogen wurde schon am zweiten Tag sehr wenig ausgefüllt. Ich denke wäre der Fragebogen Teil des Aufgabenblatts gewesen hätten die Studenten dem Fragebogen mehr Aufmerksamkeit geschenkt. Was ich als sehr positiv empfunden habe war die lange Planungsphase. So war mir vor den Beobachtungen besser bewusst, was ich beobachten wollte und warum ich dies tat.

Allerdings ist zu erwähnen, dass eine solche Fallstudie schwierig ist wenn man dies zuvor noch nicht gemacht hat. Durch wenig Erfahrung ist man sich nicht bewusst wie genau man beobachten möchte und was vor allem wichtig ist. Da ich mich gegen eine Videoaufzeichnung entschieden hatte, schrieb ich jedes von mir beobachtete Detail auf. Auch dies empfand ich als positiv, da ich denke, das eine Videoaufzeichnung die natürliche Atmosphäre beeinflusst hätte. Die Studenten hätten sich gerade zu Beginn unterbewusst anders als es gewöhnlich der Fall gewesen wäre verhalten. Ich konnte am ersten Tag bemerken, dass sich die Studenten anders verhielten, wenn ich sie beobachtete. Dies konnte man an der Körperhaltung erkennen. Es wurde sich plötzlich aufgerichtet und umgeguckt. Die Fragen wurden leiser gestellt oder das Gespräch wurde unterbrochen und beide starrten auf den Monitor. Ich achtete sehr darauf die Studenten möglichst unauffällig zu beobachten und ich denke das das Aufstellen oder Mitführen einer Videokamera dies erschwert hätte. Außerdem kann ich weiterempfehlen die Fragebögen in gedruckter Form auszugeben. So hatte ich doch das ein oder andere Mal die Gelegenheit die Studenten auf das Ausfüllen des Fragebogens aufmerksam zu machen und sie zu überreden. Es fiel sofort auf wenn ein leerer Fragebogen an einem Platz liegen blieb oder mir nicht abgegeben wurde. Bei einem Online-Fragebogen wäre dies nicht der Fall gewesen und ich denke die Studenten wären dort auch weniger motiviert gewesen diesen auszufüllen. Da in der zweiten Woche die Aufgaben schwieriger wurden, blieb den Studenten oft nicht die Zeit die Fragebögen auszufüllen und nur wenige blieben gern länger um diesen auszufüllen. Ich denke es wäre von großem Vorteil wenn der Professor die gesamte Vorlesung so konzipieren würde, das Paarprogrammierung verpflichtend ist und dieses auch abgeprüft wird. Dies ist sicherlich schwer wenn nicht alle in einem Raum zusammenarbeiten oder Saros verwenden. Allerdings könnte dies in den Übungen durch unvorhergesehenes Auswählen zum Vorrechnen an der Tafel der Studenten abgeprüft werden. Auch empfand ich als Vorteilhaft das Planen der Beobachtungsgegenstände. So konnte ich mich jeden Tag auf einen bestimmten Beobachtungsgegenstand konzentrieren und konnte meinen Fokus darauf setzten. Es gelang mir jedoch gegen Ende immer weniger den Fokus auf nur einen Beobachtungsgegenstand zu legen und so schrieb ich sämtliche Eindrücke auf. Zu Anfang war ich allerdings erleichtert die Beobachtungsgegenstände geplant zu haben und ein Konzept zu haben. Auch würde ich die vorerst stille Beobachtung weiterempfehlen, wenn man die Möglichkeit hat die Paare über einen längeren Zeitraum zu beobachten. Erst in der zweiten Woche machte ich Interviews und sprach die Studenten an. Ich hatte das Gefühl zu diesem Zeitpunkt eher eine ehrliche Antwort zu bekommen, da die Studenten mich mittlerweile besser kennengelernt hatten. Allerdings wäre es wie in Kapitel 6.1.2 beschrieben besser gewesen, wenn man vereinzelt Studenten immer wieder zwischen den einzelnen Aufgaben nach ihrer Stimmung gefragt hätte und nicht nur im Fragebogen, der im Anschluss ausgefüllt wurde. Ich denke so hätte man eine genauerer Antwort auf die Frage bekommen. Zu Beginn hatte ich auch geplant die Studenten etwa zur Hälfte als Paarprogrammierer und zur Hälfte als Einzelarbeiter einzuteilen. So erhoffte ich mir ausgeglichene und vergleichbare Ergebnisse. Durch einen Tipp ließ ich mich darauf ein die Studenten allein entscheiden zu lassen ob sie das Konzept Paarprogrammierung ausprobieren wollten und so entschied sich eine große Mehrheit für die Paarprogrammierung. Dies zeigt einerseits, dass das Konzept für die Studenten vorteilhaft erscheint, andererseits sehen wir beispielsweise in Abschnitt 6.1.2 in Abbildung 5 das dieses ungleiche Verhältnis Probleme bei der Auswertung verursachen kann und Ergebnisse verfälscht werden. Wenn die Studenten nicht unter einem solchen Zeitdruck gestanden hätten, wären etwas intensivere Interviews gut gewesen. Es könnte in nachfolgenden Arbeiten untersucht werden ob Konflikte tatsächlich negativ sind und inwiefern sie sich auf die Zusammenarbeit auswirken. Konflikte könnten auch einen positiven Nutzen wie beispielsweise die Verbesserung der Codequalität haben. Außerdem könnte man den Rollenwechsel und das Verhalten der Paare in Zusammenhang mit den Persönlichkeiten der Studenten genauer untersuchen. Auch die Frage wie man

mit dem Wissensunterschied der Paare umgeht und wie man diesen in der Gruppenzusammenstellung berücksichtigt wäre eine sehr interessante Untersuchung. Die Ablenkung, die bei Einzelarbeitern stattfindet, könnte in einer späteren Arbeit auch noch einmal genauer untersucht werden.

Literatur

[1] Arbeitsgruppe Software Engineering. Eclipse plugins tested with saros. *(http://www.saros-project.org/CompatiblePlugins) Zuletzt abgerufen am 21.05.2012*, 2011.

[2] Arbeitsgruppe Software Engineering. Installation and set-up of saros. *Freie Universität Berlin, Institut für Informatik (http://www.saros-project.org/InstallUsing) Zuletzt abgerufen am 21.05.2012*, 2012.

[3] Arbeitsgruppe Software Engineering. Saros-home. *Freie Universität Berlin, Institut für Informatik (http://www.saros-project.org) Zuletzt abgerufen am 26.05.2012*, 2012.

[4] Pearl Brereton, Mark Turner, and Rumjit Kaur. Pair programming as a teaching tool: a student review of empirical studies. *22nd Conference on Software Engineering Education and Training*, pages 240–247, 2009.

[5] Alexander Bätz. Comparison of software products using software engineering metrics. *Seminarvortrag, Universität Paderborn (http://is.uni-paderborn.de/fileadmin/Informatik/AG-Engels/Lehre/WS0809/SE/Sonstiges/Seminar/Version1.0/Seminar.NAQ.Alexander.B%C3%A4tz.v1.0.pdf), Kapitel 4, Zuletzt abgerufen am 21.05.2012*, 2009.

[6] Gerardo Canfora, Aniello Cimitile, and Corrado Aaron Visaggio. Lessons learned about distributed pair programming: what are the knowledge needs to address ? *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003.

[7] Alistair Cockburn and Laurie Williams. The costs and benefits of pair programming. *(http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF) Zuletzt abgerufen am 21.05.2012*, pages 1–11, 2000.

[8] Brian Hanks, Charlie McDowell, and David Draper & Milovan Krnjajic. Program quality with pair programming in cs1. *ACM SIGCSE Bulletin*, pages 176–180, 2004.

[9] Jan Kettner. Persönlichkeitsprofile und paarprogrammierung. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/072_Kettner07_personality_traits_pp.pdf) Zuletzt abgerufen am 21.05.2012*, 2007.

[10] Lehrstuhl Software-Systemtechnik, BTU Cottbus. Software-entwicklung nach der "cleanroommethode. *(http://www.informatik.tu-cottbus.de/~rust/vorlesungen/PSP/cleanroom.pdf) Zuletzt abgerufen am 21.05.2012*, 2004.

[11] Raymond Lister, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, Beth Simon, and Lynda Thomas. A multi-national study of reading and tracing skills in novice programmers. 2004.

[12] Christoph Lofi. cgqm - ein zielorientierter ansatz für kontinuierliche, automatisierte messzyklen. *Collaborative Software Development Laboratory, University of Hawai'i Manoa (http://www.l3s.de/~lofi/publications/metrikon05-cGQM.pdf) Zuletzt abgerufen am 21.05.2012*, 2005.

[13] Emilia Mendes, Lubna Basil Al-Fakhri, and Andrew Luxton-Reilly. A replicated experiment of pair-programming in a 2nd-year software development and design computer science course. *ACM SIGCSE Bulletin*, 38, 2006.

[14] Harald A. Mieg and Matthias Näf. Experteninterviews in den umwelt- und planungswissenschaften. eine einführung und anleitung. *(http://www.mieg.ethz.ch/education/Skript_Experteninterviews.pdf) Zuletzt abgerufen am 21.05.2012*, 2005.

[15] Matthias M. Müller, Frank Padberg, and Walter F. Tichy. Ist xp etwas für mich ? empirische studien zur einschätzung von xp. *Fakultät fšur Informatik, Universität Karlsruhe, (http://subs.emis.de/LNI/Proceedings/Proceedings64/GI-Proceedings.64-20.pdf) Zuletzt abgerufen am 21.05.2012*, 2005.

[16] Jerzy Nawrocki and AdamWojciechowski. Experimental evaluation of pair programming. *European Software Control and Metrics*, 2001.

[17] John T. Nosek. The case for collaborative programming. *Communications of the ACM*, 41:105–108, 1998.

[18] [o.A]. Methoden empirischer sozialforschung. *(http://www.referate10.com/referate/Politik/3/Methoden-empirischer-Sozialforschung-reon.php) Zuletzt abgerufen am 21.05.2012*, [o.J].

[19] Briony J. Oates. *Researching Information Systems and Computing.* SAGE, 2006.

[20] Wojciech Polcwiartek. Charakterisierung des entwurfsprozesses bei der paarprogrammierung. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/149_polcwiartek_entwurfprozessPP_thesis.pdf) Zuletzt abgerufen am 21.05.2012*, 2009.

[21] Prof. Dr. Prechelt. Vorlesungsskript - softwaretechnik - teil 2. *Kapitel 18*, 2008.

[22] Prof. Dr. Prechelt. The cleanroom method. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/V-SWT2-2011/23_cleanroom.pdf) Zuletzt abgerufen am 26.02.2012*, 2011.

[23] Prof. Dr. Prechelt. Technisches schreiben. *Aus dem Skriptum Äusgewählte Kapitel der Softwaretechnik"(https://www.mi.fu-berlin.de/wiki/pub/SE/SeminarRegeln/Technisches_Schreiben.pdf) Zuletzt abgerufen am 25.05.2012.*, pages 198–209, [o.J].

[24] Prof. Dr. Lutz Prechelt. Die cleanroom-methode. *(http://page.mi.fu-berlin.de/prechelt/swt2/node61.html), Kapitel 7, Zuletzt abgerufen am 21.05.2012*, 1999.

[25] Marc Rintsch. Konzeption und explorative evaluation eines prozesses zur technischen betreuung des einsatzes eines werkzeugs zur verteilten, kollaborativen softwareentwicklung in echtzeit im betrieblichen umfeld (dpp xiv). *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/152_rintsch_saros-tech-betreuung-betrieb-thesis-concept.pdf) Zuletzt besucht am 21.05.2012*, 2009.

[26] Marc Rintsch. Technisches projektmanagement im opensource-projekt saros. *Bachelorarbeit, Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering, (http://www.inf.fu-berlin.de/inst/ag-se/theses/Rintsch10-saros-techn-projektmngt.pdf)*, 2010.

[27] Edna Rosen. Verteilte paarprogrammierung im industriellen umfeld - etablierung und qualitative empirische untersuchung. *(https://www.inf.fu-berlin.de/w/SE/ThesisDPPManagement) Zuletzt abgerufen am 21.05.2012*, 2009.

[28] Edna Rosen. Verteilte paarprogrammierung im industriellen umfeld etablierung und qualitative empirische untersuchung. *Diplomarbeit, Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering, (http://www.inf.fu-berlin.de/inst/ag-se/theses/Rosen09-DPP-management.pdf) Zuletzt abgerufen am 21.05.2012*, 2009.

[29] Edna Rosen. Verteilte paarprogrammierung im industriellen umfeld etablierung und qualitative empirische untersuchung. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/159_rosen_DPPMgmt-thesis.pdf) Zuletzt aufgerufen am 21.05.2012*, 2009.

[30] Edna Rosen, Stephan Salinger, and Christopher Oezbeck. Project kick-off with distributed pair programming. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/183_rosen_project_kickoff_DPP.pdf) Zuletzt abgerufen am 21.05.2012*, 2010.

[31] T. Schlesinger. Forschungsmethoden der sportwissenschaft. *(http://www.tu-chemnitz.de/hsw/sportwissenschaft/sportsoziologie/dateien/Qualitative_Verfahren-Beobachtung.pdf) Zuletzt abgerufen am 21.05.2012*, 2008.

[32] Prof. Dr. Detlef Sembill and Dr. Jürgen Seifried. Implementation innovativer unterrichtskonzepte als führungsaufgabe. *(http://www.uni-bamberg.de/fileadmin/uni/fakultaeten/sowi_lehrstuehle/wirtschaftspaedagogik/Dateien/Forschung/Publikationen/SL_Symposium_short.pdf) Zuletzt abgerufen am 21.05.2012*, 2004.

[33] Alan Shaw. Extending the pair programming pedagogy to support remote collaborations in cs education. *Information Technology: New Generations, Third International Conference*, pages 1095–1099, 2009.

[34] Christophe Sokol. Einführung von saros in alp 2. *Studienarbeit, Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering*, 2011.

[35] Eike Starkmann. Verteilte paarprogrammierung in open source software projekten: Bisherige erkenntnisse und erfahrungen im bereich der kontaktaufnahme. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/164_starkmann_dpp_status.pdf) Zuletzt abgerufen am 21.05.2012*, 2009.

[36] Eike Starkmann. Verteilte paarprogrammierung in open source software projekten. *Diplomarbeit, Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering (http://www.inf.fu-berlin.de/inst/ag-se/theses/Starkmann10-DPP-OSS.pdf)*, 2010.

[37] Eike Starkmann. Verteilte paarprogrammierung in open-source-softwareprojekten. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/181_starkmann_DPPIX-thesis.pdf) Zuletzt abgerufen am 21.05.2012*, 2010.

[38] Ulrich Stärk. Empirische bewertung agiler softwareentwicklung controlled case study. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/075_ControlledCaseStudy.pdf) Zuletzt abgerufen am 21.05.2012*, 2007.

[39] Ulrich Stärk. Empirische untersuchungen des side-by-side programming. *(http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-BSE/114_staerk_side-by-side.pdf) Zuletzt besucht am 21.05.2012*, 2008.

[40] Miao Wang. Pair programming. *(http://www.miaowang.de/studium/papers/PP_Slides.pdf) Zuletzt abgerufen am 21.05.2012*, 2006.

[41] Miao Wang. Pair programming. *Ausarbeitung eines Seminarvortrags, Freie Universität Berlin, Institut für Informatik, (http://www.miaowang.de/studium/papers/PP_Paper.pdf) Zuletzt abgerufen am 21.05.2012*, 2006.

[42] Wikipedia. Goal question metric. *(http://de.wikipedia.org/wiki/Goal_Question_Metric) Zuletzt abgerufen am 27.05.2012*, 2012.

Abbildungsverzeichnis

1 Überblick über die Beteiligung an den Fragebögen 30
2 Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der ersten Woche bei den Paarprogrammierern 31
3 Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der ersten Woche bei den Einzelarbeitern 32
4 Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der zweiten Woche bei den Paarprogrammierern 34
5 Übersicht der Arbeitsatmosphäre und Schwierigkeit des Aufgabenblatts der zweiten Woche bei den Einzelarbeitern 35
6 Verwendete Ressourcen der Paarprogrammierer zur Problemlösung in der ersten Woche 36
7 Verwendete Ressourcen der Einzelarbeiter zur Problemlösung in der ersten Woche . . 37
8 Verwendete Ressourcen der Paarprogrammierer zur Problemlösung in der zweiten Woche 38
9 Verwendete Ressourcen der Einzelarbeiter zur Problemlösung in der zweiten Woche . . 39
10 Beteiligung der Studenten an den Tests 48
11 Ergebnis des ersten Tests 49
12 Ergebnisse des zweiten Tests 49

Tabellenverzeichnis

1 Übersicht der Beobachtungsannahmen . . . 20
2 Positive Eigenschaften von Paarprogrammierung . . . 23
3 Negative Eigenschaften von Paarprogrammierung . . . 23
4 Ursachen die das Lösen des Aufgabenblatts erleichterten . . . 23
5 Gründe den Inhalt der Vorlesung durch Paarprogrammierung besser zu verstehen . . . 24
6 Angaben von Paarprogrammierern zu positiven Eigenschaften von Paarprogrammierung am ersten Tag . . . 40
7 Angaben von Paarprogrammierern zu negativen Eigenschaften von Paarprogrammierung am ersten Tag . . . 41
8 Angaben von Paarprogrammierern zu positiven Eigenschaften von Paarprogrammierung zwischen dem ersten und dem letzten Tag . . . 42
9 Angaben von Paarprogrammierern zu negativen Eigenschaften von Paarprogrammierung zwischen dem ersten und dem letzten Tag . . . 43
10 Angaben von Paarprogrammierern zu positiven Eigenschaften von Paarprogrammierung am letzten Tag . . . 44
11 Angaben von Paarprogrammierern zu negativen Eigenschaften von Paarprogrammierung am letzten Tag . . . 45
12 Angaben von Einzelarbeitern zu negativen Eigenschaften vom einzeln arbeiten am ersten Tag . . . 45
13 Angaben von Einzelarbeitern zu positiven Eigenschaften vom einzeln arbeiten zwischen dem ersten und letzten Tag . . . 46
14 Angaben von Einzelarbeitern zu negativen Eigenschaften vom einzeln arbeiten zwischen dem ersten und letzten Tag . . . 46
15 Angaben von Einzelarbeitern zu positiven Eigenschaften vom einzeln arbeiten am letzten Tag . . . 47

10 Anhang

10.1 Fragebogen zum Kurs "Softwareprozesse

Fragebogen zur Paar-Programmierung

Erläuterungen von Abkürzungen:

PP = Pair Programming – Paar-Programmierung
DPP = Distributed Pair Programming – verteilte Paar-Programmierung

1. Hast du schon Erfahrungen mit PP/DPP gemacht?
   ○ Ja
   ○ Nein

   a) Welche positiven Eigenschaften konntest du daraus ziehen?

   b) Welche negativen Eigenschaften konntest du daraus ziehen?

2. Fiel dir das Lösen des Zettel leichter?
   ○ Ja
   ○ Nein

   a) Was glaubst du war die Ursache dafür?

3. Wie viel Spaß hat es dir gemacht mit deinem Partner das Übungsblatt zu lösen?

   ○ Ich hatte sehr viel spaß
   ○ Ich hatte spaß
   ○ Es ging so
   ○ Ich hatte kaum spaß
   ○ Es war schrecklich

4. Fiel es dir schwer mit deinem Partner zu kommunizieren?
   - ○ Nein, überhaupt nicht
   - ○ Es ging so
   - ○ Ja, es fiel mir schwer

5. Konntest du mit deinem Partner Dinge, die du nicht verstanden hast offen klären?
   - ○ Nein
   - ○ Es ging so
   - ○ Ja

6. Bist du der Meinung durch das Anwenden von PP/DPP mehr zu lernen bzw. den Inhalt der Vorlesung besser zu verstehen?
   - ○ Ja , weil _____
   - ○ Nein, weil _____

7. Glaubst du, dass das Erlernen von PP/DPP für das späterer Berufsleben von Vorteil ist?
   - ○ Ja , weil _____
   - ○ Nein, weil _____

8. Welche Voraussetzungen müssten deiner Meinung nach in der Universität geschaffen werden, um PP/DPP sinnvoll einsetzen zu können?

9. Hattest du den Eindruck, dass das Lösen des Übungszettels schneller ging?
   - ○ Ja
   - ○ Nein

10. Denkst du, dass PP/DPP für das Lösen der Übungsaufgaben sinnvoll ist?
    - ○ Ja , weil _____
    - ○ Nein, weil _____

11. Wenn du Saros kennst, wie ist dein Eindruck von Saros bzw. wie zufrieden bist du mit der Software?
    - ○ Ich bin zufrieden
    - ○ Ich bin weder zufrieden noch unzufrieden
    - ○ Ich bin unzufrieden

    a) Wieso hast du für die Sitzung Saros nicht verwendet?

    b) Gibt es Verbesserungsvorschläge?

10.2 Beobachtungsvorlage für die Teilnehmer des Kurses "Softwareprozesse"

Beobachtungsvorlage

Eckdaten

Gruppe:
Datum:
Uhrzeit (Von-Bis):
Übungsblatt:

Zeit

Vorbereitungszeit:
Zeit für die Aufgaben:

Alle Aufgaben erledigt:
Konzentrierte Arbeitszeit:

Zwischengespräche/Ablenkungen:

Motivation

Wie war die Atmosphäre zu Beginn (ersten 15 min):

Wie war die Atmosphäre zwischendurch:

Wie war die Atmosphäre am Ende:

1

Wie oft wurde gelacht:

Blickkontakt:

Kommunikation:

Rollenverteilung:

Hilfestellungen:

Notizen

10.3 Übersicht über die Beobachteten Gruppen des Blockkurses "Programmieren in Java"

Gruppen-bezeichnung	Name	Programmier-erfahrung	Beschreibung
Die Normalen	Heiko	etwa 100 Zeilen	Beide arbeiteten gut zusammen.
	Frank	etwa 500 Zeilen	
Die Drei	Linda	etwa 100 Zeilen	Linda und Nele arbeiteten gut zusammen, ließen sich allerdings von dem Einzelarbeiter Pawel oft ablenken.
	Nele	etwa 100 Zeilen	
	Pawel	keine	
Der Ruhige	Karl	etwa 500 Zeilen	Er war sehr still und arbeitete konzentriert ohne die Hilfe der anderen Studenten in Anspruch zu nehmen.
Die Respektvollen	Saskia	keine	Obwohl beide ein unterschiedliches Niveau an Programmiererfahrungen mitbrachten, arbeiteten sie sehr gut zusammen.
	Sven	etwa 500 Zeilen	
Die Stillen	Henrik	etwa 100 Zeilen	Beide kommunizierten sehr wenig miteinander. Henrik hatte die ersten Tage den Driver, während Alesandro still und angestrengt daneben saß.
	Alesandro	keine	
Die Einzelarbeiter	Sophie	keine	Die erste Woche schienen beide sehr gut zusammen zu arbeiten, allerdings ergaben die Fragebögen etwas anderes und beide arbeiteten in der zweiten Woche getrennt.
	Finn	etwa 500	
Die Fairen	Johannes	etwa 100 Zeilen	Beide arbeiteten sehr gut zusammen und kommunizierten sehr gut miteinander.
	Martin	etwa 100 Zeilen	
Die Abwesenden	Tim	etwa 100 Zeilen	Tim wirkte oft abwesend und Hannes hatte zwei Wochen lang den Driver.
	Hannes	etwa 500 Zeilen	
Die Kämpfer	Angélique	keine	Beide hatten die ersten Tage große Schwierigkeiten mit dem Lösen der Übungsblätter, verbesserten sich allerdings stark.
	Mylos	keine	
Die Dreiergruppe	Paul	etwa 100 Zeilen	Die ersten Tage arbeitete Paul einzeln, da er keinen Partner fand. Als Kai einen halben Tag lang fehlte lernte Paul Richard kennen und arbeitete mit ihm zusammen. Am nächsten Tag hatte Paul den Raum gewechselt und arbeitete mit beiden zusammen, obwohl Dreiergruppen nicht gestattet waren.
	Richard	etwa 500 Zeilen	
	Kai	etwa 100 Zeilen	
Der Hilfsbereite	Jack	1000 Zeilen und mehr	Jack war Einzelarbeiter, da er der Meinung war allein schneller zu arbeiten. Allerdings half er den restlichen Paarprogrammierern im Raum stets und blieb noch, obwohl er mit dem Lösen des Übungsblatts bereits fertig war. Er fragte auch selten den Tutor nach Hilfe.
Die Einzelgänger	Jan	keine	Beide arbeiteten von Anfang an nicht gut zusammen und kommunizierten sehr wenig miteinander.
	Luka	1000 Zeilen und mehr	
Die Gutgelaunten	Nina	etwa 100 Zeilen	Beide arbeiten gut zusammen, baten allerdings häufig einen Einzelarbeiter um Hilfe.
	Mia	keine	
Die Lustigen	Jenny	keine	Ralf hatte immer den Driver und die erste Woche hatten sie einige Schwierigkeiten mit dem Lösen des Aufgabenblatts.
	Ralf	keine	

Der Fragensteller	René	keine	Er arbeitete die erste Woche einzeln und hatte ab und zu einen Partner, wobei sie dann auch nicht zusammen arbeiteten. Er lenkte die restlichen Paare im Raum häufig ab und fragte viel.
Die Perfekten	Malte	1000 Zeilen und mehr	Beide wendeten das Konzept Paarprogrammierung perfekt an und wechselten sich regelmäßig ab.
	Max	etwa 500 Zeilen	
Die Unbekannten	Dirk	1000 Zeilen und mehr	Jasmin war die ersten Tage sehr still und zurückhaltend, die änderte sich allerdings nachdem sich beide etwas besser kennenlernten. Dirk hatte die zwei Wochen über den Driver.
	Jasmin	etwa 100 Zeilen	
Die Freundinnen	Lisa	etwa 500 Zeilen	Beide arbeiteten gut zusammen. Die erste Woche wechselten sie sich mit der Driver-Rolle ab. Gegen Ende hatte Lara öfter die Driver-Rolle.
	Lara	etwa 500 Zeilen	
Die Zurückhaltenden	Andreas	keine	Bastian hatte immer die Driver-Rolle, erklärte seinem Partner allerdings auch viel.
	Bastian	etwa 500 Zeilen	
Die Spaßvögel	Patrick	etwa 500 Zeilen	Beide wechselten sich mit der Driver-Rolle ab und kommunizierten gut.
	Florian	etwa 100 Zeilen	
Die Freunde	Jakob		
	Nicole		
Die Vorbildlichen	Emma	keine	Beide wendeten Paarprogrammierung vorbildlich an und gingen aufeinander ein.
	Noah	etwa 500 Zeilen	
Die Ordentlichen	Kevin	etwa 500 Zeilen	Beide wechselten sich mit der Driver-Rolle ab, oft tippte allerdings einer mehr als der Andere.
	Niklas	keine	

10.4 Ausführliche Beschreibung der Beobachten die ersten Tage im Kurs "Programmieren in Java"

10.4.1 Erster Tag

Am ersten Tag sollte die Vorbereitungszeit, die Rollenverteilung und Hilfestellung, sowie Einzelarbeiter auf Kommunikation untersucht werden.
Auf dem ersten Übungsblatt sollten sich die Studenten mit Datentypen auseinandersetzen und diese verstehen. Hierzu gab es eine Aufgabe, ähnlich der ersten Aufgabe des Tests, in der die Studenten Ausdrücke auswerten und den dazugehörigen Datentyp angeben sollten. Diese Aufgaben sollte ohne Hilfe des Computers auf dem Blatt gelöst werden. Die zweite Aufgabe war auch einer Aufgabe im Test ähnlich und bestand darin den Inhalt der Variablen nach den angegebenen Zuweisungen anzugeben. Bei der dritten Aufgabe sollte ein Programm, welches in der Vorlesung vorgestellt wurde, modifiziert werden. Für die vierte und letzte Aufgabe bekamen die Studenten ein Beispielprogramm, in dem sie einige Veränderungen vornehmen sollten, sodass die damit gezeichneten Augen die sich in Richtung des Mauspfeils bewegten, verändert werden.

Zu Beginn warteten die Teilnehmer und ich vor den Arbeitsräumen auf die Tutoren, die dann die Räume aufschlossen. Die Studenten verteilten sich selbstständig auf die vier zur Verfügung stehenden Räume. Nachdem ich die Listen an die Tutoren verteilt hatte und nochmals um Namensschilder bat, fing ich im vierten Raum mit den Beobachtungen an. Ich fragte einen Studenten, ob ich mich neben ihn setzen kann, um ihn zu beobachten und Jack willigte ein. Von diesem Platz aus konnte ich insgesamt fünf Studenten beobachten. Ich beobachtete einen Einzelarbeiter in Hinsicht auf Kommunikation mit seinem Umfeld. Er saß allerdings abgewandt von der restlichen Gruppe im Raum und arbeitete völlig selbstständig und beanspruchte die Hilfe des Tutors äußerst selten. Jack konnte bereits vor diesem Kurs einige Programmiererfahrungen sammeln, wie sich anhand des Umfragebogens herausfinden ließ. Er war einer der sehr erfahrenen Studenten, die bereits zuvor etwa 1000 Zeilen und mehr implementiert hatten. Auch benötigte er für die erste Aufgabe nur drei Minuten Vorbereitungszeit, in der er das Übungsblatt herunterlud, sowie einen Block und einen Stift herausholte und war nach weiteren neun Minuten mit der ersten Aufgabe fertig. Nach weiteren sechs Minuten löste er die zweite Aufgabe. Er arbeitete sehr konzentriert und zielgerichtet. Einmal in der Zeit als ich ihn beobachtete nahm er für eine kurze Frage die Hilfe des Tutors in Anspruch, da er sehr leise redete verstand ich die Frage nicht genau, es ging wahrscheinlich um den Vorlesungsstoff in den Folien, da er diese vorher kurz durchgeguckt hatte und dann eine Frage stellte. Anschließend dachte er einige Minuten nach und arbeitete dann weiter.

Außerdem konnte ich ein Paar Jenny und Ralf beobachten, die bisher noch keinerlei Programmiererfahrungen sammeln konnten, wie es sehr oft mit einem Einzelarbeiter Namens René kommunizierte, der ebenfalls im Programmieren noch unerfahren war. Sie diskutierten zu Dritt über die erste Aufgabe, wobei sich René zu dem Paar setzte. Das Paar nahm schon am ersten Tag die Hilfe des Tutors öfters in Anspruch. Es wurde über die Bit-Operatoren, die für die erste Aufgabe benötigt wurde gesprochen. Zudem beobachtete ich, wie sie Besuch von einem befreundeten Teilnehmer Dirk aus einem anderen Raum bekamen, der sich kurz hinter sie stellte und auf den Monitor schaute. Nach einem kurzen Gespräch kehrte er zu seiner ihm zugeteilten Übungspartnerin Jasmin zurück.

Bei Dirk und Jasmin fiel mir auf. dass die Partnerin zu Beginn sehr still war und Dirk den Driver übernahm. Dirk verfügte schon zu Beginn über eine sehr hohe Programmiererfahrung von etwa

mehr als 1000 Zeilen, während Jasmin weniger Erfahrung mitbrachte und etwa 100 Zeilen zuvor programmierte. Sie wechselten sich auch den ganzen Tag über nicht ab, wobei von keinem der Beiden die Initiative dafür ausging. Sie schienen beide zufrieden mit der momentanen Rollenverteilung zu sein. Allerdings kann man nicht behaupten, dass die etwas stillere Partnerin nicht aufmerksam genug war oder abwesend wirkte. Sie machte ihren Partner auch auf einen Fehler aufmerksam. Wenn Dirk ihr etwas erklärte oder sagte, nickte Jasmin oft allerdings nur. Ich nehme an, da sich die beiden Studenten vorher noch fremd waren, das sie sich erst einmal näher kennenlernen mussten und deshalb am ersten Tag jeder etwas still für sich gearbeitet hat. Zwischendurch kam ein Student aus dem anderen Raum, den Dirk zuvor besuchte, auf die Beiden zu und unterbrach sie bei ihrer Arbeit. Sie redeten über die letzte Aufgabe und sprachen sich ab, wie lange sie noch programmieren wollten, wobei Jasmin ruhig daneben saß und zuhörte.

Im selben Raum konnte ich zwei Paare dabei beobachten, wie sie sich gegenseitig untereinander halfen. Als sie auf keine Lösung kommen fragen sie gemeinsam den Tutor zur letzten Aufgabe. Bei einem dieser Paare Lara und Lisa, die beide vorher etwa 500 Zeile Code implementierten, wurde gleich am ersten Tag ein relativ gleichverteilter Rollenwechsel vollzogen. Während ich Lisa zuerst als Driver beobachtete und sie mehr die Führung übernahm, wechselten sie sich nach etwa 25 Minuten ab und Lara übernahm den Driver. Obwohl Lisa Oberserver war, leitete sie oft die Gespräche und stellte dem Tutor Fragen zur letzten Aufgabe. Sie bekommen außerdem Hilfe von einem Paar, welches etwas entfernt von ihnen sitzt und sogar aufsteht und um den Tisch läuft, um ihnen etwas zu erklären. Sie geben ihnen öfters Tipps und versuchen den beiden Frauen zu helfen. Lisa gibt den Jungs eine Rückmeldung: “ Compiler kann nicht kompilieren.” und sie beraten sich. Die Jungs Patrick und Florian verfügten bereits über ähnlich viel Programmiererfahrungen. Patrick hatte bisher etwa 500 Zeilen Code geschrieben und Florian etwa 100. Driver Florian fragte Patrick ob sie die Augen wirklich so lassen wollten und schlug vor etwas in die Mitte zu machen. Patrick verhielt sich sehr ruhig. Nach etwa 15 Minuten bekam ich einen Rollenwechsel der beiden mit, wobei Patrick die Tastatur förmlich an sich riss.

Nina verfügte bereits über ein wenig Programmiererfahrung von etwa 100 Zeilen und Mia war noch völlig unerfahren. Als ich die Beiden beobachtete hatte Nina die Driver-Rolle und fragte Mia etwas über Datentypen. Da auch Mia ihr nicht weiterhelfen konnte schauten sie sich die Folien an und ließen sich vom Tutor die Datentypen Int und Float erklären. Ich beobachtete sie oft dabei wie sie ihr Wissen mit den Folien aus der Vorlesung auffrischten, wobei Nina die Maus bediente und durch die Folien blätterte. Als sie die ersten beiden Aufgaben lösen, schreiben beide die Lösungen jeweils auf ein Blatt, ansonsten schien eher Nina das Lösen der Aufgaben anzuführen, wobei sie Mia auch etwas erklärt. Beide brauchten etwa eine Vorbereitungszeit von sechs Minuten, in der sie über ihre Blätter gebeugt sitzen, sich die Vorlesungsfolien anschauen und mäßig miteinander kommunizieren. Es fällt auf, dass an diesem Tag die meisten Fragen zur letzten Aufgabe gestellt wurden, bei der die Studenten durch ausprobieren das Aussehen von Augen, die sich nach der Bewegung der Maus richten, verändern sollten. Auch bei einem weiteren Paar konnte ich beobachten, dass derjenige der den Driver hatte den Tutor rief und ihm eine Frage stellte, sowie Dinge erklärte.

Emma, die bisher keine Erfahrungen im Programmieren sammeln konnte und Noah, der etwa 500 Zeilen Code erstellte, bereiteten sich bei einer Aufgabe vor, indem sie sich den Code zusammen anschauten. Ich konnte sehen, dass sie viele verschiedene Dinge auf ihrem Computer geöffnet hatten, zwischen denen sie auch häufig wechselten und versuchten den Code zu verstehen. Auch dieses

Paar fragte den Tutor etwas zur vierten Aufgabe. Während meiner Beobachtung hatte Noah die Driver-Rolle, jedoch brachte sich Emma sehr gut ein und machte deutlich darauf aufmerksam, dass sie bei den Augen eine Wunschfarbe hätte und wies ihren Partner an, diese zu ändern. Sie versucht ihn anzuleiten und ihm zu zeigen wo er welche Attribute ändert, woraufhin Noah sie aufmerksam macht, dass ihre Idee so nicht funktioniert und umzusetzen ist. Sie versuchte ihn erneut anzuleiten und ändert etwas im Programm. Anschließend fragt sie ihn, wo man etwas spezielles ändern könnte. Zusammen suchten sie im Code nach der passenden Stelle. Noah versuchte Emma etwas zu erklären und sie probierten es dann anschließend gemeinsam aus. Dies war für mich das auffälligste Paar an diesem Tag, da sie das Konzept der Paarprogrammierung sehr gut und fair umsetzten. Auch hatte ich das Gefühl, dass beide sehr gut aufeinander eingingen.

Sophie hatte im Gegensatz zu Finn noch keinerlei Programmiererfahrungen. Finn hatte schon etwa 500 Zeilen Code produziert. Beide zusammen stellen sie dem Tutor eine Frage. Jedoch fragte Sophie anschließend noch einmal bei ihren Sitznachbarn nach. Finn hatte die Driver-Rolle und Sophie saß daneben und war sehr bemüht sich einzubringen.

Angélique und Mylos verfügten beide über keine Programmiererfahrung. Der Tutor ließ sich von beiden die gelösten Aufgaben erklären. Es fiel auf, dass Angélique ebenfalls die Aufgabe gut erklären konnte, obwohl Mylos oft die Driver-Rolle hatte. Angélique versuchte sich oft einzubringen und nahm ab und zu die Maus in die Hand. Ihr kam die Idee etwas in dem Programm der vierten Aufgabe zu ändern. Obwohl Mylos dagegen war, änderte sie die Größe der Augen, wobei ihr ein Fehler unterläuft und eine Augenbraue nicht mehr angezeigt wird.

Ihnen Gegenüber saß ein weiteres Paar. Alesandro verfügte ebenfalls über keine Programmiererfahrungen und Henrik hatte etwa 100 Zeilen Code geschrieben. In Hinsicht auf die mitgebrachten Erfahrungen unterschied sich also ihr Niveau nicht sehr stark. Henrik hatte die Driver-Rolle und erklärte Alesandro was er machte. Alesandro war jedoch sehr still und schaute angestrengt auf den Bildschirm. Im Fragebogen gab Alesandro auch an, dass er weder Spaß noch keinen Spaß hatte und die Arbeitsatmosphäre als normal und zielstrebig einschätzte. Ihm fiel das Lösen des Übungsblatts nicht leicht, sowie auch das Kommunizieren mit seinem Partner, da es ihm zu schnell ging. Auch merkte er an, dass ihm der Partner keine richtigen Erklärungen lieferte und er nur die Lösung hatte. Allerdings erwähnte er dies seinem Partner gegenüber nicht. Auch war er nicht ganz zufrieden mit der Lösung der Aufgaben, da sie zwar alles gelöst hatten, er allerdings diese nicht ganz verstanden hatte.

Es gab auch ein Paar welches ich für zwei Einzelarbeiter hielt. Jan verfügte über keine Programmiererfahrungen und Luka war einer der wenigen, die bereits zuvor mehr als 1000 Zeilen Code geschrieben hatten. Beide saßen getrennt an zwei Computern, unterhielten sich aber viel. Als ich die Beiden nach ihrem Namen fragte, fragte ich sie warum sie nicht zusammenarbeiteten, wenn sie sich schon so häufig austauschten, wobei Luka erklärte und Jan die Fragen stellte. So fand ich heraus, dass sie eigentlich Paarprogrammierung ausprobieren wollten und machte sie darauf aufmerksam, dass sie dazu nur einen Computer bräuchten.

Ein Student kam aufgrund einer Klausur sehr verspätet an und hatte für alle Aufgaben etwa eine Stunde Zeit. Die meisten Teilnehmer waren zu dieser Zeit bereits am zusammenpacken. Es stellte sich heraus, dass Paul lieber einen Partner hätte, was jedoch zu diesem Zeitpunkt nicht mehr mög-

lich war. Auch am nächsten Tag fand sich niemand, der Paarprogrammierung machen wollte und so arbeitete Paul vorerst einzeln. Er bekam eine sehr intensive Betreuung von Tutor, da er der Einzige noch verbliebene Student im Raum war und sehr viele Fragen stellte. Durch die anstehende Klausur verpasste er die Hälfte der Vorlesung und kam deshalb nur schwer mit den ihm gestellten Aufgaben zurecht. Der Tutor erklärte ihm wie die Bit-Operatoren funktionieren, vor allem geht er näher auf das Oder und das Zweierkomplement ein. Der Tutor erklärte ihm wie die Werte auf der Vorlesungsfolie zustande kommen. Durch den Zeitmangel schlug der Tutor Paul vor ihm erst einmal die letzten beiden Aufgaben zu erklären, sodass er zu Hause mit dem Compilieren des Codes zurecht kommen würde. Dabei fällt auf, dass Paul sehr unsicher ist und er gibt zu, dass er sehr unerfahren mit dem Umgang mit Computern ist. Beispielsweise zeigt der Tutor ihm, wie man ein Verzeichnis anlegt und sich darin bewegt. Ich fragte den Tutor, ob er allen Studenten so viel gezeigt hat bei den Aufgaben, er verneinte dies und teilte mir mit, dass die anderen Studenten mehr selbst probiert haben. Da jedoch Paul die Zeit fehlte, half er ihm. Außerdem erklärt der Tutor ihm die Prioritäten bei Bit-Operatoren, da Paul nicht klar ist, warum er auf einen anderen Wert kommt als in den Folien beschrieben. Es folgten fünf Minuten, in denen Paul nachdachte und in den Folien blätterte, bis er wieder den Tutor rief und um Hilfe bat. Der Tutor erklärt ihm die Negation, bedingte Ausdrücke und den Code aus der dritten Aufgabe, den Paul aus den Folien falsch kopiert hatte. Des weiteren zeigt der Tutor ihm einige Konsolen befehle mit denen er den Code kompilieren kann. Zur Vorbereitung guckt sich Paul etwa 17 Minuten lang die Folien der Vorlesung an und versucht nebenbei die Aufgabe auf dem Blatt zu lösen. Allerdings ist er mehr damit beschäftigt immer wieder in die Folien zu schauen. Dann guckt er sich 12 Minuten die Aufgabenstellung, sowie die Folien an und googled. Für die vierte Aufgabe bereitete er sich 10 Minuten mit dem Tutor vor.

10.4.2 Zweiter Tag

An diesem Tag wurde die konzentrierte Arbeitszeit und die Stimmung untersucht. Dabei habe ich mir wieder einen Überblick über alle Teilnehmer verschafft. Auffällig war, das an diesem Tag das konzentrierte Arbeiten oft durch die Tutoren unterbrochen wurde. Die Tutoren kontrollierten oft die Lösungen der Aufgaben und fragten häufig nach, ob die Studenten gut voran kamen oder Hilfe brauchten. Dies kam wahrscheinlich zu Stande, da die Tutoren an diesem Tag selten gefragt wurden und nichts zu tun hatten. Also gingen sie öfters herum und schauten sich den Code der Studenten an und gaben ihnen Tipps. Viele Ablenkungen kamen auch zu Stande, da in dem Originalprogramm welches die Studenten erweitern sollte einige Fehler waren. Daher machten die Tutoren oft Ansagen und auch Frau Prof. Esponda kam und machte auf Fehler aufmerksam.
In der ersten Aufgabe sollte die Mersennen-Zahl mit Hilfe einer vorgegebenen Klasse, die in der Vorlesung besprochen wurde, berechnet werden. In der zweiten Aufgabe sollte nach der Eingabe von einer Seitenlänge ein Dreieck mit Hilfe einer vorgegebenen Formel berechnet werden, die die Eingabe der Seitenlänge verwendete. Auch sollte ein vorgegebenes Programm, welches einen Glücksspieler simulierte, verändert werden und um einige Funktionen erweitert werden. In der vierten Aufgabe wurde dazu aufgefordert einige Klassen von der Veranstaltungsseite zu kopieren. Einige leere Klassen sollten noch selbständig von den Studenten implementiert werden. Auch in der letzten Aufgabe sollten die Studenten einige Dateien kopieren und fehlende Klassen implementieren. Es sollten anschließend einige Bilder von Mosaiken entstehen.

Heiko und Frank arbeiteten sehr konzentriert. Beide hatten schon vorher Erfahrungen im Programmieren sammeln können, wobei Frank etwas mehr Erfahrungen hatte und schon etwa 500 Zeilen Code

programmierte, während Heiko 100 Zeilen Code schrieb. Beide lächelten und hatten ab und an Blickkontakt, vor allem wenn sie über die Aufgabe diskutierten. Frank zeigt Heiko etwas auf dem Monitor und sie denken nach. Zwischenzeitlich wurde es bei den Beiden sehr ruhig und sie lasen viel, während Frank mit dem Kabel der Maus rumspielte und etwas vor sich hin nuschelte. Ansonsten kommunizierten beide sehr gut miteinander und sprachen sich ab. Die Rollenverteilung schien sehr ausgewogen. Beide lachten über eine Aufgabe die sie bearbeiteten, da sie den Fehler nicht fanden. Zwischendurch freuen sich beide sichtlich, da sie für das Lösen der Aufgabe nur etwas abtippen mussten. "Dit find ich jut", sagt Heiko während er in die Hände klatscht. Sie lachen viel, als die Aufgabe funktioniert. Auch herrscht eine ausgelassene Stimmung, als das Bild eines Mosaiks auf dem Bildschirm erscheint und merkwürdig aussieht. Am Ende war zu bemerken, dass sich beide nicht einig waren. Während der eine Schluss machen wollte, fragte der Andere noch: "Meinst du nicht das schaffen wir noch irgendwie?". Der Partner schüttelte den Kopf und war der Meinung, dass sie die Aufgabe nicht mehr schafften. Außerdem bestand großes Interesse daran Essen zu gehen: "Zeit was zu essen.". Nach Zustimmung des Partners gingen beide: "Das seh ich auch so.".

Linda und Nele hatten beide zuvor etwa 100 Zeilen Code geschrieben. Als ich sie anfänglich beobachtete schienen sie sehr konzentriert, was sich allerdings nach etwa vier Minuten schon änderte, als sie den Tutor fragten und sich anschließend mit einem Einzelarbeiter, Pawel besprachen. Pawel hatte noch keinerlei Erfahrung im Programmieren. Sie bekamen häufig an diesem Tag Hilfe vom Tutor, auch Frau Prof. Esponda gab ihnen Ratschläge. Sehr auffällig war, dass sie Pawel sehr oft halfen. Allerdings machte auch Pawel auf Fehler aufmerksam, beispielsweise auf eine fehlerhafte Variablendeklaration. Trotz der Ablenkungen schafften die beiden Frauen es sich sehr schnell wieder auf die Aufgabe zu konzentrieren. Ich beobachtete Pawel hingegen sehr oft dabei wie er zu anderen Paaren schaute und nicht sehr konzentriert war. Leider füllte dieser Teilnehmer keinen Fragebogen aus, sodass ich hier nur meine Beobachtungen wiedergeben kann. Linda und Nele hatten bislang nicht sehr viel Blickkontakt, allerdings freuten sich beide gemeinsam als sie die Aufgabe fertig bekamen und streckten sich erleichtert. Sie waren etwa zwei Minuten eher fertig als Pawel, mit dem sie sehr oft diskutierten. Einmal kippte die Stimmung allerdings ins Negative. Die Situation entstand schnell und war plötzlich da. Die Diskussion wurde lauter und angeregter und der Observer hatte eine negative Mimik, da sie wollte, dass der Driver das Programm testete und dieser das nicht tat. Auch Pawel war bei der Diskussion beteiligt, allerdings nur als stiller Beobachter. Beide Frauen sahen sich angenervt an und fingen an zu lachen. Dies entschärfte die vorher negative Stimmung und die Beiden arbeiteten wie gewohnt weiter und lächelten über weitere Probleme. Zwischenzeitlich fing Linda an zu lachen, da ihr auffiel, dass sie einiges aus dem Programm gelöscht hatte, woraufhin Nele und Pawel ebenfalls anfingen zu lachen. Nele gab Linda einen Tipp, den Linda ausprobierte und der funktionierte. Währenddessen hatte Pawel Schwierigkeiten beim Lösen einer Aufgabe. Er lehnte sich mit der Hand an der Stirn mit dem Ellenbogen auf den Tisch und schnaufte gut hörbar. Daraufhin kam der Tutor und half ihm. Gegen Ende ließ sich Linda von Pawel so sehr ablenken, dass Nele allein weiter programmierte, während Linda Pawel half. Dann wird auch Nele von Pawel unterbrochen und sie diskutieren über die Aufgabe. Gegen Ende scheint Linda müde zu sein, gähnt des Öfteren und schaut sich im Raum um. Nele übernahm dadurch immer mehr die Driver-Rolle und programmierte weiter. Zuvor waren die Rollen allerdings sehr gerecht verteilt und beide kommunizierten viel mit einander. Pawel teilte seine gute Stimmung den beiden Frauen mit: "Yeah, dritte Aufgabe.". Zum Schluss schrie Nele: "Nein!", da Linda etwas falsch abgespeichert hatte und beide fingen auf einen Kommentar von Pawel an zu lachen. Beide Frauen freuen sich wieder laut, als sie mit einer weiteren Aufgabe fertig sind. Dann bemerkte eine der Beiden, dass sie nur noch 10 Minuten Zeit hatten für eine weitere

Teilaufgabe, woraufhin sich beide schnell einig waren, dass es egal war und sie noch etwas in der Zeit schaffen würden. Es herrschte eine ausgelassene fröhliche Stimmung, beide arbeiteten konzentriert und zügig. Derjenige der den Observer hatte sagte dem Driver, was er tippen sollte. Als sie auch diese Variante des Mosaiks fertigstellten jubelten beide wieder äußerst ausgelassen und lautstark. Linda bemerkt, dass sie noch eine Minute Zeit hatten und die letzte Teilaufgabe nicht mehr schaffen würden in der Zeit. Daraufhin macht der Tutor sie auf den Fragebogen aufmerksam und meinte, dass sie diesen allerdings in einer Minute schaffen würden. Beide gaben einen enttäuschten Laut von sich, schnauften und füllten den Fragebogen aus.

Karl war Einzelarbeiter und brachte bereits viel Erfahrung im Programmieren mit. Er schrieb zuvor etwa 500 Zeilen Code und arbeitete im Kurs stets allein und forderte nur ab und zu die Hilfe vom Tutor an. Er war sehr still, zurückgezogen und arbeitete sehr konzentriert. An diesem Tag beobachtete ich ihn oft beim Lesen und einmal halfen ihm der Tutor und die Professorin bei der falschen Keyboardfunktion. Als er auf ein Problem stößt lächelt er kurz, ansonsten saß er konzentriert an seinen Aufgaben. Am Ende als er eine Aufgabe erfolgreich gelöst hatte sah man, dass er sich kurz freute. Dann arbeitete er wieder an der nächsten Aufgabe weiter.

Mit Saskia und Sven gab es ein Paar, welches von den Programmiererfahrungen sehr weit auseinander lag. Sven hatte bereits etwa 500 Zeilen Code geschrieben und Saskia war noch unerfahren. Ab und zu wurde auch durch den Fragebogen deutlich, dass Sven sich unterfordert fühlte und die Aufgaben allein schneller gelöst hätte. Bei diesem Paar war das Besondere, dass sie sich trotz des enormen Wissensunterschieds gut verstanden und mit den Rollen abwechselten. Sven erklärte Saskia die Aufgaben und Lösungsideen und ließ sie oft allein knobeln, obwohl er die Lösung schon wusste. Er sorgte auch immer dafür, dass Saskia alles verstand und auch selbst ausprobierte, sowie die Driver-Rolle regelmäßig übernahm. Beide arbeiteten zusammen und sehr konzentriert. Als ich mit meinen Beobachtungen anfing, hatte Saskia den Driver und Sven war Observer. Er erklärte ihr was sie falsch machte und gab ihr Tipps. Sie waren häufig durch den Tutor abgelenkt der sie nach dem Vorankommen fragte und sie beispielsweise einmal auf Codekonventionen aufmerksam machte. Bislang stellte nur Sven dem Tutor die Fragen, obwohl er sich in der Observer-Rolle befand. Allerdings ist zu beachten, dass er trotzdem das Lösen der Aufgabe anführte und eher die Führung übernahm. Man merkte ihm an, dass er bereits über mehr Erfahrungen im Programmieren verfügte. Trotzdem brachte Saskia sich oft ein und gab auch gute Ratschläge. Sie kommunizierten häufig und gut miteinander. Auch fragte Sven bei Saskia immer nach um sicher zu stellen, dass sie alles verstanden hatte. Saskia lacht auf, als sie bei der Ausführung des Programms ein Fehler auftritt und sagt zu Sven: "Ich führ doch nur deine Anweisungen aus.". Anschließend diskutieren beide über die Lösung der Aufgabe und denken nach. Als ich die Beiden später noch einmal beobachtete war Sven nicht am Platz und Saskia versuchte eigenständig die Aufgabe zu lösen. Sie dachte viel nach und las einiges nach. Sie war kurzzeitig durch ihr Mobiltelefon abgelenkt und fragte den Einzelarbeiter Karl etwas über die Anmeldung eines Kurses. Sie unterhielten sich auch über Leistungskurse aus der Schulzeit, woraufhin sich Linda und Emma an dem Gespräch beteiligten, die zufällig mitgehört hatten. Als Saskia allein programmierte wirkte sie sehr angespannt. Dies drückte sich durch ihre Mimik aus. Die Augen waren leicht zusammengekniffen, die Hand hielt sie zur Faust geballt vor dem Mund und sie starrte den Monitor minutenlang an. Sie schnaufte laut und fing an zu tippen. Etwa 10 Minuten später half Karl ihr, der neben ihr saß und bereits dabei war nach Hause zu gehen. Er diktierte ihr Codefragmente, wobei er hinter ihr stand und ihr auch auf dem Monitor etwas zeigte. Kurze Zeit später schien sie sehr erleichtert zu sein, da sie die Aufgabe doch noch geschafft hatte und zeigte sie dem Tutor.

In einem anderen Raum gab es ein Paar welches mir gleich auffiel. Obwohl der Wissensstand der Beiden nicht sehr stark von einander abwich, bemerkte man starke Defizite des einen Partners. Zu Beginn arbeiteten sie konzentriert, dachten nach und lasen viel. Henrik hatte wieder die Driver-Rolle und fragte den Tutor, sowie die Professorin um Hilfe, während Alesandro sich weiterhin sehr still verhielt. Die Körperhaltung von Alesandro war oft zum Monitor vorgebeugt, wobei der Arm mit dem Ellenbogen auf den Tisch gestützt war und die Hand den Mund verdeckte. Er starrte konzentriert und angestrengt den Monitor an. Mir viel auf, dass Henrik vieles Tat, ohne es Alesandro zu erklären, oder sich darum zu kümmern, es ihm verständlich zu machen. Alesandro nahm sich zwischenzeitlich einen Block, wandte sich leicht von Henrik und dem Monitor ab und versuchte etwas aufzuschreiben. Im Allgemeinen war die Stimmung sehr angespannt und verbissen. Als die Professorin einen Fehler aufdeckt, der in dem vorgegebenen Programm enthalten war, waren beide etwas erleichtert, jedoch immer noch angespannt. In den Fragebögen wurde jedoch angekreuzt, dass sie die Stimmung als normal empfanden und Henrik die Arbeitsatmosphäre zusätzlich noch als bremsend empfand. Alesandro gab sogar an, er hätte bei dem Lösen der Aufgaben mit seinem Partner Spaß gehabt, während Hendrik ankreuzte, dass er weder Spaß noch keinen Spaß hatte. Auch fiel Alesandro das Lösen des Aufgabenblatts wieder nicht leicht und das Kommunizieren mit dem Partner fiel ihm immer noch schwer. Diesmal gab er allerdings an, dass er Dinge, die er nicht verstand mit seinem Partner klären konnte und sie zusammen diskutiert haben. Henrik fiel das Kommunizieren mit seinem Partner auch schwer, da er das Gefühl hatte, er sei ihm unterlegen und wüsste nicht viel. Auch war es ihm so nicht möglich mit ihm Dinge zu klären, die er nicht verstanden hatte. Erstaunlich ist, dass Alesandro im Fragebogen angab, dass er der Meinung war, dass sein Partner mehr Arbeit in das Übungsblatt reingesteckt hatte als er, da er zu langsam war. Er suchte offensichtlich die Fehler ausschließlich bei sich und war überzeugt davon, dass seine fehlenden Vorkenntnisse die Ursache für das schlechte Zusammenarbeiten waren. Auch war er wieder nicht zufrieden mit der Lösung, da er nicht alles verstanden hatte. Das Paar arbeitete offensichtlich nicht gut zusammen. Dies lag meiner Meinung nach aber nicht nur an Alesandro der sehr Leistungsschwach war, sondern auch an Hendrik, der sehr wenig erklärte und eher für sich vor sich hin programmierte, ohne auf den Partner einzugehen.

Ich beobachtete noch einige andere Paare an diesem Tag, die Beobachtungen habe ich hier nicht detailliert weiter beschrieben, weil dies den Rahmen sprengen würde. Insgesamt beobachtete ich 46 Studenten.

10.5 Fragebogen zur Gruppeneinteilung des Blockkurses "Programmieren in Java"

Umfrage zur Paar-Programmierung

Name, Vorname: _____

Email-Adresse: _____

1. Welchen Studiengang besuchst du?

   _____

2. Hast du schon vor diesem Kurs Programmiererfahrungen sammeln können?

   ○ Ja
   ○ Nein

   a) Wenn ja, wie viele Zeilen Code hast du circa produziert?

   ○ keine
   ○ etwa 100
   ○ etwa 500
   ○ 1000 und mehr

   b) Wenn ja, hast du schon Erfahrungen mit Paar-Programmierung gemacht?

   ○ Ja
   ○ Nein

3. Würdest du lieber allein arbeiten oder Paar-Programmierung ausprobieren?

   ○ Allein arbeiten
   ○ Paar-Programmierung ausprobieren

   a) Wenn du Paar-Programmierung ausprobieren möchtest und einen „Lieblingspartner/in" hast, gebe bitte
   hier seinen/ihren Namen an:

   _____

   Vielen Dank für die Beantwortung der Fragen!
   Alle deine Daten werden vertraulich behandelt und nicht an Dritte weitergegeben.

   Ich erkläre mich damit einverstanden, dass meine Daten zur Forschung und Beobachtung von Verhalten bei Paar-Programmierung anonymisiert aufgezeichnet und verwendet werden.

   _____
   Unterschrift

10.6 Fragebogen für Einzelarbeiter des Blockkurses "Programmieren in Java"

Die Fragen die unter Allgemein auftauchen wurden nur am ersten und am letzten Tag gestellt, ansonsten wurde der Fragebogen in verkürzter Form ohne die allgemeinen Fragen ausgeteilt.
Mit Frage 11 hatten die Studenten einige Probleme und verstanden diese falsch. Gefragt war ob sich die Studenten vorbereiteten und wenn wir lange sie sich auf das Übungsblatt vorbereiteten. Die Studenten verstanden die Frage so, dass sie angaben wie lange sie für den Übungsblatt brauchten. Als ich die Frage änderte fiel dies allerdings wenigen Teilnehmern auf, da die Studenten den Fragebogen kaum mehr richtig lasen. So wurde das Ergebnis verfälscht, was nicht besonders schlimm ist, da die Studenten sich kaum vorbereiteten und es auch nur schwer konnten.

Fragebogen "Programmieren in Java"

Erläuterungen von Abkürzungen:

PP = Pair Programming – Paar-Programmierung
DPP = Distributed Pair Programming – verteilte Paar-Programmierung

Name, Vorname: _____

Allgemein

---

1. Warum hast du dich dafür entschieden Paar-Programmierung nicht auszuprobieren?
   _____
   _____
   _____
   _____
2. Denkst du, dass man durch das Anwenden von PP/DPP mehr lernen bzw. den Inhalt der Vorlesung besser verstehen kann?
   ○ Ja , weil _____

   ○ Nein, weil _____
3. Denkst du, dass PP/DPP für das Lösen der Übungsaufgaben sinnvoll wäre?

   ○ Ja , weil _____

   ○ Nein, weil _____
4. Glaubst du, dass das Erlernen von PP/DPP für das späterer Berufsleben von Vorteil ist?
   ○ Ja , weil _____

○ Nein, weil _____

5. Welche positiven und negativen Eigenschaften vom allein arbeiten konntest du beim Lösen des Übungszettels feststellen?

6. Fiel dir das Lösen des Zettels leicht?
   ○ Ja
   ○ Nein

   a) Was glaubst du war die Ursache dafür?

7. Wie viel Spaß hat es dir gemacht das Übungsblatt zu lösen?

   ○ Ich hatte sehr viel Spaß
   ○ Ich hatte Spaß
   ○ Es ging so
   ○ Ich hatte kaum Spaß
   ○ Es war schrecklich

8. Wie schätzt du deine Arbeitsweise und Arbeitsatmosphäre ein?
   (mehrfach Antworten erwünscht)
   ○ lustig
   ○ normal
   ○ angespannt
   ○ konzentriert
   ○ nicht bei der Sache
   ○ zielstrebig
   ○ bremsend
   ○ angenehm
   ○ unangenehm

9. Hattest du Fragen oder Probleme bei der Lösung der Aufgaben?
   ○ Nein
   ○ Ja

   a) Wenn ja, welche Probleme hattest du?

b) Wie hast du die Probleme die während deiner Arbeit aufgetaucht sind gelöst?

○ Nachgelesen
○ Den Tutor gefragt
○ Andere Studenten gefragt
○ Nachgedacht
○ Ausprobiert
○ _____

c) Denkst du, dass ein Partner dir beim Lösen der Probleme geholfen hätte und du sie schneller gelöst hättest?

○ Nein, weil _____
○ Ja, weil _____

10. Hattest du den Eindruck, dass das Lösen des Übungszettels allein schneller ging als im Team?
    ○ Ja, weil _____
    ○ Gleich lang, weil _____
    ○ Länger, weil _____

11. Wie hast du dich auf den Übungszettel vorbereitet und wie lange hast dafür ungefähr gebraucht?
    _____
    _____

12. Bist du mit der Lösung deiner Aufgaben zufrieden?
    ○ Ja, weil _____
    ○ Es geht so, weil _____
    ○ Nein, weil _____

10.7 Fragebogen für Paarprogrammierer des Blockkurses "Programmieren in Java"

Auch hier wurde nur am ersten und am letzten Tag der Block mit den allgemeinen Fragen gestellt.
Fragebogen zur Paar-Programmierung

Erläuterungen von Abkürzungen:

PP = Pair Programming – Paar-Programmierung
DPP = Distributed Pair Programming – verteilte Paar-Programmierung

Name, Vorname: _____

Allgemein

---

1. Warum hast du dich dafür entschieden Paar-Programmierung auszuprobieren?
   _____
   _____
   _____
   _____

2. Bist du der Meinung durch das Anwenden von PP/DPP mehr zu lernen bzw. den Inhalt der Vorlesung besser zu verstehen?
   ○ Ja , weil _____

   ○ Nein, weil _____

3. Denkst du, dass PP/DPP für das Lösen der Übungsaufgaben sinnvoll ist?

   ○ Ja , weil _____

   ○ Nein, weil _____

4. Glaubst du, dass das Erlernen von PP/DPP für das späterer Berufsleben von Vorteil ist?
   ○ Ja , weil _____

   ○ Nein, weil _____

---

5. Welche positiven und negativen Eigenschaften konntest du beim Lösen des Übungszettels bzgl. PP/DPP feststellen?

6. Fiel dir das Lösen des Zettels leicht?
   ○ Ja
   ○ Nein

   a) Was glaubst du war die Ursache dafür?

7. Wie viel Spaß hat es dir gemacht mit deinem Partner das Übungsblatt zu lösen?
   ○ Ich hatte sehr viel Spaß
   ○ Ich hatte Spaß
   ○ Es ging so
   ○ Ich hatte kaum Spaß
   ○ Es war schrecklich

8. Wie empfandest du die Arbeitsatmosphäre? (mehrfach Antworten erwünscht)
   ○ lustig
   ○ normal
   ○ angespannt
   ○ konzentriert
   ○ nicht bei der Sache
   ○ zielstrebig
   ○ bremsend
   ○ angenehm
   ○ unangenehm

9. Fiel es dir schwer mit deinem Partner zu kommunizieren?
   ○ Nein überhaupt nicht, weil _____
   ○ Es ging so, weil _____
   ○ Ja es fiel mir schwer, weil _____

10. Konntest du mit deinem Partner Dinge, die du nicht verstanden hast und Probleme offen klären?
    ○ Nein, weil _____
    ○ Es ging so, weil _____
    ○ Ja, weil _____

11. Hattest du Fragen oder Probleme bei der Lösung der Aufgaben?
    ○ Nein
    ○ Ja

    a) Wenn ja, welche Probleme hattest du?

b) Wie hast du die Probleme die während deiner Arbeit aufgetaucht sind gelöst?
○ Nachgelesen
○ Mit dem Partner besprochen
○ Den Tutor gefragt
○ Andere Studenten gefragt
○ Nachgedacht
○ Ausprobiert
○ _____

12. Hattest du den Eindruck, dass dein Partner und du gleich viel Arbeit in das Übungsblatt gesteckt habt?
○ Nein, weil _____
○ Ja, weil _____

13. Wie hast du dich auf den Übungszettel vorbereitet und wie lange hast dafür ungefähr gebraucht?
_____
_____

14. Hattest du den Eindruck, dass das Lösen des Übungszettels zusammen schneller ging?
○ Ja, weil _____
○ Gleich lang, weil _____
○ Länger, weil _____

15. Bist du mit der Lösung deiner Aufgaben zufrieden?
○ Ja, weil _____
○ Es geht so, weil _____
○ Nein, weil _____

10.8 Beobachtungsvorlage für den ersten Tag

Diese stelle ich hier nur schemenhaft vor und kopiere nur einen kleinen Auszug.

Vorbereitungszeit: Studenten die Paarprogrammierung anwenden benötigen eine längere Vorbereitungszeit, während sich allerdings die Zeit, die für das implementieren der Aufgaben gebraucht wird verkürzt.
(14:30 – 15:30, 16:00 – 17:30)

Name	Uhrzeit	Details

Kommunikation(Einzelarbeiter): Einzelarbeiter in einem Raum kommunizieren trotzdem und geben sich auch teilweise Hilfestellungen. (Von 14:00 – 14:30, 15:30-16:00, 17:30 – 18:00)

Name	Uhrzeit	Details

Hilfestellung: Einzelarbeiter fragen öfter den Tutor um Rat, als Studenten, die PP anwenden. (Von 14:00 – 14:30, 15:30-16:00, 17:30 – 18:00)

Name + (Driver/Observer)	Uhrzeit	Details	Ressource der Hilfestellung (Tutor, ..

Rollenverteilung

Name	Uhrzeit	Rolle	Details

10.9 Beobachtungsvorlage für den zweiten Tag

Zu diesem Zeitpunkt hatte ich mir einen Sitzplan gemacht und wusste bereits größtenteils die Namen der Studenten. Also passte ich meine Beobachtungsvorlage an, trug die Namen bereits ein und sortierte diese ihren Gruppen zu, sodass ich pro Blatt die Namen einer Gruppe hatte. Die Namen wurde hier entfernt und durch Beispielnamen ersetzt, um Anonymität zu gewährleisten.

Stimmung: Bei Paarprogrammierern wird nur eine kurzzeitige Ablenkung stattfinden und die meiste Zeit über konzentriert gearbeitet. Wenn extrem gute oder extrem schlechte Stimmung herrscht sind die Teilnehmer abgelenkt und arbeiten nicht konzentriert.

Raum 1	→ gute Stimmung (lachen, häufiger Blickkontakt, rege Kommunikation) → schlechte Stimmung (weggucken, schmollen, Schweigen)
Müller, Steffi & Meier, Hans	
Mustermann, Lena & Knapp, Ulrich	
Raum 2	→ gute Stimmung (lachen, häufiger Blickkontakt, rege Kommunikation) → schlechte Stimmung (weggucken, schmollen, Schweigen)
Ahorn, Susi & Stenge, Franz	

Auf der Rückseite befand sich dann eine weitere Tabelle, wo ich die anderen Beobachtungen des Tages eintrug. So musste ich nur das Blatt wenden und konnte zu der gleichen Gruppe Beobachtungen über das konzentrierte Arbeiten eintragen.

Konzentriertes Arbeiten: Bei Paarprogrammierern wird nur eine kurzzeitige Ablenkung stattfinden und die meiste Zeit über konzentriert gearbeitet. Wenn extrem gute oder extrem schlechte Stimmung herrscht sind die Teilnehmer abgelenkt und arbeiten nicht konzentriert.

Raum 1	→ Konzentrierte Arbeitszeit (lesen, nachdenken, tippen, Notizen/Skizzen machen, recherchieren) → Ablenkung (Witze machen, plaudern über Freizeit usw., Essen holen gehen)
Müller, Steffi & Meier, Hans	
Mustermann, Lena & Knapp, Ulrich	
Raum 2	→ Konzentrierte Arbeitszeit (lesen, nachdenken, tippen, Notizen/Skizzen machen, recherchieren) → Ablenkung (Witze machen, plaudern über Freizeit usw., Essen holen gehen)
Ahorn, Susi & Stenge, Franz	

Für den vierten Tag passte ich die Beobachtungsvorlagen noch einmal an und hatte nun eine Tabelle wie oben. Allerdings hatte ich nun jede Gruppe auf einem Blatt und auf der Rückseite die gleiche Tabelle. Ich schrieb die Beobachtungen des Tages nicht mehr verteilt auf die Blätter sondern in ein Feld der Tabelle, da ich vorher merkte, dass ich beim Eintragen durcheinander kam und beispielsweise bei Stimmung etwas geschrieben hatte, was unter konzentriertem Arbeiten hätte stehen sollen.

Für die letzten zwei Tage hatte ich keine Beobachtungsvorlage und schrieb einfach alles auf ein Blatt. Ich entschied mich gegen die vorherigen Vorlagen, da ich nun auf einzelne Personen genauer eingehen wollte und die bisherige Beobachtungsvorlage ineffizient genutzt wurde.

10.10 Fragebogen des letzten Tages für die Paarprogrammierer

Fragebogen zur Paar-Programmierung

Erläuterungen von Abkürzungen:

PP = Pair Programming – Paar-Programmierung
DPP = Distributed Pair Programming – verteilte Paar-Programmierung

Name, Vorname: _____

Allgemein

---

1. Warum hast du dich dafür entschieden Paar-Programmierung auszuprobieren?
   _____
   _____
   _____
   _____
2. Kanntest du deinen Partner schon vorher?
   ○ Ja
   ○ Nein
3. Denkst du, dass die Stimmung in deinem Raum gut war?
   ○ Ja
   ○ Nein

   a) Was war die Ursache dafür?

4. War die Stimmung besser als in anderen Räumen?
   ○ besser
   ○ schlechter

   a) Was war die Ursache dafür?

5. Bist du der Meinung durch das Anwenden von PP/DPP mehr zu lernen bzw. den Inhalt der Vorlesung besser zu verstehen?
   ○ Ja , weil _____

   ○ Nein, weil _____

6. Denkst du, dass PP/DPP für das Lösen der Übungsaufgaben sinnvoll ist?

   ○ Ja , weil _____

   ○ Nein, weil _____

7. Glaubst du, dass das Erlernen von PP/DPP für das späterer Berufsleben von Vorteil ist?
   ○ Ja , weil _____

   ○ Nein, weil _____

---

8. Welche positiven und negativen Eigenschaften konntest du beim Lösen des Übungszettels bzgl. PP/DPP feststellen?

9. Fiel dir das Lösen des Zettels leicht?
   ○ Ja
   ○ Nein

   a) Was glaubst du war die Ursache dafür?

10. Wie viel Spaß hat es dir gemacht mit deinem Partner das Übungsblatt zu lösen?
    ○ Ich hatte sehr viel Spaß
    ○ Ich hatte Spaß
    ○ Es ging so
    ○ Ich hatte kaum Spaß
    ○ Es war schrecklich

11. Wie empfandest du die Arbeitsatmosphäre? (mehrfach Antworten erwünscht)
    ○ lustig
    ○ normal
    ○ angespannt
    ○ konzentriert
    ○ nicht bei der Sache
    ○ zielstrebig
    ○ bremsend
    ○ angenehm
    ○ unangenehm

12. Fiel es dir schwer mit deinem Partner zu kommunizieren?
    ○ Nein überhaupt nicht, weil _____
    ○ Es ging so, weil _____
    ○ Ja es fiel mir schwer, weil _____

13. Konntest du mit deinem Partner Dinge, die du nicht verstanden hast und Probleme offen klären?
    ○ Nein, weil _____
    ○ Es ging so, weil _____
    ○ Ja, weil _____

14. Hattest du Fragen oder Probleme bei der Lösung der Aufgaben?
    ○ Nein
    ○ Ja

    a) Wenn ja, welche Probleme hattest du?

    b) Wie hast du die Probleme die während deiner Arbeit aufgetaucht sind gelöst?
    ○ Nachgelesen
    ○ Mit dem Partner besprochen
    ○ Den Tutor gefragt
    ○ Andere Studenten gefragt
    ○ Nachgedacht
    ○ Ausprobiert
    ○ _____

15. Hattest du den Eindruck, dass dein Partner und du gleich viel Arbeit in das Übungsblatt gesteckt habt?
    ○ Nein, weil _____
    ○ Ja, weil _____

16. Wie hast du dich auf den Übungszettel vorbereitet und wie lange hast dafür ungefähr gebraucht?
    _____
    _____

17. Hattest du den Eindruck, dass das Lösen des Übungszettels zusammen schneller ging?
    ○ Ja, weil _____
    ○ Gleich lang, weil _____
    ○ Länger, weil _____

18. Bist du mit der Lösung deiner Aufgaben zufrieden?
    ○ Ja, weil _____
    ○ Es geht so, weil _____
    ○ Nein, weil _____

10.11 Fragebogen des letzten Tages für die Einzelarbeiter

Fragebogen "Programmieren in Java"

Erläuterungen von Abkürzungen:

PP = Pair Programming – Paar-Programmierung
DPP = Distributed Pair Programming – verteilte Paar-Programmierung

Name, Vorname: _____

Allgemein

---

1. Warum hast du dich dafür entschieden Paar-Programmierung nicht auszuprobieren?
   _____
   _____
   _____
   _____

2. Denkst du, dass die Stimmung in deinem Raum gut war?
   ○ Ja
   ○ Nein

   a) Was war die Ursache dafür?

3. War die Stimmung besser als in anderen Räumen?
   ○ besser
   ○ schlechter

   a) Was war die Ursache dafür?

4. Denkst du, dass man durch das Anwenden von PP/DPP mehr lernen bzw. den Inhalt der Vorlesung besser verstehen kann?
   ○ Ja , weil _____

   ○ Nein, weil _____

5. Denkst du, dass PP/DPP für das Lösen der Übungsaufgaben sinnvoll wäre?

○ Ja , weil _____

○ Nein, weil _____

6. Glaubst du, dass das Erlernen von PP/DPP für das späterer Berufsleben von Vorteil ist?
   ○ Ja , weil _____

   ○ Nein, weil _____

7. Welche positiven und negativen Eigenschaften vom allein arbeiten konntest du beim Lösen des Übungszettels feststellen?

8. Fiel dir das Lösen des Zettels leicht?
   ○ Ja
   ○ Nein

   a) Was glaubst du war die Ursache dafür?

9. Wie viel Spaß hat es dir gemacht das Übungsblatt zu lösen?

   ○ Ich hatte sehr viel Spaß
   ○ Ich hatte Spaß
   ○ Es ging so
   ○ Ich hatte kaum Spaß
   ○ Es war schrecklich

10. Wie schätzt du deine Arbeitsweise und Arbeitsatmosphäre ein?
    (mehrfach Antworten erwünscht)
    ○ lustig
    ○ normal
    ○ angespannt
    ○ konzentriert
    ○ nicht bei der Sache
    ○ zielstrebig
    ○ bremsend
    ○ angenehm
    ○ unangenehm

11. Hattest du Fragen oder Probleme bei der Lösung der Aufgaben?
    ○ Nein
    ○ Ja

    a) Wenn ja, welche Probleme hattest du?

    b) Wie hast du die Probleme die während deiner Arbeit aufgetaucht sind gelöst?

    ○ Nachgelesen
    ○ Den Tutor gefragt
    ○ Andere Studenten gefragt
    ○ Nachgedacht
    ○ Ausprobiert
    ○_____

    c) Denkst du, dass ein Partner dir beim Lösen der Probleme geholfen hätte und du sie schneller gelöst hättest?

    ○ Nein, weil _____
    ○ Ja, weil _____

12. Hattest du den Eindruck, dass das Lösen des Übungszettels allein schneller ging als im Team?
    ○ Ja, weil _____
    ○ Gleich lang, weil _____
    ○ Länger, weil _____

13. Wie hast du dich auf den Übungszettel vorbereitet und wie lange hast dafür ungefähr gebraucht?

    _____
    _____

14. Bist du mit der Lösung deiner Aufgaben zufrieden?
    ○ Ja, weil _____
    ○ Es geht so, weil _____
    ○ Nein, weil _____

10.12 Übersetzung des Raymond-Lister-Tests

Für den Kurs "Programmieren in Java" übersetzte ich den Test, damit die Studenten eine deutsche Fassung ausfüllen konnten. Im Original wurde der Test in der englischen Sprache verfasst.

Test

Frage 1:

```
int[] x = {2, 1, 4, 5, 7};
int limit = 3;
int i = 0;
int sum = 0;

while ( (sum<limit) && (i<x.length)){
      ++i;
      sum += x[i];
}
```

Welchen Wert hat die Variable i nachdem das Codefragment ausgeführt wurde?

a) 0
b) 1
c) 2
d) 3

Frage 2:

```
int[] x1 = {1, 2, 4, 7};
int[] x2 = {1, 2, 5, 7};
int i1 = x1.length-1;
int i2 = x2.length-1;
int count = 0;

while ((i1 > 0 ) && (i2 > 0 ))
{
      if ( x1[i1] == x2[i2] )
      {
            ++count;
            --i1;
            --i2;
      }
      else if (x1[i1] < x2[i2])
      {
            --i2;
      }
      else
      { // x1[i1] > x2[i2]
            --i1;
      }
}
```

Welchen Wert hat „count“, wenn die While-Schleife beendet ist.

a) 3
b) 2
c) 1
d) 0

Frage 3:

```
int [] x = {1, 2, 3, 3, 3};
boolean b[] = new boolean[x.length];

for ( int i = 0; i < b.length; ++i )
      b[i] = false;

for ( int i = 0; i < x.length; ++i )
      b[ x[i] ] = true;

int count = 0;

for (int i = 0; i < b.length; ++i )
{
      if ( b[i] == true ) ++count;
}
```

Was enthält „count“ wenn der Code ausgeführt wurde?

a) 1
b) 2
c) 3
d) 4
e) 5

Frage 4:

```
int[ ] x1 = {0, 1, 2, 3};
int[ ] x2 = {1, 2, 2, 3};
int i1 = 0;
int i2 = 0;
int count = 0;

while ( (i1 < x1.length) &&
      (i2 < x2.length))
{
      if ( x1[i1] == x2[i2] )
      {
            ++count;
            ++i2;
      }
      else if (x1[i1] < x2[i2])
      {
            ++i1;
      }
      else
      { // x1[i1] > x2[i2]
            ++i2;
      }
}
```

Was enthält „count“ wenn der Code ausgeführt wurde?

a) 0
b) 1
c) 2
d) 3
e) 4

Frage 5:

```
int[ ] x = {0, 1, 2, 3};
int temp;
int i = 0;
int j = x.length-1;

while (i < j)
{
      temp = x[i];
      x[i] = x[j];
      x[j] = 2*temp;
      i++;
      j--;
}
```

Welche Werte hat das Array „x“, nach der Ausführung des Codes?

a) {3, 2, 2, 0}
b) {0, 1, 2, 3}
c) {3, 2, 1, 0}
d) {0, 2, 4, 6}
e) {6, 4, 2, 0}

Frage 6:

Die folgende Methode „isSorted“ sollte true zurückgeben, wenn das Array aufsteigend sortiert wurde. Andernfalls gibt die Methode false zurück:

```
public static boolean isSorted(int []x)
{
      //missing code goes here
}
```

Welches der folgenden Codefragmente ist das fehlende Codefragment der Methode „isSorted“?

```
(a)   boolean b = true;
      for (int i=0 ; i<x.length-1; i++)
      {
            if ( x[i ] > x[i+1] )
                  b = false;
            else
                  b = true;
      }
      return b;
```

(b)
```
for (int i=0; i<x.length-1; i++)
{
      if (x[i ] > x[i+1] )
            return false;
}
return true;
```

(c)
```
boolean b = false;
for (int i=0; i<x.length-1; i++)
{
      if (x[i] > x[i+1] )
            b = false;
}
return b;
```

(d)
```
boolean b = false;
for (int i=0;i<x.length-1;i++)
{
      if (x[i ] > x[i+1] )
            b = true;
}
return b;
```

(e)
```
for (int i=0;i<x.length-1;i++)
{
      if (x[i ] > x[i+1] )
            return true;
}
return false;
```

Frage 7:

```
int[] x = {2, 1, 4, 5, 7};
int limit = 7;
int i = 0;
int sum = 0;

while ( (sum<limit) && (i<x.length) )
{
      sum += x[i];
      ++i;
}
```

Welchen Wert hat die Variable „i“ nach der Ausführung des Codes?

a) 0
b) 1
c) 2
d) 3
e) 4

Frage 8:

Wenn zwei Zahlen vom Typ integer in einem Array (nicht zwangsläufig aufeinanderfolgende Zahlen in einem Array) nicht in der richtigen Reihenfolge sind (die Zahl, die als erstes in dem Array vorkommt ist größer als die Zahl, die an zweiter Stelle steht), dann nennt man das eine Umkehrung. Z.B. ein Array „x“ mit folgenden Werten:

4 5 6 2 1 3

Es gibt 10 Umkehrungen in dem Array, wie:

x[0]=4 > x[3]=2
x[0]=4 > x[4]=1
x[0]=4 > x[5]=3
x[1]=5 > x[3]=2
x[1]=5 > x[4]=1
x[1]=5 > x[5]=3
x[2]=6 > x[3]=2
x[2]=6 > x[4]=1
x[2]=6 > x[5]=3
x[3]=2 > x[4]=1

Der Code beabsichtigt die Umkehrungen in dem Array „x“ zu zählen:

```
int inversionCount = 0;

for ( int i=0 ; i<x.length-1 ; i++ )
{
      for xxxxxx
      {
            if ( x[i] > x[j] )
                  ++inversionCount;
      }
}
```

Wenn die Ausführung des Codes beendet ist, beinhaltet die Variable „inversionCount“ die Anzahl der Umkehrungen in dem Array „x“.
Deshalb sollte das „xxxxxx“ im Code ersetzt werden mit:

a) (int j=0 ; j<x.length ; j++)
b) (int j=0 ; j<x.length-1; j++)
c) (int j=i+1; j<x.length ; j++)
d) (int j=i+1; j<x.length-1; j++)

Frage 9:

Der Code kopiert von einem Array vom Typ integer „array1“ gerade Zahlen vom Typ integer in ein Array „array2“ .
Z.B. wenn „array1“ folgende Zahlen beinhaltet:

array1: 4 5 6 2 1 3

dann beinhaltet nach dem Kopieren „array2“ in den ersten drei Stellen:

array2: 4 6 2

Der folgende Code nimmt an, dass „array2“ groß genug ist um alle geraden Zahlen von „array1“ zu speichern:

```
int a2 = 0;

for ( int a1=0 ; xxx1xxx ; ++a1 )
{
      // if array1[a1] is even
      if ( array1[a1] % 2 == 0 )
      {
            // array1[a1] is even,
            // so copy it
            xxx2xxx;
            xxx3xxx;
      }
}
```

Das fehlende Stück Code „xxx1xxx“, „xxx2xxx“ und „xxx3xxx“ sollte ersetzt werden mit:

a)
```
a1<array1.length
++a2
array2[a2] = array1[a1]
```

b)
```
a1<array1.length
array2[a2] = array1[a1]
++a2
```

c)
```
a1<=array1.length
array2[a2] = array1[a1]
++a2
```

d)
```
a1<=array1.length
++a2
array2[a2] = array1[a1]
```

Tipp: in allen vier Optionen oberhalb, der zweite und dritte Part sind gleich, nur umgedreht.

<u>Frage 10</u>:

```
int[] array1 = {2, 4, 1, 3};
int[] array2 = {0, 0, 0, 0};
int a2 = 0;

for (int a1=1; a1<array1.length; ++a1)
{
      if ( array1[a1] >= 2 )
      {
            array2[a2] = array1[a1];
            ++a2;
      }
}
```

Nach der Ausführung des Codes, beinhaltet Array „array2“ welche Werte?

a) {4, 3, 0, 0}
b) {4, 1, 3, 0}
c) {2, 4, 3, 0}
d) {2, 4, 1, 3}

Frage 11:

Angenommen ein Array von Integer enthält eine Null oder mehrere verschiedene positive Zahlen vom Typ integer, in aufsteigender Ordnung, gefolgt von einer Null.
Z.B.:

```
    int[] s = {2, 4, 6, 8, 0};
or int[] s = {0};
```

Betrachte das folgende Codefragment, wo die Sequenzen von „xxxxxx“ mit dem korrekten Java-Code ersetzt werden

```
int pos = 0;

while ( (xxxxxx) && (xxxxxx) )
      ++pos;
```

Angenommen eine Variable „e“ vom Typ integer enthält eine positive ganze Zahl (integer). Die Absicht des Codes oben ist die Stelle in „s“ zu finden, die besetzt ist von einem Wert, der in „e“ steht.
Formal: wenn die while-Schleife terminiert, ist die Variable „pos“ folgendermaßen festgelegt:

1. Wenn der Wert in „e“ auch in dem Array steht, dann enthält „pos“ den Index der Position.
   Z.B. wenn e = 6 und s = {2,4,6,8,0}
   dann sollte „pos“ = 2 sein.
2. Wenn der Wert, der in „e“ steht nicht in dem Array ist, aber der Wert in „e“ kleiner als die Werte im Array sind, dann enthält „pos“ den Index der kleinsten Position in dem Array, wo der Wert größer ist als in „e“.
   Z.B. wenn e = 7 und s = {2,4,6,8,0},
   dann ist „pos“ = 3.
3. Wenn der Wert der in „e“ enthalten ist größer ist als die Werte in „s“, dann enthält „pos“ den Index der Position die die Null beinhaltet.
   Z.B. wenn e = 9 und s = {2,4,6,8,0},
   dann ist „pos“ = 4.

Die korrekten booleschen Bedingungen für die While-Schleife sind:

```
(a) (pos < e) && (s[pos] != 0)
(b) (pos != e) && (s[pos] != 0)
(c) (s[pos] < e) && ( pos != 0)
(d) (s[pos] < e) && (s[pos] != 0)
(e) (s[pos] != e) && (s[pos] != 0)
```

Frage 12:

Die Frage bezieht sich auf die vorherige Frage.
Angenommen wir finden eine Position in dem Array „s“ welche den gleichen Wert enthält wie in „e“. Wir möchten nun Code schreiben, der die Zahl in dem Array löscht, aber die aufsteigende Sortierung der restlichen Zahlen in dem Array behält.
Z.B.:

```
s = {2, 4, 6, 8, 0};
e = 6;
pos = 2;
```

Das gewünschte Ergebnis ist, die 6 von „s“ zu entfernen:

```
s = {2, 4, 8, 0, 0};
```

Betrachte folgendes Codefragment, wo „xxxxxx“ mit dem korrekten Java-Code ersetzt wird:

```
do {
      ++pos;
      xxxxxx;
} while (s[pos] != 0 );
```

Das richtige Ersatz für „xxxxxx“ ist:

```
(a) s[pos+1] = s[pos];
(b) s[pos] = s[pos+1];
(c) s[pos] = s[pos-1];
(d) s[pos-1] = s[pos];
(e) Keine der oben genannten.
```

Lösung:

1	2	3	4	5	6	7	8	9	10	11	12
c	b	c	e	a	b	d	c	b	a	d	d

10.13 Test des Blockkurses “Programmieren in Java”

WS 2011
Prof. Dr. Margarita Esponda

Programmieren in Java

Quiz: Vorkenntnisse

20. Februar 2012

1. Aufgabe

Was ist der Wert und Typ folgender Java-Ausdrücke?

Ausdruck	**Wert**	**Typ**
2/3+1/3		
7%3==1		
4/5.0+1.0/5		
2^1		
~11		

2. Aufgabe

Was ist der Wert der Variablen *a*, *b* und *c*, nachdem folgende Anweisungen ausgeführt worden sind?

```
int a = 3, b = 0;

c = --b + a++;
```

a:_____ b:_____ c:_____

3. Aufgabe

```
int n = 123456789;
int m = 0;

while  (n != 0) {
        m = (10 * m) + (n % 10);
        n = n/10;
}
```

Welchen Wert haben die Variablen *m* und *n*, nachdem die **while**-Schleife ausgeführt worden ist?

m: _____

4. Aufgabe

Eine Annäherung der Zahl π kann mit Hilfe folgender Reihe wie folgt berechnet werden:

$$R_n = \sum_{t=1}^{v} \frac{1}{t^2} = \frac{1}{1^2} + \frac{1}{2^2} + \ldots + \frac{1}{v^2} \approx \frac{\pi^2}{6}$$

Vervollständigen Sie den Inhalt folgender statischer Methoden, die die Zahl π bei Eingabe von **n** annähern soll. Verwenden Sie dafür eine **for**-Schleife (keine **while**-Schleife).

```
public static double pi ( int n ){



} // end of the function pi
```

5. Aufgabe

Betrachten Sie folgenden Java-Code:

```
int[] x = {2, 5, 1, 3};
int[] y = {0, 0, 0, 0};
int count = 0;

for (int i=1; i<x.length; ++i)
{
        if ( x[i] >= 2 )
        {
                y[count] = x[i];
                ++count;
        }
}
```

Nach der Ausführung des Codes, welche Werte beinhaltet das *y*-Array?

a) {5, 1, 3, 0}
b) {5, 3, 0, 0}
c) {2, 5, 3, 0}
d) {2, 5, 1, 3}

6. Aufgabe

Betrachten Sie folgende Klassendefinitionen:

```
public class Circle {

  static final double PI = 3.1416;
  private static int counter;
  double x, y, radius;

  public Circle( double x, double y, double radius ) {
          this.x = x;
          this.y = y;
          this.radius = radius;
          counter++;
  }

  public void setX(double x) {
          this.x = x;
  }

  public void setY(double y) {
          this.y = y;
  }

  public static int getCounter(){
          return counter;
  }

  public double area(){
          return PI*radius*radius;
  }

}// end of class Circle
```

Schreiben die den Name folgender Variablen- und Methoden-Sorten:

a) Instanz-Variablen _____

b) Klassenvariablen _____

c) lokale Variablen? _____

c) Instanz-Methoden? _____

d) Klassen-Methoden? _____

e) Konstanten? _____

e) Schreiben Sie eine main-Methode, in dem ein Circle-Objekt erzeugt wird, das x-, y-Koordinaten verändert und die Fläche des Circle-Objekts berechnet und ausgibt.

7. Aufgabe

Welche der folgenden Zuweisungen sind in Java erlaubt? Welche sind erlaubt, aber können Fehler verursachen?

a) Circle c = new Object();

b) ColorCircle cc = new Circle();

c) Circle c = new ColorCircle();

d) Object o = new Circle();

e) Circle c1 = (Circle) new Object();

8. Aufgabe

Was ist **falsch** über Variablen in Java?

- ☐ Variablendeklarationen innerhalb von Methoden dürfen keinen Modifizierer haben.
- ☐ Der Gültigkeitsbereich von Variablen erstreckt sich über den ganzen Block, in dem sie definiert worden sind.
- ☐ Klassenvariablen können nur innerhalb von Klassenmethoden zugegriffen werden.
- ☐ Instanz-Variablen können innerhalb von Klassenmethoden modifiziert werden.

Welche Behauptung über Konstruktoren ist **falsch**?

- ☐ Konstruktoren können nicht vererbt werden.
- ☐ Konstruktoren können Ausnahmefehler erzeugen.
- ☐ Konstruktoren müssen nicht immer definiert werden.
- ☐ Konstruktoren können sich nicht gegenseitig aufrufen.

Welche der folgenden Aussagen über die *main*-Methode ist **falsch**?

- ☐ Ein Java-Programm kann nur eine main-Methode beinhalten.
- ☐ Die main-Methode, mit der ein Java-Programm gestartet wird, ist immer *static*.
- ☐ Eine Java-Klasse kann mehrere Methoden, die main heißen, beinhaltet.
- ☐ Die main-Methode innerhalb einer Klassendefinition kann nur die statischen Methoden der Klasse direkt aufrufen.

Welche der folgenden Aussagen über Vererbung in Java ist **falsch**?

- ☐ Durch Vererbung können Eigenschaften und Methoden auf neue Klassen übernommen werden.
- ☐ In einer Unterklasse können Variablen der Oberklasse überschrieben werden.
- ☐ *private*-Variablen sind nur in Unterklassen sichtbar.
- ☐ In Java gibt es nur einfache Vererbung.

Welche der folgenden Aussagen über Unterklassen in Java ist **falsch**?

- ☐ Alle Klassen in Java sind Unterklassen der Object-Klasse.
- ☐ Eine Unterklasse von einer abstrakten Klasse muss alle abstrakten Methoden der Oberklasse implementieren.
- ☐ Aus den als *final* bezeichneten Klassen können keine Unterklassen definiert werden.
- ☐ Eine abstrakte Klasse wird in einer Unterklasse konkretisiert, wenn alle ihre abstrakten Methoden implementiert werden

Welche Aussage über abstrakte Klassen ist **falsch**?

- ☐ Abstrakte Klassen dienen als Strukturierungsmittel von Software.
- ☐ Konstruktoren in abstrakten Klassen dürfen nur innerhalb von Konstruktoren einer Unterklasse aufgerufen werden.
- ☐ Konstruktoren in abstrakten Klassen dürfen nicht als **public** deklariert werden.
- ☐ Abstrakte Klassen dürfen keine Instanzvariablen beinhalten.

Welche der folgenden Aussagen über Interfaces ist **falsch**?

- ☐ Interfaces dienen zur besseren Strukturierung von Java-Bibliotheken
- ☐ Objekte können nicht von einem Interface erzeugt werden.
- ☐ In Interfaces können Instanzvariablen deklariert werden.
- ☐ Alle Methoden eines Interface sind abstrakt.

Mit der *throw*-Anweisung wird ein Ausnahmefehler (Exception)

- ☐ behandelt
- ☐ geworfen
- ☐ delegiert
- ☐ erzeugt

Was ist **falsch** über Java-Objekte?

- ☐ Der Zugriffsversuch auf Komponenten eines nicht existierenden Objekts verursacht eine *NullPointerException* in Java.
- ☐ Ein Objekt ist eine Instanz einer Klasse.
- ☐ Objekte aus abstrakten Klassen sind abstrakte Objekte.
- ☐ Ein Objekt kann seinen lokalen Zustand ändern.

Was ist **wahr** über Java-Objekte?

- ☐ Objekte können nie andere Objekte löschen.
- ☐ Ein Objekt weiß immer, zu welcher Klasse es gehört.
- ☐ Objekte können in Java auch ohne Klassendefinitionen existieren.
- ☐ Objekte können nicht andere Objekte beinhalten.

Was ist **falsch** über den *private*-Modifizierer?

- ☐ *private*-Variablen sind nur innerhalb der Klasse sichtbar.
- ☐ Methoden können als *private* deklariert werden.
- ☐ Konstruktoren können als *private* deklariert werden.
- ☐ *private*-Methoden können auch innerhalb von Unterklassen verwendet werden.

Was ist **falsch** über Methoden?

- ☐ Statische Methoden können Instanzvariablen nicht direkt verändern.
- ☐ Instanzmethoden können nur bezüglich eines Objekts aufgerufen werden.
- ☐ Methoden können auch innerhalb anderer Methoden programmiert werden.
- ☐ Methoden definieren das Verhalten von Objekten.

Was ist **falsch** über Exceptions?

- ☐ Ausnahmen (Exceptions) sind unerwartet auftretende Laufzeitfehler.
- ☐ Das Erzeugen einer Exception im Programm führt immer zum Programmabsturz.
- ☐ Eine Ausnahme (Exception) in Java ist ein Objekt.
- ☐ Die Behandlung von Ausnahmen (Exceptions) kann mit Hilfe der **throws**-Anweisung an die anrufende Methode delegiert werden.

Was ist **falsch** über Klassen- und Instanzvariablen?

- ☐ Innerhalb eines Java-Programms wird nur ein Exemplar von jeder Klassenvariablen erzeugt.
- ☐ *final*-Variablen sind automatisch auch Klassenvariablen.
- ☐ Klassenvariablen können nur innerhalb von Klassenmethoden verändert werden.
- ☐ Instanzvariablen können nicht innerhalb von Klassenmethoden verändert werden.

8. Aufgabe

Betrachten Sie folgende drei Klassen. Was wird während der Ausführung der main-Methode der Tester-Klasse ausgegeben?

```
public class Tester {

  public void start(){
        A a = new A("A");
        B b1 = new B("Anna");
        B b2 = new B("Bob");

        b2.m1();
        a.m1();
        b1.m1();

        b1.m2();
        a.m2();

        b1.m3();
        a.m3();
  }

  public static void main( String[] args){
        Tester t = new Tester();
        t.start();
  }

}// end of class Tester

public class A {

  public A(){
        System.out.println("A");
  }

  public void m1(){
        System.out.println("A.m1");
  }

  public void m2(){
```

```
            System.out.println("A.m2");
    }

    public void m3(){
            m1();
    }

  }// end of class A

  public class B extends A{

    public B(String s){
            System.out.println(s);
    }

    public void m1(){
            System.out.println("B.m1");
    }

    public void m2(){
            super.m2();
            System.out.println("B.m2");
    }

  }// end of class A
```

Ausgabe:

_____

_____