

# Multilevel Peer-Review zur Bewertung von Einzelleistungen in Programmierprojekten

Kai Eckert

Abschlussprojekt im Rahmen des  
Baden-Württemberg Zertifikats für Hochschuldidaktik  
Hochschuldidaktikzentrum (HDZ)  
Universität Mannheim  
Juni 2010

## 1 Einführung

In diesem Artikel beschreiben wir den Einsatz von Peer-Review-Verfahren auf zwei verschiedenen Ebenen (multi-level), um die Leistungen von Studenten in Übungsgruppen zu bewerten.

### 1.1 Motivation

Vorlesungsbegleitende Übungen in der Informatik bestehen häufig ganz oder teilweise aus Programmieraufgaben, die die Studenten allein oder in Gruppen bearbeiten müssen. Je nach Umfang der erforderlichen Implementierung und den Freiheitsgraden, die die Studenten dabei haben, ist die Bewertung dieser Arbeiten sehr zeitaufwändig, insbesondere, wenn der Dozent den Studenten eine detaillierte Rückmeldung zur eingereichten Lösung geben möchte.

Erfolgt eine derartige Rückmeldung nicht, sondern nur eine einfache Bewertung, inwieweit die gestellte Aufgabe durch die eingereichte Lösung erfüllt wurde, so wird ein großer Teil der potentiellen Lernerfolge einer praktischen Übung nicht genutzt. Das ist insbesondere deshalb problematisch, weil praxisnahe Programmieraufgaben für die Studenten mit einem erheblichen Zeitaufwand verbunden sind und es daher angebracht ist, die so investierte Zeit auf die bestmögliche Art in Lernerfolge umzuwandeln.

Programmieren lässt sich am besten durch Ausprobieren und das Studieren von fremdem Code erlernen. Legt man diese Aussage zu Grunde, so folgt, dass man die Lernziele am besten unterstützt, wenn die Studenten nicht nur an ihrer eigenen Lösung arbeiten, sondern auch die Lösungen anderer Studenten studieren. Im Rahmen dieses Projekts wollen wir dieses Studieren fremder Lösungen fest in den Korrektur- und Bewertungsprozess integrieren, um so den

Lernerfolg der Studierenden zu maximieren und gleichzeitig die Zeit des Dozenten in optimaler Weise zu nutzen, um diesen Lernerfolg auf allen Ebenen zu gewährleisten.

## 1.2 Realisierung im Projekt

Neben der theoretischen Erarbeitung der Schritte, die zur Umsetzung eines Peer-Review-Verfahrens in einer Programmierübung notwendig sind, geben wir in diesem Artikel auch die Erfahrungen wider, die sich bei der praktischen Umsetzung ergeben haben.

Zum Einsatz kam das Verfahren in der Übung zur Vorlesung "Digitale Bibliotheken", die im Herbst-/Wintersemester 2009 an der Universität Mannheim gehalten wurde. Die Vorlesung ist stark an der Praxis orientiert und behandelt ein breites Themenspektrum zum Aufbau und Betrieb von sogenannten Digitalen Bibliotheken. Neben den rechtlichen, wirtschaftlichen und psychologischen Rahmenbedingungen wird insbesondere auf die Techniken des Information Retrieval eingegangen. Diese bilden auch den Schwerpunkt der vorlesungsbegleitenden Übung, in der die Studierenden nach und nach ein eigenes Retrieval-System aufbauen.

Die Vorlesung ist noch sehr neu und wurde zum zweiten Mal gehalten. Wir haben mit 15-20 Teilnehmern gerechnet, leider waren es letztlich aber nur vier Studierende, die an der Vorlesung teilgenommen haben. Dadurch war es nicht möglich, das hier vorgestellte Verfahren vollständig umzusetzen. Um trotzdem erste Erfahrungen sammeln zu können, haben wir uns entschieden, mit zwei Zweiergruppen zu arbeiten und die Bewertung der Gruppen untereinander wie im Folgenden beschrieben durchzuführen. Auf die gegenseitige Bewertung der Gruppenmitglieder innerhalb einer Gruppe mussten wir naheliegenderweise verzichten, da keine Anonymität bei der Bewertung gegeben wäre.

## 2 Einordnung

Im Folgenden beschreiben wir einige grundlegende Überlegungen zur Gruppenarbeit und zum Peer-Review und geben einen Überblick über andere Veröffentlichungen zum Thema, die in unsere Konzeption mit eingeflossen sind.

### 2.1 Gruppenarbeit

Die Frage, ob Gruppenarbeit im Rahmen einer universitären Arbeit, bei der es letztlich immer um die Bewertung einzelner Studenten geht, überhaupt wünschenswert ist, geht über den Rahmen dieser Ausarbeitung hinaus.

Unabhängig davon sind Gruppenarbeit und Gruppenabgaben in vielen Veranstaltungen gängige Praxis, insbesondere in großen Veranstaltungen, bei denen die Korrektur von Einzelabgaben kaum zu leisten wäre. Von der Bewertung dieser Abgaben hängt dann meistens auch nur die Zulassung zu einer abschlie-

ßenden Klausur ab, mit der die zu vergebende Note für jeden Studenten einzeln ermittelt wird.

Es gibt gute Gründe, Projekte oder Übungen in Gruppenarbeit durchführen zu lassen: Willmot und Crawford zeigen die Parallelen zur Wirklichkeit in der Arbeitswelt auf, in der Teamarbeit immer wieder gefordert ist. Und diese Teamarbeit müssen die Studenten lernen und üben: “Studenten werden besser in der Teamarbeit durch Praxiserfahrung und [...] obwohl die Anleitung zur Teamarbeit ihre Daseinsberechtigung hat, ist Erfahrung effektiver.” [14]

## 2.2 Peer Assessment in der Literatur

Einen sehr ähnlichen Ansatz zu unserem Vorhaben verfolgen Sitthiworachart und Joy, die in einem Programmierkurs ein webbasiertes System zum Peer-Review eingesetzt haben. Allerdings arbeiten die Studenten hier nicht in einer Gruppe, sondern einzeln. Jede Abgabe wird anschließend von drei Mitstudenten bewertet, somit wird von jedem Studenten erwartet, drei andere Arbeiten selbst zu bewerten. Der ganze Prozess ist anonymisiert, kein Student weiß, wer seine Arbeit bewertet und welche Arbeiten er gerade bewertet.

In [11] gehen die Autoren darauf ein, wie dieses Peer Review Verfahren von den Studenten wahrgenommen wird. So ist zum Beispiel Anonymität bei der Bewertung ein wichtiges Kriterium, um eine Bewertung von Freunden zu vermeiden. Es gibt aber auch generelle Bedenken, dass eine Bewertung durch Mitstudenten qualitativ einer Bewertung durch den Dozenten entspricht. Auf diese Fragestellungen gehen die Autoren in [12] weiter ein, in dem sie die Übereinstimmung von Ergebnissen der Studenten und der Dozenten untersuchen. Sitthiworachart und Joy betonen auch, dass es wichtig ist, die Bewertung selbst zu bewerten, weil sonst vielfach der Anreiz fehlt, Zeit und Mühe in die Bewertung der Mitstudenten zu investieren.

Gehring [5] beschreibt ein webbasiertes System, das ebenfalls zur Bewertung von Einzelleistungen durch Mitstudenten genutzt wird. Hier wird den Studenten die Möglichkeit gegeben, die eigene Lösung anhand des Feedbacks nachzubessern, bevor dann letztlich eine Note vergeben wird. Die Bewertungen werden anschließend ebenfalls im Peer-Review-Verfahren durch weitere Studenten bewertet.

Auf Besonderheiten bei der gemeinsamen Erstellung von Textabgaben geht Speck [13] ein. Durch das Verfassen von Kritiken zu den Texten anderer Gruppen werden die Studenten in den Bewertungsprozess eingebunden. Speck geht auch auf Fragen zur gerechten Bewertung der einzelnen Gruppenmitglieder ein. Auf Aspekte der Gruppendynamik und die Bildung von Gruppen für das kollaborative Schreiben gehen auch Connery und Vohs [4] näher ein.

Chinn [1] nutzt Peer-Assessment zusätzlich zu traditionellen Übungsaufgaben in einer Algorithmen-Vorlesung. Jeder Student nimmt einmal an einer Gruppenaufgabe teil, um ein Problem zu lösen. Die Lösungen werden innerhalb der Vorlesung publiziert und alle Studenten müssen Bewertungen für die Lösung abgeben. Dadurch, dass alle anderen Aufgaben weiterhin traditionell in Einzelabgabe gefertigt und durch den Dozenten bewertet werden, kann Chinn die

Ergebnisse der Peer-Bewertungen mit den Abgaben der Studenten vergleichen. Er kommt zu dem Schluss, dass es eine hohe Korrelation zwischen der Qualität der Abgaben und der Qualität der Bewertungen im Peer-Review-Verfahren gibt.

Ebenfalls Erfahrungen innerhalb einer Algorithmen-Vorlesung beschreibt Papaöaskari [8]. Sie nennt als Gründe für ein Peer-Review-Verfahren unter anderem, dass Studenten oft in engen Bahnen denken, wenn sie ihre eigene Lösung erarbeiten und so von anderen Lösungen profitieren können, die andere Wege nutzen. Außerdem erkennen Studenten den Wert von guter Dokumentation und klarem Programmierstil, wenn sie mit den Abgaben anderer Studenten konfrontiert werden.

Im Bereich der Software-Entwicklung lässt sich Peer-Review sehr gut über das Qualitätsmanagement bei realen Software-Projekten motivieren. Insbesondere ist das effiziente Testen von Software eine eigenständige Aufgabe, ähnlich dem Verfassen von Textkritiken. Clark et al. beschreiben entsprechende Projekte zum Peer-Testing [2] und Peer-Review [3].

Eine kritische Betrachtung zur Peer-Bewertung liefert Kennedy [6], der unter anderem zum Ergebnis kommt, dass der Aufwand, der für die Peer-Bewertung betrieben werden muss, von den eigentlichen Lerninhalten ablenkt.

Eine Einweisung in die Umsetzung von Peer-Review-Verfahren gibt Race [9]. Er motiviert insbesondere auch das gruppeninterne Peer-Review: "Es ist relativ klar, wie man die Produkte studentischer Gruppenarbeit bewertet. Schwierig ist es, die Gruppenarbeit selbst zu bewerten und den relativen Beitrag der einzelnen Gruppenmitglieder zur Entwicklung des bewerteten Produkts." (S. 17, Übers. d. d. Verf.). Und weiter auf der selben Seite: "Wenn es um die Bewertung der Einzelbeiträge zur Gruppenarbeit geht, sind die einzigen Personen, die wissen, was die Einzelbeiträge waren, die Studenten selbst".

Eine ähnliche Einführung für das Fach Biologie findet sich bei Orsmond [7], ergänzt um mehrere Fallbeispiele.

Eine konkrete Software zur Unterstützung von gruppeninternem Peer-Review ist Web-PA. Da die Software in unserem Projekt wegen der zu geringen Anzahl der Studenten nicht zum Einsatz kommen konnte, konnten wir keine Erfahrungen dazu sammeln. Ergebnisse dazu finden sich aber bei Willmot und Crawford [14], sowie bei Robinson [10].

### 3 Multilevel Peer-Review

Peer-Review Ansätze werden in der Literatur in den meisten Fällen verwendet, um Leistungen einzelner innerhalb einer Gruppe zu differenzieren. Diese Form des Peer-Reviews verwenden wir ebenfalls innerhalb der einzelnen Gruppen, um aus der Gesamtleistung der Gruppe individuelle Einzelleistungen zu berechnen. Neben diesem Bewertungseffekt soll dadurch vor allem die Motivation der einzelnen Studenten gesteigert werden, sich in die Gruppe einzubringen und ein Mitläufertum innerhalb einer Gruppe möglichst vermieden werden.

Der Kern dieses Projekts ist jedoch die Einbeziehung der Studenten bei der eigentlichen Bewertung der eingereichten Lösungen. Dazu bewertet eine Gruppe

die Ausarbeitung einer anderen Gruppe und gibt dazu detailliertes Feedback. Auf dieser Ebene verhalten sich also die einzelnen Gruppen wie Peers.

Im folgenden gehen wir auf dieses Peer-Review zwischen den Gruppen (Inter-Group Peer-Review) näher ein, bevor anschließend das Review-Verfahren innerhalb der einzelnen Gruppen (Intra-Group Peer-Review) erläutert wird.

### **3.1 Inter-Group Peer-Review**

In diesem Abschnitt beschreiben wir den Kern unserer Veranstaltungsplanung. Man kann das Peer-Review zwischen den Arbeitsgruppen zwar als eine spezielle Form des Peer Assessments sehen, so wie sie in der Literatur vielfach beschrieben wird, aber es gibt einige Aspekte, die beim Peer-Review zwischen Gruppen zu beachten sind, die in den üblicherweise beschriebenen Szenarien keine Rolle spielen. Auf diese Besonderheiten gehen wir im folgenden ein:

#### **3.1.1 Implikationen für die Makroplanung einer Veranstaltung**

Beim Inter-Group Peer-Review übernimmt eine Arbeitsgruppe für eine andere Arbeitsgruppe die Rolle des Korrektors. Diese ungewohnte Rolle verlangt von den Studenten einigen Aufwand, der gerade am Anfang noch schwer einzuschätzen ist. Es ist daher angebracht, den Zeitraum für die Korrektur ausreichend lang zu wählen, auf jeden Fall weit länger, als man für die Korrektur durch den Übungsleiter einplanen würde.

Der Zeitraum kann sich weiter verlängern, wenn der Übungsleiter noch eine Nachkorrektur vornimmt. Auf die verschiedenen Möglichkeiten zur Einflussnahme durch den Übungsleiter gehen wir in Abschnitt 3.3 näher ein.

Das bedeutet, dass zwischen dem Beginn und dem endgültigen Abschluss einer Arbeitseinheit mehrere Wochen liegen und sich die Arbeitsabschnitte deutlich überschneiden. In Abschnitt 4 zeigen wir eine exemplarische Zeitplanung, wie wir sie im Rahmen unserer Veranstaltung eingesetzt haben.

#### **3.1.2 Die Zuordnung der Gruppen**

Das entscheidende Problem beim Inter-Group Peer-Review ist die Zuordnung der Gruppen zueinander. Prinzipiell gibt es die folgenden Möglichkeiten, die Zuordnung umzusetzen:

1. Jede Gruppe korrigiert alle anderen Gruppen, optional bewertet sie sich dabei auch selbst. Dieses Setup entspräche genau dem Intra-Group Peer-Review, aber der Korrekturaufwand wächst dadurch enorm. Eine denkbare Abwandlung wäre es, nur eine bestimmte Anzahl Gruppen als Korrektor für eine Gruppe zuzuordnen.
2. Je zwei Gruppen bewerten sich gegenseitig, die Kombination bleibt über die gesamte Veranstaltung bestehen.
3. Je zwei Gruppen bewerten sich gegenseitig, nach jeder Arbeitseinheit wird neu kombiniert.

4. Jede Gruppe bewertet eine andere Gruppe, keine gegenseitige Bewertung, Kombinationen bleiben bestehen.
5. Jede Gruppe bewertet eine andere Gruppe, keine gegenseitige Bewertung, nach jeder Arbeitseinheit wird neu kombiniert.

Zuordnung 1 mit allen Gruppen hat den Vorteil, vollkommen fair zu sein, weil es keinerlei Seiteneffekte durch die Gruppenzuordnungen geben kann. Vor allem zwei Punkte sprechen allerdings gegen diese Form der Zuordnung:

Erstens der insgesamt erhöhte Arbeitsaufwand, weil bei  $N$  Gruppen jede Abgabe  $N - 1$  mal korrigiert würde. Dabei ist auch zu berücksichtigen, dass die Korrekturen von den Gruppen ja auch in irgendeiner Form nachbereitet werden müssen. Dabei ist davon auszugehen, dass sich viele Korrekturhinweise überschneiden, was dazu führt, dass diese Arbeit teilweise als unnötig empfunden wird.

Zweitens wird durch diese Form der Zuordnung ein Hauptvorteil des Peer-Reviews zunichte gemacht: Eine Gruppe kann sich dann nicht mehr die Zeit nehmen, detailliert auf die Lösung der anderen Gruppe einzugehen, sondern muss ähnlich wie der Übungsleiter zügig alle Gruppenergebnisse durchgehen. Umgekehrt ist davon auszugehen, dass mit der erhöhten Verantwortung, als einzige Gruppe eine andere zu korrigieren, auch die Motivation steigt, Zeit und Energie in die Korrektur zu investieren.

Eine Abschwächung dieser Einwände lässt sich erreichen, in dem man eine bestimmte Anzahl von Gruppen als Korrektor einer Gruppe bestimmt. In diesem Fall muss man allerdings wie bei den anderen Gruppierungen Entscheidungen treffen, was die folgenden zwei Aspekte angeht:

1. Bewerten sich Gruppen gegenseitig?
2. Bleiben die Gruppenzuordnungen über die ganze Veranstaltung bestehen?

Eine gegenseitige Bewertung der Gruppen kann dazu führen, dass die Gruppen sich bei der Bewertung der anderen Gruppe von deren Bewertung für die eigene Gruppe beeinflussen lassen. Dadurch ist die Unabhängigkeit bei der Bewertung der anderen Gruppe unter Umständen nicht mehr gegeben.

Auf der anderen Seite verbindet die gegenseitige Bewertung zwei Gruppen sehr stark, insbesondere, wenn die Gruppenzuordnungen nicht verändert werden. Das kann sich positiv auf die Motivation zur Bewertung der anderen Gruppe auswirken. Nicht zuletzt erleichtert die gegenseitige Bewertung, dass sich Gruppen zu Feedback-Gesprächen treffen, da für zwei Gruppen nur ein gemeinsamer Termin gefunden werden muss.

Der zweite Aspekt betrifft die Frage, ob die Zuordnung der Gruppen über die Veranstaltung geändert werden sollte. Bleiben die Zuordnungen bestehen, ergeben sich sehr starke Verbindungen zwischen den zugeordneten Gruppen, insbesondere, wenn diese sich auch noch gegenseitig bewerten.

Handelt es sich bei den Aufgabenstellungen um aufeinander aufbauende Programmieraufgaben, dann spricht für eine feste Zuordnung insbesondere der ge-

ringere Zeitaufwand bei der Korrektur, da sich die Gruppe nicht jedes Mal auf gänzlich neue Lösungsansätze einstellen muss.

Andererseits kann eine feste Gruppenzuordnung zu Problemen führen, wenn ungeeignete Gruppen aufeinandertreffen. Dem lässt sich am besten entgegenen, wenn schon bei der Zusammensetzung der Gruppen dafür gesorgt wird, dass die Gruppen eine ähnlich Struktur und Leistungsfähigkeit haben. Das lässt sich im Vorfeld allerdings nur schwer erreichen, insbesondere, wenn die Studenten ihre Gruppen selbst bilden können.

Das Hauptproblem sind Gruppen, die deutlich leistungsschwächer sind, als die anderen Gruppen. Es ist davon auszugehen, dass sich die schlechte Leistung auch auf die Korrektur der anderen Gruppe auswirkt. Für die andere Gruppe ergibt sich daraus der Nachteil, dass sie mit einer undankbaren Korrekturaufgabe konfrontiert sind und dafür nur weniger nützliches Feedback für die eigene Arbeit erhalten.

Solche Auswirkungen auf eine einzelne Gruppe lassen sich vermeiden, indem die Gruppen nach jeder Arbeitsphase durchgewechselt werden. In jedem Fall muss durch den Übungsleiter gegebenenfalls eine schwache oder fehlende Korrektur bei einer Gruppe durch entsprechende Betreuung kompensiert werden, damit es hier zu keinem Einbruch der Motivation bei einer so betroffenen Gruppe kommt.

## 3.2 Intra-Group Peer-Review

Nachdem im letzten Abschnitt für jede Gruppe eine Bewertung erstellt wurde, geht es in diesem Abschnitt darum, diese Gruppenbewertung innerhalb einer Gruppe nach den einzelnen Gruppenmitgliedern zu differenzieren. Auf diese Weise erhält jeder Student eine individuelle Bewertung seiner Leistung.

### 3.2.1 Überblick

Wenn in der wissenschaftlichen Literatur von Peer-Review oder Peer-Assessment die Rede ist, so ist überwiegend dieses Review innerhalb einer Gruppe gemeint. Die Ausdifferenzierung der einzelnen Leistungen ist fakultativ und kann selbstverständlich auch entfallen.

Neben der Möglichkeit, Bewertungen für einzelne Studenten zu erstellen, gibt es noch weitere Gründe für die Durchführung des Peer-Reviews innerhalb einer Gruppe<sup>1</sup>:

- Entwicklung von Schlüsselqualifikationen
  - Teamarbeit,
  - Kommunikation,
  - Zwischenmenschliche Beziehungen,
  - Organisation.

---

<sup>1</sup>Weitere Informationen dazu finden sich auf der folgenden Website: <http://webpa.wiki.sourceforge.net/Supplementary+Information+About+Peer+Assessment>

- Erweiterte Motivation durch
  - aktive Teilnahme am Lernprozess,
  - Einbeziehung des Studenten bei der Bewertung.
- Verbesserte Akzeptanz der Rückmeldungen und Bewertungen durch
  - verbessertes Verständnis für die Bewertungsanforderungen,
  - die gezielte Stärkung der eigenen Leistung.

### 3.2.2 Beschreibung des Verfahrens

Das Ziel des Intra-Group Peer-Reviews ist die Ausdifferenzierung der Gruppenleistung auf die einzelnen Gruppenmitglieder. Dazu bewerten sich die Gruppenmitglieder gegenseitig hinsichtlich ihrer jeweiligen Leistung innerhalb der Gruppe.

Bei der Bewertung dieser Einzelleistungen geht es nicht um die Bewertung der fachlichen Qualifikation des Studenten, diese erfolgt allein über die Bewertung der eingereichten Lösung der gesamten Gruppe. Stattdessen wird hier bewertet, wie sich der Student in die Gruppe eingebracht hat und damit, inwiefern er zum Gruppenergebnis beigetragen hat. Dieser Beitrag kann natürlich durch ein überdurchschnittliches Fachwissen erfolgt sein, aber ebenso zählen hier Fähigkeiten, die Gruppe als ganzes zu motivieren, Diskussionen zu moderieren oder die Übernahme von Teilverantwortung, wie das abschließende Zusammenstellen und Einreichen der Lösung.

Das hier beschriebene Verfahren soll also dazu beitragen, die einzelnen Facetten der Teamarbeit, die alle zum Gesamtergebnis einer Gruppe beitragen, zu bewerten. Dazu beantworten die Studenten Fragen, die auf diese Facetten abzielen:

- Hat der Teilnehmer regelmäßig an Gruppentreffen teilgenommen?
- Hat der Teilnehmer sein Fachwissen in die Gruppe eingebracht?
- Hat der Teilnehmer eigene Ideen zur Lösung entwickelt?
- Hat der Teilnehmer an der Entwicklung der Lösung teilgenommen?
- Hat der Teilnehmer an der schriftlichen Ausarbeitung mitgearbeitet?
- Wie ist die Leistung des Teilnehmers relativ zur Gruppe zu bewerten?
- Hat der Teilnehmer unabhängig von den fachlichen Inhalten zur Gruppenarbeit beigetragen?
- Hat der Teilnehmer sich auch bei der Bewertung der anderen Gruppe eingebracht?



Diese Fragen können und sollen natürlich an die jeweiligen Erfordernisse angepasst werden. Wichtig erscheint uns vor allem, dass sowohl die fachlichen als auch die überfachlichen Beiträge der Teilnehmer zur Gruppenarbeit bewertet werden. Die Fragen, die ja den Teilnehmern spätestens nach der ersten Bewertungsrunde bekannt sind, sollen auch motivieren, sich in allen Facetten der Gruppenarbeit einzubringen. Dadurch können auch schwächere Studenten motiviert werden, sich einzubringen, selbst wenn sie fachlich weniger zur Lösung beitragen konnten.

### **3.3 Einflussnahme durch den Übungsleiter**

Wird eine Veranstaltung nach dem hier vorgestellten Peer-Review Konzept organisiert, nimmt der Übungsleiter vor allem die Rolle des Moderators ein, da ein Großteil der Bewertungs- und Korrekturarbeit auf die Studenten übertragen wird. Insbesondere bei Problemen tritt er auch als Schiedsrichter auf und muss dabei gegebenenfalls in die Bewertungen der Gruppen eingreifen können.

Darüber hinaus ist es auch denkbar, dass er aktiv an der Bewertung teilnimmt, indem er beispielsweise eine eigene Bewertung für die Arbeit jeder Gruppe abgibt und diese in das Endergebnis eingerechnet wird.

Ein anderer Ansatz, den wir in unserem Fall verfolgen, ist die Bewertung der Korrekturleistung einer Gruppe durch den Übungsleiter. Dadurch wird ein zusätzlicher Anreiz für die Gruppen geschaffen, Energie in die Korrekturarbeit zu investieren. Ausserdem erhält der Übungsleiter dabei einen guten Überblick über die Arbeiten der einzelnen Gruppen, sein Einarbeitsaufwand reduziert sich in den meisten Fällen allerdings erheblich, da er auf den Korrekturergebnissen der korrigierten Gruppe aufbauen kann. Außerdem kann der Übungsleiter so Gruppen erkennen, die kein ausreichendes Feedback für die andere Gruppe liefern und das entsprechend ausgleichen.

Kriterien für die Bewertung der Korrekturabgaben können sein:

- Ist die objektive Einschätzung der Abgabe korrekt:
  - Erfüllung der gestellten Aufgabe
  - Erfüllung der vorgegebenen Kriterien.
- Ist das Feedback verständlich geschrieben?
- Sind Lob und Kritik in der gegebenen Form angemessen?
- Stimmt der Dozent bei der Vergabe der Punkte mit den Studenten überein?

## **4 Makroplanung der Veranstaltung und konkrete Umsetzung des Peer-Review Verfahrens**

In diesem Abschnitt beschreiben wir den äußeren Rahmen der Veranstaltung, in der wir das hier beschriebene Verfahren umgesetzt haben.

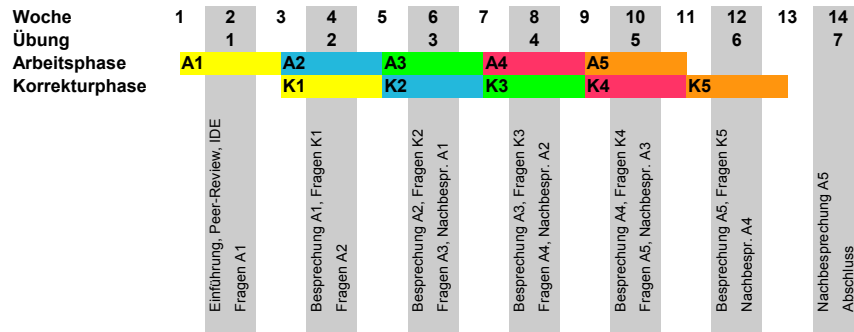


Abbildung 1: Planung der Veranstaltung. Basierend auf einem Semester von 14 Wochen und einer zweiwöchentlichen Übung. Die Übung untergliedert sich in fünf Arbeits- und Korrekturphasen.

#### 4.1 Makroplanung

Abbildung 1 zeigt einen beispielhaften Zeitablauf für die Übung. Hier haben wir ein Semester von 14 Wochen zugrundegelegt. Die Übungsveranstaltungen finden nur zweiwöchentlich statt.

#### 4.2 Generelle Umsetzung

Die Studenten konnten sich ihre Arbeitsgruppe aussuchen. Dadurch wollen wir erreichen, dass funktionierende Arbeitsstrukturen innerhalb der Studenten erhalten bleiben können. Die Größe der Arbeitsgruppe wird vorgegeben, idealerweise vier Studenten pro Gruppe, das kann natürlich je nach Gesamtzahl der Studenten variieren. In unserem Fall waren wir gezwungen, Zweiergruppen zu bilden, was generell vermieden werden sollte, weil dann keine anonyme Bewertung innerhalb der Gruppe stattfinden kann. Aus diesem Grund haben wir auf die Umsetzung des Intra-Group Peer-Review verzichtet.

Grundsätzlich werden die Gruppen nach jeder Arbeitsphase neu zugeordnet, jede Gruppe korrigiert die Arbeit genau einer anderen Gruppe. Dabei bewerten sich die Gruppen nie gegenseitig. Das hat den Vorteil, dass keine gerade Anzahl von Gruppen gebraucht wird. Ein wichtiges Lernziel der Veranstaltung soll sein, dass die Gruppen Einblick in eine möglichst große Zahl von Lösungsansätzen erhalten.

Leider mussten wir auch hier Abstriche machen, da wir nur zwei Arbeitsgruppen zur Verfügung hatten. In diesem Fall war also die Zuteilung fest, bei gegenseitiger Bewertung.

Um die Motivation zur Bewertung der anderen Gruppe weiter zu erhöhen, haben wir es in der Übung explizit erlaubt, in der Folgeabgabe auf dem Code der anderen Gruppe aufzusetzen, anstatt mit dem eigenen Code weiterzuarbeiten.

Auch eine teilweise Übernahme von fremdem Code war gestattet. Diese Vorgehensweise halten wir für sehr sinnvoll, da so nie bei einer Gruppe das Gefühl entstehen kann, den Anschluss an die anderen verloren zu haben, was gerade bei aufeinander aufbauenden Aufgaben der Fall sein kann.

### 4.3 Richtlinien zur Korrektur

Um eine einheitliche Bewertung zwischen den Gruppen zu erreichen und den Lerneffekt im Hinblick auf Qualitätssicherung in der Softwareentwicklung zu maximieren, geben wir genaue Kriterien vor, nach denen sich die Bewertung richtet. Für jedes Kriterium sind Punkte zu vergeben, die dann automatisch zu einer Gesamtbewertung verrechnet werden.

- Erfüllt die (vorliegende) Lösung die gestellte Aufgabe?
- Ist der Quelltext gut dokumentiert?
- Ist der Quelltext einheitlich formatiert?
- Ist die zusätzliche Dokumentation ausreichend/gut?
- Wurden gängige Standards (OOP, MVC...) eingesetzt?
- Ist die Umsetzung möglichst einfach gehalten?
- Kann die Umsetzung flexibel weiterentwickelt werden?

Dabei ist entscheidend, dass ein Teil der Bewertung für objektiv zu bewertende Kriterien erfolgt, die auch von einer schwächeren Gruppe sehr gut erfüllt und kontrolliert werden können.

Neben der Bewertung der Lösung ist ein ausführliches Feedback an die andere Gruppe wichtig. Auf der folgenden Seite findet sich der Bewertungsbogen, der im Rahmen der Übung genutzt wurde.

**Punktevergabe.** Die maximale Anzahl der zu vergebenen Punkte richtet sich nach der jeweiligen Aufgabe und wird auf dem Aufgabenblatt angegeben. Für jede Teilaufgabe ist ein eigener Bewertungsbogen auszufüllen.

# Bewertungsbogen

Übungsblatt Nummer \_\_\_\_\_ Aufgabe \_\_\_\_\_ Max. Punkte \_\_\_\_\_

Namen der Korrektoren

---

Namen der Gruppenmitglieder der Gruppe, die hier bewertet wird:

---

Bitte dieses Blatt für **jede Teilaufgabe getrennt** ausfüllen und bis zur Abgabefrist beim Betreuer, im Sekretariat oder in der Übung abgeben. **Zusätzlich** zu diesem Formular ist ein Bericht in angemessenem Umfang (etwa eine Seite) zu erstellen, in dem detailliert auf die einzelnen Abzüge eingegangen wird und möglichst konstruktives Feedback für die andere Gruppe gegeben wird.

	<i>Kriterium</i>	<i>Ja</i>	<i>Weitgehend</i>	<i>Teilweise</i>	<i>Nein</i>	<i>Punktabzug</i>
1	Erfüllt die (vorliegende) Lösung die gestellte Aufgabe?					
2	Ist der Quelltext gut dokumentiert?					
3	Ist der Quelltext einheitlich formatiert?					
4	Ist die zusätzliche Dokumentation ausreichend/gut?					
5	Wurden gängige Standards (OOP, MVC...) eingesetzt?					
6	Ist die Umsetzung möglichst einfach gehalten?					
7	Kann die Umsetzung flexibel weiterentwickelt werden?					

Erreichte Punktzahl: \_\_\_\_\_

Datum, Unterschriften der Korrektoren:

---

## 4.4 Abschließende Bewertung

Alle Abgaben der Gruppen wurden zum definierten Termin an den Dozenten und gleichzeitig an die andere Gruppe zur Korrektur gesendet. Die Ergebnisse der Korrekturen werden von den Gruppen zunächst an den Dozenten geschickt. Dieser bewertet die Korrekturleistung jeder Gruppe und vergibt darauf Punkte. Diese Punkte werden zum Gesamtergebnis der Gruppe auf die eigene Abgabe hinzugezählt (in unserem Fall wurde die gleich maximale Anzahl Punkte für die Korrektur vergeben, so dass die Abgabe und die Korrektur zu je 50% in die Gesamtbewertung eingingen).

Das Gesamtergebnis einer Gruppe für eine Arbeitsphase besteht also aus dem Wert, den sie von der zugewiesenen Korrekturgruppe erhalten hat, und dem Wert, den der Übungsleiter als Bewertung ihrer Korrekturarbeit vergeben hat.

Beispiel für Gruppe A und Gruppe B:

- Schritt 1: Gruppe A und Gruppe B geben ihre Lösungen ab.
- Schritt 2:
  - Gruppe A bewertet Gruppe B und vergibt 7 Punkte
  - Gruppe B bewertet Gruppe A und vergibt 9 Punkte
- Schritt 3:
  - Der Dozent bewertet die Bewertung der Gruppe A (für Gruppe B) und vergibt 8 Punkte
  - Der Dozent bewertet die Bewertung der Gruppe B (für Gruppe A) und vergibt 5 Punkte
- Schritt 4:
  - Gruppe A erhält  $9 + 8 = 17$  Punkte.
  - Gruppe B erhält  $7 + 5 = 13$  Punkte.

## 5 Zusammenfassung

In diesem Artikel haben wir die Grundlagen eines Peer-Review-Verfahrens beschrieben, das auf zwei Ebenen (Multi-Level) zum Einsatz kommen kann, um die Gruppenarbeit von Studenten in Übungen zu bewerten: Einmal die Bewertung der Abgabe einer Gruppe durch eine andere Gruppe (Inter-Group Peer Review) und die Bewertung der Arbeit der einzelnen Gruppenmitglieder innerhalb einer Gruppe zur Differenzierung der Einzelleistungen (Intra-Group Peer-Review).

Der Einsatz eines solchen Verfahrens begegnet dem Problem, dass bei klassischer Gruppenarbeit keine gerechte Einzelbewertung der Studenten stattfinden kann. Gleichzeitig unterstützt er die Lernziele einer Programmier-Übung, da er

die Auseinandersetzung mit fremdem Code und die Reflektion anderer Lösungswege als dem eigenen fördert.

Aufgrund der geringen Teilnehmerzahl konnte das Verfahren leider nicht vollständig umgesetzt werden, so dass wir uns auf das Inter-Group Peer-Review beschränken mussten. Eine systematische Auswertung ist bei nur vier Teilnehmern nicht zielführend, so dass wir uns hier auf die subjektiven Eindrücke des Dozenten beschränken. Die Studenten wurden zum ungewohnten Einsatz des Peer-Reviews befragt und äußerten sich alle positiv darüber. So hielt sich der zusätzliche Aufwand für die Erstellung der Korrekturen im Rahmen, wobei anzumerken ist, dass der Zeitaufwand insgesamt für die Lösung der gestellten Aufgaben schon als relativ hoch anzusehen ist, so dass der Korrekturaufwand zumindest im Vergleich weniger ins Gewicht fiel, als es bei anderen Veranstaltungen der Fall sein mag.

Auf den Lernprozess hatte das Peer-Review ebenfalls positiven Einfluss. Die Studenten empfanden die Auseinandersetzung mit der Lösung der anderen Gruppe als hilfreich, um die eigene Lösung besser einschätzen zu können. So nahm die eine Gruppe die Abgabe der anderen Gruppe zum Ansatz, bei der nächsten Abgabe den eigenen Code noch einmal komplett zu überarbeiten, um besser auf die zukünftigen Erweiterungen vorbereitet zu sein.

Für den Dozenten war der Korrekturaufwand allerdings merklich reduziert im Vergleich zum Vorjahr. Alle Aufgaben und Korrekturen wurden von den Studenten pünktlich abgegeben und waren generell von sehr guter bis guter Qualität. Diese Einschätzung kann aber kaum verallgemeinert werden, da die Anzahl der Studenten sehr klein war und es unter diesen keine schwachen Studenten und damit auch keine problematische Gruppe gab.

Insgesamt bewerten wir diese ersten Erfahrungen trotzdem als Erfolg und werden das Verfahren in Zukunft bei dieser und ähnlichen Veranstaltungen weiter einsetzen.

## Literatur

- [1] D. Chinn. Peer assessment in the algorithms course. In *ITiCSE '05: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, pages 69–73, New York, NY, USA, 2005. ACM.
- [2] N. Clark. Peer testing in Software Engineering Projects. In *Proceedings of the sixth conference on Australasian computing education*, pages 41 – 48, 2004.
- [3] N. Clark, P. Davies, and R. Skeers. Self and peer assessment in software engineering projects. In *Self and Peer Assessment in Software Engineering Projects*, pages 91–100, 2005.
- [4] B. A. Connery and J. L. Vohs. Group Work and Collaborative Writing. Technical report, University of California Davis, XXXX.

- [5] E. F. Gehring. Electronic peer review and peer grading in computer-science courses. *SIGCSE bulletin*, 33(1):139, 2001.
- [6] G. J. Kennedy. Peer-assessment in group projects: is it worth it? In *Proceedings of the 7th Australasian conference on Computing education*, pages 59–65, 2005.
- [7] P. Orsmond. *Self-and Peer Assessment - Guidance on practice in the Biosciences*. The Higher Education Academy, Centre for Bioscience, Leeds, 2004.
- [8] M.-A. Papalaskari. Peer evaluation in an algorithms course. In *ITiCSE '03: Proceedings of the 8th annual conference on Innovation and technology in computer science education*, pages 274–274, New York, NY, USA, 2003. ACM.
- [9] P. Race. *A briefing on self, peer & group assessment*. LTSN, York, 2001.
- [10] C. L. Robinson. Self-and Peer-Assessment in Group Work. In *13th SEFI European Seminar on Mathematics in Engineering Education, Buskerud University College, Kongsberg, Norway, June 11-14, 2006*, pages 42–50, 2006.
- [11] J. Sitthiworachart and M. Joy. Effective peer assessment for learning computer programming. volume 36, pages 122–126, New York, NY, USA, 2004. ACM.
- [12] J. Sitthiworachart and M. Joy. Computer support of effective peer assessment in an undergraduate programming class. *Journal of Computer Assisted Learning*, 24(3):217–231, 2008.
- [13] B. W. Speck. *Facilitating Students' Collaborative Writing. ASHE-ERIC Higher Education Report. Jossey-Bass Higher and Adult Education Series*. Jossey-Bass, 2002.
- [14] P. Willmot and A. Crawford. Peer review of team marks using a web-based tool: an evaluation. *Engineering Education: Journal of the Higher Education Academy Engineering Subject Centre*, 2(1):59–66, 2001.