Effekte automatischer Bewertungen für Programmieraufgaben in Übungs- und Prüfungssituationen

Michael Striewe, Michael Goedicke
Specification of Software Systems
Institut für Informatik und Wirtschaftsinformatik
Universität Duisburg-Essen, Campus Essen
{michael.striewe,michael.goedicke}@s3.uni-due.de

Abstract: Dieser Beitrag beschreibt und analysiert ein exemplarisches System zur automatischen Bewertung von Programmieraufgaben. Aus den Ergebnissen des mehrjährigens Einsatzes des Systems im Rahmen universitärer Lehrveranstaltungen werden Aussagen zur Qualität automatischer Bewertungen abgeleitet. Auf der Basis einer empirischen Untersuchung werden die Effekte automatischer Bewertungen auf die Lernenden in Übungs- und Prüfungsszenarien untersucht, um die Vorteile automatischer Bewertungen in Relation zu möglichen negativen Effekten setzen zu können.

1 Einleitung

Systeme zur automatischen Bewertung von Programmieraufgaben werden in verschiedener Form an einigen Universitäten angeboten. Neben verschiedenen technischen Details der Realisierung unterscheiden sie sich auch auf konzeptioneller Ebene im Bezug auf den Einsatzzweck, dem Verhältnis zwischen automatischer (Vor-)Korrektur und manueller (Nach-)Korrektur und die kontrollierbaren Aufgabeninhalte. Insbesondere unterscheiden sie sich auch darin, ob sie lediglich dazu dienen, Fehler in Lösungen aufzudecken oder ob sie auch in der Lage sein sollen, diese Fehler durch das Anbieten von Lösungshinweisen zu erläutern.

Am Beispiel eines konkreten Prüfungssystems, das sowohl in Übungs- als auch Prüfungssituationen seit drei Jahren im Einsatz ist, werden in diesem Beitrag zunächst mögliche Lösungen für die oben genannten Aspekte dargestellt und mit anderen existierenden Systemen verglichen. Anschließend werden aus den Erfahrungen im Einsatz des Systems Aussagen über die Qualität automatischer Bewertungen abgeleitet. Zudem werden die Effekte des eingesetzten Systems auf die Lernenden mit Blick auf verschiedene Einsatzszenarien anhand einer umfangreichen empirischen Untersuchung analysiert.

Prüfungssysteme für Programmieraufgaben eignen sich besonders als Untersuchungsgegenstand, da Programmierung Kreativität erfordert und fördert [Rom07]. Prüfungen in diesem Bereich erfordern daher offene Frageformen, bei denen es keine begrenzte Anzahl an möglichen Lösungen zur Auswahl gibt [Ros04]. Während Multiple-Choice-Aufgaben, Lückentexte und ähnliches zumindest nur eine endliche Menge von gültigen Lösungen zulassen, gibt es bei Programmieraufgaben im Allgemeinen unendlich viele Programme,

die den Anforderungen der Aufgabenstellung genügen. Die technischen Anforderungen an ein System zur automatischen Bewertung von Lösungen sind daher höher als bei anderen Aufgabentypen, wenn eine akzeptable Qualität der Bewertung und ein positiver Effekt auf die Lernenden erreicht werden soll.

2 Automatische Prüfung von Programmieraufgaben

Bei der automatischen Prüfung von Programmieraufgaben geht es darum, ohne manuelle Eingriffe eines Lehrenden zu entscheiden, ob eine Lösung den Anforderungen der Aufgabe entspricht. Grundsätzlich kann zwischen statischen und dynamischen Verfahren unterschieden werden. Statische Verfahren untersuchen Programmcode, ohne ihn auszuführen, und überprüfen beispielsweise die syntaktische Korrektheit sowie die Existenz oder die Abwesenheit bestimmter Programmkonstrukte. Dynamische Prüfverfahren führen Programmcode mit verschiedenen Eingaben aus und vergleichen die Ausgabe mit einer Musterlösung. Beide Verfahren ergänzen sich und sind in der Lage, Fehler zu entdecken, die dem jeweils anderen Verfahren verborgen bleiben.

Aus der Kombination der beiden Techniken ergibt sich insbesondere die Möglichkeit, die Ergebnisse einer Überprüfung mit Hinweisen zur Behebung eines Fehlers zu versehen. Beispielsweise kann ein dynamischer Test eine unerwartet lange Laufzeit eines Programms feststellen und es wegen des Verdachts auf eine Endlosschleife abbrechen. Ein zugehöriger statischer Test kann feststellen, dass in einem Schleifenkonstrukt im Programmcode ein Abbruchkriterium fehlt. Auf diese Weise erhält der Lernende nicht nur einen Hinweis auf ein beobachtetes konkretes Fehlverhalten des Programms, sondern auch auf eine mögliche Ursache im Programmcode bzw. umgekehrt nicht nur einen Hinweis auf einen Fehler im Programmcode, sondern auch einen Hinweis auf die mögliche Auswirkung dieses Fehlers. Prüfungssysteme, die derartige Techniken einsetzen, eigenen sich somit mutmaßlich sowohl für den Übungsbetrieb als auch für die Vorkorrektur von Prüfungen. Im Übungsbetrieb dienen die Meldungen dazu, dem Lernenden Hinweise auf eine zielgerichtete Verbesserung seiner Lösung zu geben, während sie als Vorkorrektur von Prüfungen als Hinweise für eine manuelle Nachkorrektur durch den Lehrenden interpretiert werden können. Je zutreffender diese Hinweise sind, umso geringer fällt der Aufwand für eine manuelle Nachkorrektur aus.

2.1 Das Prüfungssystem JACK

Das Prüfungssystem JACK (Java Checker) [SGB08] wurde zur Durchführung von Übungen und Testaten in der Programmiersprache Java für Studierende im ersten Semester entwickelt. Es ist derzeit ausschließlich zu diesem Zweck im Einsatz, kann aber prinzipiell auch auf andere Programmiersprachen oder gänzlich andere Aufgabentypen (z.B. UML-Diagramme) erweitert werden.

Lehrende können über eine web-basierte Oberfläche Aufgaben definieren, die aus einer

beliebigen Anzahl Quellcode- oder Bytecode-Dateien bestehen. Vier Einstellungen sind dabei möglich: (1) Der Quellcode wird den Lernenden mit der Aufgabenstellung zur Verfügung gestellt und muss nach der Bearbeitung abgegeben werden; (2) Der Quellcode wird den Lernenden zur Verfügung gestellt, soll aber nicht verändert und daher auch nicht mit abgegeben werden; (3) Der Dateiname dient nur als Platzhalter für eine Datei, die vom Lernenden vollständig anzulegen und mit abzugeben ist; (4) Der Bytecode wird für die Durchführung dynamischer Tests im Prüfungssystem benötigt und ist für den Lernenden nicht sichtbar. Ebenfalls für den Lernenden grundsätzlich nicht sichtbar sind die nötigen Dateien für die Durchführung statischer Tests.

Für den Übungsbetrieb können Aufgaben nach der Erstellung direkt freigegeben werden, so dass die Lernenden sie über eine eigene web-basierte Oberfläche auswählen und die nötigen Dateien herunterladen können. Das Hochladen der fertigen Lösungen und das Einsehen der Bewertungen durch das System erfolgen ebenfalls über diese Oberfläche. Für den Prüfungsbetrieb importiert der Lehrende eine Liste der Teilnehmer und ordnet ihnen individuell Aufgaben und ein eindeutiges Einmalpasswort (TAN) zu. Der eigentliche Prüfungsbetrieb kann dann ebenfalls über die Web-Oberfläche für Lernende erfolgen, wobei die Eingabe der TAN notwendig ist und damit die Auswahl einer Aufgabe sowie die sofortige Einsicht in der Bewertung durch JACK entfällt. Alternativ kann ein Plug-In für die Entwicklungsumgebung ECLIPSE nach Eingabe der TAN direkt auf den Server zugreifen und Dateien von dort herunterladen sowie zur Abgabe hochladen. Die Einsicht in die bewerteten Prüfungen erfolgt nach Deaktivieren des Prüfungsbetriebs über die web-basierte Oberfläche wie im freien Übungsbetrieb.

Abgegebene Lösungen können grundsätzlich vom Lehrenden eingesehen und mit manuellen Nachbewertungen und Korrekturen versehen werden. Die automatische Bewertung der abgegebenen Lösungen erfolgt durch eine separate Systemkomponente, die auf einem zweiten Server betrieben werden kann. Dadurch kann die Störung des Systems durch die Ausführung von böswilligem Code in den Lösungen verhindert werden. Die Prüfkomponente ruft unbearbeitete Lösungen aus der Datenbank des Systems ab, führt die vom Lehrenden eingestellten Tests durch und schreibt deren Ergebnisse zurück in die Datenbank.

Dynamische Tests führen Methodenaufrufe auf der Lösung aus und vergleichen die Rückgabewerte mit einer Vorgabe. Die Programmierung dieser Tests erfolgt durch normalen Programmcode in einer eigenen Klasse in Java, so dass beliebig komplexe Anfragen gestellt werden können, um präzise Meldungen über Fehler in der Lösung zu erzeugen. Exceptions, die während des Testlaufs auftreten, werden automatisch abgefangen und vor der Ausgabe um allgemeine Hinweise ergänzt, welche typischen Situationen zu dieser Exception geführt haben könnten. Programmläufe, die in eine Endlosschleife gelangen oder aus anderen Gründen nicht rechtzeitig terminieren, werden nach 30 Sekunden automatisch abgebrochen und mit einer entsprechenden Meldung als fehlerhaft bewertet.

Statische Tests werden als Mustersuche und Transformation des Syntaxgraphen durchgeführt und dementsprechend als Graphtransformationsregeln formuliert [KG06, KG08]. Je nach Ziel des einzelnen Tests greifen diese Regeln bei der Existenz oder der Abwesenheit bestimmter Muster und fügen Fehlermarkierungen in den Syntaxgraphen ein. Auch hier ist eine präzise Steuerung möglich, die Abhängigkeiten zwischen Fehlern berücksichtigt.

Je nach Komplexität der Lösung und der Prüfregeln sowie der Leistungsfähigkeit des verwendeten Servers können die statischen Tests zwischen 20 Sekunden und zwei Minuten in Anspruch nehmen.

Grundsätzlich sind beide Testverfahren optional und unabhängig von einander. Jeder Lösung wird somit maximal je ein Ergebnis des statischen Tests und des dynamischen Tests sowie optional ein manuelles Ergebnis durch den Lehrenden zugeordnet. Eine Lösung gilt als korrekt, wenn alle durchgeführten automatischen Tests sie als korrekt beurteilen und der Lehrende diese Bewertung nicht manuell überschreibt oder wenn der Lehrende eine negative automatische Bewertung überschreibt und die Lösung als korrekt markiert. In allen anderen Fällen gelten sie als fehlerhaft. Im Übungsbetrieb fand praktisch keine manuelle Nachkontrolle der automatischen Bewertungen durch JACK statt. Überprüfungen durch den Lehrenden zielten lediglich auf die Behebung technischer Fehler, durch die das System eine Lösung gar nicht beurteilen konnte. Da der Übungsbetrieb über die web-basierte Schnittstelle abgewickelt wurde und Lernende die Aufgaben mit beliebigen Programmierwerkzeugen bearbeiten konnten, traten Schwierigkeiten mit fehlerhaft hochgeladenen Dateien oder falschen Dateiformaten vereinzelt auf und erforderten eine entsprechende manuelle Behandlung. Durch die Verwendung des Plug-Ins für ECLIPSE traten vergleichbare Probleme im Prüfungsbetrieb nicht auf.

2.2 Verwandte Prüfungssysteme

Zahlreiche weitere Prüfungssysteme setzen ähnliche Techniken und Konzepte ein, von denen im Folgenden die markantesten Variationen ohne Anspruch auf Vollständigkeit vorgestellt werden.

Der an der Universität Passau entwickelte “Praktomat” [KSZ02] bietet automatische Tests für Java, C++ und Haskell an. Die statischen Tests beschränken sich dabei auf den Einsatz des externen Werkzeugs CHECKSTYLE [Che], das die Einhaltung von Formatrichtlinien überwacht. Ergänzend zur automatischen Korrektur bietet das System die Möglichkeit an, dass Lernende die Lösungen anderer Teilnehmer einsehen und kommentieren können. Die Lösung ist damit insbesondere für den kooperativen Übungsbetrieb konzipiert und weniger für den Einsatz in Prüfungssituationen. Dieselben Techniken wie der “Praktomat” nutzt auch das System ASB [MOSS07] der Fachhochschule Trier, wobei dort besonderer Wert auf die Wiederverwendbarkeit von Einstellungen für Bewertungen gelegt wird.

Das Projekt DUESIE [HQW08] der Universität Siegen prüft Programme in Java und SML sowie UML Diagramme. Das System bietet eine rein web-basierte Lösung auf der Basis von PHP5 an und integriert sämtliche Prüftechniken über externe Werkzeuge. Dabei ist über das Werkzeug PMD [PMD] auch eine Einbindung statischer Tests möglich. Diese sind etwas weniger mächtig als die in JACK verwendeten Graphtransformationen, aber prinzipiell ähnlich geeignet, Hinweise auf Fehler zu generieren. Bei der Bewertung von Diagrammen wird der Grad der Übereinstimmung mit einer Musterlösung berechnet, wobei mit dieser Technik keine Hinweise auf konkrete fehlende oder überflüssige Elemente in den Diagrammen gegeben werden können.

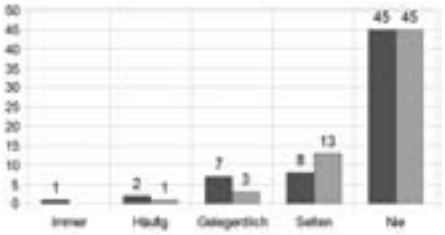
3 Beobachtungen und empirische Untersuchungen

Eines der häufigsten Argumente für den Einsatz automatischer Bewertungssysteme ist in der Regel die erwartete Einsparung beim Personaleinsatz, die die Betreuung zahlreicher Übungen und Prüfungen mit geringem Zeitaufwand ermöglichen soll. Diese Erwartung betrifft jedoch vor allem die Sicht der Lehrenden. Es ist daher ebenfalls zu untersuchen, welche zusätzlichen Effekte auf die Lernenden der Einsatz automatischer Bewertungssysteme hat. Insbesondere ist von Interesse, welche Qualität einer Bewertung durch automatische Systeme erzielt werden kann. Sollten positive Effekte durch eine deutlich sinkende Qualität der Bewertungen erkauft werden, erscheint der Einsatz automatischer Bewertungssysteme wenig sinnvoll, da damit das Ziel formativer wie summativer Prüfungen verfehlt würde.

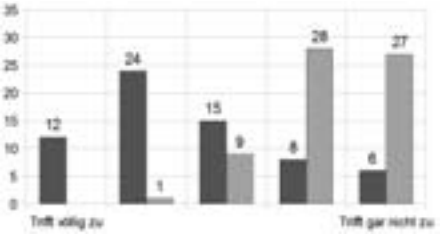
Neben der statistischen Auswertung des Einsatzes von JACK über zwei Jahre wurde zur Feststellung der Sicht der Lernenden im Wintersemester 2008/09 an der Universität Duisburg-Essen unter den Teilnehmern der Lehrveranstaltung "Programmierung" am Campus Essen eine Umfrage durchgeführt. Die Veranstaltung richtet sich hauptsächlich an Erstsemester der Informatikstudiengänge. Zum Zeitpunkt der Umfrage hatten die Teilnehmer im Verlauf des Semesters sechs Testate unter Prüfungsbedingungen unter Verwendung von JACK abgelegt sowie zusätzlich die Möglichkeit gehabt, als Vorbereitung auf jedes Testat freiwillig ein Übungsprojekt durch das System prüfen zu lassen. Zudem hatte zu Beginn des Semesters ein Probetestat unter Prüfungsbedingungen stattgefunden, das allerdings nur der Eingewöhnung diente und keine sonstigen Konsequenzen hatte. Von insgesamt 325 Teilnehmern, die die Veranstaltung zu Beginn des Semesters besuchten, nahmen leider nur 65 an der Umfrage teil, was der üblichen abnehmenden Frequentierung von Lehrveranstaltungen zum Ende der Vorlesungszeit geschuldet ist. Es wurde allerdings bewusst auf eine Befragung der Teilnehmer zu einem früheren Zeitpunkt verzichtet, damit Erfahrungen aus vielen verschiedenen Aufgabenstellungen im Verlaufe des Semesters in das Ergebnis der Umfrage mit einfliessen konnten und nicht nur der erste Eindruck der Lernenden erfasst wurde.

3.1 Qualität automatischer Bewertungen

Wie in Abschnitt 2.1 beschrieben, können in JACK die Ergebnisse automatischer Bewertungen durch den Lehrenden manuell überschrieben werden. Die Qualität der automatischen Bewertungen lässt sich folglich daran messen, wie häufig die automatischen Tests zu einem einstimmigen Ergebnis gekommen sind und wie häufig ein Lehrender ein Ergebnis überschreiben musste. Im Wintersemester 2007/08 war dies bei insgesamt 2933 Lösungen von Testaten in 261 Fällen (9%) notwendig. In 241 Fällen lag dies an zu rigoros eingestellten statischen Tests, die korrekte Lösungen als falsch markierten. Durch die sorgfältigere Gestaltung statischer Tests konnte die Zahl manueller Eingriffe im folgenden Jahr bei insgesamt 2721 Lösungen von Testaten auf 53 Fälle (2%) gesenkt werden. Die statischen Tests ließen dabei zwar häufiger als im Vorjahr Lösungen zu, die fehlerhaft waren ("false positives"), dies konnte jedoch durch die dynamischen Tests vollständig ab-



(a) Einschätzung der Lernenden bezüglich der Aussagen "JACK hat irrtümlich Fehler angezeigt, die keine waren." (dunkel) und "JACK hat irrtümlich Lösungen akzeptiert, die nicht korrekt waren." (hell)



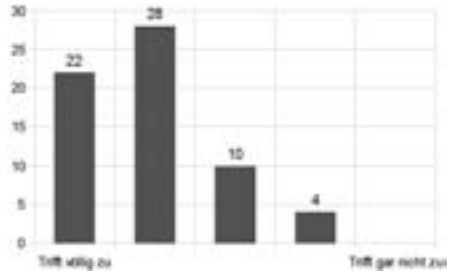
(b) Einschätzung der Lernenden bezüglich der Aussagen "JACK kann Fehler mindestens genauso gut finden wie ein Mensch." (dunkel) und "JACK kann Fehler mindestens genauso gut erklären wie ein Mensch." (hell)

Abbildung 1: Auswertung der Befragung von 65 Personen zur Einschätzung der allgemeinen Qualität der Prüfung von Lösungen durch JACK.

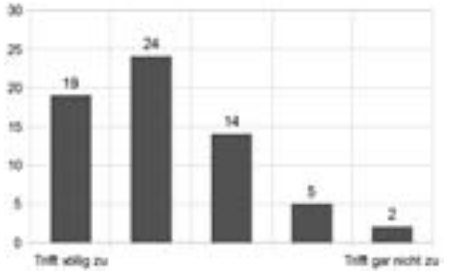
gefangen werden. Es kann daher festgestellt werden, dass durch die sorgfältige Einstellung dynamischer und statischer Tests eine Qualität der Bewertung erreicht werden kann, die kaum manuelle Überprüfung und Veränderung benötigt. Allerdings ist auch anzumerken, dass sich zu großzügige statische Tests mit vielen "false positives" negativ auf die Menge der gegebenen Fehlerhinweise auswirken. Das System ist damit zwar in der Lage, Fehler durch dynamische Tests zu erkennen, gibt jedoch seltener Hinweise auf die konkrete Fehlerstelle im Programmcode.

Genau diese Beobachtungen spiegeln sich auch in der Befragung der Lernenden wider. Im Wintersemester 2008/09 gaben lediglich 3 der 65 Befragten an, dass JACK bei ihren Testaten immer oder häufig Fehler angezeigt habe, die keine waren (siehe Abbildung 1(a)). Zudem gab keine Person an, dass JACK immer Lösungen akzeptierte, die nicht korrekt waren. Dementsprechend bezeichneten insgesamt 36 Befragte die Aussage "JACK kann Fehler mindestens genau gut finden wie ein Mensch" als weitgehend oder völlig zutreffend (siehe Abbildung 1(b)). Ein umgekehrtes Bild ergab sich wie erwartet bei der Frage, ob JACK die Fehler auch genauso gut erklären könne wie ein Mensch. Nur eine Person bezeichnete diese Aussage als weitgehend zutreffend, während die deutlich überwiegende Mehrheit sie für kaum oder gar nicht zutreffend hielt. Neben der erwarteten Tatsache, dass eine geringe Menge statischer Tests zu wenig aussagekräftigen, erklärenden Fehlerhinweisen führt, zeigt dieses Ergebnis auch, dass die Lernenden bei der Benutzung eines automatischen Bewertungssystems sehr genau zwischen dem reinen Auffinden eines Fehlers und einer ausführlichen Erklärung unterscheiden können. Für den Übungsbetrieb wird in Abschnitt 3.3 noch genauer auf diesen Aspekt eingegangen werden.

Als Zwischenfazit kann festgestellt werden, dass für kreative Prüfungsleistungen wie Programmieraufgaben eine automatische Bewertung möglich ist, die auf der rein inhaltlichen Ebene kaum menschliche Eingriffe erfordert. Da somit negative Effekte auf die Qualität weitgehend ausgeschlossen werden können, kann mit der Untersuchung weiterer positiver oder negativer Effekte fortgefahren werden.



(a) Einschätzung der Lernenden bezüglich der Aussage “JACK ist im Übungsmodus insgesamt betrachtet nützlich.”



(b) Einschätzung der Lernenden bezüglich der Aussage “JACK ist im Testatbetrieb insgesamt betrachtet nützlich”

Abbildung 2: Auswertung der Befragung von 65 Personen zur Einschätzung des Nutzens von JACK in verschiedenen Einsatzfeldern.

3.2 Allgemeine Effekte automatischer Bewertungen

Als allgemeine Effekte automatischer Bewertungssysteme wird im Folgenden zunächst auf die Auswirkungen eingegangen, die unabhängig von der Verwendung im Übungs- oder Prüfungsbetrieb entstehen. Bei Studierenden der Informatik soll die erste Annahme sein, dass keine grundsätzliche Skepsis gegenüber automatischen Bewertungssystemen besteht. Die Aussage “Systeme wie JACK sind mir unheimlich” fanden 7 Befragte völlig oder weitgehend zutreffend. 12 hielten sie für kaum und 35 für gar nicht zutreffend. 11 Personen äußerten sich ohne Tendenz. Eine grundsätzlich zurückhaltende Einstellung gegenüber automatischen Bewertungen scheint daher wie erwartet nicht gegeben zu sein. Als weitere naheliegende Hypothese kann erwartet werden, dass der Geschwindigkeitsvorteil gegenüber menschlichen Bewertungen auch von den Lernenden als Vorteil angesehen wird. Tatsächlich stimmten der Aussage “Die schnellen Prüfungen durch JACK sind ein großer Vorteil gegenüber menschlichen Prüfungen” 25 Befragte völlig zu. 24 Personen hielten die Aussage für weitgehend zutreffend. 11 Befragte hielten sie für kaum oder gar nicht zutreffend. 7 Befragte äußerten keine Tendenz. Der Geschwindigkeitsvorteil wird demnach auch von den Lernenden klar als Vorteil gesehen.

Mit der Feststellung eines Vorteils ist jedoch noch nicht geklärt, ob dieser oder weitere potentielle Vorteile auch als nützlich erachtet werden. Zum Nutzen von JACK insgesamt wurden den Teilnehmern der Umfrage daher vier Aussagen zur Auswahl gestellt, von denen sie die am ehesten zutreffende auswählen sollten. 56 Personen entschieden sich für die Aussage “JACK ist in der eingesetzten Form nützlich, sollte aber weiter verbessert werden”. 5 Personen hielten JACK bereits für so nützlich, dass keine Verbesserungen nötig seien. Die verbleibenden 4 Personen wählten die Aussage “JACK ist in der eingesetzten Form nicht nützlich, könnte es mit Verbesserungen aber werden”. Niemand entschied sich dafür, dass JACK nicht nützlich sei und daher abgeschafft werden sollte. Die Vorteile automatischer Bewertungen werden demnach deutlich als nützlich erachtet, und es besteht zudem ein großes Interesse daran, diese Systeme weiter einzusetzen und zu verbessern.

Zur weiteren differenzierten Analyse für den Übungsbetrieb und den Prüfungsbetrieb wur-

de die Frage nach dem Nutzen von JACK noch einmal getrennt für diese Einsatzbereiche gestellt (siehe Abbildung 2). Die Aussage "JACK ist im Übungsmodus insgesamt betrachtet nützlich" beurteilten dabei 50 Befragte als völlig oder weitgehend zutreffend. Ein leicht verschobenes Bild ergab sich für die Aussage "JACK ist im Testatbetrieb insgesamt betrachtet nützlich". Hier erteilten 43 Personen volle oder weitgehende Zustimmung. Die leicht unterschiedlichen Ergebnisse motivieren, die Auswirkungen in beiden Einsatzbereichen getrennt genauer zu untersuchen.

Im Vergleich zu identischen Befragungen zum oben bereits aufgeführten System ASB oder zu einem automatischen Bewertungssystem für mathematische Beweise (EASy) [GBK08] ergeben sich zwei Abweichungen. Zum einen ist das Ergebnis für JACK insgesamt etwas positiver, wobei der Unterschied im direkten Vergleich zu ASB etwas deutlicher ist. Zum anderen fällt die Abweichung zwischen Übungs- und Prüfungsbetrieb bei JACK genau umgekehrt aus wie bei EASy. Eine allgemeine Beurteilung, die sowohl unabhängig vom Einsatzbereich als auch von den verwendeten Techniken ist, scheint daher schwierig zu sein.

3.3 Effekte im Übungsbetrieb

Um den Übungsbetrieb trotz des Verzichts auf manuelle Nachkorrektur der automatischen Bewertungen nicht völlig ohne menschliche Betreuung stattfinden zu lassen, wurde eine regelmäßige Betreuung durch studentische Hilfskräfte angeboten. Diese hatten jedoch keinen manipulierenden Einfluss auf die Ergebnisse in JACK. Aus den Einschätzungen der Lernenden lässt sich daher ableiten, welche Effekte der Einsatz automatischer Bewertungen anstelle manueller Korrekturen im Übungsbetrieb hat. Ein wichtiges Ziel von JACK war es, aussagekräftige Meldungen zu generieren, die ebenso hilfreich sind wie manuelle Anmerkungen. Eine wichtige Tendenz dazu kann aus der Frage abgeleitet werden, ob Lernende sich die Meldungen von JACK durch andere Personen erklären lassen mussten. 39 Befragte gaben an, dass dies nie der Fall war und 18 Personen benötigten selten Erklärungen. Lediglich eine Person ließ sich die Meldungen immer erklären und eine weitere häufig. 6 Personen benötigten gelegentlich Erklärungen. Offenbar kann daher der Aufwand für persönliche Betreuung im Übungsbetrieb durch Systeme mit ausreichend qualitativen Meldungen tatsächlich gesenkt werden, ohne den Lernenden damit eine benötigte Hilfestellung zu entziehen.

Die Funktion des Übungsbetriebs soll an dieser Stelle jedoch nicht auf das Aufzeigen richtiger und falscher Lösungen beschränkt werden. Daher wurden die Leistungen von JACK noch detaillierter untersucht und in die Bereiche "Verständnis der Aufgaben", "Lösung der Aufgaben" und "Motivation" unterteilt. Da in Abschnitt 3.2 bereits festgestellt wurde, dass keine grundsätzliche Skepsis der Lernenden gegenüber JACK besteht und die schnelle Bewertung als Vorteil angesehen wird, kann zumindest für die Motivation ein positives Ergebnis erwartet werden. Da aus der abgegebenen Lösung nicht ohne weiteres ersichtlich ist, ob die Aufgabe grundsätzlich missverstanden wurde und entsprechende Verständnishinweise nötig sind, können dagegen kaum Auswirkungen auf das Verständnis der Aufgaben erwartet werden.

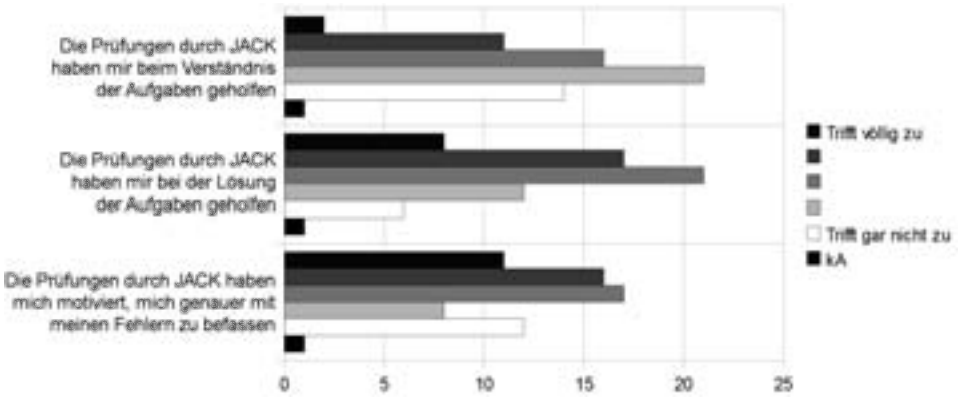


Abbildung 3: Auswertung der Befragung von 65 Personen zum Nutzen von JACK in verschiedenen Aspekten des Übungsbetriebs.

Abbildung 3 fasst die Ergebnisse zusammen. Die Aussage “Die Prüfungen durch JACK haben mir beim Verständnis der Aufgaben geholfen” bezeichnete eine deutliche Mehrheit als kaum zutreffend oder gar nicht zutreffend. Ein positiveres Ergebnis ergab sich für die Aussage “Die Prüfungen durch JACK haben mir bei der Lösung der Aufgabe geholfen”. Hier ist die Mehrheit unentschieden mit leichter Tendenz zur Zustimmung. Das Verständnis der Aufgabe kann demnach durch eine automatische Bewertung der Lösungen kaum gefördert werden. Dagegen sind die gegebenen Hinweise zumindest teilweise in der Lage, korrigierend auf den gewählten Lösungsweg einzuwirken. Diese Schlussfolgerung wurde durch mehrere stichprobenartige Vergleiche mehrerer Lösungen je eines Lernenden zu je einer Aufgabe überprüft. Dabei konnte tatsächlich festgestellt werden, dass Lernenden nach dem ersten misslungenen Lösungsversuch systematisch ihre Lösungen verbesserten, um einen Fehler nach dem anderen zu beheben.

In Einzelfällen konnte dabei auch beobachtet werden, dass Lernende zunächst eine offensichtlich unzureichende Lösung abgaben, um sich einen Überblick über die Fehlermeldungen des Bewertungssystems zu verschaffen. Vermutlich bestand in diesen Fällen gar kein Interesse daran, die Aufgabenstellung zu verstehen, sondern lediglich eine Lösung zu erzeugen, die genau den Anforderungen des Bewertungssystems entspricht. Da je nach Auslastung des Servers mehrere Minuten auf das Ergebnis der Bewertung gewartet werden musste, verlor dieses wettkampfartige Programmieren gegen JACK aber für die Lernenden in der Regel schnell seinen Reiz. Allerdings ist dieses Vorgehen auch ein Zeichen der Motivation, das automatische Bewertungssystem aktiv zu nutzen und mit seinen Vorzügen einer schnellen Bewertung aktiv in den Lernprozess einzubinden. Die Aussage “Die Prüfungen durch JACK haben mich motiviert, mich genauer mit meinen Fehlern zu befassen” beurteilten dementsprechend auch 27 Personen als völlig oder weitgehend zutreffend. Ob die gesteigerte Motivation ausschließlich und unmittelbar auf die schnelle Bewertung der Lösungen zurückzuführen ist, kann mit dieser Frage nicht geklärt werden. Ein grundsätzlich positiver Effekt auf die Motivation der Lernenden kann aber dennoch als vorhanden angesehen werden.

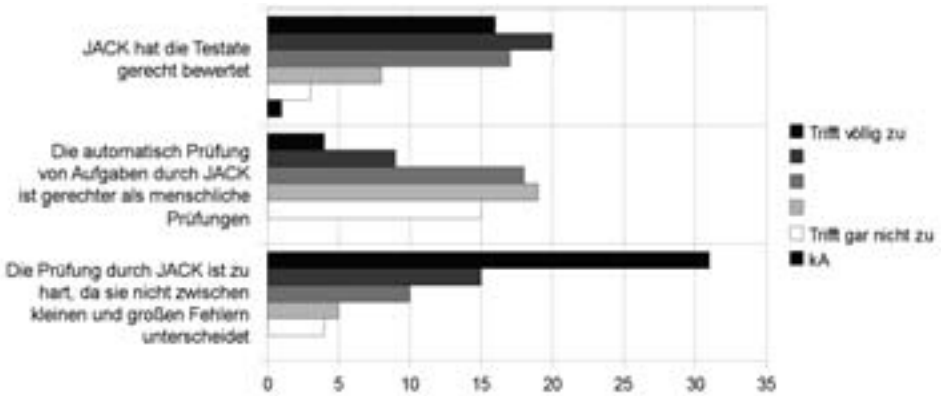


Abbildung 4: Auswertung der Befragung von 65 Personen zur Gerechtigkeit von JACK im Prüfungsbetrieb.

Als Ergebnis für den Übungsbetrieb kann festgestellt werden, dass mit dem Einsatz von JACK positive Effekte erzielt und die angepeilten Ziele erreicht werden konnten. Der Einsatz von JACK wirkt motivierend und ist in der Lage, ohne zusätzliche menschliche Eingriffe fehlerhafte Lösungen so weit zu beurteilen und zu erklären, dass die Lernenden selbständig einen Lernfortschritt erzielen können.

3.4 Effekte in Prüfungssituationen

Ein wichtiges Kriterium aus Sicht der Lernenden ist in Prüfungssituationen die Gerechtigkeit einer Prüfung. Die automatische Bewertung für Programmieraufgaben darf beispielsweise nicht den Eindruck erwecken, individuelle kreative Leistungen würden nur ungenügend berücksichtigen und stattdessen ausschließlich nach der Nähe zu einer vorgegeben Musterlösung bewerten. Da JACK andere Techniken verwendet, kann erwartet werden, dass die Prüfung als gerecht erachtet wird. Da JACK zudem jede Lösung genau denselben Tests unterwirft und nicht mit sich diskutieren lässt oder Ermüdungserscheinungen zeigt, die zum Übersehen von Fehlern führen, kann sogar erwartet werden, dass automatische Bewertungen als gerechter als menschliche Bewertungen bezeichnet werden.

Abbildung 4 fasst die Ergebnisse zusammen. Die Mehrheit der Befragte bezeichneten die Aussage “JACK hat die Testate gerecht bewertet” als völlig oder weitgehend zutreffend. Überraschenderweise ergibt sich bei der Aussage, dass JACK gerechter arbeite als ein menschlicher Korrektor, ein umgekehrtes Bild. Diese Aussage wird von der Mehrheit für kaum oder gar nicht zutreffend gehalten. Zwischen einer manuellen und einer automatischen Bewertung scheint es daher im Empfinden der Lernenden nicht zwingend einen Unterschied zu geben. Einen direkten Vergleich, wie ein menschlicher Korrektor die Testate bewertet hätte, hatten die Teilnehmer allerdings nicht. Ein besonderes Vertrauen oder ein gesteigertes Misstrauen in die Gerechtigkeit automatische Bewertungen kann aus den Ergebnissen ebenfalls nicht abgeleitet werden.

Das im Wintersemester 2008/09 verwendete Prüfungsverfahren sah vor, dass mindestens 3 von 7 Testaten bestanden werden mussten, um zur Klausur zugelassen zu werden. Jedes Testat konnte entweder mit einer fehlerfreien Lösung bestanden oder mit einer fehlerhaften Lösung nicht bestanden werden. Eine Punktevergabe, bei der auch eine gewisse Anzahl weitgehend fehlerfreier Lösungen zur Klausurzulassung ausreichend gewesen wäre, wurde nicht vorgenommen. Obwohl dieses Vorgehen völlig unabhängig davon ist, ob die Lösungen der Testate automatisch oder manuell bewertet werden, betrachtete die überwiegende Mehrheit der Befragten die Aussage "Die Prüfung durch JACK ist zu hart, da sie nicht zwischen kleinen und großen Fehlern unterscheidet" als völlig oder weitgehend zutreffend. Umgangssprachlich lässt sich die Meinung der Lernenden über JACK zusammen mit dem vorherigen Ergebnis zu einem "hart aber fair" zusammenfassen. Das Ergebnis legt aber auch dar, dass von den Lernenden nicht unterschieden wird, welche Effekte durch die automatische Bewertung der Lösungen induziert werden und welche sich aus den vom Lehrenden gewählten Prüfungsmodalitäten ergeben. Eine derartige Unterscheidung kann allerdings von Lernenden auch nicht erwartet werden, da sie eine genaue Kenntnis des gesamten Prüfungsvorgangs voraussetzt. Es erscheint daher sinnvoll, dass ein automatisches Bewertungssystem so flexibel wie möglich auf unterschiedliche Prüfungsmodalitäten einstellbar sein sollte und dass diese unabhängig von der Nutzung eines technischen Systems erklärbar sein sollten. Ferner ist nicht auszuschließen, dass eine negative Einstellung gegenüber einem automatischen Prüfungssystem auch zu einer negativen Einstellung zu den gewählten Prüfungsmodalitäten führen könnte.

Wenn das Bewertungsschema mehrheitlich trotz einer insgesamt als ausreichend gerecht empfundenen Prüfung als zu hart beurteilt wird, stellt sich zwangsläufig die Frage, ob seine Anwendung in einer Prüfung zusätzlichen Stress erzeugt. Die Aussage "Die Benutzung von JACK im Testat erzeugt zusätzlichen Stress" wurde sehr unterschiedlich beurteilt. 17 Personen hielten sie für völlig oder weitgehend zutreffend, 12 für kaum zutreffend und 17 für gar nicht zutreffend. 17 Personen äußerten sich ohne Tendenz und 2 machten gar keine Angabe. Eine eindeutige Konsequenz ist hieraus nicht abzuleiten. Allerdings äußerten 23 Personen volle Zustimmung für die Aussage, dass ihnen das Probetestat zu Beginn des Semesters geholfen habe, sich ohne Stress auf den Umgang mit JACK einzustellen. Weitere 16 Personen stimmten der Aussage weitgehend zu und nur 11 hielten sie für kaum oder gar nicht zutreffend. 15 Personen äußerten sich ohne Tendenz. Durch eine sorgfältige Vorbereitung kann daher offenbar zusätzlicher Stress durch die Verwendung automatischer Bewertungssysteme vermieden werden.

Für den Prüfungsbetrieb kann zusammenfassend festgestellt werden, dass keine gewichtigen negativen Effekte auftreten. Vielmehr werden automatische Bewertungen in Prüfungssituationen als weitgehend gleichwertig zu manuellen Bewertungen wahrgenommen. Die Vorteile einer schnellen (Vor-)Korrektur können damit voll zur Geltung kommen.

4 Zusammenfassung

In diesem Beitrag wurde ein konkretes System zur automatischen Bewertung von Programmieraufgaben vorgestellt und in seinen Auswirkungen umfassend analysiert. Es

konnte festgestellt werden, dass es mit demselben System möglich ist, sowohl Übungen ohne nennenswerten manuellen Korrekturaufwand durchzuführen als auch Prüfungen zu verarbeiten, die mit manuell bewerteten Prüfungen vergleichbar sind. Der Einsatz automatischer Bewertungssysteme kann daher für kreative Prüfungsleistungen im Bereich der Programmierung als sinnvoll und gewinnbringend beurteilt werden.

Die gewonnenen Ergebnisse motivieren, die vorgestellten technischen Grundlagen von JACK systematisch zu erweitern, um auch in anderen Einsatzfeldern im Fach Informatik oder in anderen Fächern vergleichbare Ergebnisse zu erzielen.

Literatur

[Che] CheckStyle Project. `http://checkstyle.sourceforge.net`.

[GBK08] Susanne Gruttmann, Dominik Böhm und Herbert Kuchen. E-assessment of Mathematical Proofs: Chances and Challenges for Students and Tutors. In *CSSE (5)*, Seiten 612–615. IEEE Computer Society, 2008.

[HQW08] Andreas Hoffmann, Alexander Quast und Roland Wismüller. Online-Übungssystem für die Programmierausbildung zur Einführung in die Informatik. In Silke Seehusen, Ulrike Lucke und Stefan Fischer, Hrsg., *DeLFI 2008, 6. e-Learning Fachtagung Informatik*, Jgg. 132 of *LNI*, Seiten 173–184. GI, 2008.

[KG06] Carsten Köllmann und Michael Goedicke. Automation of Java Code Analysis for Programming Exercises. In *Proceedings of the Third International Workshop on Graph Based Tools*, Jgg. 1 of *Electronic Communications of the EASST*, 2006.

[KG08] Carsten Köllmann und Michael Goedicke. A Specification Language for Static Analysis of Student Exercises. In *Proceedings of the International Conference on Automated Software Engineering*, 2008.

[KSZ02] Jens Krinke, Maximilian Störzer und Andreas Zeller. Web-basierte Programmierpraktika mit Praktomat. In *Workshop Neue Medien in der Informatik-Lehre*, Seiten 48–56, Dortmund, Germany, 2002.

[MOSS07] Thiemo Morth, Rainer Oechsle, Hermann Schloß und Markus Schwinn. Automatische Bewertung studentischer Software. In *Workshop "Rechnerunterstütztes Selbststudium in der Informatik", Universität Siegen, 17. September 2007*, 2007.

[PMD] PMD Project. `http://pmd.sourceforge.net/`.

[Rom07] Ralf Romeike. Three Drivers for Creativity in Computer Science Education. In *Proc of Informatics, Mathematics and ICT: a 'golden triangle'. Boston, USA*, 2007.

[Ros04] Jürgen Rost. *Lehrbuch Testtheorie - Testkonstruktion*. Huber, 2., vollst. überarb. und erw. Auflage, 2004.

[SGB08] Michael Striewe, Michael Goedicke und Moritz Balz. Computer Aided Assessments and Programming Exercises with JACK. Bericht 28, ICB, University of Duisburg-Essen, 2008.