

Vergleich von Papierklausuren und elektronischen Prüfungen

Diskussion der Vor- und Nachteile von elektronischen Prüfungen am Beispiel von Programmierklausuren

Bastian Küppers,¹ Ulrik Schroeder²

Abstract: Papierklausuren und elektronische Prüfungen unterscheiden sich in wesentlichen Aspekten, welche sowohl Formalitäten als auch Modalitäten der Prüfungen betreffen. Dieser Artikel stellt den Lebenszyklus einer Papierklausur dar und arbeitet heraus, dass auch bei einer Papierklausur bereits digitale Bestandteile im Lebenszyklus vorhanden sind. Darauf aufbauend werden Vor- und Nachteile von elektronischen Prüfungen diskutiert. Besonderes Augenmerk wird dabei auf die Sicherheit von elektronischen Prüfungen, sowohl im Allgemeinen als auch in Abgrenzung zur Papierklausur, gelegt. Als roter Faden werden im Verlaufe des Papers immer wieder Bezüge zu Klausuren in Programmierveranstaltungen hergestellt.

Keywords: Papierklausuren; E-Assessment; elektronische Prüfungen; Programmierklausuren

1 Einleitung

Hochschullehre ist heutzutage zunehmend digitalisiert [Ho16]. Dieser Trend erstreckt sich von der Verwendung von Notebooks und Tablets in Übungen bis hin zum Einsatz von Learning Management Systemen oder Apps in den Vorlesungen [PSD16]. Auch neben dem eigentlichen Studium sind Prozesse zur Selbstverwaltung der Studierenden zunehmend digital [PD14]. Nicht in gleichem Maße Teil der Digitalisierung sind jedoch Prüfungen, obwohl E-Assessment bereits erfolgreich für Selbsttests und formative Prüfungen eingesetzt wird [Ho16]. Ganz speziell im Bereich der Informatik birgt die Einführung von e-Assessment allerdings das Potenzial zum Studium gehörende Prüfungen zu verbessern. Der Hauptgrund hierfür ist sicherlich, dass das Werkzeug zur Durchführung von e-Assessment, nämlich der Computer, auch der Gegenstand des Studiums selbst ist. Im Folgenden wird zunächst der Stand der Forschung kurz beleuchtet. Anschließend werden Programmierklausuren, stellvertretend für Prüfungen im Informatikstudium, auf ihr Verbesserungspotenzial durch die Verwendung von e-Assessment untersucht. Dabei werden sowohl technische, als auch nicht-technische Aspekte beleuchtet und analysiert.

¹ RWTH Aachen University, IT Center, Seffenter Weg 23, 52074 Aachen, Deutschland kueppers@itc.rwth-aachen.de

² RWTH Aachen University, Learning Technologies Research Group, Ahornstraße 55, 52074 Aachen, Deutschland schroeder@informatik.rwth-aachen.de

2 Stand der Forschung

Es gibt bereits eine Vielzahl an Werkzeugen, die eine elektronische Abfrage von Wissen bzw. Fachkompetenz ermöglichen. Dies sind im einfachsten Fall Learning Management Systeme (LMS), wie z.B. Moodle³ oder ILIAS⁴. Diese stellen eine Vielzahl von Fragetypen bereits. “[Diese] reichen von [...] Multiple-Choice-Fragen über Zuordnungsfragen und zufällige Kurzantworten bis hin zu drag & drop, Lückentexten und Freitext“ [Mi15].

Für die Überprüfung von Programmierfertigkeiten existieren ebenfalls einige, web-basierte Werkzeuge, wie z.B. JACK [SBG09], DUESIE [Hqw08] oder Praktomat [KSZ02]. Almutka diskutiert in [AM05] die Eigenschaften solcher Werkzeuge und Keuning et. al. vergleichen viele der existierenden Systeme in [KJH16]. Da es sich bei den existierenden Werkzeugen um web-basierte Anwendungen handelt, sind diese nicht per-se für die Verwendung in Klausuren geeignet. Eine entsprechende Absicherung der Computer muss zusätzlich vorhanden sein, um Betrugsversuche während einer Klausur zu unterbinden. Wie diese Absicherung aussehen muss, hängt von den konkreten Rahmenbedingungen ab, in denen eine Klausur durchgeführt wird (siehe dazu auch Abschnitt 6).

3 Bestandteile einer Prüfung

Zu einer Prüfung gehören, vom Beginn der Erstellung bis zur Einsicht der Korrektur durch Studierende viele Bestandteile bzw. Prozessschritte. Diese werden im Folgenden kurz erläutert.

Erstellung Zunächst müssen Aufgaben gemäß der zu prüfenden Kompetenzen entwickelt und klausurgerecht formuliert und dokumentiert werden. Dieser Schritt ist in den meisten Fällen bereits digitalisiert, indem Textverarbeitung, Grafikedatoren und sicher auch Werkzeuge der jeweiligen Domäne, z.B. Programmierumgebungen oder UML-Editoren verwendet werden. Ferner wird häufig auch eine Tabellenkalkulation zur Verteilung der Punkte auf Aufgaben zum Einsatz kommen und häufig wird ein Rückgriff auf Aufgaben der Vorjahre erfolgen, die variiert und angepasst werden. Die einzelnen Aufgaben müssen dann zusammengefügt, papiergerecht formatiert und zu einer druckfähigen Version einer Klausur inklusive Deckblatt und ggf. Anhängen zusammengestellt werden.

Für die **Durchführung** der Klausur wird sie (ggf. in Varianten) gedruckt, damit die Studierenden sie bearbeiten können. Das Bearbeitungsmedium bestimmt damit auch die Art der möglichen Aufgaben. Da Papier kein interaktives bzw. adaptives Medium ist, können keine multimedialen Darstellungen, z.B. Animationen genutzt werden. In derselben Weise sind die Studierenden bei der Bearbeitung der Klausur eingeschränkt, da ihnen nur ein

³ <http://moodle.org/>

⁴ <https://www.ilias.de>

Stift und Lineal zur Verfügung stehen. Gerade im Bereich der Programmierung kann mit diesen Hilfsmitteln ein "natürlicher" Ablauf der Lösung von Programmieraufgaben nicht in gewohnter Weise durchgeführt werden. Das bedeutet, dass die in der Übung trainierte Situation keinen direkten Bezug zur Bearbeitung der Klausur hat. Dennoch sind auch in der Programmierung unterschiedlichste Aufgabenformate denkbar, wie z.B. Multiple Choice, Lückentext oder Freitext als "herkömmliche" Aufgabenformate. Darüber hinaus sind Aufgaben mit mehr Fachbezug denkbar, wie das Entwickeln kleiner Programme oder die detaillierte Beschreibung eines gegebenen Algorithmus, die keine digitalen Hilfsmittel erfordern. Zudem sind Stift und Papier nach wie vor die flexibelsten und schnellsten Darstellungsmittel und bieten gegenüber einem Tablet oder Rechnertastatur klare haptische Vorteile.

Für die **Korrektur** wird klassisch der Rotstift verwendet, um Fehler zu markieren und Punkte zu notieren. Speziell bei Programmieraufgaben ist es dabei oftmals sehr mühselig, die Funktionsweise eines Sourcecodes nachzuvollziehen. Daher kommt es nicht selten vor, dass Teile der Lösungen der Klausuraufgaben wieder digitalisiert werden, um sie in einer Programmierumgebung auf dem Computer zu testen. Oftmals stellt zudem die Handschrift der Studierenden ein Problem dar, sodass eine digitale Einreichung von Lösungen auch dem Zweck der besseren Lesbarkeit dient. Die Punkte, welche die Studierenden in ihren Klausuren erreicht haben, werden zudem in aller Regel digital in einer Tabellenkalkulation festgehalten, um korrekt aufzuaddieren und Klausuraushänge zu erstellen. Die Ergebnisse der Korrektur können dann elektronisch an Prüfungsämter übermittelt werden und z.B. in der Lernplattform als elektronischer Aushang bekannt gegeben werden.

Bei der **Einsicht** der Korrektur haben die Studierenden die Möglichkeit ihre Klausuren inklusive der eingefügten Markierungen, Kommentare und Punkte einzusehen und eventuell strittige Aspekte bei der Bewertung zu diskutieren. Bei einigen Klausuren werden daher in der Regel noch Bewertungen angepasst. Diese werden sowohl auf dem Klausurdokument selbst, als auch in den assoziierten digitalen Dokumenten (Listen für Aushänge, Übermittlung der Resultate etc.) festgehalten.

Archivierung Nach der Einsicht muss die Klausur datenschutzgerecht archiviert werden, um der allgemeinen Dokumentationspflicht Genüge zu tun.

Zusammenfassend lässt sich feststellen, dass dem gesamten Lebenszyklus einer Klausur ein stetiger Wechsel zwischen *digital* und *analog* inhärent ist. Dieser stetige Medienbruch kann durch die Einführung von e-Assessment reduziert werden. Dies ist wünschenswert, da jeder Wechsel potenziell eine Fehlerquelle darstellt. Welche bzw. ob es noch weitere Vorteile gibt und wie diese aussehen, wird in den folgenden Absätzen beschrieben.

4 Vorteile durch e-Assessment

Die Verwendung von e-Assessment kann den Lebenszyklus einer Prüfung homogener gestalten und darüber hinaus weitere Vorteile nutzen, die in den folgenden Abschnitten erläutert werden.

4.1 Verwendung fachspezifischer Werkzeuge

In Veranstaltungen werden in Übungen und Praktika, welche ergänzend zur Vorlesung stattfinden, oftmals fachspezifische Werkzeuge vorgestellt. Im Fall einer Programmier-Veranstaltung sind dies Entwicklungsumgebungen (*engl. integrated development environment, IDE*), wie z.B. Eclipse (<http://www.eclipse.org>) oder Netbeans (<http://netbeans.org>). Diese Entwicklungsumgebungen verfügen in der Regel über einen Editor, der Quellcode zur besseren Lesbarkeit farblich hervorhebt, einen Compiler und einen Debugger. Für komplexere Projekte stehen außerdem oftmals Möglichkeiten zur Projektverwaltung und zum kollaborativen Arbeiten, z.B. mit git, zur Verfügung. Entwicklungsumgebungen werden nicht nur im universitären Umfeld zu Lehrzwecken genutzt, sondern ebenso in Firmen zur kommerziellen Softwareentwicklung genutzt. Den Studierenden wird so bereits im Studium die Verwendung eines wichtigen Werkzeugs vermittelt. In einer herkömmlichen Papierklausur stehen den Studierenden diese Werkzeuge, deren Verwendung sie durch Übungen und Praktika gewohnt sind, nicht zur Verfügung. Dadurch entsteht für die Studierenden ein Bruch zwischen der Lehr- bzw. Lernphase und der Klausur. Durch eine Umstellung des Prüfungsmodus auf e-Assessment kann dieser Bruch entfernt werden, sodass sich Studierende in der Klausur in derselben Situation wiederfinden, die vorher geübt worden ist. Zudem müssen die Studierenden so auch die Beherrschung einer Entwicklungsumgebung, welche wie bereits erwähnt Teil des späteren Arbeitsalltages sind wird, in der Klausur unter Beweis stellen.

Natürlich ist die Verwendung von fachspezifischen Werkzeugen nicht auf Programmierklausuren beschränkt. Auch Veranstaltungen mit anderen Inhalten, wie z.B. Datenbanken oder Internettechnologien, können diesen Ansatz verfolgen. Ebenso wenig ist die Verwendung fachspezifischer Werkzeuge auf die Informatik beschränkt. Architekten oder Maschinenbauer müssen beispielsweise mit CAD-Programmen umgehen können, was sie in einer elektronischen Prüfung unter Beweis stellen können.

4.2 Vereinfachte Korrektur

Dieselben Werkzeuge, die den Studierenden während einer elektronischen Prüfung zur Verfügung stehen, können ebenfalls für die Korrektur genutzt werden. Für das Beispiel Programmierklausur bedeutet dies, dass die Korrigierenden ebenfalls eine Entwicklungsumgebung verwenden können, ohne die Klausuren vorher selbst durch Abtippen aufwändig digitalisieren zu müssen. Die Verwendung einer Entwicklungsumgebung bei der Korrektur

senkt den Aufwand für die Korrigierenden erheblich, da einige Aufgaben bereits von der Entwicklungsumgebung übernommen werden, andere Aufgaben werden drastisch vereinfacht. Syntaxfehler werden beispielsweise vollautomatisch erkannt und ausgegeben bzw. werden in den Lösungen der Studierenden gar nicht mehr vorkommen. Für die semantischen Aspekte stehen den Korrigierenden weitere Werkzeuge, wie z.B. ein Debugger oder Module für Unit-Testing zur Verfügung. Damit lässt sich leicht nachvollziehen, ob die eingereichten Lösungen der Studierenden die Anforderungen der Aufgabenstellung erfüllen bzw. wo Fehler liegen, sofern dies nicht der Fall sein sollte [JM15].

Zusätzlich zu den hier beschriebenen Vorteilen, kommt eine deutlich erhöhte Lesbarkeit der Abgaben. Somit lässt sich der Korrekturaufwand auch über die Verwendung fachspezifischer Werkzeuge hinaus noch einmal reduzieren.

4.3 Höheres Prüfungsniveau

Die bereits beschriebenen Vorteile durch den Einsatz fachspezifischer Werkzeuge während der Korrektur stehen natürlich auch den Studierenden während der Klausur zur Verfügung. Syntaxfehler können also beispielsweise bereits während der Bearbeitung der Aufgabe vermieden werden. Betrachtet man Krathwohls revidierte und erweiterte Fassung von Blooms Taxonomie kognitiver Lernziele (Abb. 1), so lässt sich feststellen, dass somit die unteren Level der Taxonomie, die bei einer Papierklausur häufig einen viel zu großen Teil der Prüfungsleistung ausmachen, nun teilweise von den zur Verfügung stehenden Werkzeugen übernommen werden.

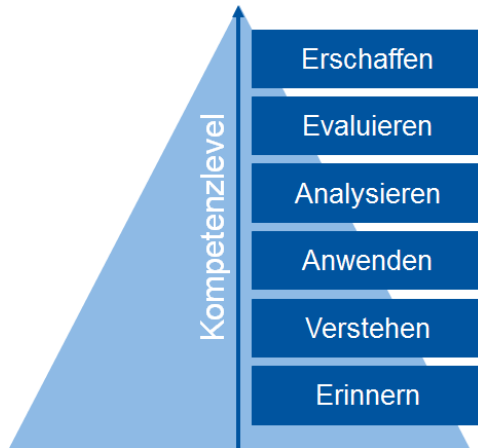


Abb. 1: Taxonomie kognitiver Lernziele nach Krathwohl [Kr02]

In Programmierklausuren können Kompetenzlevel wie z.B. “Erinnern“ in den Hintergrund treten, da Entwicklungsumgebungen Funktionen wie automatische Vervollständigung oder eine eingebaute API-Dokumentation bieten. Durch die Verwendung weiterer Tools, wie z.B. eines Debuggers, können zudem die oberen Kompetenzlevel mehr fokussiert werden.

Es wird beispielsweise einfacher komplexere Aufgabenstellungen umzusetzen, da ein Algorithmus nicht mehr vollständig auf Papier entwickelt werden muss, sondern einzelne Teile des Algorithmus bzw. eines komplexeren Programmsystems während der Entwicklung bereits getestet und nötigenfalls Fehler behoben werden können. Dies bildet zudem deutlich stärker den Arbeitsablauf im späteren Berufsalltag nach, sodass die Ergebnisse einer solchen Programmierklausur für einen potenziellen Arbeitgeber weitaus aussagekräftiger sind.

Theoretisch könnte es durch die Verwendung einer Entwicklungsumgebung in einer Programmierklausur zu *trial and error* bei den Studierenden kommen, also dem (systematischen) Ausprobieren unterschiedlicher Lösungswege. Da eine Entwicklungsumgebung den Studierenden direkt Feedback darüber gibt, ob ein Lösungsansatz das gewünschte Resultat liefert, kann so der richtige Lösungsweg in einer Menge von möglichen Lösungswegen identifiziert werden. Damit das beschriebene Vorgehen zum Erfolg führen kann, müssten die Studierenden zumindest in der Lage sein mehrere Lösungsansätze, darunter den richtigen, anzugeben. Ob dieses Vorgehen daher, vor allem in der knapp bemessenen Zeit einer Klausur, wirklich einen Vorteil liefert, bleibt an dieser Stelle fraglich.

Über Programmierklausuren hinaus sind die hier genannten Vorteile ebenfalls vorhanden. Soll beispielsweise im Rahmen einer Klausur im Fach Software Engineering ein UML-Diagramm eines komplexeren Programmsystems erstellt werden, kann der Fokus bei einer elektronischen Prüfung auf dem Verständnis des Programmsystems und den Zusammenhängen der einzelnen Module liegen, anstatt auf der Fähigkeit der Studierenden formal korrekte UML-Diagramme zu zeichnen. Fragen wie z.B. in welche Richtung ein Pfeil zeigt oder ob eine Pfeilspitze ausgefüllt sein muss stellen sich nicht, da diese Details, welche der untersten Stufe der Kompetenzlevel nach Krathwohl zuzuordnen sind, von einer Software zur Erstellung von UML-Diagrammen übernommen werden.

4.4 Innovative Prüfungsformate

Durch die Verschiebung des Fokus der Kompetenzlevel der Prüfungsaufgaben werden innovative Prüfungsformate möglich. Normalerweise sollen im Rahmen von Programmierklausuren anhand einer gegebenen Aufgabenstellung kleine Programme bzw. eher sogar nur einzelne Komponenten innerhalb eines vorgegebenen Lösungsansatzes entwickelt werden. Dazu muss in der Regel ein Konzept von den Studierenden erstellt werden, dessen zugehöriger Quellcode dann als Lösung der Klausuraufgabe aufgeschrieben wird. Aufgrund der Zeitbeschränkungen und mangelnder Hilfsmitteln ist die Aufgabe so reduziert, sodass das Konzept nicht explizit aufgeschrieben werden muss, sondern direkt als Quellcode aufgeschrieben werden kann. Diese Herangehensweise entspricht der klassischen Vorgehensweise bei der Softwareentwicklung. Modernere Methoden sind z.B. Test-driven Development [Be15]. Dabei werden zunächst Testfälle für den Algorithmus entworfen und anschließend der Algorithmus entworfen, sodass er nach und nach immer mehr dieser Testfälle ohne Fehler durchläuft. Im Rahmen einer elektronischen Klausur ist es möglich,

dass den Studierenden vorgefertigte Testfälle zur Verfügung gestellt werden, welche für die Entwicklung des Algorithmus genutzt werden können. Oder Studierende sollen als erweiterten Kompetenznachweis selbst Testfälle erstellen.

Auch jenseits von Programmierklausuren bieten elektronische Klausuren, eventuell sogar noch mehr als bei Programmierklausuren, das Potential für innovative Prüfungsformate. Hierbei birgt speziell der Einsatz von Medien, wie z.B. Audio- oder Videoclips, ein großes Potenzial. In der Medizin, beispielsweise, können unter Einsatz moderner Medien fallbasierte Prüfungen entworfen werden, bei denen die Studierenden z.B. Herztöne hören und beurteilen müssen [Ba09].

4.5 Vereinfachte Archivierung

Im Falle einer Papierklausur bedeutet die Archivierung einer Klausur einen nicht unerheblichen Aufwand, sowohl bezüglich Personal aber auch bezüglich Ressourcen. Zunächst muss genügend Platz vorhanden sein, um alle Klausuren einer Hochschule über den vorgesehenen Zeitraum zu lagern. Zudem sollte ein Archivierungssystem vorhanden sein, sodass eine archivierte Klausur wiedergefunden werden kann. Nicht zuletzt muss jemand die Klausuren anhand dieses Archivierungssystems archivieren und für den Fall, dass eine Klausur nach der Archivierung noch einmal eingesehen werden muss, beispielsweise im Rahmen eines Rechtsstreits, wieder aus dem Archiv holen. Bei elektronischen Klausuren ist der Vorgang des Archivierens automatisierbar, da die Klausuren bzw. die Lösungen der Studierenden bereits digital vorliegen. Das bedeutet, dass für die Archivierung der Klausuren die vorhandenen Datensätze nur noch anhand einer vorab definierten Struktur auf einem geeigneten Speichermedium gespeichert werden müssen. Für den Fall, dass eine Klausur noch einmal eingesehen werden muss, können die vorhandenen Datensätze einfach durchsucht und das entsprechende Dokument eingesehen werden. Damit entfallen große Teile des Aufwandes zur Archivierung einer Klausur.

5 Nachteile durch E-Assessment

Neben den bereits vorgestellten Vorteilen von e-Assessment gibt es auch Nachteile, welche in diesem Abschnitt kurz dargestellt werden.

5.1 Inflexibilität

Für bestimmte Aufgabenstellungen sind Computer nur bedingt geeignet. So sind beispielsweise die Erstellung von Zeichnungen oder die Annotation von Texten im Vergleich zur Papierklausur aufwändiger. Für die Anfertigung einer Zeichnung sollte beispielsweise ein spezielles Zeichentablet verwendet werden, da die Anfertigung mit einer normalen Computermaus nur schwer bzw. die Anfertigung mit dem Touchpad eines Notebooks beinahe unmöglich ist.

5.2 Abhängigkeit von Infrastruktur

Zur Durchführung von e-Assessment ist eine funktionierende Infrastruktur notwendig. Speziell Stromversorgung und meist auch Netzwerkzugriff müssen prinzipiell gewährleistet sein, damit e-Assessment reibungslos ablaufen kann. Bei der Verwendung von Notebooks kann ein kurzzeitiger Stromausfall zwar kompensiert werden, dies kann jedoch nicht als zuverlässige Backup-Strategie angesehen werden. Sofern die Klausur in einem Online-Prüfungssystem abgenommen wird, muss auch durchgängig Netzwerkzugriff gewährleistet sein, damit die Antworten im Prüfungssystem hinterlegt werden können.

5.3 Aufwand

In Abhängigkeit vom speziellen Prüfungsformat kann die Umstellung von Papierklausuren auf e-Assessment einen hohen Aufwand bedeuten. Wenn beispielsweise ein Antwort-Wahl-Verfahren eingesetzt werden soll, muss zunächst ein Fragepool erstellt werden. Dieser sollte im besten Fall qualitätsgesichert sein, um Aussagen über den Schwierigkeitsgrad einer Klausur treffen zu können. Speziell, wenn nur die Blaupause einer Klausur vorgegeben ist und die einzelnen Fragen für die einzelnen Studierenden zufällig aus dem Fragepool ausgewählt werden, muss der Fragepool qualitätsgesichert sein, um die Vergleichbarkeit der Klausuren zu gewährleisten.

6 Sicherheit

Das Thema Sicherheit wird immer wieder im Zusammenhang mit Klausuren diskutiert, nicht nur, wenn es um e-Assessment geht. Daher soll in diesem Abschnitt die Sicherheit von Papierklausuren und e-Assessment gegenüber gestellt werden.

Die Motivation hinter Betrug in einer Klausur ist in den meisten Fällen der eigene Vorteil. Dabei geht es nicht unbedingt darum, besser zu sein, als die anderen Klausurteilnehmer, sondern oftmals darum sich selbst einen Vorteil gegenüber der Situation zu verschaffen, die Klausur ohne Hilfe schreiben zu müssen. Gründe für die Notwendigkeit von Hilfe während der Klausur gibt es viele, beispielsweise zu wenig Zeit zum Lernen oder generelle Verständnisprobleme in Bezug auf den Lehrstoff. Wie eine Hilfe aussehen kann ist sehr unterschiedlich und reicht vom einfachen Spickzettel hin zur Verwendung des Smartphones während der Klausur.

Zunächst scheint e-Assessment anfälliger für die Verwendung unerlaubter Hilfsmittel zu sein, da durch die Verwendung eines Computers in einer Prüfung sicherlich ohne geeignete Gegenmaßnahmen deutlich mehr Möglichkeiten zum Betrug bestehen, als in einer Papierklausur. Bei Verwendung eines geeigneten Klausursystems sind viele dieser Möglichkeiten, wie z.B. die unerlaubte Verwendung des Internets, nicht ohne Weiteres möglich. Welche Software eingesetzt werden kann und welche Möglichkeiten diese bietet

hängt dabei auch von den Rahmenbedingungen ab. Wird ein zentral verwalteter PC-Pool zur Durchführung verwendet, kann das Klausursystem bei entsprechender Konfiguration im Rahmen eines zeitlich beschränkten Zugriffs durch die Studierenden vollständig abgesichert werden. Werden stattdessen die Geräte der Studierenden zur Durchführung verwendet muss für eine Betriebssicherheit deutlich mehr Aufwand betrieben werden, da die Studierenden vor der Klausur uneingeschränkten Zugriff auf die Geräte haben. Dadurch könnten Angriffsmöglichkeiten auf die Klausursoftware bestehen bleiben. Die Ausnutzung dieser Schwachstellen würde aber ein tiefgreifendes Verständnis für den Aufbau und die Funktionalität des Klausursystems voraussetzen. Daher ist fraglich, ob Studierende, die in der Lage sind diese Schwachstellen auszunutzen, nicht mit dem regelkonformen Ablegen der Prüfung besser beraten sind.

Durch die Verwendung von e-Assessment stehen aber nicht nur den Studierenden neue Wege des Betrugs zur Verfügung, sondern den Prüfenden stehen auch neue Möglichkeiten zur Feststellung des Betrugs offen, sodass sich das Bild der anfälligeren elektronischen Prüfung weiter relativieren lässt. Beispielsweise können während der Prüfung Logfiles angelegt werden, sodass sich im Verdachtsfall die Aktionen der Studierenden während der Prüfung nachvollziehen lassen. Natürlich können auch diese Mechanismen theoretisch wiederum Schwachstellen aufweisen, die ausgenutzt werden könnten, jedoch verschiebt sich das Kosten-Nutzen-Verhältnis weiter zu Ungunsten derjenigen, die durch Betrug versuchen, einen Vorteil zu erzielen.

Im Endeffekt wird sich keine Klausur - egal ob Papier oder elektronisch - zu 100% absichern lassen. Daher könnte sich eine Art Wettlauf zwischen Studierenden und Prüfenden ergeben, ob Schwachstellen schneller gefunden und ausgenutzt werden können, als sie geschlossen werden können. Die Alternative dazu ist, dass sich eine Akzeptanz für die verbliebenen Schwachstellen von elektronischen Prüfungen einstellt, wie es auch bereits bei Papierklausuren der Fall ist. Die Entwicklung dieser Akzeptanz könnte durch ein Umdenken von wissenbasierten Prüfungen hin zu kompetenzorientierten Prüfungen unterstützt werden, die beispielsweise als Open-Book Prüfung konzipiert werden können, sodass den Studierenden von vorneherein sämtlich Informationen zur Verfügung stehen und der Anreiz zum Betrug deutlich gesenkt wird. Für Programmierklausuren ist dies definitiv möglich, indem z.B. APIs für Programmiersprachen oder ähnliche Informationen bereitgestellt werden, dafür die Aufgaben aber komplexer gestaltet werden (vgl. auch 4.3).

7 Forschungsausblick

Trotz aller Vorteile ist e-Assessment nicht flächendeckend etabliert [Ho16], oft auch aus finanziellen Gründen. Die Anschaffung und Administration einer geeigneten IT-Infrastruktur ist teuer, sowohl in Bezug auf Sachkosten, als auch in Bezug auf Personalkosten [BEH09][Bü10]. Da der Großteil der Studierenden bereits selbst Endgeräte besitzt, die sich potenziell für e-Assessment eignen [Da15][Po15][Wi16], ist Bring Your Own Device (BYOD) eine mögliche Lösung für diese Kostenproblematik. Grundsätzlich gibt es verschiedene Ansätze, wie BYOD implementiert werden kann [KS16]. Die Unterschiede zwischen diesen Ansätzen

betreffen hauptsächlich die Software, welche während einer Klausur eingesetzt wird, und die Netzanbindung der Geräte der Studierenden. Speziell bei der Durchführung von Prüfungen auf Geräten der Studierenden ist Sicherheit ein relevantes Thema. Ein bekannter Weg, um Computer für einen definierten Zeitraum abzusichern, ist ein sogenanntes Lockdown-Programm. Sobald solch ein Programm gestartet wurde, sorgt es dafür, dass die Studierenden nur noch zugelassene Programme, wie z.B. eine Entwicklungsumgebung, starten und nur noch zugelassene Webseiten besuchen können. Es gibt bereits einsatzfähiger Software, welche die beschriebene Funktionalität bietet [Fr10][FSZ12]. Als Beispiele können, neben Anderen, der SafeExamBrowser der ETH Zürich oder der LockDown Browser der Firma Respondus genannt werden. Es gibt in dieser Art von Software jedoch einige Angriffspunkte [Sø16], die beim Einsatz in einem BYOD-Szenario zwingend ausgeräumt werden müssen. Daher soll im Rahmen eines Forschungsvorhabens evaluiert werden, ob die existierenden Angriffspunkte ausgeräumt werden können, oder ob neue Ansätze gefunden werden müssen, um e-Assessment in einem BYOD-Szenario hinreichend abzusichern. Eine weiterführende Diskussion zu diesem Thema wird von Küppers et. al. in [Kü17] gegeben. Neben dem Thema Sicherheit sollen die in diesem Artikel als Nachteile dargestellten Punkte beforcht und auch dafür innovative Lösungen entwickelt werden.

8 Zusammenfassung und Ausblick

In diesem Artikel wurden die Unterschiede zwischen Papierklausuren und elektronischen Klausuren mit dem Fokus auf Klausuren in der Programmierung beleuchtet. Dabei wurden Vorteile und Nachteile aufgezeigt und diskutiert. Speziell das Thema Sicherheit wurde dabei adressiert, da dies ein wiederkehrendes Thema ist, wenn es um e-Assessment geht. Über die reine Digitalisierung von Klausuren hinaus, bietet e-Assessment das Potenzial der Prüfung höherer Taxonomiestufen. Speziell der letztgenannte Punkt ist eine deutliche qualitative Verbesserung gegenüber traditionellen Papierklausuren. Daher ist geplant, die hier angesprochenen Vorteile von elektronischen Programmierklausuren im Rahmen einer prototypischen Implementierung, die innovative Lösungen für die mit den Nachteilen angedeuteten Herausforderungen bietet, umzusetzen und deren Mehrwert zu evaluieren. Dabei sollen insbesondere die sicherheitsrelevanten Fragen beforcht werden.

Literaturverzeichnis

- [AM05] Ala-Mutka, Kirsti: A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2):83–102, 2005.
- [Ba09] Baumann, Martin; Steinmetzer, Jan; Karami, Mazdak; Schäfer, Gereon: Innovative electronic exams with voice in- and output questions in medical terminology on a high taxonomic level. *Medical Teacher*, 31(10):e460–e463, 2009.
- [Be15] Beck, Kent: Test-driven development: By example. A Kent Beck signature book. Addison-Wesley, Boston, 20. printing. Auflage, 2015.

- [BEH09] Biella, Daniel; Engert, Steffi; Huth, Dieter: Design and Delivery of an E-assessment Solution at the University of Duisburg-Essen. In: Proceedings EUNIS 2009. EUNIS Proceedings, 2009.
- [Bü10] Bücking, Jens: eKlausuren im Testcenter der Universität Bremen: Ein Praxisbericht. <https://www.campussource.de/events/e1010tudortmund/docs/Buecking.pdf>, 2010 [Abgerufen am 21.06.2017].
- [Da15] E. Dahlstrom et. al.: Undergraduate Students and IT. <https://library.educause.edu/~media/files/library/2015/8/ers1510ss.pdf?la=en>, 2015 [Abgerufen am 21.06.2017].
- [Fr10] Frank, Ariel J.: Dependable distributed testing: Can the online proctor be reliably computerized? In (Marca, David A., Hrsg.): Proceedings of the International Conference on E-Business. SciTePress, S.l., 2010.
- [FSZ12] Frankl, Gabriele; Schartner, Peter; Zebedin, Gerald: Secure online exams using students' devices. In: 2012 IEEE Global Engineering Education Conference (EDUCON). S. 1–7, 2012.
- [Ho16] Hochschulforum Digitalisierung: The Digital Turn: Hochschulbildung im digitalen Zeitalter. <https://hochschulforumdigitalisierung.de/sites/default/files/dateien/Abschlussbericht.pdf>, 2016 [Abgerufen am 21.06.2017].
- [HQW08] Hoffmann, Andreas; Quast, Alexander; Wismüller, Roland: Online-Übungssystem für die Programmierausbildung zur Einführung in die Informatik. In: DeLFI. 2008.
- [JM15] Jara, Nicole; Molina Madrid, Manuel: Bewertungsschema für eine abgestufte Bewertung von Programmieraufgaben. In (Pongratz, Hans, Hrsg.): DeLFI 2015 - die 13. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V, GI-Edition Lecture Notes in Informatics Proceedings, S. 233–240. Ges. für Informatik, Bonn, 2015.
- [KJH16] Keuning, Hieke; Jeuring, Johan; Heeren, Bastiaan: Towards a Systematic Review of Automated Feedback Generation for Programming Exercises. In: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. ACM, S. 41–46, 2016.
- [Kr02] Krathwohl, David R.: A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice*, 41(4):212–218, 2002.
- [KS16] Küppers, Bastian; Schroeder, Ulrik: BRING YOUR OWN DEVICE FOR E-ASSESSMENT - A REVIEW. In (Gómez Chova, Luis; López Martínez, Agustín; Candel Torres, Ignacio, Hrsg.): International Conference on Education and New Learning Technologies. EDULEARN proceedings. IATED, S. 8770–8776, 2016.
- [KSZ02] Krinke, Jens; Störzer, Maximilian; Zeller, Andreas: Web-basierte Programmierpraktika mit Praktomat. *Softwaretechnik-Trends*, 22(3), Oktober 2002.
- [Kü17] Küppers, Bastian; Kerber, Florian; Meyer, Ulrike; Schroeder, Ulrik: Beyond Lockdown - Towards Reliable e-Assessment. DeLFI 2017 [noch nicht erschienen], 2017.
- [Mi15] L. P. Michel et. al.: Digitales Prüfen und Bewerten im Hochschulbereich: Arbeitspapier Nr. 1. https://hochschulforumdigitalisierung.de/sites/default/files/dateien/HFD%20AP%20Nr%201_Digitales%20Pruefen%20und%20Bewerten.pdf, 2015 [Abgerufen am 21.06.2017].

- [PD14] Politze, Marius; Decker, Bernd: RWTHApp: from a requirements analysis to a service oriented architecture for secure mobile access to personalized data. In: Proceedings EUNIS 2014. Jgg. 20 in EUNIS Proceedings, 2014.
- [Po15] Poll, Harris: Student Mobile Device Survey 2015: National Report: College Students. <http://www.pearsoned.com/wp-content/uploads/2015-Pearson-Student-Mobile-Device-Survey-College.pdf>, 2015 [Abgerufen am 21.06.2017].
- [PSD16] Politze, Marius; Schaffert, Steffen; Decker, Bernd: A secure infrastructure for mobile blended learning applications. In: Proceedings EUNIS 2016. Jgg. 22 in EUNIS Proceedings, S. 49–56, 2016.
- [SBG09] Striewe, Michael; Balz, Moritz; Goedicke, Michael: A Flexible and Modular Software Architecture for Computer Aided Assessments and Automated Marking. In: Proceedings of the 1st International Conference on Computer Supported Education 2009. Lisboa, Portugal, S. 54–61, 2009.
- [Sø16] Søggaard, Thea Marie: Mitigation of Cheating Threats in Digital BYOD exams. Masterarbeit, Trondheim, 2016.
- [Wi16] Willige, Janka: Auslandsmobilität und digitale Medien: Arbeitspapier Nr. 23. https://hochschulforumdigitalisierung.de/sites/default/files/dateien/HFD_AP_Nr23_Digitale_Medien_und_Mobilitaet.pdf, 2016 [Abgerufen am 21.06.2017].