

Please Vote Now! – A Field Report on the Audience Response System PINGO

Dennis Kundisch, Jürgen Neumann, Darius Schlangenotto

Abstract: Active learning allows lecturers to let students interactively participate in class. Audience response systems (ARS) are one way to realize active learning as they enable lecturers to simultaneously ask every student subject-related questions. In this article, we present our experiences with the web-based ARS PINGO, which has been developed and hosted at Paderborn University. Using PINGO as an example for ARS, we (1) present possible use cases and analyze actual user behavior, (2) describe the characteristics of the underlying software architecture, and (3) discuss challenges regarding the establishment of a long-term operation for a successful e-learning project.

Bitte stimmen Sie jetzt ab! – Ein Erfahrungsbericht über das Audience Response System PINGO

Dennis Kundisch¹, Jürgen Neumann², Darius Schlangenotto³

Abstract: Handlungsorientierte Formen der Didaktik erlauben es Dozenten die Lernenden zu aktiver und selbstständiger Mitarbeit anzuregen. Solch handlungsorientierte Formen der Didaktik lassen sich u. a. durch elektronische Audience Response Systeme (ARS) realisieren, die es Dozenten ermöglichen während einer Präsenzveranstaltung Studierende hinsichtlich einer themenbezogenen Problemstellung zu befragen. In diesem Beitrag werden die Erfahrungen mit dem an der Universität Paderborn entwickelten und betriebenen webbasierten ARS PINGO diskutiert. Am Beispiel von PINGO werden (1) Einsatzszenarien und die tatsächliche Nutzung durch Dozenten vorgestellt, (2) Besonderheiten der Softwarearchitektur eines solchen ARS hervorgehoben sowie (3) die Herausforderungen der Verstärkung eines erfolgreichen eLearning-Projekts fokussiert.

Keywords: Audience Response System, PINGO, Lehr-/Lerndesign, Erfahrungsbericht

1 Motivation

In vielen Fachdisziplinen werden die Inhalte auch heute noch vorwiegend über den Weg der klassischen Frontalvorlesung vermittelt. In stark nachgefragten Studienfächern, führt dies häufig zu sehr hohen Teilnehmerzahlen und den Studierenden bleibt oft nur die Rolle passiver Rezipienten, die sich den Vortrag des Dozenten⁴ anhören und sich die dargebotenen Inhalte auf unterschiedlichen Wegen, nach der Vorlesung, individuell erschließen. Diese Form des Lernens ist, angesichts mittlerweile gut fundierter Erkenntnisse der Lehr-/Lernforschung, nicht optimal [G196]. Handlungsorientierte Formen der Didaktik erlauben es Dozenten die Lernenden zu aktiver und selbstständiger Mitarbeit anzuregen [Ha98]. Solch handlungsorientierte Formen der Didaktik lassen sich u. a. durch elektronische *Audience Response Systeme* (ARS)⁵ realisieren, die es Dozenten ermöglichen während einer Präsenzveranstaltung Studierende hinsichtlich einer themenbezogenen Problemstellung zu befragen.

¹ Universität Paderborn, Warburger Straße 100, 33098 Paderborn, Dennis.Kundisch@wiwi.upb.de

² Universität Paderborn, Warburger Straße 100, 33098 Paderborn, Juergen.Neumann@wiwi.upb.de

³ Universität Paderborn, Warburger Straße 100, 33098 Paderborn, Darius.Schlangenotto@wiwi.upb.de

⁴ Es sind stets Personen männlichen und weiblichen Geschlechts gleichermaßen gemeint; aus Gründen der einfacheren Lesbarkeit wird im Folgenden nur die männliche Form verwendet.

⁵ Auch als Live Feedback System, Audience Response System oder Personal Response System bezeichnet [Ku13]. Ein Überblick über die Historie von ARS findet sich in [JS02].

Ein ARS, welches sich mit mehr als 12.000 registrierten Dozenten durch eine hohe Marktdurchdringung auszeichnet, ist das kostenlos angebotene, webbasierte ARS PINGO. Das System wird seit 2011 an der Universität Paderborn entwickelt, unter <https://pingo.upb.de> als gehosteter Service sowie unter <https://github.com/PingoUPB> als Open Source Version bereitgestellt und kontinuierlich hinsichtlich verschiedener didaktischer Einsatzszenarien evaluiert (z. B. [Zo16], [Be14], [Be12]). PINGO ermöglicht die Umsetzung unterschiedlicher Lehr-/Lerndesigns und zeichnet sich durch eine intuitive Bedienbarkeit aus. Technisch wurde das System für den Einsatz in Großveranstaltungen ausgelegt, um Lehrenden die Möglichkeit zu geben auch Veranstaltungen mit mehr als 1.000 Zuhörern interaktiver zu gestalten. In diesem Beitrag werden ausgewählte Erfahrungen der vergangenen sechs Jahre aus didaktischer, technischer und organisatorischer Sicht zusammengefasst. Damit wird das Ziel verfolgt, eLearning-Verantwortlichen und Webentwicklern nutzenstiftende Anregungen bei der Durchführung von ähnlichen Projekten zur Verfügung zu stellen.

2 ARS und Lehr-/Lerndesigns

Im (Hoch-)Schulkontext dienen ARS insbesondere dazu Vorlesungen und Übungen interaktiver zu gestalten, um Studierende aus ihrer häufig eher passiven Rolle herauszuholen und diese aktiv in die Veranstaltung einzubeziehen. ARS erlauben hierbei die Umsetzung unterschiedlichster Lehr-/Lerndesigns [Ku13] und können z. B. zur Klausurvorbereitung (insb. Multiple-Choice-Klausuren), Wissens-/Meinungsabfrage, Motivation neuer Inhalte und zum Sammeln von Veranstaltungsfeedback eingesetzt werden. Darüber hinaus lassen sich die speziell für ARS entwickelten und fundiert evaluierten Lehr-/Lerndesigns *Peer Instruction* [Ma97] und *Class-Wide Discussions* [Du96] realisieren.

Bei *Peer Instruction* stellt der Vortragende nach einem kurzen Impulsreferat eine Multiple-Choice-Frage. Bei weniger als 80 % korrekten Antworten, vertieft der Dozent entweder das Impulsreferat (< 30 % richtige Antworten) oder lässt die Teilnehmer mit ihren Nachbarn diskutieren (Peer Discussion bei > 30 % und < 80 % richtiger Antworten). Anschließend wird die Frage erneut zur Abstimmung gestellt und die neuen Umfrageergebnisse ein weiteres Mal ausgewertet. In der Literatur wird eine positive Korrelation zwischen dem Einsatz von Peer Instruction und positiven Rückmeldungen in studentischen Evaluationen, unabhängig von der Gruppengröße, postuliert [LGP17]. Darüber hinaus legen diverse Studien nahe, dass der Einsatz von Peer Instruction positive Auswirkungen auf die Leistungen in Abschlussklausuren hat (z. B. [SPS13]). Bei *Class-Wide Discussions* werden die Umfrageergebnisse zu einer gestellten Frage im gesamten Plenum unter Moderation durch den Vortragenden diskutiert [Du96]. Im Vergleich zu Peer Instruction wird hier also nicht nur mit den unmittelbaren Nachbarn diskutiert, sondern die gesamte Zuhörergruppe einbezogen.

3 ARS PINGO

3.1 Funktionalität

Mit PINGO lassen sich u. a. die in Kapitel 2 genannten Lehr-/Lerndesigns umsetzen. Neben Single- und Multiple-Choice-Fragen, die v. a. bei Peer Instruction und Class-Wide Discussions Verwendung finden, bietet PINGO weitere Flexibilität durch zusätzliche Fragetypen. So können auch Freitextfragen und Fragen mit einer numerischen Eingabe (bspw. Schätzfragen) an das Publikum gestellt werden. Vortragende können Fragen vorab im System vorbereiten (inkl. LaTeX-Formelunterstützung) oder ad-hoc an das Plenum senden. Das System bietet zudem die Möglichkeit vorbereitete Fragen in anderen Formaten (z. B. XML für Moodle oder QTI für Ilias) zu im- oder exportieren.

Abbildung 1 gibt einen Überblick über ein typisches Nutzungsszenario eines webbasierten ARS wie PINGO. Dozenten stellen eine Frage an das Plenum (Abb. 1, Schritt 1). Studierende antworten in einem vorgegebenen Zeitfenster auf die Frage (Schritt 2). Anschließend erfolgt die Auswertung der Ergebnisse und die Diskussion im Plenum (Schritt 3). Die bspw. an eine Peer Discussion anschließende Wiederholungsfrage wird ebenfalls durch PINGO unterstützt. Dabei werden die Ergebnisse der Wiederholung im Vergleich zu den ursprünglichen Ergebnissen angezeigt und es ist ebenfalls möglich zu betrachten, wie viele Teilnehmer von einer bestimmten Antwortmöglichkeit zu einer anderen gewechselt sind. Um eine sinnhafte Integration von PINGO in unterschiedlichste Veranstaltungen zu ermöglichen, stellen wir vielfältige didaktische Materialien (z. B. <http://go.upb.de/pingoDidaktik> oder <https://youtu.be/KK22QMb0MFA>) zur Verfügung.

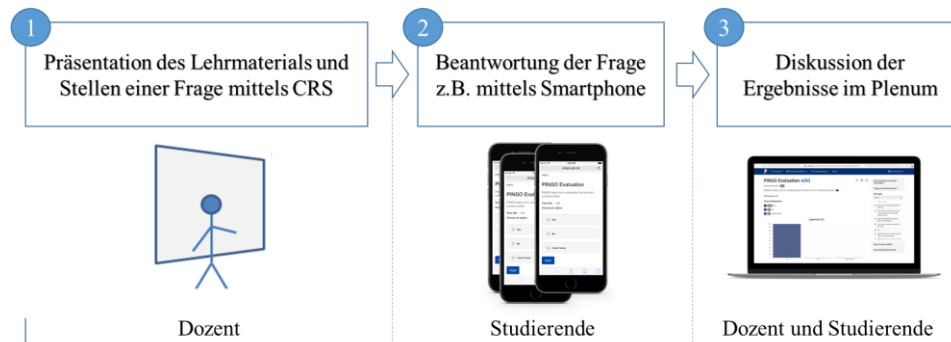


Abb. 1: Nutzungsszenario eines webbasierten ARS

Neben den klassischen Umfragefunktionalitäten bietet PINGO diverse Funktionalitäten zur Kollaboration zwischen den Dozenten. So können Dozenten ihre Fragen öffentlich anderen Nutzern zur Verfügung stellen und Fragen kommentieren. In PINGO werden zudem mehrere Fragen innerhalb einer Veranstaltung, oft über mehrere Termine, gestellt und die zugehörigen Ergebnisse in dieser Veranstaltung gespeichert. Im System ist es

deshalb möglich, eine solche Veranstaltung mit mehreren Dozenten zu teilen, sodass jeder Dozent Fragen stellen und Ergebnisse einsehen kann. Darüber hinaus erlaubt die Zusatzapplikation *PINGO Remote* (für Windows und Mac), PINGO ohne die Verwendung eines Webbrowsers und dem damit verbundenen Hin- und Herwechseln zwischen Präsentation und Browser zu steuern. So können Fragen durch ein kleines, über der Präsentation (bspw. PowerPoint oder PDF) liegendes Fenster gestartet werden. Derzeit wird zudem an einer vollständigen PowerPoint Integration gearbeitet.

Im Vergleich zu diversen anderen ARS zeichnet sich PINGO durch die Möglichkeit zur Vorbereitung und Wiederverwendung von Fragen, den Einsatz mittels beliebigen internetfähigen Endgeräten und eine hohe Nutzeraktivität aus [Ve14]. Das System bietet jedoch keine Möglichkeit in Echtzeit die Vortragsweise des Lehrenden zu bewerten [Ve14] oder mittels Learning Analytics den eigenen Lernfortschritt zu analysieren.

3.2 Softwarearchitektur & Betrieb

Webbasierte ARS wie PINGO benötigen eine angepasste Softwarearchitektur, die es erlaubt, zeitgleich stattfindende Abstimmungen mit hunderten von Teilnehmern an unterschiedlichsten Standorten weltweit durchzuführen. Abbildung 2 zeigt die Serveranfragen, welche im Rahmen eines Nutzungsszenarios entstehen. Derzeit finden in Spitzenlastzeiten hunderte Umfragen mit tausenden von Teilnehmern parallel statt. Diese Leistungsdaten erfordern die Bereitstellung eines skalierbaren Systems, welches eine Vielzahl paralleler Serveranfragen verarbeiten kann.

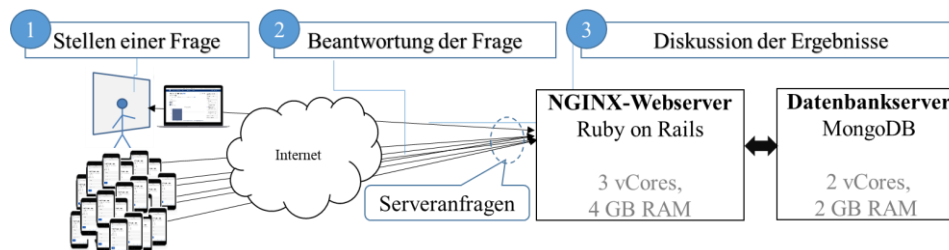


Abb. 2: Nutzungsszenario & Softwarearchitektur

Die Softwarearchitektur von PINGO besteht im Kern aus einer mit dem Web-Framework Ruby on Rails entwickelten Webapplikation, welche über einen NGINX Webserver zur Verfügung gestellt wird (für Details, siehe [WN13]). Als Datenbankserver nutzt PINGO die NoSQL-Datenbank MongoDB (für Details, siehe [Ba16]). Derzeit wird die Gesamtarchitektur über den deutschen Hosting-Anbieter *maxcluster* in einem Rechenzentrum in Frankfurt bereitgestellt. Die Webapplikation und der Datenbankserver werden hierbei auf separaten Servern betrieben, um so Leistungsempässe zu vermeiden und das Potential der NoSQL-Datenbanktechnologie vollumfänglich auszuschöpfen.

3.3 Nutzung

Seit der Entwicklung im Jahr 2011 wurden mit PINGO von den mehr als 12.000 registrierten Dozenten mehr als 170.000 Umfragen durchgeführt und mehr als drei Millionen Antworten wurden über das System abgegeben. Pro Monat stimmen derzeit mehr als 100.000 Personen ab und uns erreichen monatlich mehr als 50 technische oder didaktische Hinweise per Email oder Telefon. PINGO wird nach wie vor vorrangig an Hochschulen eingesetzt (89 % aller Nutzer). Die Zahl der Nutzer, die PINGO im Schul- (5 %) oder Unternehmensumfeld (6 %) einsetzen, steigt jedoch kontinuierlich.

4 Projekterfahrungen

4.1 Nutzerverhalten

Die weite Verbreitung von PINGO ist aus unserer Sicht auf zwei wesentliche Faktoren zurückzuführen. Zum einen verfügt PINGO über ein niederschwelliges Design und einen Funktionsumfang, der direkt auf die Umsetzung von einigen wenigen ARS-basierten Lehr-/Lerndesigns abgestimmt wurde. Wir haben bewusst darauf verzichtet, zahlreiche weitere Interaktionsmöglichkeiten (bspw. Zwischenfragen des Publikums, Live-Feedback zum Vortragstempo etc.) in PINGO zu integrieren. Zum anderen hat sich die Softwarearchitektur in der Praxis bewährt, sodass wir in der Lage sind, allen Nutzern ein hoch-performantes System zur Verfügung zu stellen, welches 24/7 zur Verfügung steht.

Im Schul- und (Hoch-)Schulkontext wird PINGO oftmals im Rahmen der Lehre eingesetzt, um Zuhörer aktiv in Veranstaltungen einzubeziehen und diese interaktiver zu gestalten. Im Unternehmenskontext nutzen Vortragende PINGO insbesondere, um Abstimmungen im Rahmen von Workshops und größeren Plenumsveranstaltungen durchzuführen. Umfrageersteller nutzen hierbei vorrangig Single-Choice- (46 % aller im System hinterlegten Fragen) oder Multiple-Choice-Fragen (40 % aller Fragen). Der Anteil an wiederholten Umfragen beträgt 13 %. Dies legt den Schluss nahe, dass Peer Instruction (und vergleichbare umfassendere Lehr-/Lerndesigns) nur ein Einsatzszenario unter vielen anderen darstellt.

Darüber hinaus nutzen Vortragende die Kollaborationsmöglichkeiten von PINGO, um gemeinschaftlich Veranstaltungen vorzubereiten. So werden 10 % aller Fragen und 8 % aller Veranstaltungen für eine Zusammenarbeit freigegeben, 13 % aller geteilten Fragen weisen mindestens einen Kommentar auf. Es zeigt sich jedoch auch, dass Fragen nur sehr selten (3 % aller im System angelegten Fragen) öffentlich allen anderen Nutzern zur Verfügung gestellt werden.

Eine zentrale Herausforderung bei der Verwendung eines ARS wie PINGO besteht aus unserer Sicht nach wie vor in der Formulierung geeigneter Fragen, insbesondere wenn es darum geht Niveau-gerecht, nicht-trivial, aber trotzdem trennscharf das Verständnis von

Konzepten und nicht nur Wissen abzufragen. So berichten uns Dozenten vielfach, dass der Einsatz von Votinglösungen in Veranstaltungen zwar sehr positiv von Zuhörern angenommen wird, jedoch hohe Anforderungen an die (Um-) Gestaltung der eigenen Lehrveranstaltung stellt. Die teilweise mit Standard-Lehrbüchern ausgelieferten Fragenkataloge (sog. „test banks“) stellen dabei teilweise einen guten Startpunkt dar (und lassen sich i.d.R. auch problemlos nach PINGO importieren). Zumindest in unserer eigenen Domäne ist trotzdem jeweils noch viel Eigenleistung nötig. Es wäre wünschenswert, wenn sich mittelfristig evaluierte themen- und fachspezifische Fragenkataloge für ARS etablieren ließen, welche kontinuierlich weiterentwickelt werden. Dies würde die Effizienz des Einsatzes von ARS im Lehrkontext aus unserer Sicht deutlich erhöhen. Plattformen wie PINGO könnten solche Kataloge zentral bereitstellen und die Weiterentwicklung über Kommentar- und Diskussionsmöglichkeiten unterstützen.

4.2 Bewertung der eingesetzten Technologien

Rückblickend erwies sich insbesondere die Wahl von Ruby on Rails als Web-Framework aufgrund der flexiblen Integrierbarkeit mit anderen Softwarekomponenten und der Verfügbarkeit diverser Bibliotheken, in Form von sogenannter *gems*, als äußerst vorteilhaft. Im Vergleich zur anfänglichen Architektur (siehe [Ku12]), wurde 2016 das Websocket-System angepasst. Das ursprüngliche *socket.io*, welches beispielsweise dazu diente die verbleibende Abstimmzeit bei allen Teilnehmern synchron anzuzeigen, wurde durch *Faye* ersetzt. Mit den steigenden Nutzerzahlen und der sich dadurch erhöhenden Zahl an parallel zu verarbeitenden Serveranfragen, zeigte sich *socket.io* wenig robust. *Faye* erwies sich in dem Anwendungskontext als weit robuster und ist in der Lage eine Vielzahl paralleler Anfragen fehlerfrei zu verarbeiten.

Daneben erwies sich der Einsatz der NoSQL-Datenbank MongoDB aufgrund ihrer Skalierbarkeit als äußerst vorteilhaft. Obwohl während besonderer Lastzeiten zwischen dem Webserver und dem Datenbankserver 10 MB/s an Daten ausgetauscht werden, verbleiben die Antwortzeiten des MongoDB Datenbankservers unter einer Millisekunde. In einer Simulation mit einem exakten Replikat der Infrastruktur, gelang es uns zudem nicht eine Leistungsgrenze zu identifizieren [Ba16]. So verblieb auch bei einer fünffach höheren Maximallast (50 MB/s) die Antwortzeit bei unter einer Millisekunde.

Die eingesetzte Softwarearchitektur bedingt, dass es bisher zu keinerlei serverseitigen Systemengpässen kam. Nutzerberichte über einen „nicht funktionierenden PINGO-Service“ ließen sich bislang ausnahmslos auf eine nicht adäquate WLAN-Ausstattung vor Ort (insb. Anzahl der Access Points in großen Hörsälen) zurückführen. Viele Hochschulen haben in den vergangenen Jahren jedoch massiv in die WLAN-Ausstattung investiert, um verschiedenste E-Learning Angebote auch in großen Hörsälen für alle Hörer nutzbar zu machen. Gleichwohl empfehlen wir jedem unserer Nutzer, vor dem Einsatz von PINGO in Großveranstaltungen, mit dem IT-Service Kontakt aufzunehmen, um zu prüfen, inwiefern die WLAN-Ausstattung die Nutzung von PINGO erlaubt.

Die Softwarearchitektur bedingt jedoch auch, dass die Einarbeitungszeit auf unserer Seite für neue Mitglieder im PINGO-Entwickler-Team vergleichsweise hoch ist. So stellt Ruby on Rails ein Webframework dar mit dem funktionale Programmierkonstrukte umgesetzt werden und MongoDB basiert auf einem nicht-relationalen Datenbankkonzept. Trotz umfangreicher Vorkenntnisse in der objektorientierten Programmierung und relationaler Datenbankmodelle, gilt es für neue Teammitglieder die fundamental anderen Grundkonzepte von Ruby on Rails und MongoDB zeitaufwändig zu erlernen. In der Einarbeitungsphase zeigt sich zudem, dass weniger (allgemeine) Vorkenntnisse dabei oftmals zweckdienlicher sind. Hochschulen, welche sich dazu entschieden haben die Open Source-Version von PINGO selbst zu hosten, berichten uns darüber hinaus von Schwierigkeiten bei der Einrichtung und Weiterentwicklung der Software, da die genutzte Architektur hohe Anforderungen an den Betrieb und die Weiterentwicklung stellen und die benötigten Kompetenzen oftmals nicht zur Verfügung stehen.

4.3 Verstetigung

Eine große Herausforderung für *erfolgreiche* E-Learning-Systeme wie PINGO besteht aus unserer Sicht darin, diese nach einer (geförderten) Projektphase zu verstetigen. Bislang erfolgen der Betrieb und die Weiterentwicklung nach Auslaufen der Projektförderung aus lehrstuhleigenen Mitteln. Aufgrund der hohen Anforderungen an ein webbasiertes ARS (z. B. 24/7-Verfügbarkeit) und der weiten Verbreitung von PINGO, stellen Nutzer vielfältige Anfragen und neue Anforderungen. Die Anfragen möchten wir idealerweise mit einem fachlich fundierten Nutzersupport mit kurzer Reaktionszeit begegnen. Viele der Weiterentwicklungswünsche – beispielsweise fragten in den vergangenen Monaten mehr als 30 Leute an, ob PINGO nicht auch Fragensequenzen unterstützen könne und ob es die Möglichkeit gäbe, PINGO in PowerPoint zu integrieren – würden wir gerne aufgreifen. Die Aufgabe eines Lehrstuhls ist jedoch i. d. R. nicht einen 24/7-Support sowie eine Auftragsentwicklungsabteilung zu unterhalten – und nicht für jeden Featurewunsch kann ein eigenes „Forschungsprojekt“ erfolgreich beantragt werden. In Gesprächen mit anderen Projektverantwortlichen wurde uns zudem klar, dass das Problem der Verstetigung zentral ist, für viele erfolgreiche universitäre Projekte und Prototypen. Mindestens zwei Handlungsbereiche drängen sich diesbezüglich auf: Zum einen sollten Projektverantwortliche selbst bereits frühzeitig geeignete Kooperationspartner (innerhalb der Universität bspw. das Rechenzentrum und das Lernzentrum) in das Projekt mit einbeziehen, die den Dauerbetrieb bei Erfolg zu einem späteren Zeitpunkt grundsätzlich sicherstellen könnten. Zum anderen gilt es Förderstrukturen zu etablieren, die eine Verstetigung deutlich besser unterstützen.

Literaturverzeichnis

- [Ba16] Bach, C. et al.: Dokumentenorientierte NoSQL-Datenbanken in skalierbaren Webanwendungen. In HMD Praxis der Wirtschaftsinformatik 53, S. 486–498, 2016.

- [Be12] Beutner, M. et al.: PINGO-Umsetzung von E-Learning in der Hochschule in Adaption und Weiterentwicklung des Peer Instruction Ansatzes-Didaktische und organisatorische Reflexion der Studierendenaktivierung in Lehrveranstaltungen. In *Kölner Zeitschrift für Wirtschaft und Pädagogik*. KWP 53, S. 3–52, 2012.
- [Be14] Beutner, M. et al.: Evaluation von Lerndesigns mit einem webbasierten Classroom Response System in der universitären Lehre. In *Proceedings der 12. e-Learning Fachtagung Informatik (DelFI 2014)*, S. 121-126, 2014.
- [Du96] Dufresne, R. J. et al.: Classtalk. A classroom communication system for active learning. In *Journal of Computing in Higher Education* 7, S. 3–47, 1996.
- [GI96] Glaserfeld, E. v.: *Radikaler Konstruktivismus: Ideen, Ergebnisse, Probleme*. Suhrkamp, Frankfurt am Main, 1996.
- [Ha98] Hake, R. R.: Interactive-engagement versus traditional methods. A six-thousand-student survey of mechanics test data for introductory physics courses. In *American Journal of Physics* 66, S. 64–74, 1998.
- [JS02] Judson, E.; Sawada, D.: Learning from past and present. Electronic response systems in college lecture halls. In *Journal of Computers in Mathematics and Science Teaching* 21, S. 167–182, 2002.
- [Ku13] Kundisch, D. et al.: Classroom Response Systems. In *Informatik-Spektrum* 36, S. 389–393, 2013.
- [LGP17] Liao, S. N.; Griswold, W. G.; Porter, L.: Impact of Class Size on Student Evaluations for Traditional and Peer Instruction Classrooms. In *Caspersen, Education (Hg.) 2017 – Proceedings of the 2017 ACM*, S. 375–380, 2017.
- [Ma97] Mazur, E.: Peer instruction: Getting students to think in class: AIP Conference Proceedings. AIP, S. 981–988, 1997.
- [PBS13] Porter, L.; Bailey Lee, C.; Simon, B.: Halving fail rates using peer instruction. In (McCauley, R. Hrsg.): *SIGCSE'13. Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, S. 177, 2013.
- [SPS13] Simon, B.; Parris, J.; Spacco, J.: How we teach impacts student learning. In (McCauley, R. Hrsg.): *SIGCSE'13. Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, S. 41, 2013.
- [Ve14] Vetterick, J. et al.: Classroom Response Systems in the Wild. Technical and Non-Technical Observations. In *International Journal of Interactive Mobile Technologies* 8, S.21–25, 2014.
- [WN13] Whittaker, M.; Neumann, J.: Develop in the fast lane with Ruby/Rails. In *Software Developer's Journal*, S. 10–14, 2013.
- [Zo16] Zoyke, A. et al.: Webbasierte Classroom Response Systeme in der Hochschullehre – Eine designbasierte Fallstudie zur Unterstützung von Lehr-/Lernprozessen aus Studierendensicht. In *Das Hochschulwesen*, 2016.