

**Woelk, Felix und Kasch, Henning (2021): Code FREAK: Automatisches Feedback für die Programmierausbildung. In: Die neue Hochschule, 2021-5, S. 28-31. <https://doi.org/10.5281/zenodo.5530450>**

## Impressum

**Herausgeber:**  
Hochschullehrerbund –  
Bundesvereinigung e. V. **h**l**b**  
Godesberger Allee 64  
53175 Bonn  
Telefon: 0228 555 256-0  
Fax: 0228 555 256-99

**Chefredakteur:**  
Prof. Dr. Christoph Maas  
Molkenbührstr. 3  
22880 Wedel  
Telefon: 04103 141 14  
christoph.maas@haw-hamburg.de  
(verantwortlich im Sinne des Presserechts  
für den redaktionellen Inhalt)

**Redaktion:**  
Dr. Karla Neschke  
Telefon: 0228 555 256-0  
karla.neschke@h**l**b.de  
in Kooperation mit der DUZ Verlags und  
Medienhaus GmbH

**Schlusskorrektur:**  
Manuela Tiller, [www.textwerk-koeln.de](http://www.textwerk-koeln.de)

**Gestaltung und Satz:**  
DUZ Verlags- und Medienhaus GmbH  
Nina Reeber-Laqua, Kronberg

**Titelbild:** skdesign/123rf.com

**Piktogramme:** S. 33, 34, 35, 36 und 39:  
123rf.com

**Herstellung:**  
Wienands Print + Medien GmbH  
Linzer Straße 140, 53604 Bad Honnef

**Verlag:**  
DUZ Verlags- und Medienhaus GmbH  
Kaiser-Friedrich-Straße 90  
10585 Berlin  
Telefon: 030 212 987-0  
info@duz-medienhaus.de  
[www.duz-medienhaus.de](http://www.duz-medienhaus.de)  
Dr. Wolfgang Heuser  
(Geschäftsführer)  
[w.heuser@duz-medienhaus.de](mailto:w.heuser@duz-medienhaus.de)

**Anzeigen:**  
DUZ Verlags- und Medienhaus GmbH  
Stefanie Kollenberg (Leitung),  
Telefon: 030 212 987-31  
Fax: 030 212 987-20  
[anzeigen@duz-medienhaus.de](mailto:anzeigen@duz-medienhaus.de)

**Erscheinung:**  
zweimonatlich

**Bezugsbedingungen:**  
Jahresabonnements für Nichtmitglieder  
45,50 Euro (Inland), inkl. Versand  
60,84 Euro (Ausland), inkl. Versand  
Probeabonnement auf Anfrage  
Erfüllungs-, Zahlungsort und Gerichtsstand  
ist Bonn.  
Verbands offiziell ist die Rubrik „**h**l**b** aktuell“.  
Alle mit Namen der Autorin/des Autors  
versehene Beiträge entsprechen nicht  
unbedingt der Auffassung des **h**l**b** sowie der  
Mitgliedsverbände.

**Redaktionsschluss dieser Ausgabe:**  
27. August 2021

ISSN 0340-448 x

# Code FREAK: Automatisches Feedback für die Programmierausbildung

**Wegen großer Teilnehmerzahlen kann Feedback durch Dozierende nicht immer individuell gegeben werden. Das Open Source System „Code FREAK“ wurde entwickelt, um Lehrende durch automatisiert erzeugtes Feedback und Bewertungen in der Programmierausbildung zu entlasten.**

| Von Prof. Dr.-Ing. Felix Woelk und Henning Kasch



Foto: Matthias Plich

**Prof. Dr.-Ing. Felix Woelk**  
Professor für Informatik

felix.woelk@fh-kiel.de, fh-kiel.de



Foto: privat

**Henning Kasch**  
Projektmitarbeiter

codefreak@fh-kiel.de, fh-kiel.de

beide:

Fachhochschule Kiel  
Grenzstraße 5  
24149 Kiel

In vielen technischen Studiengänge wie Elektrotechnik, Maschinenbau, Mechatronik oder Informatik werden grundlegende Programmierkenntnisse gefordert. Darüber hinaus finden sich diese zunehmend auch in nicht technischen Curricula (z. B. Medien, Wirtschaft) wieder. Dies führt häufig dazu, dass die Ausbildung in der Programmierung in sehr großen Gruppen durchgeführt werden muss. Trotzdem sollen Studiengänge vermehrt auf fachliche Kompetenzen, d. h. die eigenständige Fähigkeit zur Umsetzung oder Anwendung einer bestimmten Tätigkeit, ausgerichtet werden (Schaper et al., 2012). In großen Veranstaltungen werden Aufgaben aufgrund der eingeschränkten Zeit der Lehrenden häufig durch Gruppen von Studierenden bearbeitet. Für die Lehrenden ist es oftmals nicht möglich, die Leistungen innerhalb dieser Gruppen zu differenzieren. Und obwohl die herausragende Stellung von Feedback für den Lernprozess unstrittig ist (Hattie and Timperley, 2007), kann daher individuelles, qualifiziertes Feedback nicht ausreichend gegeben werden.

Häufig ist Feedback deswegen ineffizient, weil es zu spät gegeben wird. Daher sollte Feedback zeitnahe erfolgen und darüber hinaus die Studierenden zur Auseinandersetzung mit der Rückmeldung animieren (Carless, 2006). Lernerfolge werden in der Gruppe begünstigt, wenn sich alle Personen aktiv mit den Aufgaben auseinandersetzen. Dies ist aber nicht immer der Fall: Das Phänomen des „Social Loafing“ (Trittbrettfahren) beschreibt, dass sich nur ein Teil der

Gruppenmitglieder aktiv mit dem Lernstoff auseinandersetzt, während der restliche Teil eine passive Rolle einnimmt. Dies gefährdet den Lern- oder Studienerfolg, weil dadurch wichtige Grundlagen fehlen.

An diesem Punkt setzt die hier vorliegende Arbeit an: Code FREAK entlastet die Lehrenden beim Geben von Feedback und Bewertung, sodass die Bearbeitung der Aufgaben individuell erfolgen kann. Es gibt mehrere Übersichtsartikel über Systeme zur automatisierten Beurteilung: Wasik et al. (2018) haben das Thema in einem breiteren Kontext betrachtet und eine Übersicht zu automatisierten Online-Beurteilungssystemen zusammengetragen. Dabei unterteilen sie die Systeme anhand verschiedener Nutzungsszenarien: Online Compiler, System für Data Mining, Ausbildung und Programmierwettbewerbe, System zur Personalgewinnung sowie Entwicklungsumgebungen. Darüber hinaus wird auch der Evaluationsprozess formal definiert: Er besteht aus den Schritten Abgabe, Beurteilung und Vergabe von Punkten. Keuning et al. (2019) fokussieren sich in ihrer Übersicht auf Systeme zur Programmierausbildung.

## Anforderungen

Das übergeordnete Ziel lautet: Jeder Studierende soll individuelles und detailliertes Feedback erhalten, um das Lernergebnis zu optimieren – selbst in großen Kohorten. Daraus ergeben sich die folgenden konkreten Anforderungen:

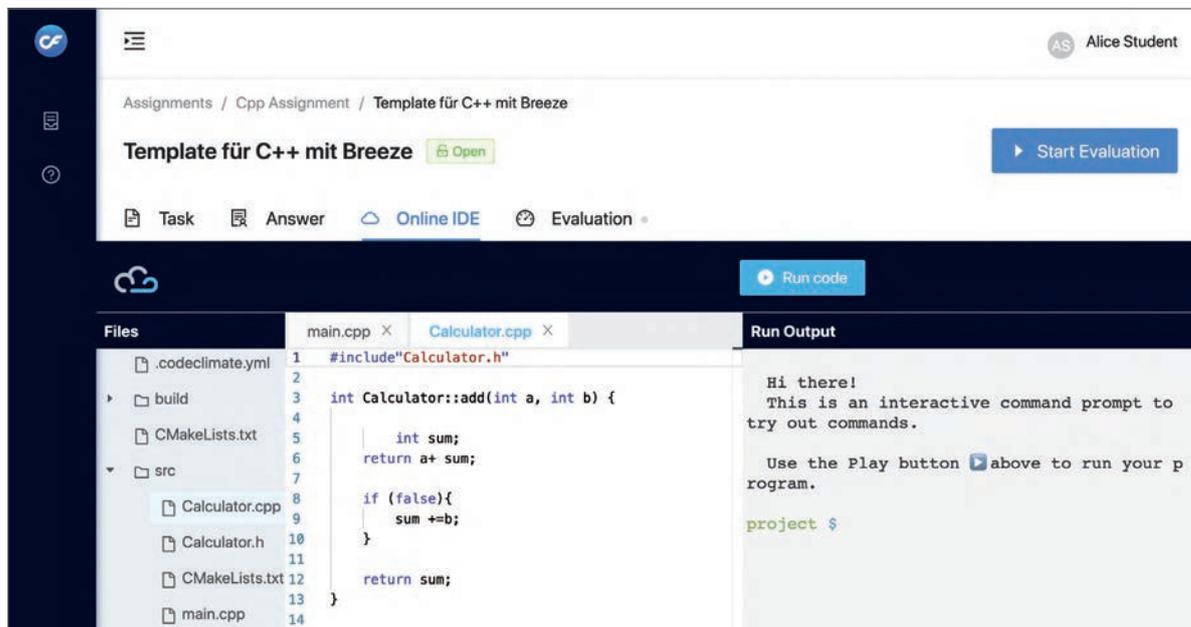


Abbildung 1: Die Online-IDE von Code FREAK zur Bearbeitung von Aufgaben.

1. Feedback soll automatisiert erzeugt werden.
2. Manuelles, komplexes Feedback soll effizient gegeben werden können.
3. Eine Entwicklungsumgebung (IDE) soll ohne Einrichtung verwendet werden können.
4. Das Erstellen von Aufgaben und Arbeitsblättern muss effizient sein.
5. Das System muss agnostisch gegenüber der konkreten Programmiersprache sein.
6. Die Komplexität der möglichen Aufgaben soll vom System nicht eingeschränkt werden.
7. Feedback soll zeitnah auf die Einreichung erfolgen.
8. Feedback soll detailliert und hilfreich sein.
9. Das User Interface (UI) soll unabhängig von Hardware und Betriebssystem sein.
10. Das UI soll intuitiv bedienbar sein.
11. Das System soll kostengünstig sein.
12. Das System soll stabil sein.
13. Das System soll einfach erweitert werden können (um neue Tools oder Sprachen).
14. Die relevanten gesetzlichen Regelungen (z. B. DSGVO) sollen erfüllt werden.
15. Die Ausführung von studentischen Abgaben soll kein Risiko darstellen.
16. Das System muss für viele Nutzer gleichzeitig nutzbar sein.
17. Das System soll sich in gängige Lernmanagementsysteme (LMS) integrieren.

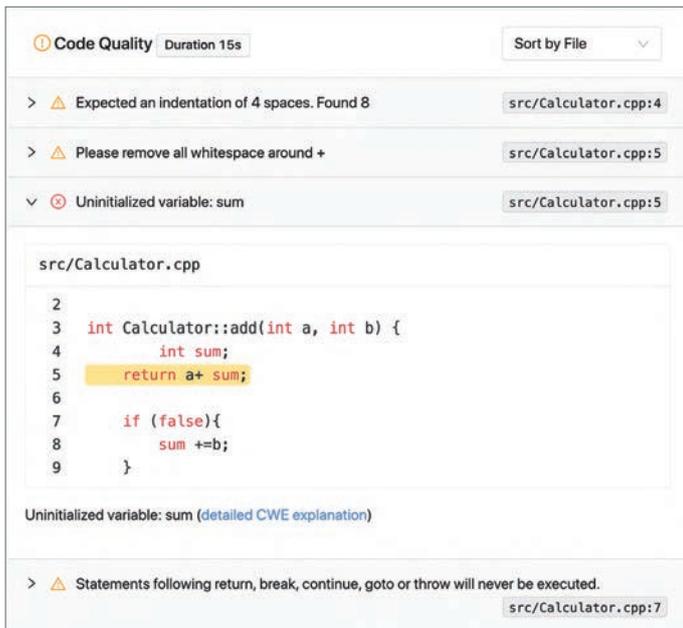
## Umsetzung

Code FREAK (Code Feedback, Review and Evaluation Kit) wurde entwickelt, da keines der wichtigsten Systeme im Markt (repl.it, codepen.io, codecademy.com, Moodle Plugin CodeRunner, Jupyter nbgrader & Aizu Online Judge) alle oben genannten Anforderungen erfüllt. Das UI von Code FREAK läuft unabhängig vom verwendeten Betriebssystem im Browser

*„Jeder Studierende soll individuelles und detailliertes Feedback erhalten, um das Lernergebnis zu optimieren – selbst in großen Kohorten.“*

und bietet unterschiedliche Sichten für Studierende und Dozierende. Die Sicht der Studierenden (siehe Abbildung 1) ist durch Reduktion auf essenzielle Funktionen besonders intuitiv bedienbar. Die Daten sind in Code FREAK wie folgt strukturiert: Die einzelnen Aufgaben (Tasks) werden in Aufgabenblättern (Assignments) zusammengefasst und so den Studierenden zur Verfügung gestellt. Aufgaben können aus einer Aufgabensammlung (Task Pool) in ein Aufgabenblatt übernommen werden. Code FREAK besteht aus fünf grundlegenden Teilen: dem Abgabesystem, einer online IDE, einem Reviewsystem, einem System für automatisches Feedback und eines zur automatischen Bewertung der Abgabe. Feedback kann auf zwei unterschiedliche Arten automatisch erzeugt werden:

- Die statische Code-Analyse (Abbildung 2) erzeugt automatisiert Hinweise zur Struktur des Quellcodes (z. B. Einrückung), zu Code Smells (z. B. unbenutzte Variablen, Länge von Funktionen, nicht erreichbarer Code, unbenutzte oder nicht initialisierte Variablen usw.) und zu potenziellen oder echten Fehlern (z. B. Hinweise zu Sicherheitslücken in den verwendeten Bibliotheksfunktionen). Hierfür werden die etablierten Tools für die verschiedenen Programmiersprachen einheitlich aggregiert.



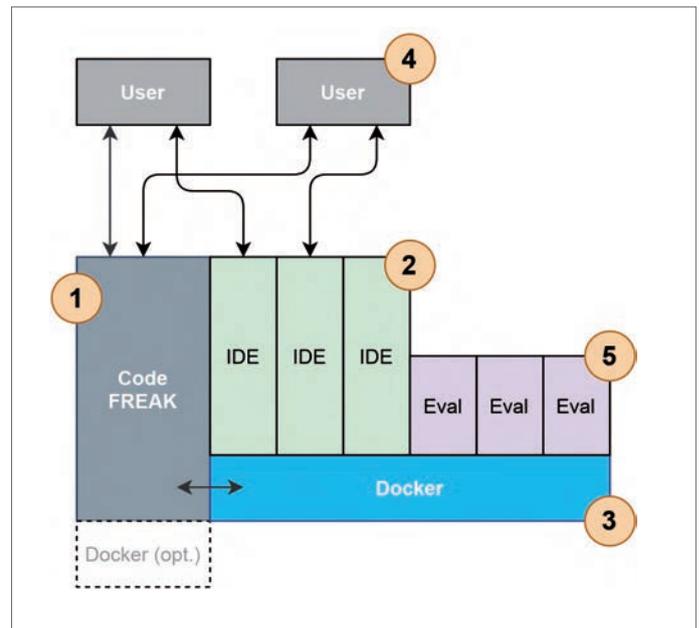
**Abbildung 2:** Das Ergebnis der statischen Codeanalyse wird von Code FREAK angezeigt (Ausschnitt).

■ Automatisierte Tests auf funktionale Korrektheit für Aufgaben werden vom Dozierenden erstellt. Dafür gibt es Vorlagen für die wichtigsten Programmiersprachen (aktuell für Java, JavaScript, C++, C# sowie Python). Das Feedback wird entweder mit einem Unit-Test-Framework (JUnit, Jest, Google Test, NUnit, pytest, etc.) oder mit einem beliebigen, ausführbaren Programm (Skript) erzeugt. Das Übersetzen und Ausführen der Abgabe ist dabei der erste und einfachste Test, der immer durchgeführt werden sollte. Zwei Beispiele für Aufgaben mit automatischem Feedback sind die Umsetzung einer Funktion zur Suche von Buchstaben in einem gegebenen Text sowie das Auflösen von Mergekonflikten bei der Verwendung von Git.

Zusätzlich zur automatisierten Rückmeldung können abgegebene Dateien manuell kommentiert werden. Code FREAK setzt sich aus vier Komponenten zusammen (Abbildung 3):

1. Backend
2. Docker Daemon
3. Container mit IDEs (eine Instanz pro Studierenden)
4. Container für Ausführung und Evaluation (je eine Instanz pro Studierenden)

Diese Konzeption als Cloud-Anwendung hat mehrere Vorteile: Die Anwendung bleibt durch die Kapselung der Container auch bei der Ausführung von nicht vertrauenswürdigen studentischen Code sicher. Ferner ist das System sowohl sehr gut horizontal skalierbar als auch einfach auf andere Programmiersprachen und Tools erweiterbar. Es existiert eine erste, rudimentäre Anbindung an Lernplattformen über die standardisierte LTI-Schnittstelle: Durch

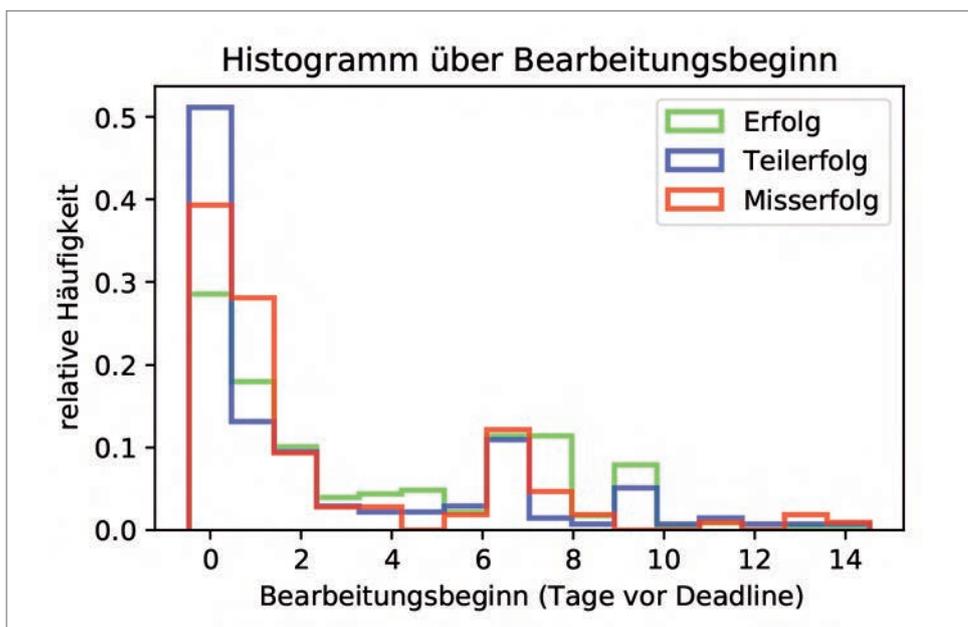


**Abbildung 3:** Architektur von Code FREAK: Jeder Studierende ist vom System getrennt. Er arbeitet in seinem eigenen IDE Container und seine Programme und Evaluationen werden ebenfalls in einem eigenen Container ausgeführt.

diese Anbindung werden Benutzerdaten inklusive der Authentifizierung übernommen und Code FREAK kann als externes Tool in Kurse eingebunden werden.

### Ergebnisse

Code FREAK wurde durch ein kleines Team aus einem Teilzeitmitarbeiter und mehreren Studierenden entwickelt. Die Entwicklung von Code FREAK wurde gleichzeitig für die Ausbildung im Studiengang Informationstechnologie genutzt: Teile des Systems sind als Projekt oder Thesis umgesetzt worden. Bislang haben über 400 Studierende und fünf Lehrende Code FREAK an der FH Kiel genutzt. Die Ausbildung erstreckte sich dabei von den Grundlagen der Programmierung mit Java oder C über Git bis hin zu fortgeschrittenen Kursen mit Python. Im kommenden Semester werden zwei neue Module mit C++ und Python hinzukommen. Einen ersten Überblick über die Nutzung des Tools liefern die Ergebnisse des Kurses ‚Programmieren‘ (1. Semester, Studiengänge Informationstechnologie sowie Elektrotechnik). Im Verlauf des Semesters wurden den Studierenden sieben verschiedene Programmieraufgaben über das System gestellt, die erste Aufgabe haben 106 Studierende abgegeben, die letzte Aufgabe nur noch 61 Studierende. Bei der Auswertung der Daten ist auffällig, dass erfolgreiche Studierende das System intensiver nutzen: Sie fangen im Mittel früher mit der Bearbeitung der Aufgaben an (Mittelwert  $\mu=145h$ , Standardabweichung  $\sigma=199$  vs.  $\mu=78h$ ,  $\sigma=129$ , siehe Abbildung 4) und nutzen die Möglichkeit zu evaluieren öfter ( $\mu=5.1$ ,  $\sigma=6.62$  vs.  $\mu=2.5$ ,  $\sigma=1.9$ ). Ein Studierender hat seine verschiedenen Lösungsvarianten für



**Abbildung 4:** Histogramm über den Beginn der Bearbeitung relativ zur Deadline in Abhängigkeit vom Erfolg. Erfolgreiche Studierende fangen im Mittel früher mit der Bearbeitung der Aufgaben an.

eine einzige Aufgabe insgesamt 74 mal vom System evaluieren lassen. Dabei hat er die Lösung sukzessive verbessert, sodass am Ende alle automatischen Tests erfolgreich bestanden wurden und die statische Code-Analyse ebenfalls keine Anmerkungen mehr lieferte.

### Fazit & Ausblick

Bereits bei der ersten Nutzung von Code FREAK zeigte sich, dass Studierende, die das automatisierte Feedback vermehrt nutzten, im Mittel ein besseres Ergebnis erzielten. Die Hoffnung ist, dass diese Studierenden als Multiplikator wirken und so die individuelle Nutzung von Code FREAK stimulieren. Die abschließende Untersuchung des Einflusses von Code FREAK auf das Lernergebnis steht allerdings noch aus. Code FREAK steht als Open Source Projekt

(GNU Affero General Public License) unter [codefreak.org](https://codefreak.org) zur Verfügung. Es befindet sich im Stadium eines fortgeschrittenen Prototypen mit sehr guter Stabilität. Insgesamt sind schon viele Anforderungen umgesetzt, allerdings ist die Usability noch nicht in allen Aspekten befriedigend.

Code FREAK wird ständig verbessert und weiterentwickelt. Geplante Funktionen sind z. B. ein Score Board über Bewertungen, verbesserter Review-Mechanismus, verbesserte Skalierung durch Container-Orchestrierung, automatische Erkennung von Plagiaten, bessere Anbindung an LMS, Unterstützung für nicht-Browser-basierte IDEs usw. Mittelfristig soll die Nutzung des Tools ausgebaut und eine Open Source Community aufgebaut werden. Auch der Aufbau einer Sharing-Plattform für Aufgaben wäre denkbar. Ferner steht der Test des Tools in einer Prüfungssituation noch aus. ■

<https://doi.org/10.5281/zenodo.5530450>

## Literatur

- Carless, C.: Differing perceptions in the feedback process. In: *Studies in higher education* Nr. 2, Jg. 31 (2006), S. 219–233.
- Hattie, J.; Timperley, H.: The power of feedback. In: *Review of educational research* Nr. 1, Jg. 77 (2007), S. 81–112.
- Keuning, H.; Jeuring, J.; Heeren, B.: A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Transactions on Computing Education* Nr. 1, Jg. 19 (2019), S. 1–43.
- Schaper, N.; Reis, O.; Johannes, W.; Horvath, E.; Bender, E.: *Fachgutachten zur Kompetenzorientierung in Studium und Lehre*, Hochschulrektorenkonferenz, Arbeitspapier, 2012.
- Wasik, S.; Antczak, M.; Badura, J.; Laskowski, A.; Sternal, T.: A survey on online judge systems and their applications. In: *ACM Computing Surveys* Nr. 1, Jg. 51 (2018), S. 1–34.