Digitale Prüfungen für Softwareentwicklung

im „Bring Your Own Device, Open Book, Open Web“-Format

Axel Böttcher

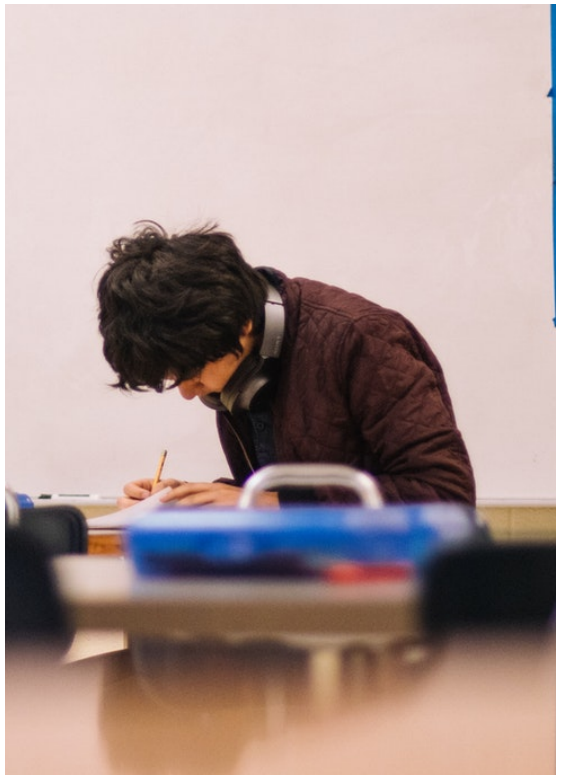
Veronika Thurner

Motivation und Ziele

Papier-basierte Prüfungen

Praktikum

Prüfung



Elektronische Prüfungen

Praktikum

Prüfung

Unsere Ziele

HM

- Constructive Alignment herstellen
- Software-✱ realitätsnah lehren und prüfen
- Möglichst automatisch bzw. automagisch unterstützt bewerten
- Randbedingungen einhalten
  - Rechtssicherheit
  - Unterschleif vermeiden
  - Technisch-/organisatorische Durchführbarkeit

Prüfung organisieren und durchführen

ExaHM

HM

- 😀 Time-boxing
- 😀 Kontinuierliche automatische git-commits

- 😐 Eingeschränkte Rechte für die Teilnehmenden
- 😐 Kein Zugang zum öffentlichen Internet
- 😐 Dedizierte Räume erforderlich

- ☹ Windows-basiert
- ☹ Vorgegebene Arbeitsumgebung

Moodle Coderunner

HM

- 😀 Time-boxing
- 😀 Rechtssichere Speicherung der Lösungen

- 😐 Kein JUnit (“Tests” per println nicht wirklich realitätsnah)
- 😐 Abhängigkeit von unserem Elearning Center

BYOD-Prüfung mit gitlab-Repositories

HM

- Eigene Rechner der Studierenden
- Bekannte Entwicklungsumgebung
- Open book, open alles (google, stackoverflow, wie im richtigen Leben)

- Kommunikation mit Anderen per Chat etc. ist nicht erlaubt

- git-Repositories lassen sich vorab erzeugen und ggfs. befüllen

BYOD-Prüfung: Variante Fernprüfung

- Aufsicht per Videokonferenz
  - Moodle-Kurs vorgeschaltet mit Aktivität *Test*
    - zur Zustimmung für VK-Aufsicht
    - zur Abgabe der Selbstständigkeitserklärung
    - Zum anschließenden kommunizieren des Zugangskennwort für zoom-Prüfungsraum
- Aufgabenstellung ...
  - ... im Repository in verschlüsseltem zip-Archiv hinterlegt oder
  - ... kurzfristig gepusht

BYOD-Prüfung: Variante Präsenzprüfung

- Aufsicht persönlich
- Aufgabenstellung ...
  - ... im Repository
  - ... austeilen auf Papier, hinterher wieder einsammeln
- Zusätzliches Material (gegebener Quelltext, Tests) im Repository, verschlüsselt, oder kurzfristig gepusht

BYOD-Prüfung mit gitlab-Repositories

- 😃 Einfacher Switch zwischen Fern- und Präsenzprüfung

- 😟 Hoher Aufwand bei Aufgabenerstellung – googeln der Lösung darf nicht ganz einfach möglich sein (ChatGPT?)
- 😟 Aufgaben zirkulieren leicht – große Aufgabensammlung erforderlich
- 😟 Push auf das Repository nach offiziellem Prüfungsende durch Rechteentzug
- 😟 Handling verschlüsselter Archive durch die Studierenden war hakelig

Täuschungsversuche

In Präsenz weniger problematisch als bei VK-Aufsicht

- Chat mit Menschen
- Chat mit ChatGPT
- Kollaborations-Plugins in der IDE (z.B: Codetogether in IntelliJ)

- Cross-checks mit Plagiats-Checkern (moss) immer einsetzbar

Measure of Software Similarity

Emotionale Ebene

- Verantwortung für Prüfungsumgebung verlagert sich teilweise auf die Prüflinge:
  - Hilflosigkeit/Panik bei Problemen
  - Schwächere Studierende kommen mit unerwarteten Schwierigkeiten im eigenen System schlechter zurecht als die anderen

- Prüfungsangst auch auf Seite der Prüfenden
  - Wird das mit zunehmender Erfahrung besser?

Prüfungsaufgaben stellen

Bloom-Levels adressieren

HM

- Hohe Levels eher kein Problem
- Wissensfrage: „Schreiben Sie eine Klassenmethode, die ...“, „Halten Sie die Prinzipien des Information Hiding ein“
  - L1: Wissen
  - Wie automatisch testen/bewerten?

Schreiben Sie eine Klassenmethode,

die ... <*Beschreibung einer sehr einfachen Funktionalität*>.

Halten Sie bei der Implementierung die gelernten Konventionen ein.

Kompetenzebenen
1: Erinnern und 2: Verstehen

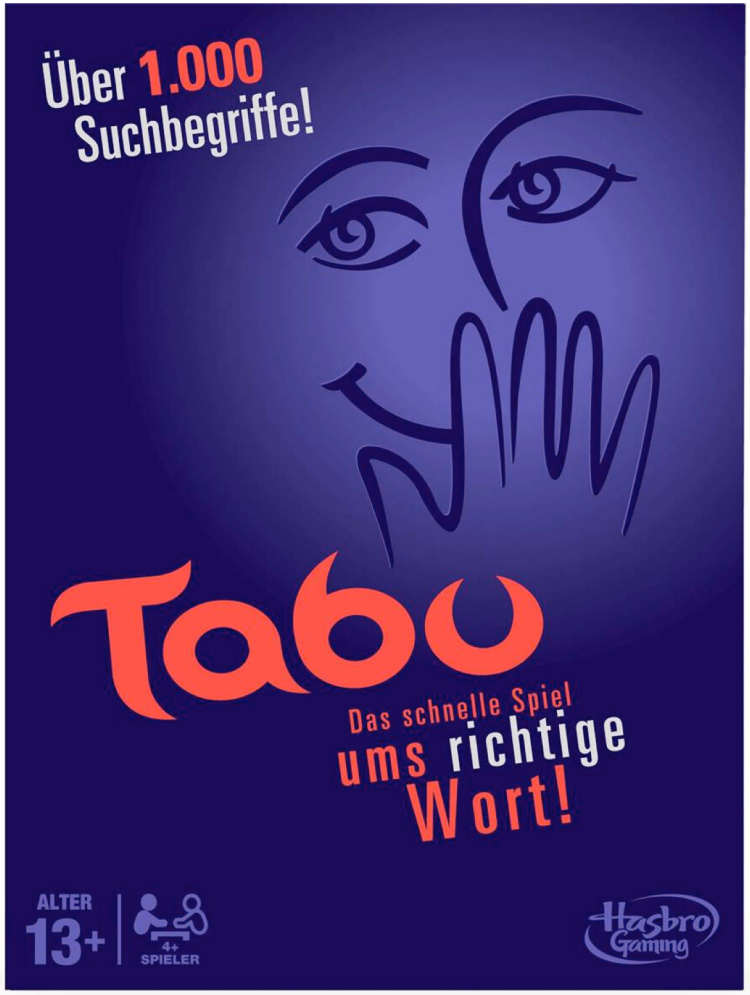
Kompetenzebenen
4: Analysieren, 5: Evaluieren und 6: Kreieren

Kompetenzebenen
1: Erinnern, 2: Verstehen und 3: Anwenden

Schreiben Sie eine Klassenmethode,

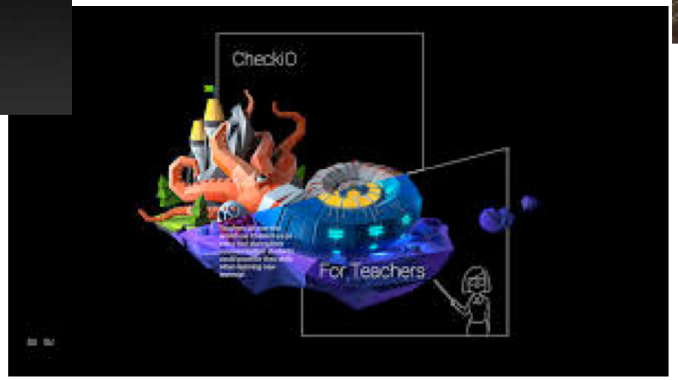
die ... <*Beschreibung einer sehr einfachen Funktionalität*>.

Halten Sie bei der Implementierung die gelernten Konventionen ein.



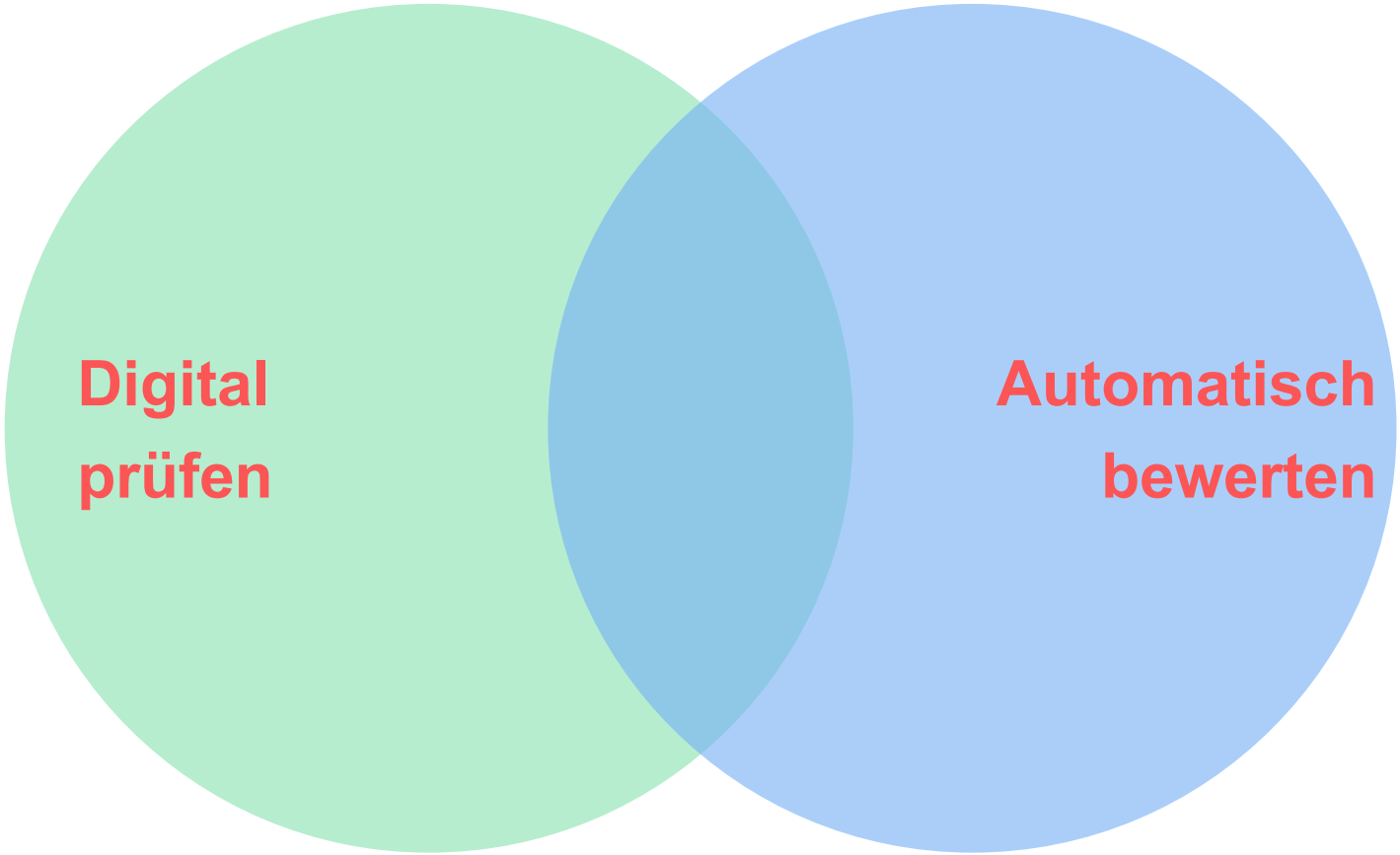
Immer neue Aufgaben: Sammlungen

Codingbat



Prüfung bewerten

Bewertung



Die Bewertung kann mit Skripten und Entwicklungsumgebung einigermaßen unterstützt werden:

- Ein Projekt mit einem Package je Person
- Zusätzliche Tests einkopierbar
- Testausführung automatisch

Überraschend viele Syntaxfehler

HM

- = statt ==
- private Attribute ohne Getter
- Blind Vorschläge der IDE akzeptiert ODER
  Vorschläge der IDE ignoriert
- ...

Können/dürfen wir Syntaxfehler heilen?

HM

- Edit-Operation (Eingriff des/der Bewertenden) kann gut- oder böswillig sein.
- Der Eingriff nicht objektivierbar.
- Selbst wenn wir Syntaxfehler korrigieren würden, treten dann meist (mehrere) funktionale Fehler zutage.

Fazit

Offene Fragen

- Ab wie vielen Teilnehmenden lohnt der Aufwand?
- Wie zuverlässig messe ich ...
  - die richtigen Kompetenzen?
  - die Kompetenzen der Prüflinge – und nicht die anderer Personen bzw. die von ChatGPT?
- Rechtssicher?
- Wie automatisiert bewerten?

Fazit

- Digital prüfen ist richtig und wichtig
- BYOD ist ein Erfolg versprechender Weg
- Wir müssen das digitale Prüfen noch üben (Studierende und Prüfende)
- Wir brauchen eine Auswertung, ob das Prüfungsergebnis zur Leistung im Praktikum passt (validieren, dass tatsächlich die **richtigen** Kompetenzen der **Prüflinge** geprüft werden)