

Classroom 50 an der BZZ

Classroom 50 ([foundation50/classroom50](https://github.com/foundation50/classroom50)) ersetzt GitHub Classroom. Es verteilt Aufgaben aus Template-Repositories, erstellt pro Lernender und Aufgabe ein privates Repository, bewertet jeden Push automatisch und sammelt die Resultate ein.

Die eigentliche Bewertung — pytest und pylint — macht an der BZZ [pygrader50](#), der Nachfolger von BZZ-Commons/pygrader. Der Übertrag nach Moodle läuft nachts aus dem Konfigurations-Repository der Klasse.

Was sich gegenüber GitHub Classroom ändert

Thema	GitHub Classroom (bis 2026)	Classroom 50 (ab August 2026)
Aufgaben verteilen	Web-Oberfläche von GitHub Classroom	gh teacher-CLI + assignments.json im Config-Repo
Bewertungs-Workflow	.github/workflows/classroom.yml in jedem Repo	zentraler Reusable Workflow, wird beim Anlegen eingesetzt
Bewertungs-Logik	BZZ-Commons/pygrader	BZZ-Commons/pygrader50, per Tag gepinnt
Resultat der Abgabe	Kommentar / Moodle-Aufruf aus dem Studi-Repo	GitHub-Release + Commit-Status + Feedback-PR
Moodle-Übertrag	direkt aus dem Studi-Repo, Token per secrets: inherit	nachts, zentral aus dem Config-Repo
Zuordnung zu Moodle	GitHub-Benutzername im Moodle-Profil (Feld <i>Pseudonym</i> bzw. <i>GitHub Classroom</i>)	Moodle-Benutzername = GitHub-Login

Die letzte Zeile ist die häufigste Stolperfalle bei der Umstellung — siehe [Noten nach Moodle übertragen](#).

Die beteiligten Repositories

Repository	Rolle
<ORG>/classroom50	Konfiguration der Klasse: assignments.json, roster.csv, scores.json, Workflows, Pages-Site. Privat.
<ORG>/<CLASSROOM>-<SLUG>-<LOGIN>	Repository einer lernenden Person für eine Aufgabe. Privat.
templates-python/<SLUG>	Startcode, Tests und Musterlösung einer Aufgabe
BZZ-Commons/pygrader50	Bewertungs-Engine und Moodle-Übertrag
foundation50/classroom50	Upstream-Projekt, Wiki

Platzhalter: <ORG> = GitHub-Organisation der Klasse (z.B. m323-ix24), <CLASSROOM> = Kurzname des Klassenzimmers, <SLUG> = Kurzname einer Aufgabe (z.B. m323-lu01-a02-imperativer-ggt).

Ablauf einer Abgabe

```

Lernende:r pusht
├─ autograder-runner.yaml          (im Studi-Repo, ruft den zentralen
Workflow)
├─ runner.py                      (von der Pages-Site des Config-
Repos)
├─ <CLASSROOM>/autograder.py (Klassen-Default der BZZ)
├─ pip install pygrader50@<TAG>
├─ python -m pygrader50
├─ result.json                    → Release, Commit-Status
├─ release-body.md                → Release-Text, Job-Summary,
Feedback-PR
├─ collect-scores (nachts)        → <CLASSROOM>/scores.json
├─ moodle-sync (nachts, 04:57) → Moodle-Notenbuch

```

Zwischen Push und Note in Moodle liegt also **eine Nacht**. Das Feedback im GitHub-Repo steht dagegen nach ein bis zwei Minuten.

Wo welche Konfiguration liegt

Der Runner sucht pro Aufgabe in **dieser Reihenfolge** und nimmt den ersten Treffer:

1. autograders/<SLUG>/autograder.py im Config-Repo — eigener Autograder für genau diese Aufgabe
2. autograders/<SLUG>/tests.json im Config-Repo — deklarative Tests von Classroom 50
3. <CLASSROOM>/autograder.py im Config-Repo — der Klassen-Default, an der BZZ pygrader50
4. nichts davon → die Abgabe wird mit **0/0** aufgezeichnet

An der BZZ ist Punkt 3 der Normalfall. Wer für eine einzelne Aufgabe eine tests.json hinterlegt, schaltet für diese Aufgabe das **Linting ab** — Punkt 2 gewinnt gegen Punkt 3. Details: [Wie bewertet wird](#).

Weiter

- [Für Lernende: Aufgabe annehmen, bearbeiten, abgeben](#)
- [Für Lehrpersonen: Klassenzimmer und Aufgaben verwalten](#)
- [Wie bewertet wird](#)
- [Noten nach Moodle übertragen](#)
- [Ein Klassenzimmer von Grund auf einrichten](#)
- [Migration von GitHub Classroom](#)
- [Problembehandlung](#)
- [CLI-Referenz](#)

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/howto/git/classroom50/start>

Last update: **2026/08/14 11:12**

