

Valutazione: estratti da
Guide to Teaching Computer Science
An Activity-Based Approach
(capitoli 9 e 10)

Anna Morpurgo

3 giugno 2015

Evaluation is not a target by itself

Evaluation is a pedagogical means

1. teachers improve their understanding of the current knowledge of their pupils;
2. learners get feedback related to their own understanding of the learned topic.

The main target should not be grading.

It should serve as a reflection both for teachers and learners with respect to the teaching and learning processes and with respect to pupils knowledge and perception.

Evaluation methods

As all pedagogical tools, evaluation methods should be varied

- ▶ tests,
- ▶ project evaluation,
- ▶ portfolio in computer science education

Principles

Principles related to evaluation:

- ▶ A single and unique way to evaluate computer science learners does not exist and different evaluation approaches are appropriate to be applied in different pedagogical situations;
- ▶ The different evaluation approaches should make sense, and when appropriate, should be explained to the pupils (who take the exam, develop the project, etc.);
- ▶ Different aspects of learners knowledge and cognitive skills should be evaluated;

Principles

Principles related to evaluation:

- ▶ Teachers feedback to pupils exercises/exams/projects may convey different messages (sometimes hidden); therefore, careful attention should be given to written feedbacks;
- ▶ Evaluation tasks should be varied in order (a) to relate to different aspects of the learned topic and different cognitive skills, and (b) to motivate learners and keep their curiosity;
- ▶ Evaluation should be conceived as an ongoing reflective process.

Tests

Tests can be written or take place in the computer lab.

The process of test handling is based on several steps:

1. The teacher constructs the test and the test evaluation rubric.
2. Students take the exam.
3. The teacher evaluates the test.
4. The teacher returns the tests to the pupils.

Test Construction

Topics that a teacher should address when constructing a test:

- ▶ the target of the test,
- ▶ the structure of the test,
- ▶ learners level,
- ▶ types of questions,
 - ▶ questions scope: many short and focused; of different scopes; a small number of wide scope questions.
 - ▶ question type: open, closed, programming tasks, etc.
- ▶ questions of different complexity levels,
- ▶ organization of the questions in the test,
- ▶ the grading policy.

Types of Questions

- ▶ Type1. Development of a Solution
- ▶ Type2. Development a Solution That Uses a Given Module
- ▶ Type3. Tracing a Given Solution
- ▶ Type4. Analysis of Code Execution
- ▶ Type5. Finding the Purpose of a Given Solution
- ▶ Type6. Examination of the Correctness of a Given Solution
- ▶ Type7. Completion a Given Solution
- ▶ Type8. Instruction Manipulations
- ▶ Type9. Efficiency Estimation
- ▶ Type10. Question Design
- ▶ Type11. Programming Style Questions
- ▶ Type12. Transformation of a Solution
- ▶ Story Questions
- ▶ Closed Questions

Evaluation rubric

Rubric: set of guidelines that a teacher uses in the grading process of a specific test. The actual preparation of evaluation rubrics encourages teachers

- ▶ to realize what the test actually checks and if it matches their pedagogical intentions;
- ▶ to verify that there is a match between the grades the pupils will get and their actual knowledge;
- ▶ to ensure (as much as it is possible) that all learners exams are checked uniformly by the same criteria.

Evaluation rubric

Rubric sharing

Teachers can share the evaluation rubric with their pupils:

- ▶ when they wish to deliver what they consider important with respect to the test content
- ▶ or when they wish to explain to their pupils how their grades were calculated.

It is not necessary to indicate all its details; rather, each teacher should select the level of detail he or she shares with his or her pupils according to the class characteristics and his or her personal pedagogical considerations. In any case, it is important that pupils be familiar with the evaluation principles.

Evaluation rubric: aspects

- ▶ Point accumulation: Should points be gathered (that is, a pupil starts with zero points and collects points according to his or her answers) or should points be reduced (that is, a pupil starts with 100 points, and mistakes reduce his or her grade)? Each approach is appropriate as long as it is based on relevant pedagogical considerations.
- ▶ What is considered a mistake: For example, if a pupil wrote a correct computer program but did not use meaningful names for methods, should points be reduced? If a student found a solution to a given problem, described it correctly, but did not implement it correctly in the programming language, should points be added?
- ▶ Evaluation of different solutions: If a question can be solved in several ways, is one answer preferable over the others? Are all the solutions accepted?

Project evaluation

Individual projects

- ▶ Teacher evaluation can be performed in two ways:
 - ▶ Formative assessment: carried out by the teacher during the entire process of project development with respect to (almost) each activity that the student performs. The purpose is to guide the pupils in the development process.
 - ▶ Summative assessment: several times during the development process, usually at the end of specific stages, to monitor the students and class progress.
- ▶ Peer assessment: for example, the pupils are divided into groups and each pupil presents his or her project to the other group members and receives their feedback.
- ▶ Individual feedback/evaluation: for example by asking each pupil to reflect on his or her work and on the way he or she plans to meet the schedule that the teacher set for the entire class.

Team Projects

General issues:

- ▶ assignment of students to groups
- ▶ coordination of teamwork
- ▶ the grading of such projects
- ▶ ways by which instructors can gain information about the contribution of individual students to the team project

Team Projects

Evaluation issues

Teamwork is essential for software development, but

- ▶ Conflicts between the contribution to the teamwork and the way by which rewards are shared may intensify.
- ▶ Software developers are usually highly motivated. This can cause conflicts between personal targets and team goals.
- ▶ Team-based rewards may cause social problems, such as the free-rider phenomenon.

Team Projects

Grading policy

It should aim at motivating both team-work and collaboration as well as the personal contribution of each team member to the project success. A proposal:

- ▶ group component (65%) whose main criterion is the meeting of the customer stories as well as the time estimations given by the students at each of the three iterations in which the projects were developed throughout the semester.
- ▶ individual component (35%), whose main criterion is the personal performance of the student with respect to his or her development tasks as well as with respect to his or her personal role in the project.

Team Projects

Grading policy

Group component (65%)

60% Answer the customer stories and meeting the schedule according to the team time estimations:

(10%) for iteration 1

(25%) for iteration 2

(25%) for iteration 3

25% Project documentation

15% Group evaluation of the academic coach

Individual component (35%)

50% Weekly reflection

Pair programming experience

Test-Driven-Development exercise

Weekly presence

25% Performance of a personal role:

Actual implementation

Further development and enhancement

25% Personal evaluation of coach

Rubric for Software Projects

Example of an evaluation rubric for software projects developed by high school pupils individually

Topic	Max pts	Actual grade	Comments
Project documentation and organization	10		
Project code	10		
Project scope	30		
Knowledge about the project and its domain	30		
Extension and changes during the lab exam	20		
Total grade	100		

Portfolio

A portfolio is a purposeful collection of students works that tells the story about learners efforts, progress, achievement, and self-reflection on his or her learning process and progress, in one or more knowledge areas. Thus the portfolio can be a pedagogical tool that:

- ▶ Integrates learning with assessment.
- ▶ Creates a continuous communication and collaboration channel between teachers and learners.
- ▶ Provides a comprehensive view of learners achievements with respect to a variety of concepts.
- ▶ Enables learners to identify both their weaknesses and their strengths.
- ▶ Encourages learners to take responsibility on their learning process.
- ▶ Enhances learners reflective skills.

Portfolio

In the case of computer science education, it is relevant to create an online portfolio. A portfolio can include

- ▶ learners individual and group projects,
- ▶ intermediate versions of these projects,
- ▶ a description of the development process of these projects,
- ▶ peer reviews
- ▶ learners presentation of their work,
- ▶ teacher observations of learners learning process and tests.