

# Reinventing Evaluations with Competency Based Assessments: a Practical Experiment with Future Computer Science Engineers

**Sébastien Combéfis<sup>1,2</sup>**    **Guillaume de Moffarts<sup>2</sup>**

<sup>1</sup>ECAM Brussels Engineering School (ECAM)

<sup>2</sup>Computer Science and IT in Education ASBL (CSITeD)

October 23, 2020



[FIE 2020, Uppsala, Sweden]



This work is licensed under a Creative Commons Attribution – NonCommercial – NoDerivatives 4.0 International License.

# Context and Motivations

- Propose more fair and **better evaluations** to students
  - Reconcile students with the idea of assessment
  - Evaluations better adapted to students' learning profile
  - Possibility to take into account personal student's projects
- Better real-time **data collection** tool for instructors
  - Help teachers to adapt their teaching
  - Possibility to individualise/personalise the learning experience

# Competency Based Assessment

- Course defined with **basic and advanced** competencies

*Basic are mandatory to succeed the course*

- List of **assessments** to cover all the competencies

- Students choose the ones better fitting their learning profile
- Several types of assessments: MCQ, quiz, project, interview...

- Students prove they master competencies with **evaluations**

*Do not fail anymore, may just miss an opportunity to grab a star*

# The Stars System

- The proposed approach is supported by an **online tool**

*Students can track their progress, teachers can get statistics*

The screenshot displays the UDKA online tool interface. At the top, there is a dark header with 'UDKA beta' and a 'Courses' dropdown. Below this is a breadcrumb trail: 'Dashboard / Courses / PL190μ'. The main heading is 'Introduction to the Go Programming Language'. To the right of the heading is a blue 'Assessments' button. Below the heading, a paragraph describes the course as an introduction to the Go programming language, a compiled C-like language created by a team at Google, which provides excellent support for network, system, concurrent and distributed programming. The course is divided into two parts: Part 1 (Variable, Control Flow and Data Structure) and Part 2 (Structure, Method, Interface and Error Handling). To the right of the text is a 'Progress' widget showing two circular progress indicators: '15% Basic' and '0% Advanced'. Below the text are 'Resources' and 'Competencies' sections. The 'Competencies' section has tabs for 'Basic' and 'Advanced'. Under the 'Basic' tab, there is a table of competencies with their descriptions and star ratings.

Competency ID	Description	Rating
GP001	Correctly use the syntax of Go programming	☆☆☆☆☆
GP101	Write, compile and execute a single source file Go program with the command line	☆☆☆☆☆
GP002	Use basic built-in data structure: array, slice and map	☆☆☆☆☆
GP401	Understand basic Go compiler errors and warnings and fix the code accordingly	☆☆☆☆☆

# Experiment

- Pedagogical device has been **tested** on nine courses  
*Future computer engineers students (bachelor and master)*
- **Adaptations** since run in current school settings
  - No more exams nor revision break, continuous work
  - 0/20 is not all basic competencies,  $10 + 2x/n$  otherwise
- **Short survey** on  $n = 38$  students
  - Students globally happy to control their learning and evaluation
  - Organisational and logistics issues have been highlighted

# Conclusion and Future Work

- Bring **competency based assessment** in higher education  
*In a pragmatic and concrete way, supported by a tool*
- Several improvements to the pedagogical device as **future work**
  - Improvements and new features for the “*stars system*”
  - Deeper analyses of collected data and students' perception
  - Propose better design of competency based courses
  - Integrate with other tools for better logistics