Agility and Criterion-Referenced Assessment for Complex Competencies
Acquisition in Computer Science: Experiments with Master Students

This paper proposes an agile assessment process cycle which revolves around three main stages: analyzing the student's proposal, student's assessment using a criteria evaluation grid defined for each problem situation. The third step is a control point executed as follows: when the criteria are not satisfactory, the student can prepare a new proposal to improve his/her work with respect to unvested criteria and then overcome their specific challenges. The loop attempts stops when the semester ends. An experiment was conducted during the first semesters of four academic years with master students. The observed results show an improvement of the complex skills related to programming "authentic" computer applications in one hand and of the relationships between students and teachers in the other hand.

Index Terms— agility, criteria-based Assessment, complex skill, computer education.

I. INTRODUCTION

The computer education is based on the programming practice. In educational progression, exercise starts with the writing, individually or in pairs, of small programs implementing the algorithmic concepts taught. Then continue by writing in groups of at least two, more complex programs to solve problems situations (PS) raised by the professional world. In the first part, the evaluation of students' work is based on the verification of the acquisition of basic skills, while the second part needs an evaluation process of the acquisition of complex skills. Generally, the assessment of the PS resolution is a simple evaluation of the results: at the end of the development cycle, the students of a group present their solution that is evaluated according to the results produced by the application, given the features expected by the customer. In this context, the difficulties faced by the students, the way they overcome obstacles and the possible improvements of the work are not evaluated and discussed. In addition, they usually get an overall score for the group and for the whole project without precisions about their failure points. Hence, we don’t give the concerned students the possibility of getting better by learning how to improve software by overcoming the detected failures. The evaluation process in this case remains a simple scoring one since it doesn’t help competencies acquisition.

In addition, these evaluation methods have various problems. First, it is not because the basic skills are learned that students are able to implement complex skills necessary for problem solving situations: having the knowledge, the how to choose and how to combine multiple elementary skills to solve a new and complex situation requires interpretation of the situation and organization of the resolution process. Moreover, in assessing the problem situation, students who have encountered difficulties in the development of the project are somewhat penalized because they did not obtain the expected result even if
their developing processes are better. This type of assessment discourages students. We observed this problematic particularly with master 2 (M2) students from computer science department at Tizi Ouzou University. These students have difficulties when doing their final dissertation, where they have to develop a project with professional partners. The Analysis of these M2 students training pathways identified some of the reasons for the difficulties they encounter. During their previous training, including M1, teaching focuses on achieving practical works built around the acquisition of disciplinary basic skills and sometimes second-level skills. When the complex skills are addressed through PS proposed at the end of the license path, the performed evaluation is strictly summative for certification: it is limited to assign an overall score to a final presentation work, regardless of cognitive pathways taken by the student and all the obstacles he/she overcame. Moreover, this mode of evaluation does not consider the student in his singularity, including obliterating his/her personal skills and ignoring the psycho affective dimensions. So, arriving in M2, even if they are able to implement complex skills, students do not have access to the meta-cognitive reflection enabling them to identify the needs to mobilize. Faced with this problem, we tried to act from the M1 level with the professional master students in "engineering of information systems".

In this paper, we consider this problem in an environment where agile practices are in full shall develop and we put on the following research question: “Can agile principles combined with criterion-referenced assessment grids used in the evaluation of students’ solutions for authentic PS improve student skills?”

To resolve this question, we first propose a method for assessment using agile principles. This method is experienced during four successive years with respectively four different promotions. In this paper, we first present the conceptual frameworks that underpin our approach related to agile methods, criterion based assessment and the competency based approach. After that; we expose the conducted experiments, the results analysis and the main findings. We conclude with a discussion about the added value of the proposed approach and the possible research perspectives.

II. Conceptual Frameworks

We will successively present here the three main frameworks of our study that are: competency based approach pedagogy, agile methods and criterion-based assessment. After that we will give the study objectives.

A. Competency Based Approach

"In a competency based pedagogy, it is expected that students are able to mobilize, to transfer, to implement an organized set of knowledge, know-how and Know how to be that allow them to perform a task" [1]. Consequently, learning styles and assessment processes must evolve to meet the objectives imposed by the requirements of this pedagogy as well as the evolution of socio-professional needs boosted by technology and globalization. In [2], the
authors distinguish three levels of competence defined as follows:

- The first level of skills or "basic skills" which consist namely performing one or more identifiable operations without difficulty or ambiguity.
- The second level of skills or "basic skills with framing" that provide facing a new problem situation, the capacity of resolution by the selection and arrangement of basic skills and knowledge with the help of a "framing" where the situation is explained.
- The third level of skills or "complex skills" used to acquire the capability of selecting and combining several elementary or other skills to solve a complex and authentic PS. Here, the interpretation of the situation and organization of the resolution process are needed.

Programming complex skills assessment is a conceptual framework for our research while the matter of assessment with competency based approach is still a problematic. We aim to propose a way to tackle this challenge in computer science that can be generalized to other domains.

B. Agile Methods and education

Agile methods are computational methods that reduce the life cycle of the software and thus accelerate its development. The idea is not to go, as in traditional methods, by a sequential set of phases to realize in which development is carried out at the end of the project. This is to develop completely from the outset a first draft, evaluate then to complete in an incremental cycle. Each increment integrates new features to the precedent one with possible improvements in the development process (learning from errors corrected in the precedent step). A key point of this process is listening to the customers who explain their needs as the prototypes or parts of the software are delivered to him.

The origin of agile methods is related to the instability of technical environments and the fact that the customers are often unable to define their needs in a comprehensive manner from the beginning of the project. The term "agile" thus refers to the ability to adapt to changing circumstances and conditions occurring specification changes during the development process.

In 2001, seventeen (17) people have defined the agile manifesto [3] which is based on four values:

(I) Individuals and interactions over processes and individuals and interactions over processes and tools.
(ii) Operational software more than exhaustive documentation.
(iii) Collaboration with clients than contract negotiation.
(iv) Adapt to change more than following a plan.

A set of twelve (12) principles derived from these values (see [4]) for more details). For example, here as two principles among the twelve:

- "Our Highest priority is to Satisfy the customer through early and
continuous delivery of valuable software"

- "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."

In computing, there are several methods for the development of applications based on agile principles. We can cite SCRUM [5], RAD (Rapid Application Development) [6], UP (Unified Process) [7] and XP (Extreme Programming) [8].

If the concept of “agility” is now at the mature level with computer science software development, in education the concept is still at its beginning. Indeed, when at the beginning of our experiments, we didn’t found works about that, these last three years there are some ones about concepts such as “agile learning” or “agile teaching”. Thus, the main works about the introduction of agile concepts in education are as follows. Derek Bruff, introduced in his blog [9] and book [10] the concepts of “agile learning” and “agile teaching” with this expression: “When talking to faculty about their teaching, I often use the phrase “agile teaching” to describe a certain kind of on-the-fly responsiveness to student learning needs in the classroom”.

For Beatty et al, “Agile teaching” refers to an approach in which instructors may shift the direction or focus of a lesson based on the cognitive needs of the students [11]. Technically, some works propose methods to conduct agile teaching. For example, Michele S. Flint implemented formative assessment using i-clickers [12].

Some researchers such as Rodulfo J. Prieto from Laboratoria, spain, think that agility is a culture: “Agile is not a methodology; it’s a way of behaving, it’s a culture, it’s a mindset”. [13]. For Laboratoria institution “The beauty of Agile being a culture is that you can apply it to any organization, and to any aspect of work. And that’s precisely what we’ve ventured out to do at Laboratoria. We are applying Agile values and principles to education”.

We also found an emergent group [14] who introduces a collaborative site to work about “agility in education” in one hand and the eduScrum [15] method in the other hand. Hence eduScrum is an adaptation of the known SCRUM method in software engineering to the education context. We noted that there are fewer works about methods using agility to conduct assessment and scaffolding for complex competencies acquisition.

In this article, it is the agile principles that we address, not the design methods used by computer science students as part of their project so for software development. These include introducing agility in mini projects evaluation by iterations on the revaluation of unveested reworked criteria. It is an “incremental” process where the students build their software by continual improvements from a step to another guided by the teacher assessment using the evaluation grid. The agile manifest is someway adapted to the conduct, by the teacher, of the “assessment project” where the students have the role of “customers”. This is on one hand; on the other hand, students are encouraged to adopt the agile principles in the development of their respective applications
required by the proposed PS.

C. Criteria-Based Assessment

Classically and practically two main types of evaluation exist: formative assessment and summative assessment. The first is performed during the activity. It aims to assess the progress made and to understand the nature of the difficulties encountered by students. It improves, corrects or adjusts the cognitive development of students. The second type is the summative evaluation: after a fixed period and on the basis of an evaluation test, it measures the acquisition of knowledge and know-how. Competency based approach supports formative assessment because it promotes interaction through a student-teacher dialogue.

If the evaluation activity is generally seen as a final decision taken on the basis of data collected, formative assessments has the aim to exploit the observations made during the evaluation process by comparing them to "external standard" or what is called evaluation referential. In this paper, we are particularly interested by the normative criterion-referenced assessment that is a kind of formative assessment.

With the normative assessment, the results are compared to a standard, to a reference (e.g., the average of a group, a performance level, a minimum or maximum, etc.). In criterion-referenced assessment, these are the different criteria that are standardized by assigning them respectively value scales.

One of the main tools of criterion-referenced assessment is the evaluation grid. This is to establish a correspondence between, the result of a task that is required of a learner (especially a complex PS) and a quality criteria of this result associated with a scale performance of several degrees (an example of an evaluation grid is given in point 3 of this paper). According to Mueller [16], for each grid, it is necessary to specify the evaluation criteria, performance levels, a scoring mechanism, a degree of importance to each criterion for the overall evaluation and possibly descriptors that guide the allocation of the performance value to each learner.

In our research work, we start from the assumption that the evaluative activity is a phenomenon capable of "activating" teaching while personalizing it. Indeed, as part of the competency based approach, assessment exceeds the role of a single certification built around a summative evaluation. This activity, to be credible and useful, as emphasized Anne Jorro [17], is characterized by its components: gestures, postures and evaluator’ skills. We are here interested to explore the formative criterion-referenced assessment for the evaluation of complex skills for the case of computer science domain.

D. Objectives of the study

In this study, we focus on a type of formative evaluation of complex PS resolution in computer science domain. This is a normative criterion-referenced assessment of complex skills based on agile principles. This evaluation was initially conducted in a "natural" or "instinctive" way in order to initiate the M1 computer science students to solve complex PS. It is the limitations encountered with the classical evaluation way cited above that led us to
abandon the traditional rating system and to imagine new approaches where the
mark becomes, as shown by Claire Tournem [18], a judgment gradually built.
Here the proposed approach follows an agile evaluation process that is running
with an agile development cycle of the student software.

The dual use of agile principles (by the teacher when assessing and by
students when developing their projects) brought us out these following
research questions:

Q1: Does the introduction of PS resolution in training improve the students’
skills?
Q2: Was the used agile assessment process easily executed by teachers and
students?
Q3: What are the factors to improve to get better results with the used
evaluation process?

Through Q1 we investigate whether the students, after the completion of
the PS in the scope of the proposed evaluation process, feel more able to
develop computer applications. The question Q2, subdivided into two sub-
questions will tell us if the agility of the process has facilitated the PS
resolution for students (through Q21) and made more relevant and effective the
assessment for teachers (through Q22). The third question Q3 is a way to get
during the experiments suggestions about possible improvements of the
process.

In the following, we will describe the experiment conducted. After that, a
synthesis about the findings is given before concluding and discussing the
research perspectives inspired by this work.

III. EXPERIMENT DESCRIPTION

We describe here the experiment conducted for four successive years by
giving its context, the pedagogical engineering, the agile assessment process
and in the end the corpus and methodology.

A. Context

The experiment was conducted with the Master 1 (M1) students from
computer science department of Tizi Ouzou University, Algeria. The students
are given, each year, to solve a complex and authentic problem situation (PS)
with subjects inspired from socio-professional needs. The idea is to offer
students authentic application development projects. A project-based teaching
has been implemented: students have to develop specific software within a
specified time and respecting the agile principles.
The objective is to assess the basic skills acquisition during years of prior training in a "complex" setting with an agile assessment adapted to the identification of third-level skills. Figure 1 shows an example of such PS topic given for M1 students of 2014-2015 academic year where students have to develop an application with some expressed needs using some given tools.

**B. Instructional Design**

During the first session of the course, the teacher introduces the topic of the mini-project, forms the groups and put on an indicative timetable. Students seize on the topic, conduct research to find the necessary and adapted tools (software, documentation, etc.) and begin programming. The group is supposed to produce a report about the project, an oral presentation a software prototype. At the time of the checkpoint date specified in the schedule, students make an oral presentation and a demonstration of their software to their teacher. This last assesses and provides personalized recommendations to the group to improve the work for the next iteration.
Students are required to cycle if they want to improve their respective works and avoid the failures detected with some criteria. The loop attempts, as shown by Figure 2, stops when the semester ends.

C. Assessment Process

The mini-project monitoring approach has three main steps: analysis of the student's proposal, assessment guided by a criteria-based evaluation grid and the proposition of improvements according to the weaknesses detected. The grid of criterion-referenced assessment is normative. For each criterion there are different level descriptors with values possible domain. The normative criterion-referenced assessment grid used in this experiment is analytical [16]. Table 1 presents the evaluation grid used for the proposed 2014-2015 PS illustrated in Figure 1. Each row of the table corresponds to an evaluation criterion for assessing the response of the learner. For each row, we have three level descriptors (insufficient, good and excellent). Each descriptor has a value interval that guides the teacher to put an estimated mark for a criterion.

The objective of the first round evaluation is to enable students to identify not yet acquired skills represented by criteria of the grid. For teachers, the criteria are used to answer to this question: what will I look at when I evaluate the work done by the learners? For students, the criteria are used to answer the question: what are the skills I need to develop to get a good performance?

Corpus and Methodology

The experiment was conducted in the previous four academic years, in 2011-2012, 2012-2013, 2013-2014 and 2014-2015 in the computer science department of Mouloud Mammeri University of Tizi Ouzou in Algeria.

To evaluate this experiment, we sent students a survey to measure their satisfaction with both the content and form of this novelty in “teaching” of complex problem situations. Meanwhile, we gathered informally feedbacks...
from teachers involved in the supervision of dissertations studies of M2
students (next master year after the one where an experiment is conducted).
Finally, we have maintained for each academic year, the evolution of the
students’ marks after each attempt, final reports and prototypes made by
students. Our analysis is based on a triangulation of data got from these
different sources for each promotion during the four years covered by this
paper.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation of Report and Presentation (a)</td>
<td></td>
</tr>
<tr>
<td>Criteria</td>
<td>Insufficient (level 1)</td>
</tr>
<tr>
<td>Structure: introduction, conclusion, bibliography.</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Synthesis on the Agile Approach</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Understanding of Junit</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Presentation of the implemented application</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Presentation of the programming environment</td>
<td>0</td>
</tr>
<tr>
<td>Evaluation of the developed application (b)</td>
<td></td>
</tr>
<tr>
<td>Criteria</td>
<td>Insufficient (level 1)</td>
</tr>
<tr>
<td>Proficiency in the chosen Java development environment</td>
<td>0</td>
</tr>
<tr>
<td>Implementation of the sentence comparator (similarity measure)</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Creation and use of the sentences database</td>
<td>0-0.5</td>
</tr>
<tr>
<td>Implementing /</td>
<td>0-0.5</td>
</tr>
</tbody>
</table>
IV. EXPERIMENT RESULTS

We collected data and performed some calculations to get the results summarized in the following for each of the three research questions considered in this study.

A. About Q1

Our interested here is whether the introduction of the PS and its evaluation mode has improved the students' skills. We gathered on one hand the students' answers to the direct question:

"Do you feel better in application development after the completion of the PS?"

The proposed possible answers are: “surely”, “a bit” and “no”. Table 2 provides a summary of the results for each year (the first four lines of the table) and the last line gives the calculated average for each possible answer about four years.

**TABLE 2. RESULTS SYNTHESIS FOR Q1**

<table>
<thead>
<tr>
<th></th>
<th>Surely</th>
<th>A bit</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-2012 (29 students and 14 groups)</td>
<td>93.1%</td>
<td>06.9%</td>
<td>0%</td>
</tr>
<tr>
<td>2012-2013 (31 students and 17 groups)</td>
<td>77.41%</td>
<td>12.9%</td>
<td>09.69%</td>
</tr>
<tr>
<td>2013-2014 (34 students and 16 groups)</td>
<td>82.40%</td>
<td>11.7%</td>
<td>05.9%</td>
</tr>
<tr>
<td>2014-2015 (48 students 18 groups)</td>
<td>94.10%</td>
<td>04.16%</td>
<td>01.74%</td>
</tr>
<tr>
<td>Average for four years</td>
<td>86.75%</td>
<td>08.91%</td>
<td>04.33%</td>
</tr>
</tbody>
</table>

We observe here that the majority is for the "surely" response. We also note differences in the percentages from a promotion to another. Table 3 gives some parameters whose combination may explain the rates indicated by Table
2. For example in 2012-2013, the groups had almost no collaboration, which caused that the PS has not been fully resolved by all. This may justify the increase in the number of iterations to improve their outcomes. Indeed, it is at the first presentation session that each group discovers what others have done and generally we noted that when one criterion is resolved by one group, all the others are motivated to try doing it on a further iteration.

We also noted by analyzing the cases that have answered "no" respectively in the years 2013 and 2014, that they were all not involved in the programming process. Indeed some have simply contributed with a theoretical aspect and the others not at all involved and participated in the presentation thanks to the help of their comrades.

TABLE 3. ADDITIONAL RESULTS FOR Q1

<table>
<thead>
<tr>
<th>Year</th>
<th>Average Number of Iterations (ANI)</th>
<th>Number of defaulters</th>
<th>Specificities of the training group</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-2012</td>
<td>1.92</td>
<td>0%</td>
<td>High cohesion and spirit of collaboration</td>
</tr>
<tr>
<td>(29 students</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groups)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2012-2013</td>
<td>2.29</td>
<td>3%</td>
<td>rather low level of cooperation, more monomials groups</td>
</tr>
<tr>
<td>(31 students</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and 17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groups)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-2014</td>
<td>2.33</td>
<td>5%</td>
<td>Level of collaboration rather low</td>
</tr>
<tr>
<td>(34 students</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and 17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groups)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014-2015</td>
<td>2.66</td>
<td>13%</td>
<td>Collaboration average, larger promotion and then groups can contain from 4 students (to avoid the help of another teacher and risk the disturb of the assessment process)</td>
</tr>
<tr>
<td>(48 students</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and 18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groups)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average for 2.3</td>
<td>05.25%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>four years</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ANI=$\sum_{j=1}^{nbG} nbjf/nbG$ (*Formula* (1))

Table 3 gives the average number of iteration (ANI) of each year. The parameter ANI is calculated with formula (1) where “nbG” is the number of...
groups, “nbIj” is the iteration number of a given group j. We noted that ANI is growing from one year to another. As the average number of iterations over the four years was 2.3, on average each student made at least two attempts. We also have the number of defaulters (students who did not do the PS) that appears proportional to the number of students in the promotion.

For Q1, we also captured a rather informal (questions / verbal responses) opinions of teachers who were supervisors of M2 end-studies projects of those students from M1 concerned by our experiment. We chose 18 teachers already all supervisors with promotions before our experiments to distinguish the possible skills evolution. About 89% of them (or 16 out of 18) thought that the introduction of the PS resolution in M1 training improves the students’ skills. The two who gave negative responses have mentored students who didn’t do the PS (defaulters or not involved in their respective groups). These Teachers also noted a certain "maturity" in mini project management: deadlines’ management, selection of collaborators, report writing, presentation capacity; choose of programming environment etc.

**B. About Q2**

The question Q2, subdivided into two sub-questions will tell us if the agility of the process has facilitated the PS resolution for students (Q21) and made more relevant and effective the evaluation process for teachers (Q22).

Q21, which is a multiple choice question, is as follows:

"The agility of the monitoring process (including discussions at the end of each course, the criteria grid and iteration possibility) allowed you to:

a-have more time to finish the PS
b-address the unmet criterion and improve your ability to develop computer applications
c-regulate your solution compared to the remarks and the criteria in the grid
d-have a better mark
e-remove some obstacles, seeing your classmates able to solve the same criterion that you did not succeed"

The subjects’ responses are summarized in Table 4 where we see that all students from the sample have chosen the responses (a) and (d). The lowest scores are for the response (c). For some students who didn’t choose the response (c), the sentence expressing this last in the survey, wasn’t enough clear for them.

**TABLE 4. RESULTS SYNTHESIS FOR Q21**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-2012 (29 students)</td>
<td>100%</td>
<td>69%</td>
<td>65.5%</td>
<td>100%</td>
<td>86.2%</td>
</tr>
<tr>
<td>2012-</td>
<td>100%</td>
<td>71%</td>
<td>74.2%</td>
<td>100%</td>
<td>83.8%</td>
</tr>
<tr>
<td>Year</td>
<td>Students</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>2013-2014</td>
<td>34</td>
<td>100</td>
<td>76.5</td>
<td>70.6</td>
<td>100</td>
</tr>
<tr>
<td>2014-2015</td>
<td>48</td>
<td>100</td>
<td>79</td>
<td>62.5</td>
<td>100</td>
</tr>
<tr>
<td>Average for four years</td>
<td></td>
<td>100</td>
<td>73.8</td>
<td>68.2</td>
<td>100</td>
</tr>
</tbody>
</table>

For teachers, the question Q22 is also a multiple choice one that is as follows: "The agility of the monitoring process (including discussions at the end of the course, the criteria grid and iterations):

A. Allowed you to follow the evolution of the students and detect better their obstacles.
B. Made you more objective and "just" in the assessment and grading of students.
C. Gave you more dynamism and enthusiasm when working.
D. Gave you more work (at least double that the conventional so the non-agile mode).
E. Allowed you to get a differential track of your students”.

Table 5 shows the results of statistics concerning question Q22. We can see that the responses A, D and E got the complete score (100%). However, four (4) among the twenty teachers didn't choose the response B. The main comment they all gave is: “The evaluation is better than with the classical assessment mode, however it is closely related to the competencies (technical skills, objectivity …) of the teacher who conducts the process”. Also, two (2) teachers didn’t choose response C. The main justification they gave is: “It depends on the teacher’s reactions and assessment qualities; otherwise the process can generate stress on both sides”.

We just note here that the twenty (20) teachers who participated in the assessment process along the four years were volunteers. They were each year surprised by the time necessary for real assessment of the work of each group particularly the review of developed prototypes respective programs. They also noted the “difficulty” for them to differentiate between students of the same group. Hence, after four years, all teachers thought that the additional work especially without any remuneration or particular grants is tiring. This fact poses the problematic of teachers’ motivation to go over problem based teaching and fulfill real assessments to get real differentiation/personalization of teaching/learning.
TABLE 5. RESULTS SYNTHESIS FOR Q22

<table>
<thead>
<tr>
<th>Responses</th>
<th>Percentage of teachers (among a population of 20) consulted during the four academic years</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100%</td>
</tr>
<tr>
<td>B</td>
<td>80%</td>
</tr>
<tr>
<td>C</td>
<td>90%</td>
</tr>
<tr>
<td>D</td>
<td>100%</td>
</tr>
<tr>
<td>E</td>
<td>100%</td>
</tr>
</tbody>
</table>

C. About Q3

Here we are looking about suggestions to improve the proposed and used evaluation process. Some factors are proposed to the subjects from the two sides represented by: teachers and students from the computer science department all involved in computer science teaching and/or learning and implicated in our different experiments. We proposed particularly Q3 as follows: “To make the process more efficient on both sides (students and teachers) to acquire complex skills, what to do?”

We collected different suggestions, the most significant and recurrent ones are as follows:

- More close and specialized tutoring from systems and/or programming experts especially to avoid some technical obstacles such as: program installation, machine configuration, tools portability, execution bugs ...)
- Propose the same teaching process during the third license training.
- Propose an individual work
- Impose iterations for those who failed from students.
- Apply the teaching way to acquire other kind of competencies such as foreign languages.
- Apply the same process for end studies projects for license and master trainings.
- Remunerate or give special grants for teachers who practice this kind of assessment
- …

V. DISCUSSION AND FINDINGS

The results of this experiment are viewed below from different dimensions as follows.

A. Agile methods in education

The didactic transposition of some agile principles gave good results. In the mini-projects developed by the students, the teacher plays the role of client. His satisfaction is measured in relation to all the criteria defined in the
evaluation grid and not only to the expected software features. These criteria include a range of pedagogical elements to define students’ skills indicators. For example, a development can meet all the functionalities required but if the developed code doesn’t meet the constraints imposed by criteria of the evaluation grid, for example the use of JAVA Servlet class type in Java programming, the teacher will not validate the corresponding criteria and consequently the developed application.

As result, the finding here is a possible adaptation of agile methods such as SCRUM to teaching achievements.

B. The role of criterion grids in learning regulation

The evaluation grid plays a fundamental role in the process. For one, it allows teachers who evaluate the work to standardize the results, to clearly identify the shortcomings of each student (differentiated evaluation) and thus to develop, given the expected competencies, a prioritized list of recommendations. Students in turn, have a sufficiently detailed feedback in order to develop by themselves a new strategy to improve their work. Finally, in the next iteration, teachers as students can finely measure the evolution of skills. The criterion-referenced assessment grids as well as the list of recommendations, therefore, become a tool for improving the production of computer programs and of complex competences in general.

C. The impact on the student progression

Individualized and adaptive management of student attempts showed very encouraging results: the majority of students were able in the end of the iterations to validate the project by developing the application requested respecting the requirements presented in the evaluation grid and the deadline. Moreover, we see through the various iterations, an improvement of skills. This is due among other things to the fact that only students who wish, continuing the iterations fulfill the process. We noted that those who are satisfied by their respective marks are not always very motivated to continue the project. However, in some cases, these lasts choose to continue to get better results to their peers.

We also found the particular case of students in a situation of failure (got bad marks with exams) in the context of knowledge and basic skills evaluation. It turns out that some of them, who did easily with smartness the project, are through the resolution of this complex SP, to bounce back in training. The self-taught nature of this activity allows them to be aware of their third level skills and allows them to get new motivation for training.

D. The impact on teachers and teaching context

Since the beginning of this experiment, teachers of M2 find that students are more comfortable in making their final thesis study and better in programming than the previous promotions.

We also found that from the social point of view, the evaluation process has brought the teacher to his/her students, close them from each other and
despite the additional efforts of both sides, all believe that the new learning environment created with this experiment is more "friendly".

E. The proposed process and socio-constructivism

The proposed process that helps students to acquire complex competencies is going with the socio-constructivist theory. Hence, the process is based on one hand on the teacher’s scaffolding and in the other hand on the students interactions each other, with teachers and with other social partners to get help when developing the project programs. For that we observed in the results of the different experiments a correlation between the social behavior of students and the acquisition of the different criteria, then on the improvement of computer science competencies. The proposed process is then a socio-constructivist one which is adequate to computer science complex skills acquisition that is a constantly evolving field.

VI. CONCLUSION

Training for acquiring complex skills, equipped with a normative criterion-formative evaluation process, can target the professionalization of learners framing the problem situations resolution process. This framework consists in our case to manage attempts to obtain a report, a presentation and a prototype as result of a resolution of the PS. This assessment process, guided by an evaluation grid with different criteria is well suited to the development of computer science complex competencies. The move towards a new attempt is well accompanied by the precision of the weaknesses and the criteria to rework.

Our experimentations along four academic years reported here showed interesting results regarding to the sensible improvement of programming skills and project management ones among the students subjects of our experiments. This agile way used to scaffold students enhanced the quality of the training and gives a dynamic and motivation to acquire complex competencies in computer science program development. The main potential constraint that can be an obstacle for the use of the proposed approach and method is the number of students. Indeed, for the amount of work generated by the process since it is in the context of learner oriented pedagogy, the management and valorization of the teachers’ additional work is necessary.

Following our work presented in this paper, we plan to work on experimentations using more agile methods principles like with SCRUM method for the complex problem situation in different areas of computer science. Particularly, we will study eduScrum and the possibilities to contribute to its development in the case of competency based pedagogies. We are also actually working on an assessment aided tool to support the process exposed in this paper. Hence, it will be possible to teachers to reuse existing criteria and execute different automated tasks such as marks calculus, candidates to future iteration selection, assessment synthesis … to help them on the evaluation grids and the learning process management.
REFERENCES


