Assessing Teaching Methods in Programming
A Case Study: Salahaddin University

Tarik A. Rashid
Software Engineering Department
College of Engineering
Salahaddin University
Hawler, Kurdistan
E-mail address:tarikrashid4@yahoo.com

Rana F. El Farra
Science Department
College of Education
SABIS University
Hawler, Kurdistan
E-mail address:relfarra@sabis.net

Abstract— Several teaching methods are practiced at the Software Engineering Department in Salahaddin University, Hawler. This paper is a review of some programming teaching methods and cooperative teaching methods based on Biggs four levels of thinking and teaching. The paper proposes applicable examples of Problem Based Learning (PBL)/assessment approach and concept mapping. It is also an assessment of the current teaching methods at the department using Biggs’s perspective. Focus group interviews and structured questionnaires show that while the first and second levels of teaching are applied in teaching the first and second year students, the third and fourth year students remain at the third level of thinking and teaching. The paper identifies independent attempts to reach the fourth level of thinking and teaching by competent students. The paper suggests that lecturers can prepare students for this transition by introducing more constructively aligned activities and motivating the students to take control. Lack of adequate teamwork skills among students is another challenge that should be incorporated in all courses offered at the department. Involving students in assessment design, implementation and evaluation is strongly recommended strategy. The main purpose of this study is to help prepare students for real work experience.

Keywords-Teaching Methods; Level of thinking;

I. INTRODUCTION

The knowledge and skills of education systems are continuously challenged by rapid and dynamic global changes. This requires education systems to prepare students with excellent knowledge, critical thinking and communication skills, pliability, and malleability to maintain efficient retention of knowledge and life-long learning skills [1].

There are two main factors that control the efficiency of learning and teaching: (1) competent and up-to-date curricula, (2) productive and proactive lecturers with successful teaching skills.

Many educational research projects, collaborations and consortiums have extensively studied and proposed various strategies to provide proficient students, curricula, lecturers and instructional methods.

In 1997, Kember identified two general approaches in teaching: lecturer centered/content approach and student centered/content approach [2].

A. Lecturer-oriented approach

The lecturer centered/content approach focuses on passing the knowledge from the lecturer to the student. The role of students is to memorize patches of knowledge and recall them on exam papers. Accordingly, the students will not be able to independently assimilate information.

The strategies of the lecturer-oriented approach include lecturing, questioning and demonstration. This approach dominated universities and schools all around the world for a long time. It is a common practice in several Middle Eastern and Asian universities, particularly in humanities subjects and in large classrooms.

The lecturer-oriented approach may be necessary at introductory level courses, particularly when teaching large groups. This may provide basic and essential knowledge preceding other approaches.

Yet, in our rapidly evolving information world, the lecturer-oriented approach has proven inadequate for many students and subjects such as engineering, science, mathematics and many more. Pre-defining, dictating and limiting students’ knowledge will restrict what the students learn, limit their horizons and creative initiatives. This approach does not help students acquire proper learning skills.

The main challenge and most essential issue that faces the learning process is to transform the students from being passive recipients to active preceptors that have a choice in not only what to learn but also in how and why [3].

A guided transition from a strict lecturer-oriented approach to a student-oriented approach may provide a potential answer to this challenge.
B. Student-oriented approach

The student centered/content approach places the student at the center. The focus is what the student does rather than what the lecturer does [4].

The theoretical standing of the student-oriented approach essentially relates to the constructive view of learning which focuses on activity, discovery, choice and independent learning [5].

This approach considers learning as the full responsibility of the learner (student), encourages students’ involvement and participation, and promotes growth and development by providing an equal relationship between students. The lecturer becomes a facilitator and the resource person, the student experiences the flow of cognitive and affective domains together which results in the student viewing him/herself differently due to the learning process [6] [5] [7].

In the student-oriented approach, the emphasis is on deep learning and understanding. Students develop an increased sense of autonomy and interdependence [8].

While, Kember’s definition can be limited, other education researchers have proposed more detailed approaches that can help assess various teaching methods, detect weaknesses and strengths points and introduce innovative solutions.

The next section is a brief discussion of the four levels of thinking and teaching proposed by Biggs [9].

II. FOUR LEVELS OF THINKING AND TEACHING

Biggs [9] provides a more detailed approach to fully understand and categorize what is appropriate and what is not to reach an effective teaching/learning process. He strongly shares the academics opinion that education is not just about acquisition of knowledge but also the ability to develop life-long learning skills and the ability to apply knowledge in the workplace and throughout all aspects of life [9].

Biggs approach includes the following four levels of thinking and teaching:

A. Level 1: What the student is.
B. Level 2: What the lecturer does.
C. Level 3: What the student does.
D. Level 4: How the student manages what he does.

A. Level 1: What the student is

This level defines the roles and responsibilities of both the lecturer and the student. While the lecturer is responsible for collecting and presenting knowledge, learning is the sole responsibility of the student. The responsibility of motivation lies completely on the student. The failure of the learning ‘teaching process is due to lack of proper students’ learning skills and talents, lack of sense of responsibility, lack of self-motivation as well as low university-entry standards.

According to this level, lecturers define what students learn. Lecturer should provide theoretical backgrounds and explanations of all delivered knowledge. The lecturer should focus on what he/she wants the student to learn. This approach is close to the lecturer-oriented approach proposed by Kember. Biggs describe this level as deficit or incomplete [9].

B. Level 2: What the lecturer does

This level states that the success of the learning process is the sole responsibility of the lecturer. The academic success of the student is achieved by completing activities designed by the lecturer. In this level learning in class depends mainly on the activities, sense of responsibility, and lecturing and demonstration skills of the lecturer.

Biggs also describes this level as deficit or incomplete. He believes that successful teaching should not be assessed by acquiring teaching skills only, but also by deploying such skills to ensure an efficient learning process [9].

C. Level 3: What the student does

This level views teaching as a learning support. Teaching, at this level, is only effective if both the learning outcome and the planned learning activity are aligned properly. This level can be related to level 2 by successful deployment of effective learning skills to create a strongly engaging environment [10].

According to Biggs, this is the active learning level that is linked to constructive alignment. Shuell defined constructive alignment as designing a teaching activity which achieves the expected learning outcomes without using traditional teaching methods [11].

Several teaching strategies that can improve both the lecturers’ and the students’ performance at this level have been proposed by the University of Glasgow. For example guided practicals, group discussions, field trips, computer assisted learning, writing newspaper articles and poster presentations [12].

D. Level 4: How the student manages what the student does

At this level, students take control. It is essential that the students would start at an affective framework within level 3 before they proceed to this level. The students at this level are responsible for their own learning. This involves intensive creative thinking and problem solving skills. It is the level at which students are prepared for real life experiences and expectations.

Suggested teaching strategies at this level could be peer mentoring, independent projects, pyramids/snowballing, debates, reflective diaries and learning journals, role play, choice in subjects for study/projects and production of mind maps [12].

Involving students in assessment design, implementation and evaluation is another strategy at this level. It allows students to develop some autonomy and decision-making in learning and assessment. Gibbs describes the involvement of students to range between setting the criteria and standards, deciding how the judgments are made and by whom [13].
a more specified approach, Brown, Rust and Gibbs have divided the assessment process to involve the student into two stages: (1) Stage one: when the task is set and (2) Stage two after the task is completed [13]. The first stage includes choosing and setting the assessment task, discussing the assessment criteria and setting the assessment criteria.

The second stage may include self-assessment, peer-assessment feedback, suggesting, negotiating and assigning both self and peer assessment grades [13].

The following section summarizes suggested programming teaching methods within Biggs levels of thinking and teaching.

III. PROGRAMMING TEACHING METHODS

A. Several language programming teaching methods

Szlávi P., Zsakó in [14] identifies several programming teaching methods. He believes that applying a single method or mixing different methods is a sole choice of the lecturer and is controlled by how creative the lecturer is. These methods include:

1) Statement-oriented: in this method the lecturer presents the language as a set of individual elements in a certain order that student should follow. This approach can be classified under the first level of thinking and teaching.

2) Software technology-oriented: a programming language teaching method where learning is achieved by developing a software methodology. This teaching method can be both applied at the third and fourth levels of thinking and teaching.

3) Task type-oriented: This is a problem solving based approach. The programming language is taught through a certain task defined by the lecturer. The lecturer can initiate this approach at the third level. Upon effective and continuous practice, the students can take control towards the fourth level of thinking and teaching.

4) Language-oriented: This programming language teaching method starts at the logic and structural units of the language. Ideally, the lecturer should present the knowledge and allow the students to practice through practical tutorials. This method is applicable at the first and second levels of thinking and teaching.

5) Action-oriented: The main aim of this method is to enable students to envisage what happens when the statements are being executed. This is a third level approach that stimulates students towards a fourth level of thinking and teaching.

6) Sample task-based method relies on presenting and analyzing many examples that enable students to acquire a programming language well. This approach fits within the first or second level of thinking and teaching.

B. Problem based learning in computer programming

Problem based learning (PBL) is one approach that can help both the lecturer and the students to successfully shift from the third level to the fourth level of thinking and teaching. It was introduced through several examples by Ramsden and Peter Cawly [15][16]. According to Duch in [17] PBL instruction in computer programming enhances the students’ abilities to analyze and solve complex problems, work cooperatively with their peers and develop effective verbal and written communication skills. Such skills are essential to help prepare the students for the real work life experience [17]. Tan in [18] has proposed the PBL model which transforms the content or curricula into a problem, the lecturer into a coach and the student into a problem solver [19].

The problem suggested by authors of this papers, is to develop an online college examination software. The design of the problem involved constructive alignment to provide a general frame work for the students to start from.
The lecturer at the second level may start with introductory lectures. The lecturer guides the students through the third level by guided practicals. The students are grouped in pairs. One student being the academic representative and the other is the software developer. The students at this level can move to the fourth level of thinking at which they practice skills of mind map and independent project to design and develop the required software. At the fourth level, the lecturer’s role is limited to a facilitator.

Assessment is an essential part of the PBL and thinking level combined approach. The assessment starts with the lecturer at the second level of thinking. The lecturer can set the criteria and standards of assessment, how and who judges the project’s outcome. The lecturer at this level can request oral presentations and set standard exams.

The lecturer then can guide the students in self-assessment to evaluate the efficiency and effectiveness of the developed software. The lecturer can also encourage peer-assessment and feedback from the academic representative. The learning outcomes are achieved by setting the problem, guiding the teaching/learning methods and allowing different methods of assessments.

Figure 2, presents a summary that combines PBL, assessment methods and levels of thinking and teaching approaches using the online college examination system.

![Concept Mapping Diagram]

**Figure 2:** Combines the PBL, assessment and levels of thinking approach

### C. Concept mapping

Concept maps are conceptual graphs that are used to visualize the relationships between different concepts. The main concepts are interconnected by nodes which represent the relationships between them [21] [1].

Concept mapping can not only be used to communicate complex ideas but also to generate them and aid in building up on existing knowledge.

Lecturers at the second level of thinking and teaching use concept maps to introduce and summarize new concepts and provide a comprehensive picture of the expected learning outcome. At the third level lecturers can use the concept mapping technique as a guideline to add new knowledge to existing knowledge. The students at the fourth level can develop their own concept map to represent a developed software or program.

![Concept Mapping Diagram]

**Figure 3:** Shows a rough concept mapping

Figure 3, represents the use of concept mapping at the second level of thinking and teaching. The lecturer generates a rough diagram that includes the initial and general picture. At the third level, as shown in Figure 4, the lecturer continues to add concepts to the map as the teaching progresses. The lecturer provides a useful tool which
strengthens critical thinking, organization and comprehensive skills of his/her students.

Concept mapping can also be successfully applied at the fourth level of thinking and teaching by allowing the students the independence of interconnecting concepts. Computer programming students can use a concept map to propose initial ideas and modify them upon testing them.

David and Rojer Johnson define the five elements of cooperative learning that ensure its effectiveness. The five elements include positive interdependence, face-to-face interaction, individual and group accountability, interpersonal and small group skills and group processing [23]. Christdusan suggests an array of strategies that help the lecturer guide and facilitate successful cooperative learning such as buzz, affinity, solution and critique groups, teach-write-discuss, role play, debates, case studies and integrated projects [24]. Gehringer [22] strongly argues that classroom should be devoted to group exercises and assignments devised for students to actively participate in educating others [22]. This study analyses the above mentioned practices of cooperative learning.

Cooperative learning can be started at the third level of thinking and teaching. The lecturer can initiate activities by intellectual scaffolding. The lecturer selects the discussion topics, raise questions/issues and devise collaborative processes to allow all group members to participate actively [24].

Upon adequate guidance, practice and assessment, the lecturer can promote the students to the fourth level of thinking and teaching. At this level, the students can suggest their own projects, formulate their own questions, discuss issues, present and explain their own viewpoints and work as independent teams on problems and projects.

A. Team projects

Corporate recruiters seek individuals with good team skills. To prepare students for successful work experience, computer science university educators should not only provide the students with knowledge but also help build and enhance their team work skills. There is considerable evidence that students working in teams develop deeper understanding and stronger retention. Group work has also proven to help struggling students [22].

B. Peer tutoring

Peer tutoring is the system of instruction in which learners help each other and learn by teaching [25]. It is successful means of active learning that motivates tutors to learn before they teach. It improves communication, leadership and interpersonal skills of both tutors and tutees. It allows struggling students to receive individual attention and stronger responsibility and involvement in the learning process [26]. Peer tutoring challenges tutors to modify presented knowledge according to the level of tutees. Peer tutoring can be practiced both inside and outside class where the tutors are the most competent students in the cohort.

The Software Engineering Department in SU does not formally include peer tutoring as part of its curricula. Peer tutoring is practiced as an unguided effort that is initiated by the students.

V. CURRENT SITUATION

Universities in Kurdistan have witnessed numerous transformations in their educational systems due to the rapid
development in information technology and networking systems. Yet, the change and improvement in teaching methods particularly, in programming subjects requires intensive assessment and analysis.

SU, Hawler, was established in 1983. It provides wide range of specialties. The university offers different courses in engineering, science, education, literature, business and economics. The focus of this paper is the software engineering program. The program consists of a four year plan that covers a broad range of subjects that starts at an introductory level and ends at the advanced level.

The software engineering program is pre-defined and approved. Students do not choose subjects. The offered subjects are updated according to the popularity and dynamicity of programming language in the market and research.

Programming subjects are taught as two-hour lectures in class and two-hour tutorials in the computer laboratory per week. The number of students per lecture is not less than sixty. The laboratory facilities accommodate thirty students on average.

The tutorials are designed to target short activities that should be completed within a certain time frame. Students are not encouraged to work in groups. Competition is the main theme. Programming subjects’ tutorials usually focus on practicing certain lecture-oriented concepts rather than working on projects to develop systems.

During the first and the second years students are given individual compulsory tasks as part of their practical marks according to their results. Additional projects are optional, only for those who are interested to communicate directly with the course lecturer. Extra bonus mark will be given according to what is submitted. This type of activity does not encourage students to work as a team.

At the fourth year, students are required to independently develop their own final projects. Almost one fourth of the students are capable of working independently.

In general, the department adapts a lecturer-centered approach. Demonstrations and programs are shown to the students in lectures rooms. Students will be given enough time in labs to ask questions and where they are guided to complete activities and tasks. The students sit for midterms and final exams.

This paper is a pilot assessment of the current teaching methods at the software engineering program. The preliminary results should detect weaknesses and strengths to improve the teaching/learning levels.

VI. METHODS

This study examines the frequency, tendency and application of the four levels of thinking and teaching at the software engineering department at SU. It is a mixed mode investigation technique that comprises two steps:

The first step includes interviewing a focus group of software engineering lecturers at SU. This step aims at collecting in depth insights, responses and opinions to help guide, design and construct a reflective survey. It also includes discussion and analysis of active examples of PBL/assessment instructional methods and concept mapping.

The focus group interview is semi-structured to ensure that all interviewees are asked the same set of open ended questions. The duration of the interview is 120 minutes. The number of focus group members is five. Groups were selected to ensure effective participation and substantial coverage of software engineering teaching methods at SU. The focus group interview emphasizes current software engineering teaching methods. Upon concluding the focus group activity, the authors defined the four levels of thinking and learning to the interviewees to help introduce and encourage the third and fourth levels of thinking and teaching at the department.

The second step includes the design of a structured questionnaire. The structured questionnaire comprised 63 questions. Questions addressed the four levels of thinking and teaching. Questionnaire included the following scale: 1-strongly disagree, 2-disagree, 3-neither agree nor disagree, 4-agree and 5-strongly agree. Questionnaires were disseminated among 130 students at the junior and senior levels at the software engineering department.

VII. RESULTS AND DISCUSSION

In the focus group discussion, the interviewees stress the importance of the first and second levels of thinking and teaching for the first and second year students. According to the focus group, the joined responsibility of the lecturer and the students is essential. At this stage interviewees believe that the lecturer-centered approach represents the main trend. Lectures are followed by guided tutorials where students should submit a hand-written assignment.

While discussing the teaching methods for the third and fourth year students, the focus group interviewees express a strong trend of the third level of thinking and teaching and strongly link its implementation to the students’ different abilities. While top students are more prepared for the third level activities, middle level students require close monitoring and feedback that may be provided through peer tutoring and group discussion. In most cases, middle level students usually gather information and specifications of design while top students link the components and present the final project of a functional value. Large numbers of students per section present an obstacle that makes the transition towards the third and fourth levels of thinking rather challenging.

The focus group discussed the PBL approach, assessment and thinking and teaching methods proposed by the authors of the paper. The purpose of the discussion is to help assess and introduce the knowledge of teaching and thinking levels at the department.

The focus group, also discussed the Software Engineering Department Annual Fair (SIDAF) as an example of the fourth level of thinking and teaching. As an annual event, projects are assessed according to certain criteria agreed upon by department. All years students are
encouraged to submit an independent project that reflects their abilities according to their levels of knowledge. The students usually submit 38-45 projects as groups or individuals. The fourth year students represent the majority of the participants. This is a strong indication to how only at the fourth year level; some students are ready to move towards the fourth level of thinking and teaching.

The focus group strongly agrees on the importance of motivating the students and providing a mutual respect environment at all levels of education.

The study measures different detailed examples of each level of thinking and teaching. Figure 5, is a graphical presentation of a wide span of teaching methods at software engineering department. The common trend shows that while the third and fourth year students may have been exposed to different methods of teaching; there is a tendency towards the third level of thinking and teaching. This may be explained by their need for more guidance and direction. Upon discussing the results with faculty members, lack of motivation may explain this trend. In the designed questionnaire, the majority of students agree that learning is achieved by a positive response to expectations of lecturers’ activities.

Figure 6, demonstrates that the students do not prefer classical teaching methods. Students are fully aware that providing knowledge is not the sole responsibility of the lecturer. They strongly agree that programming is not a talent. It is more of a skill that can be acquired by proper practice and guidance. The majority of the students in the third and fourth years prefer smaller class rooms than large ones.

The students are aware that solving theoretical problems only in learning programming language is inadequate. Learning a programming language involves practice and guidance as well as independence. This reflects a strong readiness among students to shift towards the third and fourth levels of thinking and teaching. This is an expected pattern among the third and fourth year students.

Figure 6: demonstrates that third and fourth year students avert classical teaching method

Figure 7, assures that students have a wide awareness and exposure to a variety of teaching methods. This figure reflects tendency towards the third level of thinking and teaching or in other words, what the student does. The students believe that they understand better from a lecturer that respects them. They acknowledge that the best evaluator of their work is the lecturer, yet they want to be evaluated by their level of engagement and interest. Students strongly believe that effective learning is achieved by guided practicals, group discussion and poster presentations. Students at software engineering department are uncertain about computer assisted learning. Innovative assessments
and lecturers’ skills deployment are essential characters of a successful lecturer.

The statistical analysis of questionnaire presented in Figure 7 shows a low tendency towards the fourth level of thinking and teaching. It is evident that the students need to be motivated and introduced to new learning methods where they are in control of their own learning. The teaching methods should start at the constructive alignment of the learning outcome and expectation by the lecturer. Students should be empowered to be responsible for their own learning. While students are supportive of independent projects they are discouraged by peer mentoring. This detects limited team work skills among students. On a positive note, the students declared a strong need and support of promoting better peer communication. The students believe that lecturers of programming languages should encourage design and creative thinking/problem solving among them. Students are enthusiastic about producing mind/concept maps and believe that they will lean better by active engagement.

Figure 7: reflects tendency towards the third level of thinking and teaching or in other words what students does

VIII. CONCLUSION

This paper is an effort to assess and identify different teaching methods at software engineering department at Salahaddin University. This section summarizes the author’s recommendations to help improve the levels of teaching and learning at the department.

It is evident that both students and academics at the department agree that the first and the second levels of thinking and teaching are suitable for early education levels. However, this study indicates that the majority of the students at the third and fourth year are more comfortable at the third level of thinking and teaching. Top fourth year students exhibit a stronger potential to succeed at the fourth level of thinking and teaching.

Lack of motivation, productive team work skills and large number of students per section impede the necessary suggested transition to the fourth level. A joint responsibility lies on lecturers, curricula developers and students to move towards the fourth level.

The actual teaching practice lies primarily in a range between the first and the third levels. The challenge viewed by the authors of this paper is to establish strong teaching skills that should allow the teaching process to move to the fourth level which is ideally the ultimate goal of higher education. While level three is based on constructive alignment and interdependence between the lecturer and the student, the fourth level is when the student is ready to choose his/her knowledge.

The lecturers’ attention should be directed towards engaging the students in effective learning activities that can prepare them for independent projects. Mimicry is certainly not acceptable, but free decisions by the students and thin outlines by the lecturers could be a good start. The third level types of activities may effectively engage the students, but does not necessarily assure the clear understanding of the learning process or objectives. At the third level, the students are not taking the responsibility on deciding what to do. Managing their learning is introduced only at the fourth level.

It is always essential to remember that reaching the fourth level requires stimulating students to make good choices. The students should experience and value good and effective learning skills by properly aligned assessments. Moving students so they are operating at the fourth level requires us to make changes in the engineering curriculum, involving students in the assessment process and the students’ approach to learning.

The aim of this study is not only to assess the current understanding and teaching practices but also to help faculty and students recognize and value characteristics of good teaching and to maintain them in higher education.

Teaching programming is one of the most challenging tasks in most universities around the world. Without enough practice and guidance students are usually not prepared to move properly to the fourth level.

Involving students in assessment design, implementation and evaluation is another strategy at this level. It allows students to develop some autonomy and decision-making in learning and assessment.

Authors of the paper highly recommend formal and guided peer tutoring at the first and second years. The positive impact and appreciation on students’ performance will lead to continuous informal peer tutoring that is initiated by the students.

Corporate recruiters seek individuals with good team skills. To prepare students for successful work experience, computer science university educators should not only
provide the students with knowledge but also help build their team work skills. There is considerable evidence that students working in teams develop deeper understanding and stronger retention. Group work has also proven to help struggling students.

However, many academic and university bodies have a broad consent that educational systems are not only ways of attainment of knowledge, but are also a means to provide guidance to the right attitudes and skills the graduates need in real life. This paper is also an attempt to empower the students’ ability of lifelong learning. Effective and well structured learning activities will guide students in the process. Nevertheless, deciding what is and what is not effective can be identified through different levels of thinking and teaching.

ACKNOWLEDGMENT

The authors would like to thank students and academics at software engineering department, Salahaddin University for their effective participation that helps enrich this study.

REFERENCES


