Software Engineering Assessments and Learning Outcomes

KHAIRUDDIN HASHIM
Software Engineering Department
Tenaga Nasional University
43009 Kajang
MALAYSIA
khairuddin@uniten.edu.my

NURUL NASLIA KHAIRUDDIN
Faculty of Computer Science and Information Technology
50603 Kuala Lumpur
MALAYSIA

Abstract: - Learning outcomes of software engineering-based courses require assessors to design assessment questions matching defined cognitive levels. This is to ensure students who take software engineering courses are assessed effectively and are imparted with the right level of knowledge and skill-sets. This paper outlines the application of Bloom’s Taxonomy in software engineering assessment and the matching of assessments with learning outcomes. Sample questions are given and categorized according to the relevant Bloom’s Taxonomy levels. This paper aims to assist software engineering teaching and learning; and improve the quality of software engineering assessment.

Key-Words: - Software Engineering Assessment, Bloom’s Taxonomy, Learning Outcomes

1 Introduction
Learning outcomes represent deliverables at the end of a course offering. It covers the required cognitive level(s) on each topic for successful completion of a course. The approach proposed is to define learning outcomes using Bloom’s taxonomy cognitive levels. Six levels of Bloom’s Taxonomy give a hierarchy of cognitive domain from knowledge to evaluation. The higher the level, the more useful it is. Through Bloom’s Taxonomy, software engineering assessment can be matched with learning outcomes. Employers look for potential employees with higher order cognitive competencies to help do comparative analysis and make good decisions. Assessment approach in a domain of study requires proper design to match assessments with learning outcomes.

2 Background
In this research, Bloom’s Taxonomy is used as a guide to define required learning outcomes and matching assessments. In this section, we give a brief explanation on Bloom’s Taxonomy, and the related education domains that have incorporated the taxonomy in their assessments.

2.1 Bloom’s Taxonomy
Bloom’s taxonomy was introduced by its originator, Benjamin Bloom, when he first published his idea on the cognitive skills taxonomy in his book [1]. Bloom defined 6 levels of cognitive domain: knowledge, comprehension, application, analysis, synthesis, and evaluation. The levels are shown in Bloom’s Taxonomy hierarchy model in Fig. 1.

Fig. 1: Bloom’s Taxonomy Hierarchy

The pyramid is ordered in such a way that the lowest level is the simplest form of recognition, while the highest level is the most abstract and complex form of cognitive skill. A more detailed explanation of each level is given below:

- Knowledge - ability to recall
- Comprehension - ability to understand
- Application - ability to use knowledge
- Analysis - ability to separate component parts
- Synthesis - ability to put parts together
- Evaluation - ability to judge value of ideas
In 2001, Anderson et al. [2] modified the taxonomy. They made a lot of changes and the new hierarchy is flattened on the top (see Fig. 2). This newly modified Bloom’s Taxonomy is well-known as the Revisited Bloom’s Taxonomy. However, this paper refers to the original version of the taxonomy.

![Fig. 2: Revised Bloom’s Taxonomy Hierarchy](image)

### 2.2 Bloom’s Taxonomy in Related Fields of Education

Numerous research have been seen in many of computer science domain of studies including computer science itself being related to Bloom’s Taxonomy. From existing research, Bloom’s Taxonomy seems to be very useful for education purposes and is still very popular even after more than 5 decades. It has been widely used for learning objectives measurement and assessment.

Bloom’s Taxonomy was applied on three computer science courses by Machanick [3]. His experience was then analyzed and he decided that Bloom’s Taxonomy-based approach works well.

But a question was raised in 2006 by Johnson et al. on whether the taxonomy is appropriate for computer science [4]. They also suggested a new hierarchy that adds another level onto Anderson et al.’s taxonomy, which is the higher application level (see Fig. 3). This is given from their idea that application is the aim of computer science teaching. However, creation of a solution to a problem entails synthesis of all knowledge and skills learnt (including application) and the final step is usually an evaluation process.

![Fig. 3: Bloom’s Taxonomy Hierarchy for Computer Science](image)

A closely related research was done by Thompson et al. earlier this year, but they focused on computer science assessment [5]. Their aim was to use Bloom’s Taxonomy to assist in designing introductory programming examinations. Samples of programming questions were given for each Bloom’s Taxonomy level.

A more recent research was done by Starr et al. which focused on specifying assessable learning objectives in computer science [6]. They believed that their idea of integrating Bloom’s Taxonomy with computer science curricular had made their faculty communication more effective and department’s assessment program stronger. Their work is actually an extension to [7] which focused specifically on human-computer interaction curricular guidelines.

Other research work that were done for specific computer science areas of education using Bloom’s Taxonomy include a test-driven automatic grading approach for programming [8], Bloom’s Taxonomy levels for three software engineer profiles [9], and Bloom’s Taxonomy for system analysis workshops [10].

### 3 Applying Bloom’s Taxonomy

During the development of an academic program, a set of learning outcomes is defined for each course. Assessments need to match learning outcomes. The requirement for learning outcomes is that they should be assessable. This ensures that students who take software engineering courses are assessed effectively and are imparted with the right level of knowledge and skill-sets.

The use of keywords defined for each cognitive level helps identify the complexity of cognitive requirement. Assessments questions are either formative or summative in nature. These can be in the form of assignments, quizzes or formal examination. In this section we look at each level of the Bloom’s Taxonomy and demonstrate suitable examples.

#### 3.1 Knowledge Level

Knowledge in this research context refers to the students’ ability to recall software engineering concepts that they have learnt in classes. Knowledge level questions include the keywords define, list, arrange, order, and state.

Below are some sample questions that fall under this level:
• What is requirements engineering?
• List 5 characteristics of good requirements.
• State four attributes of well engineered software.
• Define four types of traceability.

3.2 Comprehension Level
Comprehension in this research context refers to the students’ ability to understand and restate or describe a learnt concept using their own words or explanation. Comprehension level questions include the keywords explain, describe, discuss, identify, review, select, and predict.

Below are some sample questions that fall under this level:
• Given the above code fragment, identify the test cases for the code fragment.
• Describe 4 types of coupling in software design.
• Describe the Pareto Principle in statistical software quality assurance.
• Discuss the amplification and propagation of defects. What steps should be taken to alleviate the above problem?

3.3 Application Level
Application in this research context refers to the students’ skill in using the theories learnt to solve new problems. Application level questions include the keywords classify, write, apply, choose, and interpret.

Below are some sample questions that fall under this level:
• A software system is to be developed for Company XYZ. The client is unsure of what the final system should be. Which software development model would be suitable for this project? Justify your choice of software development model.
• Given the above description of a search function, write the full specification of function Search. The specification must also define error predicates.
• The following statements are the informal description of a system for a shop. Analyse the statements and then transform them into a formal specification by using Z.
• Given the following requirements, classify them to be either functional or non-functional requirement:
  i. Security
  ii. Feature to calculate cost of item based on current discount policy.
  iii. Reliability

3.4 Analysis Level
Analysis in this research context refers to the students’ ability to separate a whole into various component parts. Analysis level questions include the keywords analyze, compare, contrast, distinguish, categorize, calculate, differentiate, and test.

Below are some sample questions that fall under this level:
• A software development house X plans to develop a software application for company Y using a leading edge technology. The estimated development time is one year. Discuss relevant risk analysis and management procedures.
• Given the above code fragment, calculate the cyclomatic complexity number using the flow graph approach.
• Given that there are five members in a democratic team, calculate the number of communication paths needed.
• Compare and contrast the waterfall model with the prototyping model.

3.5 Synthesis Level
Synthesis in this research context refers to the students’ ability to relate learnt software engineering concepts and produce a new idea. Synthesis level questions include the keywords create, construct, design, develop, manage, organize, plan, predict, and propose.

Below are some sample questions that fall under this level:
• Based on the project and product attributes given above, propose the check-points and check-lists for the project.
• Given the basic requirements of the project, propose suitable features of the final system.
• Develop a SQA Plan for a software development project which is defined in the attached document.
• Design the architecture of the software system based on the requirements defined in the Software Requirement Specification document.

3.6 Evaluation Level
Evaluation in this research context refers to the students’ ability to judge, critic and decide on value of ideas or materials. Evaluation level questions
include the keywords argue, debate, recommend, prioritize, justify, rate, and decide.

Below are some sample questions that fall under this level:

- Given the above solution to the pre-defined problem, decide if it is a suitable solution. Justify your answer.
- Given four approaches to solving the given problem, rate the solutions in the context of system reliability.
- Given two possible solutions, A and B, to solving the given software development problem, decide on the best solution. Give your justification.
- Given three possible approaches to implement the defined system, discuss the advantages and disadvantages of each approach.

4 Conclusion
We have presented the application of Bloom’s Taxonomy in software engineering assessment. The higher the cognitive level, the more demanding it is in terms of cognitive complexity and competency. The approach presented will help assessors in designing assessments and matching assessments to learning outcomes which are critical activities in ensuring that graduates coming out of a program have acquired the required knowledge and cognitive competency levels.

References: