Application of Bloom's Taxonomy in Software Engineering Assessments

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

KHAIRUDDIN HASHIM
Software Engineering Department
Tenaga Nasional University
43009 Kajang
MALAYSIA

Abstract: - Bloom's Taxonomy has been utilized in many fields of studies. It has also been used in computer science education but research on the application of Bloom's Taxonomy into software engineering curricula has not been done much. This paper outlines software engineering assessment using Bloom's Taxonomy. 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 education.

Key-Words: - Software Engineering Assessment, Bloom's Taxonomy

1 Introduction

Bloom's Taxonomy is a cognitive skills taxonomy which has been applied on many education fields, including computer science. The taxonomy is widely used in almost any known education field. It is still accepted and proved to be successful in assisting education practices and teachings.

Assessment of learning outcomes for software engineering courses can be improved effectively through proper application of the taxonomy. This paper introduces an approach to improving software engineering assessments using Bloom's Taxonomy.

2 Background

In this research, Bloom's Taxonomy is used as a guide to define software engineering education practices. In this section, we give a brief explanation on Bloom's Taxonomy, and the related education fields that have incorporate the taxonomy in their assessment.

2.1 Bloom's Taxonomy

Benjamin Bloom was the originator of this taxonomy, as 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.

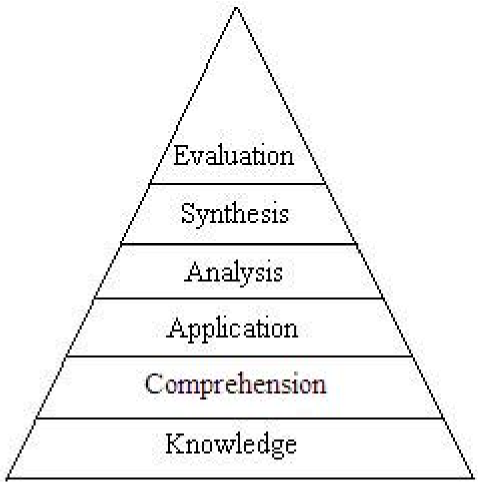

Fig. 1: Bloom's Taxonomy Hierarchy

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

Bloom's Taxonomy was then modified in 2001 by Anderson et al. [2]. 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.

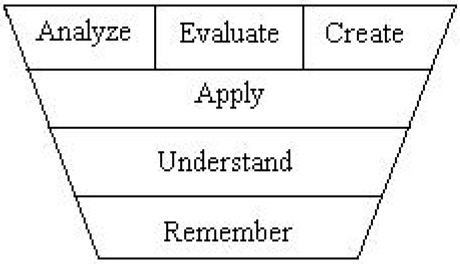

Fig. 2: Revised Bloom's Taxonomy Hierarchy

2.2 Bloom's Taxonomy in Related Fields of Education

Software engineering is a part of computer science. Numerous research have been seen in many of computer science field 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 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.

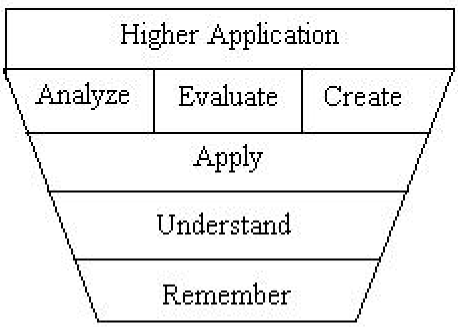

Fig. 3: Bloom's Taxonomy Hierarchy for Computer Science

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 in Software Engineering Assessment

Learning outcomes of SE-based courses require assessors to deliver questions with different cognitive levels. This is to ensure students who take SE courses are assessed effectively and are imparted with the right level of knowledge and skill-sets. 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 a global variable?
- List 5 reserved words in C programming.
- State four attributes of well engineered software.
- Define four types of traceability.

3.2 Understanding 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:

- Identify the value of `x` after running this program fragment:

```
x=0; y=0;
while (y<50) {x++; y=y+5}
```

- Predict the output of this program fragment:

```
i=0;
while (i<=10) {
  printf("%3d %3d\n", i, 50-i);
  i++;
}
```

- Describe 4 types of coupling in software design.
- Describe the Pareto Principle in statistical software quality assurance.

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:

- Write a `for` loop that produces this output:

```
0    1
1    2
2    4
3    8
4    16
5    32
6    64
```

- Write an `if` statement to compute and display the average of a set of n numbers. Calculation should only be done if n is greater than 0, or else an error message should be prompted.
- 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 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:

- Differentiate between .call by value and call by reference.
- Differentiate `printf` function calls for displaying prompts and for echoing data.
- 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:

- Construct a complete C program that reads text strings from a text file into a suitable data structure, sorts the list in ascending order, displays the list on the screen and stores the list in sorted order into the text file.
  Justify your choice of data structure.
- Write a C program that accepts integer inputs from the screen, computes the total and average values; and displays the values on the screen.
- 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 two solutions to the stated programming problem, rate the solutions in terms of efficiency and readability.
- Which of the two algorithms, bubblesort or quicksort, is more efficient? Justify your answer.
- 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 possible advantages and disadvantages of each approach.

4 Conclusion

We have presented the application of Bloom's Taxonomy in software engineering education. This will help educators in designing their questions for software engineering assessments, given the level of question types. It will also help to assess and ensure that software engineering students' knowledge level and skills acquired are as defined by the learning outcomes.

References:

[1] Bloom, B., *Taxonomy of Educational Objectives: Book 1 Cognitive Domain*, Longman, 1956.

[2] Anderson, L. & Krathwohl, D., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, Addison-Wesley, 2001.

[3] Machanick, P., Experience of Applying Bloom's Taxonomy in Three Courses, *Proc. Southern African Computer Lecturers' Association Conference*, 2000, pp.135-144.

[4] Johnson, C. & Fuller, U., Is Bloom's Taxonomy Appropriate for Computer Science?, *Proceedings of the 6th Baltic Sea Conference on Computing Education Research (AICPS '06)*, Vol. 276, 2006, pp. 120-123.

[5] Thompson, E., Luxton-Reilly, A., Whalley, J., Hu, M., Robbins, P., Bloom's Taxonomy for CS Assessment, *Proceedings of 10th Australian Computing Education Conference on Research and Practice in Software Engineering (ACE2008)*, Vol. 78, pp. 155-161.

[6] Starr, C., Manaris, B., Stavley, R., Bloom's Taxonomy Revisited: Specifying Assessable Learning Objectives in Computer Science, *Proceedings of SIGSCE 2008*, 2008, pp. 261-265.

[7] Manaris, B. & McCauley, R., Incorporating HCI into the Undergraduate Curriculum: Bloom's Taxonomy Meets the CC'01 Curricular Guidelines, *34th ASEE/IEEE Frontiers in Education Conference*, 2004, pp. T2H/10 - T2H/15.

[8] Buckley, J. & Exton, C., Bloom's Taxonomy: A Framework for Assessing Programmers' Knowledge of Software System, *Proceedings of the 11th IEEE International Workshop on Program Comprehension (IWPC'03)*, 2003, pp. 165-174.

[9] Hernán-Losada, I., Pareja-Flores, C., Velázquez-Iturbide, A., Testing-Based Automatic Grading: A Proposal from Bloom's Taxonomy, Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, 2008, pp. 847-849.

[10] Borque, P., Buglione, L., Abran, A., April, A., Bloom's Taxonomy Levels for Three Software Engineer Profiles, *Proceedings of the 11th Annual International Workshop on Software Technology and Engineering Practice (STEP'04)*, 2004, pp. 123-129.

[11] Yadin, A., Implementation of Bloom's Taxonomy on Systems Analysis Workshops, *Proceedings of the AIS SIG-ED IAIM 2007 Conference*, 2007.