

A Practical Review of ABET Accreditation Criteria

The case of Software Engineering Programs

Noureddine Abbadeni and **Abdullah Alghamdi**

King Saud University

College of Computer and Information Sciences

Department of Software Engineering

Riyadh, KSA

nabbadeni@ksu.edu.sa; ghamdi@ksu.edu.sa

Abstract

In this paper, we pass in review the ABET accreditation criteria for Engineering programs, with a focus on Software Engineering programs. We consider accreditation criteria for the 2011-2012 accreditation cycle. Based on a practical experience we gained while we were leading our department through the ABET accreditation process, we discovered and noted some practical issues with some accreditation criteria. We elaborate on some important practical gaps in these accreditation criteria and discuss some potential solutions to these gaps. We discuss only accreditation criteria for Software Engineering programs that we consider as not well specified: program educational objectives, student outcomes, curriculum, assessment and continuous improvement, and program criteria. Other accreditation criteria, such as students, faculty, facilities, and institutional support are quite straightforward and thus will not be discussed in this paper.

1. Introduction

Accreditation of academic programs is a peer-reviewed and voluntary process used by academia to assess and evaluate the quality of their degree programs. In United States, and in some other countries, the accreditation board of engineering and technology (ABET) is becoming the leader in accrediting Engineering, Computing, Technology, and Applied Science programs [1, 8].

ABET is structured in four main commissions; each covers a list of relevant programs and defines a list of relevant accreditation criteria. ABET commissions are: Engineering, Computing, Technology, and Applied Sciences. Accreditation criteria are respectively Engineering Accreditation Criteria (EAC) for engineering programs, Computing Accreditation Criteria (CAC) for computing programs, technology accreditation criteria (TAC) for technology programs, and Applied Sciences Accreditation Criteria (ASAC) for applied sciences programs [2].

ABET accreditation criteria are divided in two categories [2]:

- General criteria, applicable to all programs in a category (e.g. Engineering programs), include students, program educational objectives (PEO), student outcomes (SO), assessment and continuous improvement, curriculum, faculty, facilities, and institutional support
- Program criteria, applicable to a specific program (e.g. Software Engineering), depend on each program. For example, a Software Engineering program must demonstrate that graduates have “the ability to analyze, design, implement, verify, maintain, and apply software systems”.

In addition to general and program criteria, ABET accreditation is based on a list of policies and procedures described in the ABET APPM (Accreditation Policies and Procedures Manual) that

must be satisfied in order to fulfill accreditation [3]. For example, in order to apply for ABET accreditation, a program must have graduates prior to the on-campus visit (or the academic year preceding the on-campus visit).

The authors led their department (Software Engineering) for more than a year and half in the preparation of ABET self-study report and other materials required by ABET towards accreditation. While conducting their work towards achieving this goal, the authors discovered a number of issues and gaps related to ABET criteria and discussed some solutions to overcome the identified issues. This paper summarizes these findings. Some criteria, such as students, faculty, facilities, and institutional support are quite well specified in our view, and thus straightforward and easily understood, while other criteria, such as program education objectives, student outcomes, assessment and continuous improvement, curriculum, and program criteria are not well specified in our view, and thus confusing. We focus especially on these “confusing” criteria. Then we elaborate on some important issues in the specification of these criteria and discuss some potential improvements to overcome these issues. This study is based on the ABET accreditation criteria for the 2011-2012 accreditation cycle.

2. Shortcomings in the specification of ABET accreditation criteria

As mentioned previously, we will focus on accreditation criteria that we consider as not well specified. These criteria are:

- Program Educational Objectives
- Student Outcomes
- Curriculum
- Assessment and Continuous Improvement
- Program Criteria

2.1 Program educational objectives

Program educational objectives are broad statements that represent the achievements students (graduates) are expected to achieve or reach few years after graduation. For example: few years after graduation, students (graduates) will be able to assume managerial and leadership positions.

ABET defines the requirements for program educational objectives for all engineering programs (including Software Engineering) as follows [2]:

“The program must have published program educational objectives that are consistent with the mission of the institution, the needs of the program’s various constituencies, and these criteria. There must be a documented and effective process, involving program constituencies, for the periodic review and revision of these program educational objectives.”

The main issue with the specification of program educational objectives is the fact that it is completely left open to each program to define its educational objectives without any guideline. This leads to some programs with educational objectives focusing on the technical side of their discipline while other programs focus on other aspects such as life-long learning, leadership, etc.

2.2 Student outcomes

While program educational objectives are broad statements that describe the achievements students (graduates) are expected to attain few years after graduation, student outcomes are narrow statements that describe the knowledge and skills students shall possess by the time of graduation.

ABET defines the requirements for student outcomes for all engineering programs (including Software Engineering) as follows [2]:

“Student outcomes are outcomes (a) through (k) plus any additional outcomes that may be articulated by the program:

- a) an ability to apply knowledge of mathematics, science, and engineering*
- b) an ability to design and conduct experiments, as well as to analyze and interpret data*
- c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability*
- d) an ability to function on multidisciplinary teams*
- e) an ability to identify, formulate, and solve engineering problems*
- f) an understanding of professional and ethical responsibility*
- g) an ability to communicate effectively*
- h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context*
- i) a recognition of the need for, and an ability to engage in life-long learning*
- j) a knowledge of contemporary issues*
- k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. “*

Issues with student outcomes can be summarized as follows:

- Student outcomes are defined in a very general way that can be misleading and allows different interpretations or even misinterpretations by different educators in the same program. For example, as pointed out in [9], it is not clear what are the skills that a student should acquire during his studies in order to become a life-long learner.
- Some important skills for software engineers are not covered in the stated outcomes (such as the ability to plan and manage software systems, the ability to analyze requirements, the ability to maintain software systems, etc.).
- Student outcomes, as mentioned previously, represent knowledge and skills students should acquire by the time of graduation. Knowledge and skills, as described in [5], are divided in several levels including knowledge, understanding, application, analysis, synthesis, and judgment. It is not clear to which level each student outcome corresponds.

2.3 Curriculum

ABET defines the curriculum requirements for all engineering programs (including Software Engineering) as follows [2]:

“The curriculum requirements specify subject areas appropriate to engineering but do not prescribe specific courses. The faculty must ensure that the program curriculum devotes adequate attention and time to each component, consistent with the outcomes and objectives of the program and institution. The professional component must include:

- a. One year of a combination of college level mathematics and basic sciences (some with experimental experience) appropriate to the discipline. Basic sciences are defined as biological, chemical, and physical sciences.*
- b. One and one-half years of engineering topics, consisting of engineering sciences and engineering design appropriate to the student's field of study. The engineering sciences have their roots in mathematics and basic sciences but*

carry knowledge further toward creative application. These studies provide a bridge between mathematics and basic sciences on the one hand and engineering practice on the other. Engineering design is the process of devising a system, component, or process to meet desired needs. It is a decision-making process (often iterative), in which the basic sciences, mathematics, and the engineering sciences are applied to convert resources optimally to meet these stated needs.

- c. *A general education component that complements the technical content of the curriculum and is consistent with the program and institution objectives.*

Students must be prepared for engineering practice through a curriculum culminating in a major design experience based on the knowledge and skills acquired in earlier course work and incorporating appropriate engineering standards and multiple realistic constraints.

One year is the lesser of 32 semester hours (or equivalent) or one-fourth of the total credits required for graduation.”

In the EAC SSR template [4], ABET requires display material, to support the curriculum, to be available during the on-site visit:

“Describe the materials (course syllabi, textbooks, sample student work, etc.), that will be available for review during the visit to demonstrate achievement related to this criterion”

Issues with curriculum requirements can be summarized as follows:

- Defining basic sciences as biological, chemical, and physical sciences and requiring 1 year of college-level mathematics and basic sciences is not clear enough. Which mathematics topics should be covered (calculus, linear algebra, probability & statistics, operations research, etc.)? Which biological, chemical, or physical topics should be covered (organic chemistry, biochemistry, genetics, mechanics, electricity, etc.)?
- Requiring one and one-half year of engineering topics is not clear enough. Again, which topics should be covered by a specific program? Are one year and one-half enough to become an engineer?
- The general education component is left open for each program without specifying a minimum number of hours or a set of core topics to be considered. This does not help to have a minimum of a common general education for programs in the same category.
- The minimum number of hours is not specified for the senior design experience (graduation project). This leads to programs in the same category having different hours and thus different weight given to graduation project. For example, some programs allocated 3 hours for it for 1 semester; other programs allocate 6 hours for it over 2 semesters.
- The requirements in terms of display materials are not clear enough. For example, it is not clear for how many semesters the display materials should be available for the on-campus visit. The list of items that should be available for each course is also not clearly defined.

2.4 Assessment and continuous improvement

Assessment covers the assessment of program educational objectives as well as the assessment of student outcomes [4]: "Assessment is defined as one or more processes that identify, collect, and prepare the data necessary for evaluation. Evaluation is defined as one or more processes for interpreting the data acquired through the assessment processes in order to determine how well the program educational objectives and student outcomes are being attained."

ABET defines the requirements for assessment for all engineering programs (including Software Engineering) as follows [2]:

"The program must regularly use appropriate, documented processes for assessing and evaluating the extent to which both the program educational objectives and the student outcomes are being attained. The results of these evaluations must be systematically utilized as input for the continuous improvement of the program. Other available information may also be used to assist in the continuous improvement of the program."

Then, some guidelines are provided in the ABET SSR template [4]:

"Although the program can report its processes as it chooses, the following is presented as a guide to help you organize your self-study report. It is also recommended that you report the information concerning your program educational objectives separately from the information concerning your student outcomes."

Program Educational Objectives

It is recommended that this section includes (a table may be used to present this information):

- *A listing and description of the assessment processes used to gather the data upon which the evaluation of each the program educational objective is based. Examples of data collection processes may include, but are not limited to, employer surveys, graduate surveys, focus groups, industrial advisory committee meetings, or other processes that are relevant and appropriate to the program.*
- *The frequency with which these assessment processes are carried out*
- *The expected level of attainment for each of the program educational objectives*
- *Summaries of the results of the evaluation processes and an analysis illustrating the extent to which each of the program educational objectives is being attained*
- *How the results are documented and maintained*

Student Outcomes

It is recommended that this section includes (a table may be used to present this information):

- *A listing and description of the assessment processes used to gather the data upon which the evaluation of each student outcome is based. Examples of data collection processes may include, but are not limited to, specific exam questions, student portfolios, internally developed assessment exams, senior project presentations, nationally-normed exams, oral exams, focus groups, industrial advisory committee meetings, or other processes that are relevant and appropriate to the program.*
- *The frequency with which these assessment processes are carried out*
- *The expected level of attainment for each of the student outcomes*
- *Summaries of the results of the evaluation process and an analysis illustrating the extent to which each of the student outcomes is being attained*
- *How the results are documented and maintained"*

Although some general guidelines are provided, the major issue with assessment, for both program educational objectives and student outcomes, is that the assessment methods and

metrics are completely left to the choice of each program. Some programs use rubrics-based approach, other programs use a holistic approach, etc. [6]. Programs also use different types of surveys and different types of metrics (average score, percentage of students achieving a satisfactory level, etc.). This creates a lot of confusion. In many cases, programs within the same college and university are using different assessment methods and metrics. Such situation, beyond the fact that it is confusing, it does not allow benchmarking between programs in the same category.

2.5 Program criteria

ABET defines the requirements in terms of program criteria for Software Engineering programs as follows [2]:

“The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program.

The curriculum must prepare graduates to analyze, design, verify, validate, implement, apply, and maintain software systems; to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems; to work in one or more significant application domains; and to manage the development of software systems. “

The main issues with program criteria are as follows:

- Important information, which is not available in the general criteria section (student outcomes, assessment, etc.) as general criteria apply to all engineering programs (and not only Software Engineering programs), is not defined in program criteria (specific to Software Engineering programs). For example, the topics or knowledge areas that must be covered in a Software Engineering program is not defined.
- Some information, provided in the program criteria, is confusing. For example, if we consider the Software Engineering programs, a program criterion is saying: “the curriculum must prepare graduates to analyze, design, verify, validate, implement, etc.”. This requirement is put under “curriculum” while this is clearly a student outcome, especially if we want to assess it in order to see whether it is attained or not.
- Computer science topics that should be covered by a Software Engineering program are not specified.

3. Potential Improvements

In order to improve the specification of the ABET accreditation criteria, and thus make them less confusing, we propose the following list of potential improvements:

- Define a minimum requirement to be covered by program educational objectives. For example one educational objective that addresses the technical side of the program, one educational objective that addresses the continuous learning and education side; etc.
- Break down student outcomes, as they are very general and subject to different interpretations or even misinterpretation, should be broken down into clearly defined items. Such breakdown should be done at the program criteria level (and not at the general criteria level) as it necessarily depends on each program.
- Link and categorize students outcomes based on the bloom's taxonomy of knowledge as defined in [5] and proposed in [9].

- Add important missing student outcomes for Software Engineering programs at the program criteria level (such as the ability to manage software systems; the ability to maintain software systems; etc.).
- Specify the minimum list of topics/subjects related to college-level mathematics and basic sciences (physics, chemistry and biology) that should be covered for Software Engineering programs.
- Specify the minimum list of Software Engineering topics/subjects or knowledge areas that should be covered in a Software Engineering program. The IEEE/ACM curriculum recommendations for Software Engineering programs [11] as well the Software Engineering Body of Knowledge (SWEBOK) [12] could be a good reference to this purpose. They should be referenced clearly by ABET in the definition of their criteria.
- Software Engineering is an engineering discipline strongly rooted in computing. Consequently a clear specification of the relevant computing topics that should be covered in a Software Engineering program is definitely needed. The same references mentioned in the previous point can be used here too.
- Define a minimum number of credit hours that should be allocated to the senior design experience.
- Define a minimum number of hours as well as a list of minimum topics that should be cover under the general education component of the curriculum.
- Specify clearly which display materials and for how many semesters should be available for review during the on-campus visit.
- Specify (if not prescribe) and standardize assessment methods and metrics to be used in the assessment and evaluation processes. This will reduce major confusion and resolve important issues facing educators it comes to define assessment methods and metrics for their programs. Standardizing assessment methods and metrics will also have a great benefit by allowing benchmarking between programs in the same category.
- Review program criteria for Software Engineering programs and clearly distinguish between curriculum requirements and specific student outcomes applicable to Software Engineering programs.

4. Conclusion

Program accreditation is becoming a very serious matter for many institutions and programs around the world in order to demonstrate and guarantee a minimum of quality in their degree programs delivery and to ensure they are up to the recognized standards in their disciplines. ABET is becoming the leading accreditation agency in USA and some other countries that follow the American model in higher education in the fields of engineering, computing, technology, and applied sciences.

The paper suggests that, based on a thorough and practical review, some of the ABET accreditation criteria are lacking precise specification in their definition. Such issues create confusion and lack of clarity and lead to different interpretations and understandings, and sometimes misinterpretations by educators. These confusing criteria are: program educational objectives, student outcomes, curriculum, assessment and continuous improvement, and program criteria. A more precise and rigorous specification of these criteria is required and will be of great benefit. In particular, specifying and standardizing assessment methods and metrics will have a serious positive impact and will allow a unified understanding and interpretation among different programs in different institutions, and consequently will allow performing benchmarking between programs in the same category.

Acknowledgements

We thank our colleagues, members of the ABET-Software Engineering Department Committee and the ABET-College Committee at the College of Computer and Information Sciences (CCIS), at King Saud University (KSU), for the fruitful discussions during our regular meetings. We thank also the College of Computer and Information Sciences for its support.

References

1. Accreditation Board of Engineering and Technology (ABET): <http://www.abet.org>
2. ABET Engineering Accreditation Criteria (EAC), 2011-2012 Accreditation Cycle: <http://www.abet.org/forms.shtml>
3. ABET Accreditation Policy and Procedures Manual (APPM), 2011-2012 Accreditation Cycle: <http://www.abet.org/forms.shtml>
4. ABET EAC self-study template (SSR template), 2011-2012 Accreditation Cycle: <http://www.abet.org/forms.shtml>
5. B. S. Bloom, M. D. Englehart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, *Taxonomy of Educational Objectives: Handbook 1: Cognitive Domain*, New York: Longman, 1956.
6. G. Rogers, "Assessment Tips", *ABET Newsletter*, August, October, and November, 2006.
7. R. M. Felder and R. Brent, "The National Effective Teaching Institute: Assessment of Impact and Implications for Faculty Development", *Journal of Engineering Education*, The American Society of Engineering Education (ASEE), Vol. 99, No. 2, 2010, pp. 121–134.
8. R. M. Felder, and R. Brent, "Designing and Teaching Courses to Satisfy the ABET Engineering Criteria", *Journal of Engineering Education*, The American Society of Engineering Education (ASEE), Vol. 92, No. 1, 2003, pp. 7-25.
9. M. Besterfield-Sacre, L. J. Shuman, H. Wolfe, C. J. Atman, J. McGourty, R. L. Miller, B. M. Olds, and G. Rogers, "Defining the Outcomes: A Framework for EC 2000", *IEEE Transactions on Education*, Vol. 43, No. 2, 2000, pp. 100-110.
10. R. M. Felder and R. Brent, "Understanding Student Differences", *Journal of Engineering Education*, The American Society of Engineering Education (ASEE), Vol. 94, No. 1, 2005, pp. 57-72.
11. Y. Wang, "The Development of the IEEE/ACM Software Engineering Curricula", *IEEE Canadian Review*, Vol. 51, No. 2, 2005, pp. 16-20.
12. A. Abran and J. Moore (Eds.), *SWEBOK: Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, 2004.