Engineering Criteria 2000: The Impact on Engineering Education

Neelam Soundarajan*
Dept. of Computer & Information Science
College of Engineering, Ohio State University
Columbus, OH 43210
e-mail: neelam@cis.ohio-state.edu

Abstract

Engineering Criteria 2000 or EC2000 is the new set of criteria that Engineering programs must satisfy in order to be accredited by the *Accreditation Board for Engineering and Technology* (*ABET*). While some of the ideas behind EC2000 are certainly likely to have a positive impact on engineering education, others, this paper argues, are questionable, and indeed may have a negative impact. The paper attempts to justify this claim by presenting some hypothetical programs and examining how they will fare under EC2000.

1 Introduction

Engineering Criteria 2000 or EC2000 is the new set of criteria that Engineering programs must satisfy in order to be accredited by the *Accreditation Board for Engineering and Technology* (*ABET*). These criteria differ considerably from the existing criteria and, as such, are likely to have a major impact on Engineering education across the country and indeed across the world, since *ABET* serves as a model that many other countries follow. The goals of this article are to examine the likely impact and to see to what extent the new criteria achieve their intended purposes.

The work that preceded the drafting of EC2000 focused on four issues (see, for example, [8]):

1. The current accreditation criteria are too long and encourage a rigid, bean counting approach that stifles innovation;

2. The current accreditation process demands excessive time commitments;

3. Accreditation visits occur too frequently;

4. The demands of ABET participation limit the volunteers seeking ABET leadership roles.

*It is important to note that the opinions expressed in this article are those of the author as an individual educator and researcher. They do *not* in any way reflect the official position of the Ohio State University, its College of Engineering, or the Dept. of Computer & Information Science.

As the work on developing the EC2000 criteria progressed, two important issues seem to have been added to this list; first, a very strong focus on identifying and documenting the *processes* used for almost every activity in an engineering program; and second, a focus on developing, documenting, and utilizing a range of *assessment mechanisms* to evaluate every aspect of the program. Interestingly, while EC2000 is less prescriptive than its predecessor in the specific courses in the curriculum or in the number of hours that must be devoted to various important topics, it is extremely prescriptive in requiring engineering programs to follow documented processes. Indeed, one could even argue that what EC2000 has done is to shift the 'bean counting' from product (the curriculum) to process. Further, anecdotal evidence indicates that EC2000 will be far more demanding than its predecessor in terms of time commitment on the part of the faculty at programs wishing to be accredited. It is true that some of the documentation required by the previous criteria, as part of the self-study reports that the programs submit, is not required by EC2000; but this is more than offset by the requirement of having to document the processes used for various activities, to establish paper trails to show that the processes are indeed being followed, to document the fact that the results of various activities are being used to improve the program, etc. Moreover, the frequency of accreditation visits under EC2000, from all indications, will be the same as under the previous criteria. If these claims are valid, and we will try to validate them in the rest of the article, then clearly EC2000 will not solve the problems identified in the focus issues listed above. Nevertheless one could claim, and indeed this is exactly what its proponents do seem to claim, that EC2000, with its stress on detailed documented processes and assessment mechanisms, will have a strong positive impact on engineering programs. In the rest of this article we will try to argue that this claim is questionable.

In the next section we summarize the EC2000 criteria as well as the existing criteria. In the third section we focus on a (hypothetical) case study created by *ABET*. In the final section we summarize our arguments and the potential problems we see with EC2000.

2 The Criteria

We will use the term Pre2000 to refer to the (existing) set of criteria that EC2000 is intended to replace. Complete details on both EC2000 and Pre2000 have been widely published by *ABET*, see for example [2]. Each set contains common criteria applicable to all engineering programs, and program specific criteria that apply to specific disciplines. By far the most important differences between the two sets are in the common criteria; there are also differences in the program-specific criteria but these tend to be less pronounced. In this article we will focus on the common criteria.

Pre2000 has seven criteria that we will refer to as P1, ..., P7. EC2000 also has seven criteria which we will refer to as E1, ..., E7. The numbering in the two sets are quite different from each other; thus P1, for example, concerns the faculty of an engineering program, while E1 is concerned with the students of the program. We start by summarizing both sets of criteria:

P1: Faculty: Faculty are required to be qualified, competent, able to communicate effectively, professionally active, and be actively involved in the program, including in student advising; the program must have the equivalent of at least three full-time faculty; and teaching loads must be reasonable and allow for professional growth.

P2: Curricular Objectives: These must include the goal of training students to be able to solve engineering problems in a practical way, being sensitive to social needs including public health and safety, and to be able to engage in life-long learning; these objectives are assumed to be met by a curriculum that includes coursework in fundamental scientific principles followed by application of these principles in engineering courses. Programs are expected to publish clear program goals and to demonstrate success in meeting these goals.

P3: Curricular Content: The curriculum must include at least three years of study in mathematics, basic sciences, humanities and social sciences, and engineering topics, where a 'year of study' is defined to be equal to 32 semester hours; of this, at least one year must be a combination of mathematics and basic sciences, at least 0.5 year of humanities and social sciences, and at least 1.5 years of engineering topics. The criterion further goes on to specify that 'mathematics' means 'beyond trigonometry' and that it must stress concepts and principles (over computation). The humanities and social sciences courses are required to provide both breadth and depth. Engineering topics must include both engineering sciences such as mechanics and thermodynamics, and engineering design, including a meaningful, major design experience that builds on the earlier part of the curriculum. In addition, the curriculum is required to pay attention to laboratory experience, computer-skills, application of probability and statistics in engineering problems, social and ethical issues in engineering practice, and developing written and oral communication skills.

P4: Student Body: The program is required to have policies that ensure that students admitted to the program are suitably qualified, that appropriate policies are in place for awarding transfer credits where appropriate, that items such as samples of student homeworks and exams are made available to the accreditation team to demonstrate the quality of student work, and that records of graduates' progress in the profession be tracked.

P5, P6, P7: Other: These concern such matters as library and computer facilities as well as the relation between the program and the rest of the University; we will not go into the details of these criteria.

Next let us turn to the EC2000 criteria.

E1: Students: The program is required to evaluate, advise, and monitor students to determine its success in meeting program objectives. The program is required to deal with transfer credits appropriately, and to ensure that all students meet all program requirements.

E2: Program Educational Objectives: The program is required to have detailed published objectives. It is also required to have a specific process, based on the needs of its constituencies, to determine these objectives and to evaluate them periodically; a curriculum and processes that ensure that these objectives are achieved; and a system of ongoing evaluation that demonstrates the achievement of these objectives and uses the results to improve the effectiveness of the program.

E3: Program Outcomes and Assessment: The program is required to demonstrate that its graduates have the ability to apply knowledge of mathematics, science, and engineering; to design and conduct experiments, and to analyze and interpret data; to design a system to meet desired needs; to function on multi-disciplinary teams; to identify, formulate, and solve engineering problems; to communicate effectively; to engage in life-long learning; to use modern engineering tools. The program is also required to demonstrate that its graduates understand professional responsibility, have a broad education, and have a knowledge of contemporary issues.

The program must have an assessment process with documented results. The program must demonstrate that the results are applied to the further development and improvement of the program. The assessment process must demonstrate that the outcomes important to the mission of the institution and the objectives of the program are being measured. *ABET* suggests using evidence such as student portfolios, nationally-normed examinations, alumni surveys that document professional accomplishments and career development activities, employer surveys, and placement data of graduates, for this purpose.

E4: Professional Component: The program must include one year of a combination of college level mathematics and appropriate basic sciences, one and half years of engineering topics (engineering sciences and engineering design), and a general education component consistent with the program objectives. The program must include a culminating, major design experience incorporating such considerations as economic, environmental, health and safety, etc.

E5: Faculty: This is similar to P1, except that the size of the faculty is required to be sufficient, but no specific number (such as 3 in P1) is prescribed.

E6, E7: Other: These are similar to P5, P6, P7 but are much less specific and detailed.

We will focus our discussion on P2 through P4, and on E1 through E4. In particular we will be interested in E2 and the second part of E3 since these embody the process and outcomes/assessment focus of EC2000. We conclude this section with some general comments about the criteria. E4 is indeed significantly less prescriptive than the corresponding P3 in terms of what the curriculum of the engineering program must contain. In this respect, EC2000 does address the 'bean-counting' issue as far as curriculum is concerned. But E2 and the second half of E3 more than make up for this by requiring specific processes and corresponding documentation that the engineering program must provide. Indeed in the hypothetical case study on the *ABET* web site which we will discuss in more detail in the next section, the program in question is complimented for having a good set of objectives and a successful program, but taken to task for failure to follow well documented processes. This is extremely disturbing; it is as if processes had become an end in themselves rather than being a means to improving the education and training of future engineers.

3 Case Study

On its web site *ABET* has provided a detailed case study [1] of a hypothetical institution, the Coastal State University (CSU), and its four engineering programs. Although CSU itself does not exist, the case study, according to *ABET*, is quite realistic. Here we briefly summarize the case study.

CSU has four engineering programs, respectively in Civil, Mechanical, Electrical, and Computer engineering. The first three are currently accredited, the fourth is a new program seeking initial accreditation. CSU had volunteered to be evaluated under EC2000 (but the final accreditation decisions were to be consistent with Pre2000). The administration of the college and the university was enthusiastic about outcomes assessment, continuous improvement, etc., but the faculty was not enthusiastic about these matters. The general feeling among faculty was that the engineering programs were very successful in achieving their goals, and that their graduates were well-regarded in industry. Because of this and because of time constraints, the process of setting goals etc. was primarily one in which the existing goals were reaffirmed (rather than 'starting with a clean piece of paper'). Briefly, these goals were to produce technically qualified graduates for employment as engineers, to produce graduates with a strong scientific basis preparing them for advanced studies, and to produce socially-aware graduates. The outcomes listed in E3 were adopted by the CSU programs as their own. For many years CSU had been using measures such as placement data, alumni surveys, performance of graduates in the FE exams, feedback from co-op employers, and performance of student teams in national competitions. CSU relied on informal rather than formal, documented, processes for using these measures to improve the programs, and cited *specific examples* [emphasis added] where such improvements could be clearly seen.

The evaluation team identified three concerns (with respect to EC2000). Briefly, these concerns were:

- E1: Although the advising system was functioning satisfactorily, the programs were not identifying their objectives for students, nor specifying what student educational objectives were being evaluated.
- E2: The programs did not have well-described processes based on the needs of their constituencies in which the objectives were being determined and periodically evaluated.
- E3: The programs did not have a system of ongoing evaluation that demonstrated that their objectives were being achieved and that the results were being used to improve the effectiveness of the programs.

The team found no concerns or deficiencies with respect to Pre2000. (There were some program-specific concerns but these were relatively minor and we will not go into those here.) One quote from the full report (from the section on the *Exit Interview*) is significant: 'The team found that the programs at CSU were clearly effective in delivering course material to the students, and that the graduates and the students in progress were very satisfied with their education. However, there was little evidence that there were any connections between the embryonic outcomes assessment processes in any of the programs and the processes of determining the efficacy, content, and objectives of their curricula.' One gets the distinct impression, although this is not stated explicitly in the case study, that had this been an actual evaluation under EC2000, these programs would very likely have been in serious difficulty getting (or retaining) accreditation.

4 Discussion

From the old adage that admonishes us not to reinvent the wheel, to the remark attributed to Newton about being able to see further by standing on the shoulders of giants, to the recent exhortations to software engineers to reuse and build on the work of others, conventional wisdom speaks to the importance of borrowing and improving on the work of others in the field. Indeed it would not be too much of an exaggeration to say that this has perhaps been the single most important basis for progress in nearly every branch of science and certainly of technology.

Is engineering education an exception to this? We have not seen any evidence of this, and it would be very surprising if it were true. Yet this seems to be one of the important assumptions underlying EC2000. Why else would *every* program be required to use an elaborate process to establish its basic objectives? If the objectives that the CSU programs had in place were reasonable, in the opinion of the CSU faculty and in the opinion of other professionals both academic and industrial, why does *ABET* insist that 'they start with a clean slate' and follow an elaborate process to arrive at their objectives? One could argue that having such a process that involves all constituencies of CSU will ensure that the objectives are uniquely tailored to the needs of those constituents. But engineering education is not a consumer good like a bag of chips that universities should tailor to the perceived desires of its (loudest) constituents. Indeed, there is potential for serious harm in doing so. Consider, for example, the experience of Stedinger [10]. He quotes from student surveys in a class where he used techniques similar to the processes required by EC2000: 'No derivations! I will never need to derive anything'; and another: 'I won't need to know any theory'. Reactions of this kind from students is indeed a common experience for many engineering faculty. Doesn't the *ABET* approach in this case require the engineering faculty to throw out all theory and derivations from the course? Of course we (as faculty and/or engineering professionals) understand the importance of these topics to the proper training of the future engineer, and one could argue that we could appeal to this understanding to decide that these topics must remain in the curriculum in spite of what the student surveys say. But then, what was the purpose of the survey? Wouldn't it be far better for the profession as a whole to decide what the objectives of engineering programs should be, and require each accredited program to meet these objectives? Adopting such common objectives will ensure, in an ever-shrinking world, that our graduates have the appropriate skills and abilities to work on engineering projects around the world and with graduates of other programs. Instead, requiring each program to develop its own unique set of objectives will, at best, result in enormous duplication of effort by programs around the country to arrive at essentially the same set of objectives; and, at worst, it may result in the training of graduates of some programs being so narrowly tailored to the needs of the extant local industry that these graduates will be unable to adapt to industries elsewhere or even locally, as local industry evolves.

Three further points should be noted. First, E4 and the first half of E3 *do* establish a common set of objectives for all programs. If these are met by a program, that will indeed ensure that graduates of the program, when they leave the program, are well on their way to becoming successful engineers. But their training could have been even better if the program did not have to divert scarce resources to 'reinventing the wheel' as required by E2 (or some of the questionable parts of E3 which we will consider shortly). Second, we are not suggesting that individual programs should not have their own unique objectives in addition to the common set; just that every program should not be required to reinvent the common set, nor should a program be required to have any additional objectives beyond those in the common set; if some objective that is not in that set is considered worthwhile by the profession as a whole, then it ought to be added to the common set. And if a program does have some unique objectives of its own, the question that *ABET* (and its evaluators) should be concerned with is whether these objectives are reasonable (for example, will they detract too much from the common set of objectives?), not the processes used in arriving at them. Third, industry, especially local industry, tends to have a rather narrow and short range focus; indeed, it must – else it would not stay in business very long. As an example from Computer Science and Engineering (*CSE*), not long ago many software houses, especially smaller ones, used to issue calls on a regular basis to *CSE* programs to train students in 'useful' languages such as '*JCL*' rather than 'waste time' on esoteric topics such as finite state machines [5] or garbage collection [6] (for memory management); yet concepts based on finite state machines are now a key component of the object oriented approach to software design, and *Java* would not exist without garbage collection, whereas it is difficult to find a software house that still uses *JCL*. This does not, of course, mean that industry is always short sighted; indeed many of the most important ideas in *CSE* (including *Java*) came from industry. But the point is that requiring every program to establish its objectives starting with a clean slate and based on the needs of its (immediate) consituencies is not only analogous to reinventing the wheel but, stretching the metaphor a bit, may result in some programs inventing the square wheel: a *CSE* program that devotes time to such topics as *JCL* at the expense of more fundamental concepts; useful perhaps to local industry on a very short term basis but definitely not training for the future.

To summarize this part of the discussion, we are suggesting that the engineering profession as a whole must decide what the important objectives are that all engineering programs must meet. Criteria E4 and the first half of E3 provide a very good set of common objectives. And to the extent that individual programs want to have additional objectives, they should be free to do so, so long as this does not detract from their meeting the common objectives.

Let us now turn to the assessment portion of EC2000, specifically the second half of E3, as well as part of E2. In a very basic sense, assessment is of course important. Having a lofty set of objectives does not mean anything if you don't achieve them. And to be sure you are achieving the objectives of your program, you must of course measure how well the students have learned the material in question and how well they are able to apply it as needed. This is something that presumably most faculty, indeed teachers at all levels, understand fully. Not surprisingly, many faculty invest a substantial amount of time and effort in designing thoughtful questions and problems that students are required to address in their examinations, or homework assignments, etc. But the jargon-laden 'education literature' (which seems to be the inspiration behind these components of EC2000) suggest that engineering faculty do not have a clue about how to evaluate how well their students have learned the material. Consider for example the following quote from [9] 'Establishing measurable objectives and evaluating their outcomes are sophisticated activities with which most engineering educators have had little or no experience.' Incredibly, the same article grants that 'traditional engineering instruction has served the nation well'. How can this be? If most educators have no experience with such a basic component of education as establishing objectives or evaluating how well their students are achieving their objectives, how could they possibly have done well?

As another example, consider the earlier quote from the CSU case study; the team found that 'CSU was effective in delivering course material to the students, and that the graduates and students were very satisfied with their education.' One would think, assuming that the course materials in the program is satisfactory, that CSU would be complimented for doing an effective job of delivering it to the students. Moreover, given the specific instances that CSU cited of improvements in the program on the basis of the results of their assessments, one would think that these assessments were serving their purpose. But the evaluators say: 'However, there was little evidence that there were any connections between the embryonic outcomes assessment processes in any of the programs and the processes of determining the efficacy, content, and objectives of their curricula.' Although the report does not say so explicitly, one gets the distinct impression that this is a serious violation of EC2000 in general, and E2, (and the second half of) E3 in particular. In other words, it looks as if the assessment methods and their documentation have become an end in themselves rather than being tools to ensure that the program is effective.

The assumption seems to be that widely used assessment techniques such as course examinations are not reliable and that more fashionable approaches such as 'focus groups, ethnographic studies, and protocol analysis' [7], with everything being fully documented, must be used. Occasionally, see for example [3], the argument is made that no single method (such as course exams) can fully test a students' learning, hence other techniques such as surveys must also be used. But we could not find any published evidence that this will help in any way. It would even seem that using two assessment techniques, call them A and B, would actually be worse than using just A if it turns out that B is not very reliable, because its results would tend to water down the results of the more reliable technique.

Indeed, the report [4] by the Joint Task Force on Engineering Education Assessment, after identifying various possible assessment techniques along these lines, goes on to caution 'How these measures are to be interpreted is an art in early development, and the task force cautions the engineering community to recognize it as such.' We would go further. Given that engineering education using conventional assessment techniques such as course exams has served the nation well, unless and until definitive evidence is available that these other assessment techniques will actually improve the quality and effectiveness of engineering programs, requiring every engineering program to implement these techniques and study their effectiveness etc., in effect turning every such program into a program in education research, would be a serious mistake. And to the extent that implementing these techniques drain limited resources (as it surely will), there is a definite risk that such a requirement will be detrimental to the quality of engineering education in the country. Note that we are *not* suggesting that individual programs, or even individual courses in these programs, should not be allowed to use assessment techniques that they find most appropriate. Just that programs should not be *required* to do so. So long as the objectives are reasonable, and so long as the program is effective in ensuring that students meet these objectives, the program should be considered successful and worthy of accreditation irrespective of the processes it uses.

The old saying goes 'the proof of the pudding is in the eating', not 'the proof of the pudding is in the recipe'. Good cooks will often alter the recipe in creative ways in preparing complex dishes. Engineering education is surely at least as creative as cooking. Requiring educators to follow documented processes and requiring them to show how these processes are functioning etc., is a recipe for disaster. It will not encourage innovation, but rather stifle it, since educators will not be able to use spur-of-the-moment creative solutions to problems they encounter. In the final analysis accreditation must attest to the product, i.e., the program objectives and how successful the program is in ensuring that its graduates meet these objectives, not the process, and we hope the engineering community will give serious consideration to redesigning EC2000 to eliminate its process-centric focus.

References

[1] ABET, EC2000 Case study: Coastal State University, Available at www.abet.org/eac.

[2] ABET, Criteria for accrediting engineering programs, Available at www.abet.org/eac.

[3] M. Aldridge and L. Benefield, A model of assessment, *ASEE Prism*, pages 22–28, May-June 1998.

[4] Task force on engineering education assessment, A general assessment framework, In *How do you measure success*, pages 17–23, ASEE Professional Books, 1998.

[5] H. Lewis and C. Papdimitriou, *Elements of the theory of computation*, Prentice-Hall, 1981.

[6] B. Meyer, *Object-Oriented Software Construction*, Prentice Hall, 1988.

[7] B. Olds and R. Miller, Assissing a course or project, In *How do you measure success*, pages 35–43, ASEE Professional Books, 1998.

[8] NEEDHA White Paper, Accreditation of engineering programs, Available at www.needha.org/acc-whitepaper.shtml.

[9] G. Peterson, A bold new change agent, In *How do you measure success*, pages 5–10, ASEE Professional Books, 1998.

[10] J.R. Stedinger, Lessons from using TQM in the classroom, *Journal of Engineering Education*, 85:151–156, 1996.