

# Proposed Change to ABET Software Engineering Program Criteria

CSAB Criteria Committee, 1 March 2014

## Introduction

The ABET software engineering program criteria supplement the Engineering Accreditation Commission (EAC) general criteria, for the evaluation of software engineering programs. As the lead society for software engineering, CSAB is preparing to propose a change to these program criteria.

Comments and suggestions should be sent to [csab@csab.org](mailto:csab@csab.org), with “SE program criteria change proposal” in the subject line.

## Proposed changes to the program criteria (February 2014)

The current draft of the proposed modifications is:

### PROGRAM CRITERIA FOR SOFTWARE AND SIMILARLY NAMED ENGINEERING PROGRAMS [DRAFT]

Lead Society: CSAB

Cooperating Society: Institute of Electrical and Electronics Engineers

These program criteria apply to engineering programs that include “software” or similar modifiers in their titles.

#### 1. Curriculum

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~~ include discrete mathematics, probability, and statistics, with applications appropriate to software engineering; computing fundamentals, software design and construction, requirements analysis, security, verification, and validation, as well as software engineering processes and tools appropriate for the development of complex software systems.

#### 2. Faculty

The program must demonstrate that faculty members teaching core software engineering topics have an understanding of professional practice in software

engineering and maintain currency in their areas of professional or scholarly specialization.

The rationale for the proposed change is:

- To replace the implied “outcome demonstration” language with “curricular topics”, to better comply with the requirement in the EAC general criteria that program criteria may only address “curricular topics” and “faculty qualifications” More information on this topic can be found below.
- Avoid repetition of “software systems”.
- Remove the reference to “supporting disciplines”, for which there seems to be no common understanding or definition.
- Remove the reference to “application domains”, since it is not clear how to translate this into a “curricular topic”.
- Make clear that, as in the IEEE program criteria for electrical engineering and computer engineering, probability and statistics must be applied to the discipline.
- Introduce computer security as a cross-cutting topic (e.g., information security, computer and network security, developing secure software) that is being given increased visibility in the current draft revision (SE2014?) of the SE2004 joint ACM/IEEE-CS software engineering curriculum guidelines.
- Replace “implementation” with “software construction”, and scramble the order of “software engineering lifecycle activities”, so the listing does not seem to imply a preference for the “waterfall” development model.
- Substitute “software engineering processes” (which can include individual, team, and organizational aspects) for “manage the development of” and remove the reference to maintaining software systems. The prior formulation might be read to imply certain “top-down” project management structures that could conflict with contemporary processes that rely on team self-management. “Managing the development of software systems” in such a hierarchical sense may not be appropriate at the undergraduate level in any case. This change is also consistent with the current draft revision to the SE2004 SE curriculum guidelines, which treats project management and evolution (including maintenance) as subtopics of software process.
- Introduce the mention of software engineering tools, since appropriate tool choice and use is an important element of software engineering.
- Restore a requirement for faculty qualifications. From 2001-2002 through 2003-2004, the software engineering program criteria required the program to demonstrate that “those faculty teaching core software engineering material have practical software engineering experience”. An earlier version of this proposal did not include a faculty requirement, but survey feedback included strong support for reintroducing one, and the CSAB Board voted on 20 February 2014 to endorse the currently proposed language. Faculty requirements are very common in engineering disciplines; of the 27 sets of EAC program criteria (2013-2014), more than two-thirds (19) include them.

## Change process

The process for making the proposed change to the software engineering program criteria is:

- [17 July 2013] Discussion of possible changes in the program criteria by the CSAB criteria committee.
- [3 November 2013] Discussion of initial proposal at a meeting of the CSAB Board.
- [13 November 2013] Further discussion of proposal by CSAB criteria committee.
- [13 December 2013] Proposal description document posted on CSAB web site, with link to an on-line survey soliciting feedback. Email notifications sent to program heads and program evaluators.
- [17-18 January 2014] Proposal presented to IEEE Committee on Engineering Accreditation Activities (CEAA), since IEEE is a cooperating society in EAC for software engineering; the committee encouraged all interested members to submit comments through the on-line CSAB survey.
- [19 February 2014] Revised proposal, incorporating constituent feedback, discussed by CSAB criteria committee.
- [20 February 2014] CSAB board approval of revised proposal.
- [March 2014] Revised proposal and explanatory text posted on CSAB web site, with email notification to program heads and program evaluators.
- [March 2014] Transmission of revised proposal to IEEE CEAA.
- [March/April 2014] Transmission of revised proposal to EAC criteria committee chair.
- [July 2014] Discussion and possible “first reading” approval of proposal by the full EAC criteria committee.
- [July 2014] Discussion and possible “first reading” approval of proposal by the full EAC.
- [October 2014] Review and possible “first reading” approval of proposal by ABET Board.
- [October 2014 – July 2015] If approved, publication for review and comment by all interested parties.
- [July 2015] Discussion and possible “second reading” approval of proposal by the full EAC criteria committee.
- [July 2015] Discussion and possible “second reading” approval of proposal by the full EAC.
- [October 2015] Review and possible “second reading” approval of proposal by ABET Board.
- [October 2015] If approved, final publication for initial use in the 2015-2016 accreditation cycle.

## History

The first version of the software engineering program criteria appeared in the “EC2000” section of the 2000-2001 Engineering Accreditation Commission (EAC) criteria, submitted by IEEE (which was then acting as the lead society). Subsequent changes were made to these criteria in the 2001-2002 and 2004-2005 cycles.

Through all this time, the software engineering program criteria (like those of most other engineering disciplines) contained “outcome” language of the form “the program must demonstrate that graduates have [a listed set of abilities]”. This type of language was in conflict with the EAC general criteria, which specified that “[r]equirements stipulated in the Program Criteria are limited to the areas of curricular topics and faculty qualifications.” (Note that this differs from the Computing Accreditation Commission general criteria, which do not contain a similar restriction.)

In the 2011-2102 cycle, the EAC made “editorial changes” to the program criteria for essentially all disciplines, substituting language like “the program must prepare graduates to [do a specified list of things, mirroring the earlier “abilities” lists]”. While this technically removed the explicit violation of the general criteria restriction, most program criteria were left with wording that sounded a lot like it was specifying student outcomes.

#### **Note on student outcomes in the EAC criteria**

In the context of the EAC restrictions on program criteria, it should be noted that the general engineering criteria explicitly address student outcomes (Criterion 3) and the need to assess and evaluate the level of outcome attainment by students (Criterion 4). For reference, the current (2013-2013) Criterion 3 outcomes are:

- (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.

While these “required” student outcomes (programs may choose to add others) are stated in general terms, it is pretty clear in practice that the intent is to interpret them in the context of a specific discipline. Thus, the absence of explicit “outcome” language in the program criteria does not imply that software engineering programs

do not need to have and assess student outcomes. Indeed, the “curricular topics” language of the program criteria provides one reference for the interpretation of the Criterion 3 student outcomes for individual programs.

While it is possible that the EAC could decide to amend the general engineering criteria, and permit engineering program criteria to address student outcomes explicitly, a change of this type would have to be proposed by the EAC Criteria Committee, with the involvement of all EAC member societies, and ratified by the commission as a whole.