

Software Engineering PEV FAQ

Last updated: Aug 6, 2008

1. What resources are available for software engineering PEVs?

The ABET web site has electronic versions of many important documents. There are two publications that should be reviewed before every visit. The first is “Criteria for Accrediting Engineering Programs” and the second is the “Accreditation Policy and Procedure Manual.” Revised versions of these documents are published each year. The new versions incorporate all changes that have received final approval from the ABET Board. The Criteria document also includes text of all proposed changes that have preliminary approval. Printed versions of these documents should be sent to you by your Team Chair. PDF versions of both documents may be downloaded from <http://www.abet.org/pev.shtml>.

Other useful documents that can be found on the same web page (see the Engineering programs section) include:

- Code of Conduct and Conflict of Interest Policies
- E60 Manual of Evaluation Process, and
- E067 PEV Workbook,

In addition, this page has a link to the secure login that enables one to change one’s volunteer biographical information, do evaluation of team chairs, and get a copy of the ABET Expense Report Form.

These resources should enable PEVs to refresh their knowledge without formal retraining. Every PEV will find it valuable to undertake a personal retraining review program before each accreditation visit (or season).

2. Where are the forms that I have to fill out?

Go to the ABET web site page <http://www.abet.org/pev.shtml> and download “E067 PEV Workbook_6-11-07.zip.” This is a zipped document. When it is unzipped all of the forms and reference material are nicely contained in a folder (and there is a Word document that permits easy access to most of the documents). This download includes almost all of the forms and reference documents that a PEV needs.

3. How do I tell ABET about myself, my availability, and my conflicts of interest?

Every EAC PEV has a user name and a password that can be used to access personal Biographical Data Form information on the ABET secure server [<https://service1.abet.org/pev/>]. Biographical Data Form information should be updated regularly. If you do not have a user name and password or if you have forgotten one or both of them, you should contact Jim Ware, jware@abet.org.

4. How should computer science courses be classified? Are they mathematics, basic science, engineering science, or engineering design?

Although it is reasonable to expect computer science courses to be engineering science and design, there is no easy answer to this question. Just because a course is labeled computer science is not a reason for placing it in any one specific category. That is the reason examination of the course display materials is an important part of the visit. One can usually assign a course to one or more categories based on the information presented in the

course syllabus, the course assignments and projects, and the course examinations. In certain cases a discussion with an instructor or some students may be useful.

Proper identification of the role of a course in the curriculum should be determined by what is actually taught, assigned and experienced in the course. It is possible (although highly improbable) that a course called Analysis of Algorithms is substantially engineering design or that a course called Senior Design Project is substantially general education. Careful examination of the course materials is very important. The course materials should support the classification made by the institution in their Self-Study.

Note also that a course labeled Software Engineering is not automatically all engineering science and engineering design. In an extreme case, a course labeled engineering could be mostly mathematics. Again, it is the responsibility of the PEV to determine what is actually taught, assigned and experienced in the course.

5. What is the role of SWEBOK, CCSE (SEEK) and other curriculum guides? How should these be used during the accreditation process?

Curriculum guides are extremely useful as resources. However, it would be inappropriate to require any institution to follow any curriculum guide exactly. We can use curriculum guides to suggest appropriate content for software engineering programs but they are not (and should not be) a final determination of appropriate content. On the other hand, an institution that has developed a software engineering curriculum without consulting any of these guides probably has some explaining to do. It is always appropriate to ask individual faculty how they have used these guides and other resources in designing their program and their courses.

Note that Criterion 8 requires a program to have “breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program.” Curriculum guides provide some indication of generally accepted community standards regarding breadth and depth. The institution needs to provide some indication of the reasons behind curriculum design decisions, indicating how the program provides both breadth and depth across the range of topics implied by the program’s title.

Note also that the curriculum guides and the Criteria require more than just a reasonable set of courses in a program. For example, Criterion 4 states “Students must be prepared for engineering practice”

6. How do I consider program objectives when evaluating a program?

ABET considers program objectives to be “broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.” Programs can use their own terminology for objectives (as well as outcomes discussed below), but they must have documented these statements for their program. In addition, they must be regularly assessing whether or not their graduates are achieving these objectives.

There are several questions related to the program’s objectives that are useful to ask.

Are the program’s objectives and the program’s curriculum compatible? The curriculum must include breadth and depth across the topics “implied by the title and objectives of the program.” It can be useful to ask faculty how they determined the courses and topics included in the program. If there is no mention of the program’s objectives, then there might be a problem.

Are the program's objectives based on the needs of the programs constituents? The needs of the program's constituents should be considered when developing a program's objectives. Note that this does not necessarily mean that the constituents have had input nor approved these objectives.

Are the programs objectives consistent with the mission of the institution? The objectives should be consistent with the mission of the college and university.

Does the institution have an assessment process that periodically documents the degree to which these objectives are attained? There should be a part of the institutions assessment process that documents to what extent its graduates are achieving its objectives.

7. How much “engineering experience” should the faculty have? How can the “engineering experience” of the faculty be determined?

The current criteria do not mandate any particular amount or type of engineering experience for faculty. However, engineering experience is one of the factors that is used to judge the overall competence of the faculty. It is important to recognize that one is judging the competence of the faculty as a whole. In addition, engineering experience is only one of nine factors that are listed in the Criteria (Criterion 5).

Interviews with faculty might include questions about the nature of their software development experience (e.g., what type of organization, what team size, what kind of product, what kind of customer, what tools were used, what were the major failures and successes, how that experience influences their teaching and the curriculum). It is reasonable to expect at least some faculty members to have experience working in large teams developing large software systems.

8. What do I do if my Team Chair thinks that software engineering is not really engineering?

This should not be an issue. Team Chairs have been trained appropriately and they should understand that engineering now encompasses many fields that were not part of engineering 20 or 30 years ago. If this problem does arise, the best strategy is to privately and quietly discuss the ways that traditional engineering practices have become an important part software engineering programs. One can explain the difference between programming and software development and show how this is similar to the difference between manufacturing/machining/fabricating and engineering. A public, confrontational, direct attack approach is likely to be unproductive and unsuccessful. It is also obviously unreasonable to expect a Team Chair to be as knowledgeable as you are about software engineering.

9. How do I deal with “Outcomes and Continuous Improvement” [Criteria 3 & 4]?

In addition to the objectives described in question 6 above, programs should have a set of outcomes that are related to these objectives. These outcomes should describe what students know and are able to do at the time they graduate. These outcomes must cover the a-k elements listed in criterion 3. Programs should regularly assess whether or not students are attaining these outcomes. In addition, the results of these assessments should be evaluated regularly. Results of the evaluation should be used to improve the program. Evidence of the assessment, evaluation and improvement should be presented by the program.

Remember that the institution is supposed to be convincing you (and the public). You should not have to do in-depth analysis or extensive data retrieval.

10. What should be done about a program that is just a computer science (or computer engineering) program with a few software engineering courses added?

You might ask some of the following questions: *Do the faculty understand the differences between software engineering and computer science (or computer engineering)? Does the software engineering experience of some faculty influence the program? Are some faculty members involved with the software engineering community? Are the objectives of the different programs actually different? Are there enough faculty members with software engineering experience? Does the software engineering faculty have control of the program?*

In general, formulating and asking appropriate questions usually resolves most issues. Using good questions is one of the best ways to develop a thorough understanding of a program. It is usually difficult to formulate and ask good questions without adequate advance preparation for the visit.

11. What about the major design experience that is required by Criterion 4?

The major design experience does not have to be a senior project, even though it usually is. It probably should be a team project. It must build on the skills and knowledge acquired in previous course work and must incorporate appropriate engineering standards and multiple realistic constraints.

12. What should be done if there are no graduates from the program?

This issue should have been handled by the ABET staff. Refer it to them.

13. This is a new program. When does accreditation become effective?

This is not your concern. If you are asked to make a recommendation about the effective date of the accreditation, you should determine when graduates who met the ABET Criteria first graduated from the program. The ABET office will use this information to make a final decision.

14. What do I do if the faculty has not really considered (a) through (k) in Criterion 3?

Be kind but firm. Gently explain what you have time to explain. Try to help the faculty understand what they need to understand. But be sure that your report identifies this as a deficiency because ignorance of a Criterion is not an excuse for not satisfying it. PEV's should be careful to apply Criterion 3 in a wholistic sense. There is no requirement that all 11 of the a-k's be attained to the same degree and there is no requirement that all students attain a-k to the same extent.

The working definition of a deficiency is that a D should be cited if the intent of the criterion in question is totally or largely unmet. Just missing out on one of the 11 a-k's does not make the criterion "totally or largely unmet."

15. How should one handle assessment when visiting a recently created program?

Waiting a year or two after the graduation of the first set of students makes it much easier to provide evidence of assessment activities. A program with only one cohort of graduates likely has not had time to do all of the assessment that the Criteria require. But there should be an assessment plan. There should be some data collection. The process by which the curriculum was developed should be documented and there should have been some review of the curriculum as it was being developed. There should also be evidence of faculty and institutional support for assessment. New programs may be in departments that already have other accredited degree programs. Programs can draw parallels between recent graduates of these other programs and SE graduates to help make the case that students are achieving outcomes.

16. What should be done before the visit?

Thoroughly read (and then re-read) the institution’s materials. There is nothing worse than a PEV who is not familiar with the contents of the institution’s reports. Fill out as much of the visit report form as possible. Do all analyses that are possible. Make a list of questions that arise and share the list with the program chair. Pay close attention to the responses of the program chair. If parts of the report are incomplete, unclear, or inadequate, notify the program chair and ask for clarification or amplification. Get as much handled in advance as you can because there will not be time to do everything during the visit.

17. How do EAC visits differ from CAC visits?

CAC	EAC
Team chair is in same field	Team chair usually in different field
Team chair handles all communication	PEV communicates with program chair
Usually two PEVs + TC for one program	One PEV for each program + TC
Usually longish exit statement by team -most everything is mentioned	Brief exit statement by TC & each PEV -only important things are mentioned

Note that some of these practices are under active review and may change in the near future.

18. How are CAC and EAC visits similar?

Surprises are avoided whenever possible. Programs are evaluated only for adherence to the Criteria. No judgments are made about an institution’s internal procedures and staff except when they are directly related to the Criteria. Final accreditation decisions are made by the Commissions, not by the team members. There is a lot of work to do in a very short time.

19. How does one handle “application domains?”

The institution needs to present appropriate information about the ability of their graduates to “work in one or more significant application domains” (Program Criteria for Software Engineering programs). If such information is not presented, then it should be requested.

20. Are there any suggested questions that PEVs could consider using?

See the attached document “Sample Discussion Questions” for some suggestions.

Sample Discussion Questions

1. How does your curriculum ensure that software engineering is presented as a broad field that draws on several different fields for its theoretical and conceptual foundations?
 2. How does your curriculum develop special domain knowledge beyond the computing discipline?
 3. How does your curriculum convey to the students the rapidity with which the discipline evolves and the need for continuing education by software engineering professionals?
 4. How do you maintain the technological currency of the curriculum?
 5. How does your curriculum prepare students for careers as international software engineers?
 6. What experiences do your students have in extracting client needs (preferably from indecisive and confused clients)?
 7. What experiences do your students have in identifying and reconciling conflicting client and system objectives?
 8. What experiences do your students have in working in various types of teams in various types of situations?
 9. Have your students been required to independently learn new technologies during their undergraduate program?
 10. What topics and courses give your students the mathematical foundations of software engineering? Why?
 11. What topics and courses give your students the computing foundations of software engineering? Why?
 12. What topics and courses give your students the engineering foundations of software engineering? Why?
 13. What topics and activities prepare your students for professional practice of software engineering? Why?
 14. Are the questions on your assessment surveys getting you accurate and informative data?
- The Software Engineering Education Knowledge (SEEK) document identifies ten knowledge areas that make up the SEEK. They are: Fundamentals (FND), Professional Practice (PRF), Software Requirements (REQ), Software Design (DES), Software Construction (CON), Software Verification & Validation (VAV), Software Evolution (EVL), Software Process (PRO), Software Quality (QUA), and Software Management (MGT).
15. How is each of the above knowledge areas incorporated in your curriculum?
 16. How successful do you consider your presentation of each of the above areas?
 17. What topics and activities prepare your students to maintain software systems? Why?

Note: These questions were prepared by one Software Engineering PEV before an EAC visit. They were submitted to the program chair before the visit. The intent was to encourage open and productive dialog during the visit.