2006-1695: ACCREDITATION -- APPLYING CMM TO SOFTWARE ENGINEERING EDUCATION

Sheryl Duggins, Southern Polytechnic State University

Accreditation – Applying CMM to Software Engineering Education

The software engineering department at Southern Polytechnic State University has decided to embark on a journey that will hopefully result in ABET accreditation. Since ABET only started accrediting software engineering programs in 2003, this is all new to us, and we find ourselves trying desperately to understand and apply the process. As anyone who has gone through this process knows, accreditation is a stamp of approval that says your department is successful at meeting its goals and has an established process for continual improvement. To be able to show this you must use assessment to collect the data, evaluation to analyze it, and incorporate the results to improve your program.

The importance of the role of process in software engineering, and of continual process improvement as we teach in CMM and now CMMI, is a deep thread running through our coursework. Yet when trying to apply those very same concepts to software engineering education, which is what accreditation is really doing, we realized we had been operating at CMM Level 1, with a set of undefined processes. We did not have stated program goals and we were not assessing how our program was doing. This paper will propose an outcomes-based process model for teaching and learning software engineering and define an initial set of maturity levels. The Maturity Process Teaching Model proposed here incorporates the ideas of operational definitions as outcomes, process-improvement, and Capability Maturity Model-based maturity levels and applies them to software engineering education. This paper explores the accreditation process as it relates to software engineering departments. It will discuss procedures that are needed to "jumpstart" the self-study and move your program up the CMM scale. Finally, the status of our efforts as well as any problems we experienced will be discussed in an effort to disseminate this information to our software engineering colleagues, since improving the quality of our programs, which accreditation helps us to do, is critical for the future of software engineering education.

1. Reality Check

The ABET literature (e.g., "Self-Study"), as well as ABET workshops all suggest that the benefits of accreditation are primarily self-awareness and self-improvement, with the "self" being one's own department. But the reality of the mission initially, is often less noble, with the actual motivation lying closer towards things like improved enrollment, greater visibility, approval from your administration, and recognition by your peers. The funny thing about the whole process is that somewhere along the line, one loses sight of the original motivations, and the most amazing thing happens: it really does become about self-awareness and self-improvement. Sure those other things would be nice to have, but ultimately, they move into a very distant second place, lagging far behind the goal of self-improvement.

One can only assume that our experience during this endeavor was fairly typical. Prior to the decision to seek ABET accreditation, our department, and more specifically, our day-to-day faculty activities relating to the teaching aspects of the job, were probably very typical and

similar to those practiced by faculty in most departments of higher education that are not either seeking accreditation or presently accredited by their field-specific accreditation body. Namely, each faculty member operated pretty much independently, since after all, we are all Ph.D.s and therefore, are experts in our fields, and know what we want and need to teach our students. The idea that we should work together with the other faculty members in our department to decide what we want to teach our students not only never occurred to us, but if someone suggested it, we would probably have taken offence. ("What does he or she think I'm doing in my classes anyway? I don't need him or her telling me what to teach in my classes.") We had university, school, and department mission statements, but they were all very vague, and didn't in any way affect how or what we taught; nor was there any accountability with respect to those mission statements. They were out there as frameworks within which we operated, but we did not have any means to show that we were actually working towards those goals. In essence, we were all designing our own courses and teaching exactly what we wanted to teach without any idea of how well we were doing collectively. Of course we did work together to design our curriculum and we frequently updated our curriculum based on the literature in our field, (e.g., Computing Curriculum for Software Engineering, SWEBOK, etc.). But what we did *not* do was identify any objectives, or goals, for our department, nor any goals or outcomes for individual courses, and consequently, we were unable to assess whether or not our students as a whole were learning what we wanted them to learn.

When we first began learning about outcomes-based assessment, it was daunting and overwhelming. The whole vocabulary was new to us, but more importantly, the underlying philosophy was new to us. There is a very steep learning curve at first. But the biggest hurdle was realizing that the day-to-day way you are doing business must change. It is a bit humbling when you become self-aware and finally comprehend that maybe you aren't doing things as effectively as you once believed. More specifically, you realize that accreditation is very similar to the Capability Maturity Model (CMM): both are used to assess the quality of the processes of an organization with a focus on using the obtained data to improve those processes. The importance of the role of process in software engineering, and of continual process improvement as we teach in CMM and now CMMI, is a deep thread running through our coursework. Yet when trying to apply those very same concepts to what we do, software engineering education, which is what accreditation really is about, we realized we had been operating at CMM Level 1. We had undefined processes, we did not have stated program goals, and we were not assessing how our program was doing. In other words, we did not practice what we taught; seeking ABET accreditation meant we had to turn our shop around like any business operating at CMM Level 1, and learn how to apply continual process improvement in our department.

2. The Role of Process

One fundamental idea that is crucial to include in our software engineering curriculum is the role of process. We teach our students the importance of process in software engineering; however, we typically fail to incorporate the very fabric of process in the way we teach our students [1]. Quality Assurance (QA) is based on the idea that improving the process by which a product is developed will result in an improved product [2]. The initial work in this area, known as Total Quality Management (TQM), was done by Deming [3] and was applied to the manufacturing community. Since his initial contribution, QA has been applied to all types of endeavors, with

the underlying assumption that having a defined and managed process will improve the resulting product. With respect to software engineering, the Software Engineering Institute (SEI) has developed the Capability Maturity Model (CMM) to assess the quality of the software processes of an organization [4].

The concept of process, or specifically, software process, is fundamental to the notion of software quality assurance. Humphrey [5] defines the software process as "the sequence of steps required to develop or maintain software" and a software process definition as "a description of this process." The software process definition should guide the individual software engineers as they work. An organization that has a well-defined process description can better coordinate the work of individuals and track their progress. As new methods are identified, they are incorporated into the process definition, facilitating learning by allowing new projects to build on prior experiences.

The Capability Maturity Model provides a way for organizations to assess the capabilities of their current software processes and to focus on improving those processes. The CMM defines five levels of progressively more mature process capability [4].

"1. Initial: The software process is characterized as ad hoc and occasionally even chaotic. Few processes are defined, and success depends on individual effort.

2. Repeatable: Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier success on projects with similar applications.

3. Defined: The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

4. Managed: Detailed measures of software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.

5. Optimizing: Continuous process improvement is enabled by qualitative feedback from the process and from piloting innovative ideas and technologies."

Most organizations are at level 1. The success of the CMM has focused attention on the successful role that a well-defined process can have on an organization. Organizations that effectively communicate and manage their processes operate more efficiently, and with experience, are more likely to improve those processes. In the terms of the CMM, those organizations mature, and reach a higher level of capability.

3. CMM and Outcomes-Based Teaching and Learning

The role of process as discussed with respect to a software organization, and CMM in particular, presents a model of successively improving stages of maturity. One of the underlying

assumptions of CMM is the idea of a defined process, or specifically, an operational definition, which Deming [3] refers to as something everyone in the organization can communicate about and work toward. Some of the benefits of using operational definitions [5] are that they enable effective communication, they enhance understanding, they facilitate reuse, and they support process evolution. ABET has utilized the concept of operational definitions in its use of the term "outcome", and when referring to "outcomes-based assessment", the underlying ideas are very similar to those of CMM.

While CMM was originally intended to apply only to software organizations, this author is proposing that the ideas can be applied to other types of processes; namely, to outcomes-based teaching and learning. Applying CMM to the process improvement ideas underlying ABET accreditation, a new outcomes-based model for teaching and learning software engineering emerges. It is suggested that the five levels of the CMM could in a very broad sense, be applicable to the maturity of outcomes-based teaching and learning as well as to software organizations. Thus the five levels: 1. Initial; 2. Repeatable; 3. Defined; 4. Managed; and 5. Optimized, of CMM, could be redefined to apply to the maturity levels of outcomes-based teaching and learning demonstrated by a software engineering department.

Paralleling Humphrey's ideas on software processes [5], a department that uses a well-defined outcomes-based process approach to teaching will be able to assess and track the learning progress of its students. Furthermore, as the department matures and identifies new, successful, teaching and learning activities, they will be easily and effectively incorporated into its outcomes -based teaching, which facilitates student learning as well as improvement by the teacher and the department, by creating new learning activities that build on prior experiences.

The Maturity Process Teaching Model (MPTM) can be used to assess the process maturity level of departments and to provide a technique for outcomes-based improvement. While the MPTM is a general purpose model and can be used for any type of outcomes-based process assessment, it is defined in terms of ABET assessment and accreditation, but the ABET specifics could easily be changed to fit any accrediting body. Like CMM [4], the MPTM has five levels of increasingly more mature outcomes-based process teaching. The levels are as follows.

1. Initial: The teaching and learning process is characterized as typically ad hoc. There are no defined learning outcomes, no program goals, and there is no process in place for improvement.

The department has no specified goals or objectives; individual courses have no specified goals or objectives; professors teach each course in whatever manner he or she chooses with no mandated collaboration with colleagues or specification of material to be covered in that course; individual students are given the opportunity to learn what his or her professor deemed important and chose to cover; after taking a course there is no process in place to ascertain what the student learned in that particular course and thus no way to know if any future changes actually improve student learning in that course; and finally when a student graduates there is no process in place to identify what knowledge the student leaves the institution with nor any mechanism to gauge whether or not the graduated students are meeting the needs of industry.

2. Repeatable: Basic outcomes-based processes are established to define and track the effectiveness of the program.

- The program has a defined mission that corresponds to the college and university mission statements.
- The program has defined program objectives that satisfy the departmental mission statement and describe those things their graduates should be able to know or do as practicing engineers.
- Program outcomes or learning outcomes are defined that describe what things students should know or be able to do at the time of graduation.
- The program outcomes must be mapped onto the program objectives as well as the ABET required outcomes A through K.
- Each course in the program has defined course goals that describe what things students in that course should have the opportunity to learn.
- The courses in which students get the opportunity to learn the program outcomes are identified.
- Both course goals and program or learning outcomes identified for each course in the program are mandated as part of the course syllabus that must be covered each time the course is offered.

3. Defined: Teaching and learning activities are documented and integrated into a defined outcomes-based process for all courses in the department. All courses incorporate defined performance criteria, use standardized and agreed upon syllabi, and utilize similar strategies of assessment based on a standard outcomes-based process for teaching and learning defined for the program.

To show how each program outcome relates to Criterion 3, it is necessary to create performance criteria that define each of the program outcomes. Performance Criteria define *what is the acceptable standard of performance?*

- Performance criteria indicate concrete measurable expectations.
- Performance criteria are developed from program outcomes.
- Performance criteria indicate what concrete actions the student should be able to perform as a result of participation in the program and state minimum criterion for evaluation.
- Performance criteria identify the knowledge and skills necessary for the mastery of program outcomes.

4. Managed: An outcomes-based process for teaching and learning is utilized and outcomes are quantitatively assessed and evaluated.

The assessment process must demonstrate that the outcomes important to the mission of the institution and the objectives of the program (including Criterion 3 a – k) are being measured. Programs must have an appropriate assessment process in place that produces documentation that demonstrates that students have achieved each and every item listed in (a) – (k). Each program measures their program outcomes by assessing performance criteria which yields quantitative data which is documented and then analyzed using their defined assessment process.

As part of these documented assessment results, programs must collect evidence supporting their data. At this level of maturity, programs must have a standard assessment process in place that includes keeping copies of student work according to an agreed upon process. Further, the assessment process must include evaluation of the collected data at the program level as well as at the course level.

5. Optimizing: Continuous outcomes-based teaching and learning process improvement is enabled by frequent qualitative feedback assessing the process, and from incorporating innovative ideas and experiences in the process.

To have an optimized assessment process, the documented assessment results provide evidence that results are applied to further development and improvement of the program in addition to demonstrating the achievement of each program outcome important to the mission of the institution and the objectives of the program, and the outcome requirements of Criterion 3 a – k. To be granted ABET accreditation, institutions must be at MPTM level 5.

For decades, education literature has been promoting the use of operational definitions for evaluating teaching effectiveness. CMM has shown that organizations that communicate effectively and utilize well-defined processes are more productive. Fundamental to CMM and QA is the notion that process improvement based on experience is possible, and as organizations mature, they attain a higher level of capability. The Maturity Process Teaching Model proposed here incorporates the ideas of outcomes as operational definitions, process-improvement, and CMM-based maturity levels and applies them to outcomes-based process teaching and learning. While the MPTM is a general purpose outcomes-based process model, it is more specifically proposed as a model to be utilized in preparation for readying a program for ABET accreditation.

As the level of maturity increases from one to five in the MPTM, the teaching process becomes more effective. The teacher is better able to create learning situations that facilitate the student learning process, and is able to do this in a defined, repeatable, and measurable way.

4. Getting Started for ABET Accreditation

Ultimately, what ABET will be looking for is to see that your software engineering department has reached level 5 in our MPTM. You need to have in place processes for continuous outcomes-based teaching and learning and you need to have quantitative and qualitative feedback which you utilize to improve the process.

The first thing you need to do is to visit www.abet.org to see the available resources and to find a workshop to attend. They will tell you to have defined, and measurable program objectives, which are things you want your graduates to be able to do or know three to five years after graduation, and program outcomes, which are things you want your students to know at the time of graduation. Both of these have to be written so they are measurable, (like software performance requirements). Then you must decide where in your curriculum students are getting the opportunity to learn those program outcomes. This is best done by a curriculum map of outcomes to courses. I suggest starting with the courses only in your department. Eventually you must map the entire curriculum, but that can wait a bit. After you decide in what courses

students will be given the opportunity to learn those outcomes, then you must decide how you are going to measure what they learned. There are a number of assessment methods to choose from that ABET will point you to. We chose to use a fairly new assessment method that we found to be very helpful in getting everyone involved in the assessment process called FCARS described in the next section. To help you get started, the following is a high-level description of the process we used.

Our Assessment Process:

1. Define Program Objectives
2. Define Program Outcomes
3. Develop Performance Criteria
    - What minimum level of performance by graduates will be acceptable?
4. Select Assessment Methods
    - What Assessment Methods will be used for each objective/ outcome?
5. Define the Timelines and Feedback Channels
    - Who gets what information and when?
6. Evaluate the Feedback
    - What does it say?
    - Is it significant?
    - What action is indicated?
7. React to the Evaluation of Feedback
    - Implement program revisions indicated by the evaluation of feedback

To further guide you in getting started, an example taken from SPSU's objectives and outcomes will be given to demonstrate one mapping between objectives, outcomes, and Criterion 3 a-k.

ABET Outcome Requirements of Criterion 3 c and e:

> (c) an ability to design a system, component, or process to meet desired needs
>
> (e) an ability to identify, formulate, and solve engineering problems

were mapped onto our following Program Objective:

> *Possess broad and solid foundations in software engineering concepts and methodologies, computer programming, and computing environments so that they can rapidly adapt to changes in technology and engage in life-long learning.*

This Program Objective is mapped onto several of our Program Outcomes, including the following:

Program Outcome #3: **Demonstrate the ability to gather, analyze, develop, verify and validate artifacts of software systems (**Satisfies Criterion 3.c & e)

To assess how well our students are learning this outcome, we defined the following performance criteria:

Performance Criteria:
Students will demonstrate the ability:
1. To apply requirements elicitation techniques and requirements analysis processes to a software system.
2. To use various methodologies and tools to analyze and develop artifacts of a software system.
3. To verify and validate software system artifacts.

Each time any course that includes Program Outcome 3 as part of its learning outcomes is taught the professor must assess student performance on the preceding criteria. This ensures that all sections of the course stress the topics and evaluate student performance using the same criteria.

5. FCARS – Faculty Course Assessment Report

The Faculty Course Assessment Report, or FCAR, is an assessment methodology that was developed by John K. Estell in the Electrical and Computer Engineering and Computer Science Department at Ohio Northern University[6]. It played a very important role in our evaluation and assessment process as it provided us with several important features. Firstly, it provided us with a standard format for our assessment reports for use in ABET criterion 3 program outcomes assessment. Secondly, it kept faculty actively engaged in and focused on assessment and evaluation as part of an on-going continuous process. Each faculty member teaching a software engineering or a computer science course produces an assessment report (FCAR) that assesses both student outcomes for the course as well as student performance for specific program outcomes that are identified as being supported by that course. Additionally it contains modifications made to the course, instructor reflections about the effectiveness of the course, and suggestions for future improvement. Thirdly, it assists in program outcome assessment by incorporating the raw data in the FCARs, which are then grouped together by program outcomes and the relevant sections can be evaluated. This allows program-level assessment to be done by evaluating relevant sections of groups of FCARS rather than processing raw student performance data. Fourthly, it gives immediate feedback to the next professor teaching a particular course about prior offerings of the course and any suggestions on improvement can be incorporated, which facilitates continuous improvement at the course as well as program level.

Each course has a Course Description that identifies the course outcomes for the course as well as the program outcomes supported by the course. Using the mapping of program outcomes to courses, the faculty decided which specific courses we would use to assess the program outcomes. The FCAR uses the process for course outcomes assessment to assist in the program outcomes assessment process for those courses in which we are collecting data.

Assessment for student outcomes for the course is done by assessing each course outcome separately. Rather than including every item on every test or project, instructors select a specific set of items to assess (e.g., question 4 on the midterm). Documentation is given stating what items were used for the assessment and the results of the assessment.

Assessment for student outcomes for the program is done by assessing each program outcome being measured separately. Just as in course assessment, rather than including every item on every test or project, instructors again select a specific set of items to assess relating to the program outcome. Documentation is given stating what items were used for the assessment and the results of the assessment.

A template for an FCAR follows. For detailed information on how to use it, please see Estell[6].

Faculty Course Assessment Report (FCAR)

Department and Course Number:
Course Coordinator:
Course Title: **Total Credits:**
Current Catalog Description:

Grade Distribution:

Modifications Made to Course:

Course Objectives:

Course Outcomes Assessment:

Program Outcomes Assessment: *(This one includes an example.)*

Learning Outcome #3: **Demonstrate the ability to gather, analyze, develop, verify and validate artifacts of software systems**

Performance Criteria:
Students will demonstrate the ability:
1. To apply requirements elicitation techniques and requirements analysis processes to a software system.

Source of Assessment: Elicitation Assignment
(Followed by the assessment data.)

Student Feedback:

Reflection:

Proposed Actions for Course Improvement:

6. Conclusion

We had our ABET visit in December and are awaiting the initial report which we expect in February. We spent two years preparing for the visit, and I strongly suggest taking at least three

years to get everything ready. It was a tremendous amount of work, and as ABET coordinator, I spear-headed the job. Just getting everyone up to speed on the process we decided to use, and instituting all the changes in the way we do our business on a day-to-day basis was really a paradigm shift (to use an overused word). You must have a complete cycle before you have the visit. That means have some outcome defined, measured after it's been used, analyze the data and find a problem or something needing change, decide on what to do and make the change, use the change and collect data after the change has been utilized for at least a term, and then evaluate the new data. That's a cycle and it is necessary to be able to show that you have a continual improvement process in place. Defining all your objectives, outcomes, performance criteria; mapping all the outcomes to courses, and outcomes to objectives; and deciding what assessment methods you are going to use, when you will use them, how you will collect data, and who will evaluate the data; all is necessary before you actually begin the first round of the process. We immersed ourselves into the process and tried to start everything at once. It resulted in a lot of process improvement before we knew what we were doing. I attended three workshops at ABET which I highly recommend doing. They helped get things straight before we implemented them. Expect to be confused, overwhelmed, and exhausted as you work your way through the process. But watch out for something else that happens along the way. You also become part of a team that might not have existed before; you learn to work with your colleagues; and most importantly, you find out that your students are learning what you want them to learn and that they are highly valued in industry after they graduate. It is a lot of work but the payoff is there. There is actually a lot to this idea of continual self-improvement, especially when you get to see the payoff: successful software engineers who graduated from your program.

In summary, an initial process-based model of teaching and learning was proposed based on the CMM and ABET accreditation for process improvement. The role of process in Quality Assurance as it relates to software engineering, and the process-based Capability Maturity Model were examined and applied to the outcomes-based teaching and learning process model described. The Maturity Process Teaching Model was presented, and the five levels of maturity were defined. Tips for getting started on accreditation were presented and a new assessment method, FCARS, which we found particularly useful, was described. Finally both the status of our accreditation efforts and the benefits we have received were discussed.

References

[1] Duggins, Sheryl (2002) "Process Teaching and Learning in Engineering Education", in Proceedings of the 2002 American Society for Engineering Education, Montreal, Canada, June 2002.

[2] Wheeler, S. & Duggins, S. (1998) "Improving Software Quality", in Proceedings of the Southeastern ACM Conference, Marietta, GA, April 1998, pp. 300-309, ACM, New York, NY.

[3] Deming, W. E. (1986) Out of the Crisis, M.I.T., Center for Advanced Engineering Studies, Cambridge, MA.

[4] Paulk, Mark C. et.al. (1993) "Capability Maturity Model for Software, Version 1.1", Software Engineering Institute Technical Report, CMU/SEI-93-TR, February 1993.

[5] Humphrey, Watts (1995) A Discipline for Software Engineering, Addison-Wesley.

[6] Estell, John K., "Faculty Course Assessment Report (FCAR) Instructions," Best Assessment Processes VI, Rose-Hulman Institute of Technology, 2004, pg. 23-28.