Development and Utilization of a Process for Incorporating Constituent Feedback Into Curriculum Improvement

Barbara D. Gannod, Gerald C. Gannod / and Mark R. Henderson
Division of Computing Studies / Dept. of Industrial Engineering
Arizona State University – East / Arizona State University – Tempe
Mesa, AZ 85212 / Tempe, AZ 85287
bgannod@asu.edu, gannod@asu.edu, Mark.Henderson@asu.edu

Abstract

The ABET accreditation criteria require that programs follow a process for defining and refining program outcomes and objectives. The process is a continuous one in which feedback is used to periodically evaluate and modify objectives based on the ever-changing needs of program constituents. Most programs agree that students, faculty, and industry professional are members of their constituencies. However, the processes for soliciting feedback from constituents, as well as the processes for incorporating feedback into curriculum (and program) improvements vary widely. This paper describes a portion of the process used at Arizona State University (ASU) to provide feedback about the Bachelor of Science in Engineering (BSE) in Computer Systems Engineering (CSE).

The Consortium for Embedded Systems (CES), a partnership between Intel, Motorola, and ASU, sponsors a structured, for-credit internship program for students in the BSE program. During the late Spring and early Summer of 2004, a committee of ASU Faculty developed and utilized a process for obtaining feedback from students involved in the CES internship program, industry mentors, and faculty that contribute to BSE curriculum. The information gathered during that process was used to assess how well the BSE program is currently meeting the needs of student and industry constituents.

This paper describes a continuous curriculum improvement process, SAGE, that is based on an engineering design process and experiences in applying the process to the BSE program. First, we describe the SAGE process and focus on the portion of the process used to solicit feedback from student interns, industry mentors and faculty. Next, we provide examples of the results from analyzing the feedback obtained. We then provide observations about factors that contributed to the quality of the information gathered as well as the strengths and weaknesses of the process. In particular, we describe which types of activities led to the most valuable feedback and which activities resulted in less useful outcomes. Finally, we discuss future directions including how this process can be modified for future use both at ASU and in other programs as well.

1 Introduction

In 2000, the Accreditation Board for Engineering and Technology (ABET) changed its evaluation process from assessment of curriculum *content* to assessment of program *outcomes*, primarily student outcomes, which is more difficult. This shift in the target has turned out to be a huge adjustment for program administrators. What had been a de rigueur process for the past 50 years now presents a challenge of obtaining and evaluating information never before formally considered by most universities. ABET specifies that a successful engineering program will produce students who can meet a set of defined requirements,[1] including difficult-to-measure capabilities such as functioning on a team and interest in life-long learning. Now universities must measure “what is learned instead of what is taught,”[1] which results in several problems: 1) departments have not traditionally measured student outcomes so no process or data exists, 2) some of the criteria are difficult to define, let alone measure, 3) it is not clear who should evaluate what students have learned: faculty, students, employers or someone else. Thus, all university engineering, computing and technology programs face a common problem: how to achieve program accreditation while simultaneously measuring student outcomes and performing continuous curriculum improvement.

While these problems are common to universities across the country, no single process for addressing the problems has been suggested or widely adopted. Consequently, individual programs are left to create and follow their own *stovepipe* process for assessing outcomes and continuously improving curriculum. Accreditation activities are often viewed as an *end*; they are necessary to ensure that a program is certified as having some level of quality. While ABET identifies what artifacts must be in place in preparation for evaluator site visits, the *means* for achieving the *end* - e.g., specific techniques and processes for collecting and assessing the quality of a program - are left for individual programs to determine. In recent communication with the outgoing chair for ABET CAC, it was indicated that approximately 50% of the issues (ranging from deficiencies to concerns) for any given program within the scope of CAC are related to assessment[2].

In this paper, we describe a curriculum assessment process used at Arizona State University (ASU) to provide feedback about the Bachelor of Science in Engineering (BSE) program in Computer Systems Engineering (CSE). Specifically, the process was designed to measure the gaps between stakeholder (e.g., student, employer, and faculty) desired knowledge and current curricula. The assessment process is part of the overarching continuous improvement process. In particular, the products produced by the assessment process are being used to redesign BSE curriculum and program requirements.

In the Spring Semester of 2004, the authors served on a *Gap Analysis* team whose primary task was to determine the effectiveness of a Motorola and Intel funded curriculum grant program for embedded systems in the Department of Computer Science and Engineering (CSE). The team consisted of faculty members both internal and external to CSE. To prepare for the analysis, the team defined a process for performing interviews of focus groups and set goals for identifying the quality of the current program and desired content that could be used to improve the BSE program in CSE. Interview data was collected from interns and industry mentor focus groups during an internship presentation day, and data was collected from CSE faculty at a subsequent meeting. The interview data was analyzed by mapping course content to student and mentor concerns and then identifying where significant gaps existed. This analysis was facilitated by a technique that was developed to map and cross-reference desired knowledge to course objectives and outcomes as well as their relation to degree objectives and outcomes.

Based upon the analysis of the data, a number of conclusions were identified and reported to the Consortium for Embedded Systems (CES)[3] board consisting of representatives from ASU, Motorola and Intel. These recommendations focused on identifying gaps within courses, degree programs, and curriculum content and were subsequently used to enact improvements in the BSE degree program.

The remainder of this paper is organized as follows. Section 2 describes the assessment process defined and used by the Gap Analysis team. Section 3 provides examples of the products that resulted from the assessment process. In Section 4, we give several observations that were made while using the process and analyzing the feedback. We also point out some strengths and weaknesses of the process. Section 5 concludes the paper and makes suggestions for future work.

2 Process

In this section we describe the *Stakeholder Assessment and curriculum Gap Elimination* (SAGE) process that was developed to assess the BSE program. The process is based on the engineering design process shown in Figure 1. The steps of the engineering design process are as follows:

1. Identify Needs – a requirements elicitation step whereby needs are identified using one or more of a wide variety of techniques
2. Establish Target Specifications – a requirements analysis step that formally defines identified requirements
3. Generate Concepts – a product conceptualization step that identifies potential solutions based on the target specifications
4. Select Concepts – a product selection step that narrows down potential solutions into those best suited to meet needs
5. Prototype and Test – a product development and testing step
6. Launch – a product release step

While the diagram shown in Figure 1 depicts a linear, waterfall-like process, the various steps can include feedback loops that allow for validation that concepts, products, prototypes, etc. meet needs and specifications identified in earlier stages of the process. Ultimately, the product of the SAGE process is an improved curriculum. We contrast this with the engineering design process, which can involve development of new products rather than improving and evolving existing ones. As such, the SAGE process assumes that a degree program has already been established and that objectives, outcomes, and topics for courses within the curriculum have been defined. Ideally, the program uses a standardized style for documenting course syllabi as would be exemplified by templates used for ABET accreditation.

As shown in Figure 2, the SAGE process uses the same basic process steps in the assessment and improvement of curricula with the following modifications.

1. Focus Groups – requirements for a curriculum are elicited using focus groups made up of constituents from a number of stakeholder groups including students, faculty, and potential employers
2. Identify Gaps – gaps between existing courses and needs elicited from focus groups are identified; these gaps are categorized to identify importance
3. Identify Solutions – recommendations and potential solutions for improving curricula are suggested
4. Request New Course Development or Course Improvement – choose solutions (e.g., new course development or improvement of existing courses) that meet needs identified in Steps 1 and 2
5. Pilot Courses – offer courses in pilot offerings
6. Launch – establish pilot courses as fixtures in the degree program

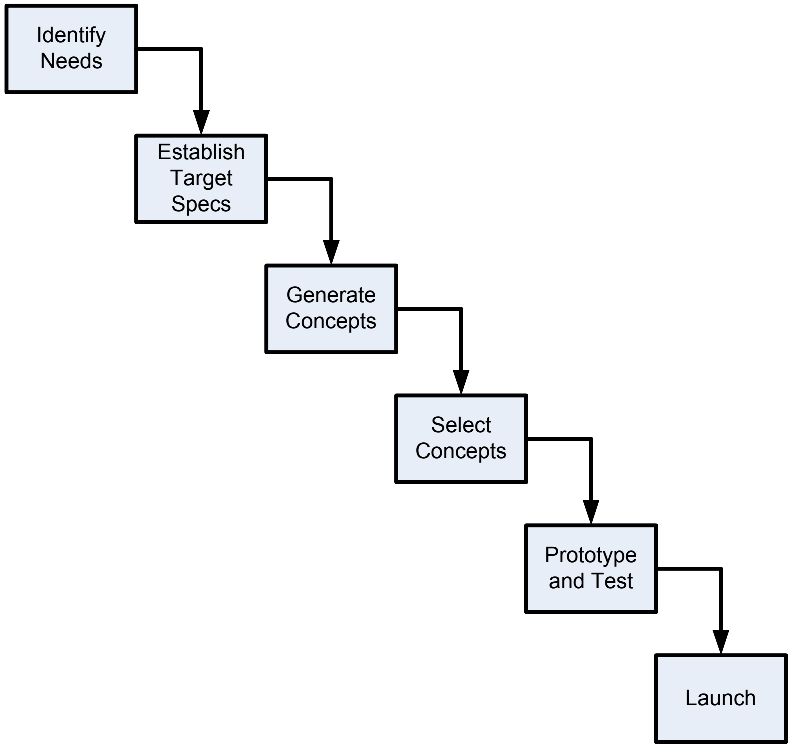

Figure 1 Engineering Design Process

While it is commonly expected that assessment of a curriculum using the SAGE process will be enacted solely within a department, there is opportunity for having independent assessment teams perform certain aspects of the process. The first two steps of the SAGE process are steps that can be performed by an independent assessment team while steps 4 – 6 are expected to be enacted by a department. Step 3 is a shared step that must be performed both by an assessment team and the assessed department.

3 Experience and Examples

In this section we describe our experiences in applying the first three steps of the SAGE process to assess embedded systems degree programs at Arizona State University. The approach was applied primarily to the Computer Systems Engineering (CSE) program and secondarily to the Computing Studies (CST) program. Specifically, we describe activities that were performed in each of the following areas:

- Constituent and Focus Groups
- Needs and Gap Analysis, and
- Identification of Solutions

Sections 3.1 and 3.2 cover methods used to collect data and analyze data, respectively. Section 3.3 discusses steps taken to provide feedback to the CSE department and their subsequent actions.

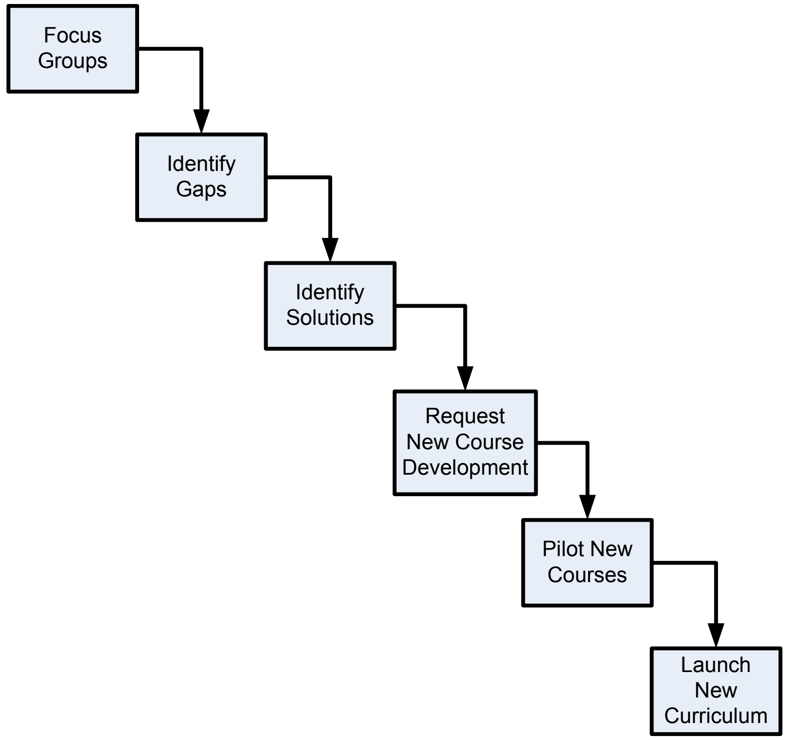

Figure 2 SAGE Process

3.1 Constituents and Focus Groups

In the context of the SAGE process, the primary purpose of focus groups is to elicit suggestions and feedback concerning a degree program. The impetus for concentrating on focus groups was rooted in the ABET process and its focus on constituents. It was our goal to make constituent feedback a primary vehicle for curriculum improvement rather than an afterthought.

We initially piloted the SAGE process at Arizona State University as an independent team assessing curricula related to an embedded systems degree program. Focus groups identified the topics and areas of study that should be in the embedded systems curriculum but were missing or under-emphasized. The assessment team then determined the desired content and compared it to the current curriculum content to find the material that should be added to conform to the stakeholders' wishes. The stakeholders in this case were:

1. Students, both current undergrads and recent graduates
2. Faculty members within the current program and also in departments peripheral to embedded systems
3. Industry which hires our graduates and provides internships for the undergraduates

The gap analysis required that the following information be audited: 1) list of desired topics in the embedded systems curriculum that are currently in the program (the CURRENT model) and 2) A definition of the desired curriculum (the DESIRED model). The difference between these two lists defines the current "gaps" in the curriculum in terms of topics taught and expectations of students and graduates. A specific emphasis is placed on the internship program and is intended to highlight a strategy for making the curriculum representative of the types of educational outcomes that ***interns*** should possess. In addition, a further emphasis was placed on those education outcomes or topics that industry expects ***graduates*** of the embedded systems curriculum to possess.

In general, we asked the student groups to give the following information:

- CURRENT curriculum: topics that they have studied in courses that were the most important in their internship jobs
- DESIRED curriculum: topics they have not covered in the curriculum, but see as valuable in their internship jobs.

We asked the mentors and faculty in two separate groups to tell us skills that are valuable and reasonable for an Undergraduate intern to know when entering the internship and those expected from new Graduates.

The two student groups consisted of 5-6 students each for a 1.5 hour session. The procedure was to introduce the goal of the group and then proceed to gather information/comments on each of the following questions:

1. *What course topics and skills were most useful during your internship?*
   - *Consider technical and non-technical skills*
   - *Give specific courses when possible*
2. *What topics or skills are not taught in the curriculum but are needed in preparation for an internship/employment?*
3. *What were the most valuable contributions from your involvement in the internship program to your educational experience?*

In order to ensure equal participation by all students, the Affinity Process was used.[4] The Affinity Process helps organize a large set of items into a smaller set of related items in an environment that allows all participants to feel free to contribute equally. The guidelines are:

1) The rules of brainstorming are followed but each idea is written (in 7 words or less, including a noun and a verb) on a self-adhesive Post-it note or card
2) Team members silently move the Post-it cards around to form closely-related idea groups
3) If disagreement exists when grouping, make copies of the contested card and place in more than one group
4) Label each group with a header card, which clearly identifies and reflects the theme of the cards
5) If there are single idea cards that don't fit well with the other ideas, have the team decide if they should be kept (they may be excellent ideas thought of only by one person).

This process was used on each of the first three questions. The results consisted of several grouped post-it notes, each with a student-generated summary header card. The individual post-it note topics will be called secondary items and the header card summary topic will be called primary topics. Multi-voting was used to rank the resulting header card topics.

Two additional focus groups were convened, one exclusively with intern industry mentors and one exclusively CSE faculty. We invited all participants to evaluate the embedded systems curriculum listing curriculum outcomes for two groups of engineers: interns and new college graduates. 15 mentors and 4 faculty participated in these sessions.

The mentor focus group consisted of 6 intern mentors from Intel and 9 from Motorola. Two additional Intel mentors provided a written response to the questionnaire but were unable to attend the interactive session. Instead of the Affinity Process, we asked the mentors the following questions:

1. *What skills are needed for interns to be successful in their internship responsibilities?*
2. *What skills are needed for new graduates?*

Answer these two questions for each of the following topics:

- *Programming Languages*
- *Tools*
- *Technologies*
- *Problem solving, Trouble shooting, Debugging*
- *Realtime concepts*
- *Reliability, fault tolerance*
- *Teaming, communication*
- *Other areas?*

The faculty members were asked the same questions as the mentors, but the responses turned out to be more general than those of the mentors and, therefore, were more useful in confirming the student and mentor concerns rather than itemizing them for a ranked analysis.

3.2 Needs and Gap Analysis

The second major step of the SAGE process is the most intensive: Needs and Gap Analysis. The analysis is focused on four major activities:

1. Student-Centered Needs Analysis
2. Mentor-Centered Needs Analysis
3. Course-Centered Analysis
4. Gap Analysis

The remainder of this section provides an example of each kind of analysis that was performed during our assessment of the embedded systems offerings for the CSE and CST programs at Arizona State University.

3.2.1 Student-Centered Needs Analysis

As mentioned earlier, students were asked to identify the *current* and *desired* curricula with respect to the experiences gained in their internships. With respect to the desired curricula, data was obtained by asking interns to identify (in rank order) those major topics or skills (and refinements of them) that the students believed *would have* most helped them in their internships but were not learned in their course work. In order to assess desired knowledge, we cross-referenced the responses received against the CSE and CST program offerings. Furthermore, we cross-referenced the responses against the mentor concerns obtained during the mentor focus group meetings. In the interview session, the students identified eighteen (18) major topics. These topics were then ranked according to importance. Table 1 shows the topics identified and their rank.

Note that some of the topic areas received a large number of votes while others received none. Those that received none indicate a topic area that likely arose out of the brainstorming activity but were later found to be less critical components of their undergraduate embedded systems

Topic	Votes
I. C/C++ and other non-Java languages	18
II. Operating Systems and Linux	17
III. Device Drivers	15
IV. Networking and Network programming	14
V. Hardware	13
VI. Architecture	11
VII. Using Linux and Scripting	6
VIII. Embedded Systems	6
IX. Compilers	5
X. S/W practices	5
XI. Analysis	3
XII. Licensing/Copyright	2
XIII. Project management	2
XIV. IT	2
XV. Technical communication	0
XVI. Business	0
XVII. International Interaction	0
XVIII. Video Game Design	0

Table 1 Desired Topics Identified by Students

instruction by the students. Those with a high ranking indicate those areas that were of most importance and required, from the perspective of the students, more significant treatment in the curriculum.

A sample of one of the topic areas is shown below in Table 2. Specifically, the table shows data regarding the area ranked highest by the students that were interviewed: C/C++ and other non-Java languages. The discussions of this topic area centered around the need to provide a stronger foundation for programming in C and C++. The first column of the table lists major topic areas and their rank votes in parentheses. The Roman numeral at the beginning of the topic area is used to index the major and minor topic areas and skills. The second column of the table is the minor topic areas or skills that were identified as being both necessary and missing from the BSE program. The rows in this column are numbered starting with "a." and in increasing alphabetical order. The third column of this table indicates whether the minor topic area was a concern also identified by industry mentors, while columns four and five indicate courses in the BSE and CST programs that address the topic, respectively.

All topic areas were analyzed in the same format. For courses that are listed in columns four and five the highlighting has the following meaning. Courses in bold font are required courses, and all other courses are technical electives. Interpretation of the table is as follows. If a topic is covered by some required course, then the need is being met and a gap exists only in the depth of coverage (a course gap). Another potential gap exists if the course is a technical elective and is not taken by a student (a program gap). Specifically, the topic is only guaranteed to be covered if the course covering the topic is required (bold face). Otherwise, coverage is not guaranteed (either because the topic is not covered in a core course or because the course is a technical elective, of which students have limited access to due to the constraints of the BSE program).

With respect to data specifically found in Table 2, a number of issues can be raised. First, C and C++ were identified as the primary languages in which students felt they lacked adequate training. The importance of this topic was reinforced by information received from industry

		Industry Mentor Related Concern	CSE Program	CST Program
I. C/C++ and other non-Java languages (18 votes)	a. Fundamentals of C/C ++ programming	✓	**CSE 240**	**CST326,** CST494
	b. Pointers in C/C++	✓	**CSE 240,** CSE 494 RT	**CST326,** CST494
	c. Object Oriented Programming in C++	✓	**CSE 240**	**CST326**
	d. inheritance	✓	**CSE 240**	**CST326**
	e. Advanced C/C++ programming	✓	CSE 494 RT	**CST326**, CST494
	f. Pointers to functions (C/c++)	✓	CSE 494 RT	
	g. Understanding C code/low level use of a high level language	✓		CST494
	h. Standard template library	✓		
	i. Writing to virtual/physical memory in C/C++	✓	CSE 494 RT	CST494
	j. Visual Basic			**CST326**
	k. Writing GUI			

Table 2 Excerpt from Student Needs Assessment Table

mentors. Note item h – Standard Template Library. The item suggests that the topic would be useful as identified by both interns and mentors, but that the topic is not identified as being covered in either program. This constitutes a *Curriculum Gap* (e.g., the content does not currently exist).

3.2.2 Mentor-Centered Needs Analysis

As mentioned earlier, industry mentors were asked to identify knowledge and skills that they expect either an intern or a graduate from a BSE program should have in order to excel in their work environment.

The data for the mentor needs assessment was collected and organized in a manner slightly different than that of the student data due to the fact that the mentors did not have a strong knowledge of the structure of the BSE degree program. Instead we identified eight major areas and asked for refinements of each as well as an opinion of whether the skills were appropriate for interns or new BSE graduates. In this way, we could identify which topics were appropriate for lower division or early upper division, and which topics were appropriate for upper division only.

3.2.3 Course-Centered Analysis

Table 3 contains three primary pieces of information related to courses in the BSE program: whether students found the topic useful for their internship, whether the topic was identified as a need by the students, and whether the industry mentors found the topic to be necessary for interns and graduates. The table is formatted in the following manner. Columns one and two contain course descriptor and course topics, respectively. The course descriptor is simply the

		Found Useful by Students	Student Gap Ref	Mentor Gap Ref
CSE 434 Computer Networks	Computer Networks and the Internet	✓	S.IV	M.iii.j-k
	Application Layer	✓	S.IV	M.iii.j-k
	Transport Layer	✓	S.IV	M.iii.j-k
	Network Layer	✓	S.IV	M.iii.j-k
	Link Layer	✓	S.IV	M.iii.j-k
	Network Security	✓	S.IV	M.iii.j-k

Table 3 Excerpt of Computer Science and Engineering Coverage by Course

course number and name. Boldface type in the descriptor indicates that the course is a required course in the BSE program. Column three indicates whether a topic covered in a current course was found to be useful by students (e.g., assessment of the *as-is*), while Columns four and five indicate whether the topic was identified as a need in the desired program by the students and mentors, respectively. The interpretation of the table is as follows.

- No checkmark in the third column indicates a topic is not considered in the scope of student interest; either the topic was not found useful or the topic was a pre-requisite to some other useful topic and thus not considered.
    - If the topic is not checked but has a reference in columns four and five, then the topic is offered but students are not getting exposure to it.
- If a topic is checked, there are a number of other possibilities:
    - If it lacks a cross-reference in columns four and five, then students found the topic useful and students and mentors found no need to further supplement the topic. That is, in those cases the topics are appropriately being covered.
    - If there is a cross-reference in column four only, then students found that the coverage was not adequately deep or needs updating.
    - If there is a cross-reference in column five only, then industry mentors found that the topic was important for embedded systems development knowledge.
    - If there are cross-references in both columns four and five, then the topic was found useful but also carries a strong recommendation for updating course content since students felt they needed more coverage of the topic, and mentors found the topic to be important.

Note that a lack of a checkmark in column three and cross-references in column four or five does not mean the topic should be eliminated; the topic is a candidate for removal but not a necessity. However, if a course lacks a significant amount of checkmarks or cross-references, that course should be the center of some discussion regarding exclusion from the BSE degree program.

The data in Table 3 is shown for CSE 434 Computer Networks. The table shows that students found the topics to be useful (although the students did not specifically state the topics in these terms) but also indicated a need to have more coverage of networking topics. Additionally, mentors stated that networking topics were important for the intern and BSE graduate perspectives. Note that this course is a technical elective rather than a required course.

3.2.4 Gap Analysis

The data, including that shown in Tables 1 – 3, was used to determine Embedded Systems *gaps*. Topics identified as needed by student interns and mentors were determined to be *gaps* if any of the following conditions were met:

1. Courses in the current curriculum state that the topic is addressed, but students report knowledge in the area is not appropriate (Course Gap).
2. Courses in the current curriculum address the topic, but students cannot take the courses due to inflexibility in program constraints (Program Gap).
3. No courses in the current curriculum address the topic (Curriculum Gap).

Table 4 identifies the gaps (Course, Program, Curriculum) found in the CSE (BSE degree) Department. Gaps are listed from most important to least important (as identified by students). A check mark indicates that the Gap specified by the corresponding column exists in the topic specified by the corresponding row. A "†" indicates that we do not have enough data to conclude that the gap exists.

Note that some of the topics were identified as having multiple gaps. For example, C/C++ is given very basic coverage (5 weeks) in a required programming languages course. Thus, C/C++ is a course gap. Additionally, one of the technical elective courses provides additional experience with C/C++ programming but is inaccessible to BSE students due to program constraints. Thus, it is identified as a program gap. Finally, treatment of the language is incomplete even considering the available courses. Thus, C/C++ is also identified as a curriculum gap. Similar reasoning applies to other topics with multiple gaps.

3.3 Identification of Solutions

The information described in the previous sections is merely an excerpt of a larger collection of data. The full collection was used to draw several conclusions and make several recommendations to the Consortium for Embedded Systems Board and ultimately to the Department of Computer Science and Engineering. The conclusions and recommendations were categorized according to their potential cost and impact to the CSE department and were used by CSE to overhaul the BSE degree program. These changes amount to enacting steps 4 – 6 of the SAGE process.

4 Observations

In this section we make several observations about factors that contributed to the quality and usefulness of the constituent feedback and discuss the impact of the results.

4.1 Rigor of Interview Process

The quality and specificity of the outcomes that resulted from the constituent interviews seemed to be directly proportional to the rigor of the interview process. A very structured process, the Affinity Process, was used to obtain feedback from the student interns. The feedback from the students was very useful in performing the curriculum gap analysis because the results were very specific both in terms of general embedded systems content as well as specific CSE course content.

Topic	BSE *Course* Gap	BSE *Program* Gap	BSE *Curriculum* Gap
C/C++ programming	✓	✓	✓
Practical experience with the Linux Operating System		✓	✓
Operating Systems Internals		✓	
Device Drivers	✓		
General Networking		✓	
Applied Networking	✓	✓	
Advanced Hardware	✓		
Modern Systems Architecture	✓		
Scripting Languages		✓	✓
Embedded Systems Design and Development		✓	
Practical use of compilers and tools	✓		
Experience with Software Engineering best practices	✓		
Business aspects			✓
Practical Experience with Project management and related issues	†		
Information Technology		✓	✓
Reliability and Fault Tolerance		✓	✓

Table 4 Gap Summary (listed from most to least important as ranked by students)

The interview process used with the industry mentors and CSE faculty was more informal and less structured. The feedback from the industry mentors was more "big picture" and less specific to the curriculum content but more specific to outcomes. The feedback was used to validate as well as complement the intern feedback. Similarly, the faculty feedback was also "big picture," but focused on implementation issues. Consequently, faculty feedback was less useful for identifying desired curriculum. Using a more formal process, like the Affinity Process, with mentors or faculty may have resulted in more detailed feedback.

4.2 Familiarity with Curriculum

Familiarity with curriculum seemed to result in an "implementation" bias. Whereas, constituents unfamiliar with specific curriculum were better able to focus on the goal. Both student interns and faculty members are familiar with the current curriculum. Thus, these constituents (but more specifically, the faculty) often made comments about how difficult it would be to change specific courses. Their knowledge of the curriculum often hindered them from focusing on what the curriculum "should be" because they were overwhelmed by the changes that would have to occur to achieve curriculum utopia. The faculty are not only familiar with the curriculum, but also the bureaucracy involved in making changes to curriculum. Thus, the faculty had an especially difficult time divorcing their feedback from practical implementation issues.

Industry mentors, on the other hand, have little or no familiarity with the current curriculum details or academic processes. Thus, they were free to focus on the ultimate goal of identifying the desired embedded systems knowledge and skills for new graduates.

4.3 Relationship between Interviewers and Interviewees

The closer the "peer" relationship was between the interviewers and interviewees, the more difficult it was to get useful feedback. The students, industry mentors, and faculty were all interviewed by the same committee that consisted of four faculty (two outside the department, one outside but a former member of the department, and one inside the department). The student interviewees were not "peers" of the interviewing committee. There was a pre-established "hierarchy" that allowed the faculty to require the students to follow specific procedures. As mentioned in Section 4.1, requiring the students to following a more rigorous process resulted in high quality feedback.

The industry mentors are considered "peers" of the faculty on the interviewing committee. The major reason a rigorous process was not used with the industry members was because the faculty committee was uncomfortable taking their peer's lunch hour away from them to "treat them like students" (asking them to follow specific rules). Consequently, the interview process that was used resulted in less specific feedback.

The faculty members were the closest "peers" of the faculty on the interviewing committee. The faculty were acquainted with one another and in some cases had worked together. In this case, the faculty being interviewed made comments such as "you know how it is" and brought up issues with a great deal of history rather than providing specific feedback. Thus, it was very difficult to obtain specific, useful feedback from the faculty group. It seems that a group of "outsiders" would be more likely to obtain quality feedback from the faculty.

4.4 Impact of Constituent Feedback

Part of the history mentioned in the previous section, is that several of the faculty had been working very hard to change specific course content and overall program requirements for at least three years. Students that were genuinely interested in the success of the program tried to voice opinions, but often did not know who to talk to. Thus, drastic changes were being advocated by only one group of program constituents (faculty) and the changes that were suggested were road blocked.

After the Gap Analysis, which included feedback from *all* constituents, was completed the department immediately reacted. During the Fall semester of 2004, the BSE program was redesigned. Several new courses have been proposed and the program requirements have been drastically changed. The proposed curriculum is farther along the bureaucratic process than it has been in the past four years. It seems that industry input (which contributes a large sum of money to ASU's Embedded Systems Consortium) was a major factor in moving curriculum reform forward.

5 Conclusions and Future Investigations

We have described a curriculum gap analysis and assessment process that was used at ASU. At this point, the process is just another "stovepipe" process for assessing curriculum as described in Section 1. For any process to truly make an impact, it must be generalized so that it can be applied to a variety of programs in Engineering, Computing, and Technology. Because the overarching continuous improvement process is based on an Engineering Design process, we believe that the SAGE process can be defined, widely adopted (and adapted), and easily followed. There are several steps that must be taken to generalize the curriculum assessment (SAGE steps 1 – 3) process described in this paper.

First, the stakeholders that provided feedback for our process had a strong tie to a formal internship program. As such, the content of the interviews and the subsequent analysis of the curricula were biased towards the internship program. While many engineering degree programs desire and strive to build strong internship components, tying the assessment process directly to whether a program has an internship option limits the potential impact of the process. To be widely useful, we must generalize the process so that it can be applied in a wide variety of programs regardless of the existence of an internship.

Additionally, during our experiences, we identified a number of opportunities for developing tools that would enhance the ability to properly assess a curriculum. To streamline the process, it is desirable to have software tools that facilitate the collection of feedback and analysis of the data once collected.

Finally, the process should be applied to other program to validate its applicability. Two of the authors are currently involved in the development and delivery of new B.S. degree programs in computing, and the other author is involved in the planning and implementation of a new B.S. in Engineering program at ASU's Polytechnic. These programs are ideal testbeds for future application of the process.

Bibliographic Information

1. Engineering Accreditation Commission, "Criteria for Accrediting Engineering Programs", ABET Report E1 11/19/03, November 2003.
2. Personal Communication, Ben Huey, Outgoing Chair, ABET Computing Accreditation Commission, October 2004.
3. B. Huey, S. Prital, D. Dannenberg, and J. Robertson, "An Arizona Ecosystem for Embedded Systems", In *Proc. of the 2001 IEEE International Performance, Computing and Communication Conference*, pages 131 – 134, April 2001.
4. Hyerle, D., "Visual Tools for Constructing Knowledge", Alexandria Virginia: Association for Supervision and Curriculum Development, 1996.

Biographical Sketches

BARBARA D. GANNOD is an assistant professor in the Division of Computing Studies at Arizona State University - East. She completed B.S. degrees in Mathematics and Computer Science and a Secondary Education degree at Calvin College in 1992. She received her Ph.D. in Computer Science from Michigan State University in 1997. Her research interests include engineering education and high-performance computing.

GERALD C. GANNOD is an assistant professor in the Division of Computing Studies at Arizona State University - East. He received the MS('94) and PhD('98) degrees in Computer Science from Michigan State University. His research interests include software product lines, software reverse engineering, formal methods for software development, software architecture, and software for embedded systems. He is a recipient of a 2002 NSF CAREER Award.

MARK R. HENDERSON is a professor of Industrial Engineering at Arizona State University in Tempe, AZ. He received the Ph.D. in mechanical engineering from Purdue University. Henderson was named a Presidential Young Investigator from 1985-90. He directs the Global Engineering Design Team for undergrads, the Nomadic Design Academy summer study abroad and is also co-director of the InnovationSpace, a transdisciplinary entrepreneurial course for undergrads.