1. Overview

The software engineering program assessment plan was developed by software engineering faculty with input from others. It provides a basis for obtaining feedback on the program, its outcomes, and its objectives, and using that feedback for making improvements. This document describes the program assessment plan. It includes the objectives and outcomes for the program, specifies assessment tools, and gives a timeline for assessing the program throughout the academic year. The plan is designed to obtain input from all constituents and to provide a structure for continuous improvement of both the program and the plan.

2. Mission Statements

This section lists the mission statements for the University, the College, and the Software Engineering Program.

2.1. University Mission Statement

The fundamental mission of UW-Platteville and the entire UW System is to serve the people of Wisconsin. This basic goal is expressed in detail in the mission statement adopted in 2002. In this statement, UW-Platteville pledges itself to:

1. Enable each student to become broader in perspective, more literate, intellectually more astute, ethically more sensitive, and to participate wisely in society as a competent professional and knowledgeable citizen.

2. Provide baccalaureate degree programs which meet primarily regional needs in arts and sciences, teacher education, business, and information technology.

3. Provide baccalaureate degree programs and specialized programs in middle school education, engineering, technology management, agriculture, and criminal justice which have been identified as institutional areas of emphasis.
4. Provide graduate programs in areas clearly associated with its undergraduate emphasis in education, agriculture, technology management, engineering, and criminal justice.

5. Provide undergraduate distance learning programs in business administration and graduate online programs in project management, criminal justice, and engineering.

6. Provide agricultural systems research programs utilizing the Pioneer Farm in partnership with businesses, universities and agencies.

7. Expect scholarly activity, including applied research, scholarship and creative endeavor, that supports its programs at the baccalaureate degree level, its selected graduate programs, and its special mission.

8. Serve the needs of all students and in particular the needs of women, minority, disadvantaged, and nontraditional students. Furthermore, to seek diversification of the student body, faculty and staff.

9. Serve as an educational, cultural, and economic development resource to southwestern Wisconsin.

These statements, along with the UW System and University Cluster mission statements, provide a guide to UW-Platteville in what it attempts and does not attempt to accomplish as an institution of higher education.

2.2. College Mission Statement

The College's objective is to ensure that its students gain the knowledge and develop the mental skills, attitudes, and personal characteristics necessary to become successful citizens and professionals who can meet the present needs of business, industry, government, and society, and the more demanding requirements of the future.

2.3. Software Engineering Mission Statement

The mission of the Software Engineering Program is to provide a quality software engineering education with significant hands-on and laboratory experience that will enable our graduates to practice their profession with proficiency and integrity.

3. Software Engineering Objectives

The objectives for the UW-Platteville software engineering program were changed in the fall of 2004. The new objectives were established by asking for input from the UW-Platteville software engineering major constituents: faculty, students, graduates, and advisory board members. Members of the advisory board included representative employers. The input from the constituents was distilled to a few broad objectives. The process went through a couple of iterations until everyone was satisfied with the results.

The current objectives, as adopted in December of 2004, are given below. These objectives were reaffirmed by the CSSE Advisory Board at the spring 2010 meeting.
1. Graduates are effective team members, aware of cultural diversity, who conduct themselves ethically and professionally.

2. Graduates use effective communication skills and technical skills to assure production of quality software, on time and within budget.

3. Graduates build upon and adapt knowledge of science, mathematics, and engineering to take on more expansive tasks that require an increased level of self-reliance, technical expertise, and leadership.

4. Outcomes

4.1. Software Engineering Outcomes
The software engineering outcomes were modified in the December of 2005 in response to the modification of the software engineering objectives. The current outcomes, as adopted in December of 2005, are given below.

Since our accreditation visit in 2006, the ABET wording of Criterion 3 was changed. We requested a clarification of that wording and the response is in Appendix D. The email outlines two choices: Adopt ABET a-k or keep the existing outcomes and map current measurements to ABET a-k. To get an outside perspective, we posed the choices to the CSSE Advisory Board at the spring 2010 meeting. The members were unanimous saying our current outcomes should be kept. Therefore, the outcomes remain the same and a mapping of our performance criteria and their measurements to ABET a-k is given in Section 4.4.

A. Foundation: Graduates shall have a strong foundation in science, mathematics, and engineering, and can apply this fundamental knowledge to software engineering tasks.

B. Development: Graduates can effectively apply software engineering practice over the entire system lifecycle. This includes requirements engineering, analysis, prototyping, design, implementation, testing, maintenance activities and management of risks involved in software and embedded systems.

C. Process: Graduates know classical and evolving software engineering methods, can select and tailor appropriate methods for projects, and can apply them as both team members and managers to achieve project goals.

D. Professionalism: Graduates are knowledgeable of the ethics, professionalism, and cultural diversity in the work environment.

E. Quality: Graduates can apply basic software quality assurance practices to ensure that software design, development, and maintenance meets or exceeds applicable standards.
F. Presentation: Graduates have effective written and oral communication skills. Graduates can prepare and publish the necessary documents required throughout the project lifecycle. Graduates can effectively contribute to project discussions, presentations, and reviews.

G. Growth: Graduates understand the need for life-long learning and can readily adapt to new software engineering environments.

4.2. ABET Outcomes

The engineering programs must demonstrate that the graduates have:

a. an ability to apply knowledge of mathematics, science, and engineering
b. an ability to design and conduct experiments, as well as 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 multi-disciplinary 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

4.3. Relationship of Software Engineering Outcomes to ABET Outcomes

<table>
<thead>
<tr>
<th>ABET Outcomes</th>
<th>A Foundation</th>
<th>B Development</th>
<th>C Process</th>
<th>D Professionalism</th>
<th>E Quality</th>
<th>F Presentation</th>
<th>G Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Relationship of Software Engineering Outcomes to ABET Outcomes
4.4. Mapping of Measurements to ABET a-k

The Outcomes Performance Criteria and Measurements are given in Appendix B. This section maps these to ABET a-k.

a. an ability to apply knowledge of mathematics, science, and engineering

A.1: Uses engineering principles to construct tests.
A.2: Uses mathematical modeling techniques to specify systems.
A.6: Uses physics formulas when developing real-time embedded systems programs.

b. an ability to design and conduct experiments, as well as analyze and interpret data

A.3: Analyzes run-time performance of algorithms.
B.2: Designs and executes an experiment using a prototype to address a technical risk.
E.5: Applies knowledge of quality assurance methods to design and optimize a quality assurance process.

B.3: Produces a system design satisfying requirements and constraints.

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

B.5: Implements software for a real-time embedded system

d. an ability to function on multi-disciplinary teams

C.3. Applies a specified process and methods to a large group project.
C.5. Manages a small team in development of a software system
D.4. Contributes to teams that include students from other disciplines

e. an ability to identify, formulate, and solve engineering problems

A.1: Uses engineering principles to construct tests.
A.4: Constructs software using classic data structures and algorithms.
A.5: Develops solutions to concurrent programming problems.
B.1: Develops requirements specifications.
B.3: Produces a system design satisfying requirements and constraints.
B.5: Implements software for a real-time embedded system

f. an understanding of professional and ethical responsibility

D.2. Evaluates the professional and ethical implications of workplace issues.
g. an ability to communicate effectively

F.1. Speaks clearly and makes use of visual tools to effectively communicate a technical topic.
F.2. Writes clearly and follows specified standards and formats where appropriate to effectively communicate technical topics.
F.3. Responds to questions and comments to clarify questions and address issues.
F.4. Reviews both written and orally presented material to address unclear, defective, or non-standard items.

h. the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context

D.2. Evaluates the professional and ethical implications of workplace issues.
D.3. Lists and describes the considerations for developing multi-national software.

i. a recognition of the need for, and an ability to engage in life-long learning

G.1. Researches new software engineering topics.
G.2. Learns new software engineering environments.

j. a knowledge of contemporary issues

G.1. Researches new software engineering topics.

k. an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

A.5: Develops solutions to concurrent programming problems.
B.2: Designs experiment using prototype to address a technical risk.
B.3: Produces a system design satisfying requirements and constraints.
C.2. Chooses an appropriate process model for a project and specifies refinements to each of the phases.
C.3. Applies a specified process and methods to a large group project.
E.1. Applies knowledge of software metrics to improve a process.
G.2. Learns new software engineering environments.

The SE Specific Criteria:

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 program must demonstrate that graduates have the ability to analyze, design, verify, validate, implement, apply, and maintain software systems; the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and
supporting disciplines to complex software systems; the ability to work in one or more significant application domains; and the ability to manage the development of software systems.

A.1: Uses engineering principles to construct tests (verify)
A.2: Uses mathematical modeling techniques to specify systems (apply discrete math)
A.3: Analyzes run-time performance of algorithms (analyze)
A.4: Constructs software using classic data structures and algorithms (implement and apply CS)
A.5: Develops solutions to concurrent programming problems (implement and apply CS)
B.1: Develops requirements specifications (analyze)
B.3: Produces a system design satisfying customer requirements (design)
B.5: Implements software for a real-time embedded system (work in real-time domain)
B.6: Maintains an existing software system (maintain)
C.2: Chooses an appropriate process model for a project and specifies refinements to each of the phases (manage)
C.4: Produces a Software Project Management Plan (manage)
C.5: Manages a small team in development of a software system (manage)
E.2: Produces a system validation plan (validation)
E.3: Produces a thorough module test (verify)
E.7: Applies knowledge of Discrete Math, Probability and Statistics to test coverage analysis (apply discrete mathematics, probability and statistics)

5. Software Engineering Constituencies

The major constituencies of the software engineering program are the students, alumni, faculty, employers, and advisory board.

6. Assessment Tools

Three categories of assessment tools are used:

- Tools to assess the program objectives
- Tools to assess the program outcomes
- Additional program assessment tools

This section specifies the details of each of the assessment tools used. The timelines for assessment as well as how the assessment data are used are contained in the Section 7.

6.1. Program Objectives Assessment

Employer and alumni surveys are used to assess the software engineering program objectives. These surveys are done annually and target two groups. The first group includes alumni who have been out of school for two years and their immediate supervisors. The second group surveyed is those alumni who have been out of school for five years and their immediate supervisors. The surveys are questions directly related to the program objectives. Appendix A contains sample alumni and employer surveys.
6.2. Program Outcomes Assessment

To assess the software engineering outcomes, two to five performance criteria are established for each outcome. For each performance criterion, at least one direct and one indirect measurement are specified. The current performance criteria and their measurements are specified in Appendix B. The methods used for assessment fall into the following categories:

In-Course Assessment: A specific item such as a test problem or quiz problem is assessed independent of the grade for the test or quiz. The assessment is done using a separate rubric with the purpose of providing a direct measurement of a specific performance criterion.

Direct Observation: The student is observed while performing some task such as giving a presentation, participating in a code inspection, etc. The assessment is done using a rubric with the purpose of providing a direct measurement of a specific performance criterion.

Course Surveys: Course learning outcomes are included in each course. Students are surveyed in each course and asked whether or not the course learning outcomes are being met. A mapping of course outcomes to performance criteria is used. This is an indirect assessment measurement.

Graduating Senior Exit Surveys: Graduating seniors provide feedback by completing a standard exit survey during the last week of the semester they are graduating. The survey is designed using the performance criteria. This is an indirect assessment measurement.

6.3. Additional Program Assessment

Besides the tools used for assessing the program outcomes and objectives, additional tools are used for general assessment and program improvement. These tools are:

1. Course Folders: A course folder is kept for each required software engineering and computer science. Course folders include the course syllabus and possibly other handouts. Furthermore, samples of graded exams, quizzes, homework, programs, and projects are included. The Software Engineering faculty perform spot-check reviews of the course folders. The primary objective of this is to continually review the course content and provide feedback regarding specific coverage areas in the curriculum.

2. Advisory Board: At least once per year, the software engineering curriculum is presented to the Computer Science and Software Engineering advisory board for comments and feedback. The software engineering and computer science faculty review suggested additions and changes to the curriculum and incorporate the changes where deemed appropriate.

3. Benchmarking: The software engineering curriculum is mapped to the Software Engineering Education Knowledge (SEEK) Computing Curricula Software Engineering Volume. In addition, the program is informally compared to ABET-accredited software engineering programs for content as well as depth and breadth.

4. Other data: Data such as job placement rates and average salaries are used to evaluate
the program against the national averages. The results obtained reflect the overall strength of the Software Engineering program.

7. **Assessment Timeline, Reporting, and Feedback**

This section specifies the timelines for assessment, the reports generated, and the feedback mechanisms used. Any comments, suggestions, and recommendations resulting from the assessment are evaluated by the SE faculty and changes are made when deemed appropriate. Changes range from modifying the way a topic is covered in a course to addition and deletion of required courses.

Below is a table that summarizes the major timelines presented in the remainder of this section:

<table>
<thead>
<tr>
<th>Table 2: Assessment Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Emp / Alum Survey</td>
</tr>
<tr>
<td>Mailed</td>
</tr>
<tr>
<td>Analyzed / Report</td>
</tr>
<tr>
<td>Suggestions / Comments</td>
</tr>
<tr>
<td>Outcomes Assessment</td>
</tr>
<tr>
<td>Performed</td>
</tr>
<tr>
<td>Analyzed / Report</td>
</tr>
<tr>
<td>Suggestions / Comments</td>
</tr>
<tr>
<td>Advisory Board</td>
</tr>
</tbody>
</table>

7.1. **Program Objectives Assessment**

The employer and alumni surveys are sent out each spring semester. During the summer, the returned survey data are tabulated and a report is written by the program coordinator. The report is made available to the SE faculty at the start of the fall semester. The report is presented at the fall meeting of the Advisory Board for comments and suggestions. If the fall meeting is not held, the results are reported at the spring meeting.

7.2. **Program Outcomes Assessment**

The indirect assessment measurements involving graduate exit surveys are performed every semester. The indirect assessment measurements involving course surveys are performed every semester that a given course meets. The direct measurements are performed according to Table 3 below. This table covers the time period between spring 2007 and fall 2012. This table will be updated after each ABET visit. The last ABET visit was in the fall of 2006. The next ABET visit is anticipated to be the fall of 2012. Additional direct measurements may be scheduled if analysis warrants it.

The results of the program outcomes assessment are aggregated once per year, during the summer, and a summary report, noting any deficiencies or areas of concern, is written by the
program coordinator. The report is made available to the SE faculty before the start of the fall semester. The SE faculty makes recommendations for improvement. Those recommendations that can be directly implemented by the SE faculty are implemented as soon as appropriate. Those recommendations requiring department or college approval to implement are ushered through the necessary faculty governance channels.

Table 3: Outcomes Direct Assessment Measurement Schedule

<table>
<thead>
<tr>
<th>Measure</th>
<th>Course</th>
<th>S07</th>
<th>F07</th>
<th>S08</th>
<th>F08</th>
<th>S09</th>
<th>F09</th>
<th>S10</th>
<th>F10</th>
<th>S11</th>
<th>F11</th>
<th>S12</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.2.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.2.b</td>
<td>3430</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.3.a</td>
<td>2630</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.3.b</td>
<td>4730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.4.a</td>
<td>4730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.5.a</td>
<td>4130</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A.6.a</td>
<td>4130</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.1.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.2.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.3.a</td>
<td>3430</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.3.b</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.4.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.5.a</td>
<td>4130</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.6.a</td>
<td>3860</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>C.1.a</td>
<td>4730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>C.2.a</td>
<td>4730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>C.3.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>C.3.b</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>C.4.a</td>
<td>4730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>C.5.a</td>
<td>3330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D.1.a</td>
<td>3330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D.2.a</td>
<td>Ph Eth</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D.3.a</td>
<td>3730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D.4.a</td>
<td>3430</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>E.1.a</td>
<td>3730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>E.2.a</td>
<td>4330</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>E.3.a</td>
<td>4730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>E.4.a</td>
<td>3730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>E.5.a</td>
<td>3730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>E.6.a</td>
<td>3730</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>F.1.a</td>
<td>4110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>F.2.a</td>
<td>4110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>F.3.a</td>
<td>4110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>F.3.b</td>
<td>4110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>F.4.a</td>
<td>4110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>F.4.b</td>
<td>4110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
7.3. Additional Program Assessment

Course folders are reviewed on an ad-hoc basis. The results are informally conveyed from the person doing the checking to the person who created the folder.

The Advisory Board usually meets every semester. Comments and suggestions from the Advisory Board are discussed as part of any changes that stem from other assessments.

Benchmarking against SEEK and other SE programs is done on an ad-hoc basis. Sometimes these results in new courses, but more typically, the result is a repackaging of topics in existing courses, if deemed appropriate.

Graduate placement and starting salaries are tracked every semester. No direct action is generally taken as a result beyond trying to understand or informally explain any trends noticed.

8. Program and Process Improvement

Throughout the year, the assessment tools will be used to determine areas that need improvement. These improvements will be implemented as agreed upon by the SE faculty. If there are program objectives that are not being met, the faculty will take steps to fix the problem. The steps include, but are not limited to:

1. Take appropriate corrective action in the specified course
2. Make curriculum changes to address the deficiencies
3. Reevaluate the program outcomes
4. Revise assessment tools and procedures

This assessment plan will be reviewed annually by the SE faculty and amended with any appropriate changes.
Appendix A – Sample Employer and Alumni Surveys

Alumni Survey

1. My current job title is _____________________________________________________________

2. I have been employed at this company for __________ years.

3. Total number of employees in our company (approximately): _____________________

4. The following describe my current responsibilities (mark all that apply):
   □ Requirements □ Design □ Programming □ Testing
   □ Embedded Systems Development □ Business Applications □ Web Development
   □ Network Administration □ Database Administration □ Software Quality Assurance
   □ Others _____________________________ (specify)

5. Have you developed software for international use?    Yes   No

6. Have you taken any additional courses since graduation? Yes   No

7. Mark the square specifying the approximate number of
   0  1-2  3-5  >5
   □ □ □ □ A. different projects you worked on within the past year.
   □ □ □ □ B. total team members you worked with in the past year.
   □ □ □ □ C. times you made a presentation within the past year.
   □ □ □ □ D. new languages, new programming environments, and/or new technologies you’ve learned and/or applied since graduation.
**Employer Survey**

The employee (don’t specify his/her name):

<table>
<thead>
<tr>
<th></th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
<th>Not Observed</th>
<th>Not Applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. is an effective team member

2. has effective communication skills

3. produces quality products, on time

4. shows awareness of cultural diversity

5. conducts themselves ethically and professionally

6. shows potential for growth within the organization
Appendix B – Outcomes, Performance Criteria, Measurements

A. **Foundation:** Graduates shall have a strong foundation in science, mathematics, and engineering, and can apply this fundamental knowledge to software engineering tasks.

A.1: Uses engineering principles to construct tests.

A.1.a. The instructor shall assess each student’s system test specification. Each student shall be responsible for developing a test specification covering at least two requirements and at least five test cases in SE 4330.

A.1.b. Student course assessment survey for SE 3730.

A.1.c. Graduate exit survey.

A.2: Uses mathematical modeling techniques to specify systems.

A.2.a. Locally-developed SE 4330 exam questions on constructing Z (or other language) specifications.

A.2.b. Locally-developed SE 3430 exam questions on constructing finite state machines.

A.2.c. Student course assessment survey for SE 4330.

A.2.d. Graduate exit survey.

A.3: Analyzes run-time performance of algorithms.

A.3.a. Locally-developed SE 2630 final exam questions on algorithm analysis.

A.3.b. Locally-developed exam questions given in SE 4730.

A.3.c. Student course assessment survey for SE 2630.

A.3.d. Graduate exit survey.

A.4: Constructs software using classic data structures and algorithms.

A.4.a. Locally-developed programming problem on exam in SE 4730.

A.4.b. Student course assessment surveys for SE 2630.

A.4.c. Graduate exit survey.

A.5: Develops solutions to concurrent programming problems.
A.5.a. Locally-developed exam or quiz questions in SE 4130.

A.5.b. Student course assessment surveys for CS 3230.

A.5.c. Graduate exit survey.

A.6: Uses physics formulas when developing real-time embedded systems programs.

A.6.a. The instructor shall assess one real-time embedded systems program for each student in SE 4130.

A.6.b. Graduate exit survey.

B. Development: Graduates can effectively apply software engineering practice over the entire system lifecycle. This includes requirements engineering, analysis, prototyping, design, implementation, testing, maintenance activities and management of risks involved in software and embedded systems.

B.1: Develops requirements specifications.

B.1.a. Assess portions of requirements specifications written for project by individual students in SE 4330.

B.1.b. Student course assessment surveys for SE 4330.

B.1.c. Graduate exit survey.

B.2: Designs and executes an experiment using a prototype to address a technical risk.

B.2.a. Assess prototypes in SE 4330.

B.2.b. Student course assessment surveys for SE 4330.

B.2.c. Graduate exit survey.

B.3: Produces a system design satisfying requirements and constraints.

B.3.a. Locally-developed exam questions in SE 3430.

B.3.b. Assessment of designs developed by groups in SE 4330 for class project.

B.3.c. Student course assessment surveys for SE 3430.

B.3.d. Graduate exit survey.
B.4: Identifies risks in software systems.

B.4.a. Locally-developed exam questions in SE 4330 for which student must identify risks and classify them according to severity.

B.4.b. Student course assessment surveys for SE 4330.

B.4.c. Graduate exit survey.

B.5: Implements software for a real-time embedded system

B.5.a. The instructor shall assess one real-time embedded systems program for each student in SE 4130.

B.5.b. Student course assessment surveys for SE 4130.

B.5.c. Graduate exit survey.

B.6: Maintains an existing software system

B.6.a. Faculty and peer assessment of each student’s contribution to the maintenance project in SE 3860.

B.6.b. Student course assessment surveys for SE 3860.

B.6.c. Graduate exit survey.

C. Process: Process: Graduates know classical and evolving software engineering methods, can select and tailor appropriate methods for projects, and can apply them as both team members and managers to achieve project goals.

C.1. Lists and describes several software process models, detailing what is done in the phases of each.

C.1.a. Locally-developed exam administered in the second course of the capstone sequence (SE 4730). Students given a list and asked to do this for three in the list.

C.1.b. Student course assessment surveys for SE 3330.

C.1.c. Graduate exit survey.

C.2. Chooses an appropriate process model for a project and specifies refinements to each of the phases.

C.2.a. Locally-developed exam administered in the second course of the capstone
sequence (SE 4730). Students given a hypothetical project and must specify a process and refinements for each stage, justifying their decisions.

C.2.b. Student course assessment surveys for SE 3330.

C.2.c. Graduate exit survey.

C.3. Applies a specified process and methods to a large group project.

C.3.a. Observation against a check-list in the capstone sequence (SE 4330, 4730). Students must apply the selected and tailored process established for the given project. It should be noted that all students work together on teams on a single large project. The students and instructor establish the process and methods to be used for the course early in the first semester, to which all students must adhere.

C.3.b. Student course assessment surveys for SE 4330.

C.3.c. Graduate exit survey.

C.4. Produces a Software Project Management Plan

C.4.a. The instructor shall assess each student’s contribution to the Software Project Management Plan produced in SE 4330.

C.4.b. Student course assessment surveys for SE 4330.

C.3.c. Graduate exit survey.

C.5. Manages a small team in development of a software system

C.5.a. Assess each student’s effectiveness as a leader in SE 3330.

C.5.b. Graduate exit survey.

D. Professionalism: Graduates are knowledgeable of the ethics, professionalism, and cultural diversity in the work environment.


D.1.a. Locally-developed exam question in the Intermediate Software Engineering Course (SE 3330)

D.1.b. Student course assessment surveys for SE 3330.
D.1.c. Graduate exit survey.

D.2. Evaluates the professional and ethical implications of workplace issues.

D.2.a. Check sheet in Philosophy 2540 class filled out by the instructor.

D.2.b. Student course assessment surveys for SE 3330.

D.2.c. Graduate exit survey.

D.3. Lists and describes the considerations for developing multi-national software.

D.3.a. Locally-developed exam question in the Software Quality Course (SE 3730)

D.3.b. Student course assessment surveys in SE 3730.

D.3.c. Graduate exit survey.

D.4. Contributes to teams that include students from other disciplines

D.4.a. Assessment by the instructor and non-SE group members of effectiveness in an Application Domain course involving a project with team members from other disciplines.

D.4.b. Graduate exit survey.

E. Quality: Graduates can apply basic software quality assurance practices to ensure that software designs, development, and maintenance meets or exceeds applicable standards.

E.1. Applies knowledge of software metrics to improve a process.

E.1.a. The instructor shall assess a team’s analysis of a hypothetical set of defect data. The team shall use appropriate metrics and decision tools to support process improvement recommendations. SE 3730 team project.

E.1.b. Student course assessment surveys in SE 3730.

E.1.c. Graduate exit survey.

E.2. Produces a system validation plan.

E.2.a. The instructor shall assess an acceptance test specification for the project developed in SE 4330. Each student shall specify acceptance tests covering an aspect of the project for which that student did not participate in developing requirements.
E.2.b. Graduate exit survey.

E.3. Produces a thorough module test.

E.3.a. The instructor shall assess one of the module tests produced by each student for the project in SE 4730.

E.3.b. Student course assessment surveys in SE 3730.

E.3.c. Graduate exit survey.

E.4. Applies knowledge of Software Standards to evaluate a process.

E.4.a. The instructor shall assess each student’s analysis of conformance of a process to a Software Standard (ISO-9xxx, CMMI, …). SE 3730 homework assignment or exam question.

E.4.b. Student course assessment surveys in SE 3730.

E.4.c. Graduate exit survey.

E.5. Applies knowledge of quality assurance methods to design and optimize a quality assurance process.

E.5.a. Locally developed SE 3730 lab exercise: process cost minimization problem.

E.5.b. Graduate exit survey.


E.7.a. Locally developed exam questions in SE 3730: test coverage problem involving combinations of inputs, defects, test strategies and likelihood of detecting a defect.

E.7.b. Graduate exit survey.

F. Presentation: Graduates have effective written and oral communication skills. Graduates can prepare and publish the necessary documents required throughout the project lifecycle. Graduates can effectively contribute to project discussions, presentations, and reviews.

F.1. Speaks clearly and makes use of visual tools to effectively communicate a technical topic.
F.1.a. Instructor shall directly observe each student’s SE 4110 presentation. SE 4110 requires each student to research and make an oral presentation on a Software Engineering topic. See CSSE 4110 – Senior Seminar –Presentation Review Form

F.1.b. Student course assessment surveys in SE 4110.

F.1.c. Graduate exit survey.

F.2. Writes clearly and follows specified standards and formats where appropriate to effectively communicate technical topics.

F.2.a. Instructor shall evaluate each student’s SE 4110 research paper. SE 4110 requires each student to research and write a technical paper compliant with MICS publication standards. See CSSE 4110 – Senior Seminar – Paper Review Form.

F.2.b. Student course assessment surveys in SE 4110.

F.2.c. Graduate exit survey.

F.3. Responds to questions and comments to clarify questions and address issues.

F.3.a. Instructor shall observe and rate how well each student presenter addresses questions asked by peers and other attendees. See SE 4110 Question and Response Form.

F.3.b. Instructor shall observe and rate how well each student responds to peer reviews of his/her Senior Seminar paper. See SE 4110 Question and Response Form.

F.3.c. Student course assessment surveys in SE 4110.

F.3.d. Graduate exit survey.

F.4. Reviews both written and orally presented material to address unclear, defective, or non-standard items.

F.4.a. Instructor shall examine each student’s formal review of two other students’ research papers for appropriateness to standards and content. See CSSE 4110 – Senior Seminar – Paper Review Form.

F.4.b. Instructor shall review each student’s questions. In SE 4110 each student shall ask seven questions during other students’ presentations during the semester. See CSSE 4110 – Senior Seminar – Presentation Question Form.
F.4.c. Instructor shall review each student’s performance during technical walkthroughs and reviews in SE 4730.


F.4.e. Graduation exit survey.

**G. Growth:** Graduates understand the need for life-long learning and can readily adapt to new software engineering environments.

G.1. Researches new software engineering topics.

G.1.a. The instructor shall examine each student’s final paper for evidence of the ability to understand and assimilate new information. In SE 4110 each student is required to research a software engineering topic, present the topics to their peers and faculty, and write a paper. See CSSE 4110 – Senior Seminar – Paper Review Form.

G.1.b. Student course assessment surveys in SE 4110.

G.1.c. Graduation exit survey.

G.2. Learns new software engineering environments.

G.2.a. Instructor shall appraise each student’s skill in using PIC programming environment as part of Lab 1 in SE 4130. A student’s grade on this lab shall be used as competency measure.

G.2.b. Student course assessment surveys in SE 4130.

G.2.c. Graduation exit survey.
Appendix C – Changes to the Assessment Plan

• Modifications from Version 2.0 to 2.1
  1. Changed all measurements using student course assessment surveys that list two courses to only list one course instead, except those that list SE 4330 / SE 4730. Left the course that is more “advanced”.
  2. Modified measurements B.1.b and B.2.b so that they specify that course surveys are used from SE 4330.

• Modifications from Version 2.1 to 2.2
  1. Added performance criteria B.5 and B.6, together with measurements for each. These additional performance criteria were needed to complete the assessment for the ABET Program Criteria for Software Engineering.
  2. Fixed a typographical error in A.2.c, which had SE 3430 but was supposed to be SE 4330.
  3. Fixed a typographical error in D.2.a, which had Ethics 2530 but was supposed to be Philosophy 2540.

• Modifications from Version 2.2 to 2.3
  1. Removed performance criterion measurement A.1.b. This one doesn’t directly get at what is wanted and is hard to test. There already is a direct measurement for this performance criterion anyway. Also, renumbered A.1.c and A.1.d to A.1.b and A.1.c.
  2. Changed E.3.a to be assessed in SE 3330 instead of SE 2630. Testing this later is more appropriate.
  3. Removed “SE 4330” from F.4.c. The only needs to be done in one of the two courses.
  4. Added a rotation schedule for the direct assessments (see Section 7.2). This was always part of the plan to do this after a few semesters in order to reduce the workload. During the ABET visit, the SE PEV strongly recommended we do so, since he believed that the amount of work we were doing was quite high.
  5. Added table numbers for each table.

• Modifications from Version 2.3 to 2.4
  1. Remove “and SE 473” from measurement C.3.b since there is no course outcome that directly corresponds to this measurement.

• Modifications from Version 2.4 to 2.5
  1. Do assessment reports once per year, in the summer, instead of twice per year.

• Modifications from Version 2.5 to 2.6
  1. Changed outcome C and added two performance criteria to include management as per the changes to the ABET program criteria for Software Engineering. Also, updated the
2. Changed the direct assessment for measurement E.3.a. to be done in SE 4730 instead of SE 3330, and updated the direct assessment rotation schedule.

3. Modified E.4.a to allow the assessment to optionally be via an exam question.

- **Modifications from Version 2.6 to 3.0**
  1. Changed all occurrences of UWP to UW-Platteville to comply with name rebranding.
  2. Added Appendix D, the response to a clarification of the change to Criterion 3.
  3. Modified Section 3 to include the reaffirmation of the objectives by the Advisory Board.
  4. Modified Section 4.1 to explain our new approach to outcomes assessment due to the wording change to ABET Criterion 3.
  5. Added Section 4.4 that maps Outcomes Performance Criteria and Measurements to ABET a-k.
  6. Modified Performance Criteria B.2 to add “and executes”. Changed measurement B.2.a to assess a prototype.
  7. Modified Performance Criteria B.3 to add “and constraints”.
  9. Replaced E.2 (which had two measurements the same as A.1) with a Performance Criterion for Validation.
  10. Added Performance Criteria A.6, E.5, and E.6, and associated measurements.
  11. Modified Table 3 to add rotations for the direct measurements for the added Performance Criteria.
Appendix D – Email from Dayne Aldridge

The email is dated: Tuesday, March 30, 2010 3:15 PM

Joe,

I am responding to the following request you sent to ABET:

We went through accreditation in 2006 and go up again in 2012. There is discussion about what the wording changes to criterion 3 mean. Currently, we have program outcomes (with performance criteria) and then show how these map to the ABET a-k outcomes. There are some that say we can no longer do that and must use a-k directly as our program outcomes (and then add in additional for the program-specific outcomes). Is there a document that clearly defines what is expected / required?

RESPONSE: We understand that many programs chose to compose their own outcomes and map them to the ABET a-k outcomes. Moreover, you may wish to continue using your outcomes as part of your assessment and evaluation activities. We advise the following for your situation. In addition to mapping your outcomes to the a-k, you should show how the assessment measurements you use map to the ABET a-k. There should be at least one measurement for each ABET outcome that provides evidence of attainment. To summarize, you may continue using your current outcomes if they adequately embrace the ABET a-k and the measurements you use provide sufficient data to evaluate attainment of each of the ABET a-k.

Be aware that newly harmonized criteria will be in effect in the fall of 2012. Although there will not be substantial changes relative to the content of the current criteria you should note the following. There will be wording changes in the definition of Program Educational Objectives. The continuous improvement activities required under the harmonized Criterion 4 will be more strongly tied to the objectives and outcomes required by Criterion 2 and 3. A draft version of the harmonized criteria is published in the back of the 2010-2011 criteria for review and comment. The final version will be approved by the ABET Board late October 2010 and published by December 1, 2010 for first application in the 2011-12 cycle. The self-study questionnaire will also change accordingly and the first version will be available for download from the ABET website by August 1, 2010.

Dayne Aldridge
Adjunct Accreditation Director for Engineering