Progress Review and Plan

By
ABET Assessment Committee
Doina Bein
Wenlin Han
Shawn Wang
Bin Cong (Chair)
Common Complains

- Do not know my roles in the process? Do not know who to ask and where to find info?
- Do not know what data to collect? Do not know how the data are used?
- Do not know how to integrate ABET into my teaching activities.

Hopefully we can help today!
Outline

1. Refinement of Assessment and Evaluation Process
2. The ABET Website
3. 2017-2018 Assessment Report
4. Continuous Improvement – what we have done
5. The Plan
We already have a solid foundation in place!

- ABET has made changes, for example, number of Student Outcomes has been reduced to 6 from 11.

- Feedback received from previous assessments and internal practices.

- ROI consideration.
CS Mission Statement

The Computer Science Department provides students with:

- A strong knowledge of computer science fundamentals and the technology of computer systems.

- Practical problem-solving skills for creating computer systems applications.

- The ability and motivation to adapt as technology advances.
Program Educational Objectives (PEOs)

- **Technical Growth:** Graduates will be successful in modern computing practices, integrate into the local and global workforce, and contribute to the economy of California and the nation.

- **Professional Skills:** Graduates will continue to demonstrate the professional skills necessary to be competent employees, assume leadership roles, and have career success and satisfaction.

- **Professional Attitude and Citizenship:** Graduates will become productive citizens with high ethical and professional standards, who make sound technical or managerial decisions, and have enthusiasm for the profession and professional growth.

Both Mission Statement and PEOs are reviewed periodically by IAD (Fall 2018), faculty, and students (clubs).
The 6 Student Outcomes (SOs)

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.

3. Communicate effectively in a variety of professional contexts.

4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

5. Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.

6. Apply computer science theory and software development fundamentals to produce computing-based solutions.
How to assess SOs met or not?

Use Performance Indicators!
# The PIs – Performance Indicators (SO and course mapping)

<table>
<thead>
<tr>
<th>Core Course CPSC</th>
<th>SLO 1</th>
<th>SLO 2</th>
<th>SLO 3</th>
<th>SLO 4</th>
<th>SLO 5</th>
<th>SLO 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>CODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>ACODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>ALG</td>
<td>ACODE</td>
<td></td>
<td></td>
<td></td>
<td>ALG</td>
</tr>
<tr>
<td>223</td>
<td>CODE, TEST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>311</td>
<td></td>
<td>WRITE,SPEAK</td>
<td></td>
<td>FDBK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>315</td>
<td></td>
<td>WRITE,SPEAK</td>
<td></td>
<td>ETH,IPSEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>323</td>
<td>DESC</td>
<td></td>
<td></td>
<td></td>
<td>COOP,FDBK</td>
<td></td>
</tr>
<tr>
<td>332</td>
<td>RESPEC</td>
<td></td>
<td></td>
<td></td>
<td>COOP,FDBK</td>
<td></td>
</tr>
<tr>
<td>335</td>
<td>DESC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ALG</td>
</tr>
<tr>
<td>351</td>
<td>HW</td>
<td>DESC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>362</td>
<td>RESPEC</td>
<td>DESC,TEST</td>
<td></td>
<td>PROC</td>
<td></td>
<td>RESPEC, DESC, Test</td>
</tr>
<tr>
<td>440</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>471</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>481</td>
<td>RESPEC</td>
<td>DESC,TEST</td>
<td></td>
<td>RESPEC,DESC,TEST,FB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exit Survey</td>
<td></td>
<td></td>
<td></td>
<td>ETH,IPSEC</td>
<td>PROC</td>
<td></td>
</tr>
</tbody>
</table>
• HW: Demonstrate understanding of the architecture of computer hardware (i.e. CPU, memory, storage, etc.), low level programming (Assembly), operating system, middleware, and computer communication protocols.

• RESPEC: Translate an informal description of a problem into a precise requirements statement and develop specifications for a software system based on requirements.

• CODE: Write syntactically-correct source code, making appropriate use of fundamental constructs such as variables, branches, loops, and functions that solves a well-posed computational problem. Understand how computers process data, how to model domain concepts and procedures as data types and code, and how to formulate a human problem as an abstract computation.

• DESC: Design software exhibiting design best practices, such as clarity, structured programming, separation of concerns, and/or design principles and patterns, and describe it clearly (using e.g. pseudocode, database schema, flowcharts, etc.)
• TEST: Determine whether a program correctly meets its requirements, either through direct observation or the use of testing tools.
• ACODE: Write syntactically-correct and more advanced, nuanced C++ programming source code that make appropriate use of object-oriented concepts such as classes, encapsulation, and templates; and includes pointers, recursion, and memory management. Write source code with clear and informative comments following some coding standards or conventions.
• COOP: Cooperate effectively on a group project.
• PROC: Demonstrate knowledge of a formalized software engineering process (e.g. spiral, waterfall, agile).
• FDBK: Demonstrate ability to make improvements after receiving constructive feedback.
• ETH: Demonstrate an understanding of professional ethics appropriate to the use or development of computer science artifacts, and social impact of computer technology
PI Definitions -3

- IPSEC: Demonstrate an understanding of intellectual property laws and ethics, software licenses, and commensurate rights. Demonstrate an understanding of security, privacy, and other ethical or legal issues, that arise in the context of computing.
- WRITE: Write a clear document which meets the needs of the intended reader(s).
- SPEAK: Deliver a clear oral presentation which meets the needs of the intended listener(s).
- ALG: Design an algorithm to solve a novel computational problem that builds upon classical techniques (e.g. data structures, discrete mathematics tools, divide-and-conquer, dynamic programming) and analyze the algorithm in terms of formalisms such as asymptotic efficiency, lower bounds, or computational complexity.
- FB: Foundational Breadth: Demonstrate knowledge and competence in such fundamental areas of computer science as algorithms, design and analysis, computational theory, computer architecture, and software engineering
Guideline: The rubrics defined provide instructions on how to assess the performance indicators for Computer Science core courses. The instructors shall design the course work so that the data needed can be collected and evaluated. The following three categories are defined:

- Unsatisfactory: unable to achieve the basic skill and/or knowledge required by the performance indicator.
- Developing: able to achieve the basic skill and/or knowledge required by the performance indicator.
- Satisfactory: able to achieve advanced skill and/or knowledge required by the performance indicator.
# Rubrics Example

<table>
<thead>
<tr>
<th>Performance Indicator</th>
<th>Unsatisfactory</th>
<th>Developing</th>
<th>Satisfactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACODE SO 2</td>
<td>Unable to develop C++ programming source code that make appropriate use of object-oriented concepts and follow acceptable coding standards or conventions.</td>
<td>Able to write C++ programming source code that make use of some object-oriented concepts and follow some coding conventions.</td>
<td>Able to demonstrate the ability to write C++ programming source code that make appropriate use of object-oriented concepts such as classes, encapsulation, and templates; includes pointers, recursion, and memory management; and follow industry coding standards or conventions.</td>
</tr>
</tbody>
</table>

Please provide feedback on PIs and Rubrics of your classes. Design the course activities to assess the PIs, for example, by exam, homework, projects, and etc.. We will see two examples later. Note: It is your PIs, your Rubrics, use your effective approach to assess.
Improve the process

- Transparent! Routine! Value driven!
- Improve the change management to ensure the completeness and consistence.
- Developing a comprehensive Handbook based on current process, guidelines, metrics, and etc.
- Automation! Automation! Automation! Automation!
  More specific refinement will be discussed later.
Outline

1. Refinement of Assessment and Evaluation Process
2. The ABET Website
3. 2017-2018 Assessment Report
4. Continuous Improvement – what we have done
5. The Plan
The ABET Website

https://tryu-fullerton-edu.github.io/assessment
Outline

1. Refinement of Assessment and Evaluation Process

2. The ABET Website

3. 2017- 2018 Assessment Report

4. Continuous Improvement – what we have done

5. The Plan
## Data Collection Schedule

<table>
<thead>
<tr>
<th>Semester</th>
<th>120</th>
<th>121</th>
<th>131</th>
<th>223</th>
<th>240</th>
<th>304</th>
<th>311</th>
<th>315</th>
<th>323</th>
<th>332</th>
<th>335</th>
<th>351</th>
<th>362</th>
<th>440</th>
<th>471</th>
<th>481</th>
<th>Exit survey</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Fall</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
Student Outcome Assessment Criteria

The following formula used to assess Student Outcomes:

1. If no survey PIs are used in a SO:
   A SO Criteria is met if the total of SAT and DEV is more than or equal to 60%.

2. If survey PIs are used in a SO:
   A weighted system is used in which 80% is assigned to Course PIs and 20% is assigned to Survey PI. A SO criteria is met if \((SAT + DEV) \times 0.8 + \text{Survey Correct} \times 0.2 \geq 60\%\)
The Results (using old 11 SOs) - 1

A: Ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline:
- 240 HW, DM
- 335 GRAPH
- 440 HW
- 471 GRAPH
- 481 DM

Result:
Satisfactory: 60%
Developing: 25%
Unsatisfactory: 15%
Sat + Dev = 85% > 60% (Criteria Met)
B. Ability to analyze a problem, and identify and define the computing requirements appropriate to its solution:

131 ALTS
332 REQ
362 SPEC
440 ALTS

Result:
Satisfactory: 73.9%
Developing: 11.7%
Unsatisfactory: 14.4%

Sat + Dev = 84.6% > 60% (Criteria Met)
C. Ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

The result:

Satisfactory: 67.4%
Developing: 19.6%
Unsatisfactory: 13%

Sat + Dev = 87% > 60% (Criteria Met)
D. Ability to function effectively on teams to accomplish a common goal

323 COOP, FDBK
332 COOP, FDBK
362 PROC
Exit Survey PROC

Result:

Course PI (80% weight):
Satisfactory: 83.2%
Developing: 12.8%
Unsatisfactory: 4%
Sat + Dev = 96%

Survey PI (20% weight):
Ave Correct %: 55.5%

Weighted outcome:
96%*0.8 + 55.5%*0.2 = 87.9% > 60% (Criteria Met)
E. Understanding of professional, ethical, legal, security and social issues and responsibilities.

315 ETH, IP, SECISS
Exit Survey ETH, IP, SECISS

Result:

**Course PI:**
- Satisfactory: 86%
- Developing: 8.4%
- Unsatisfactory: 5.6%

Sat + Dev = 94.4% > 60% (Criteria Met)

**Survey PI:**
- Ave Correct%: 73.7%

Weighted Outcome = 94.4%*0.8 + 73.7%*0.2 = 90.2% > 60% (Criteria Met)
F. Ability to communicate effectively with a range of audiences.

121 CMNT
240 CMNT
311 WRITE, SPEAK
315 WRITE
362 SPEAK

Result:
Satisfactory: 84.3%
Developing: 11.3%
Unsatisfactory: 4.4%

Sat + Dev = 95.6% > 60% (Criteria Met)
G. Ability to analyze the local and global impact of computing on individuals, organizations, and society.

315 INF, IMP
Exit Survey INF, MILE, IMP

Result:

Course PI:
Satisfactory: 98.2%
Developing: 0.6%
Unsatisfactory: 1.2%
Sat + Dev = 98.8%

Survey PI:
Ave Correct %: Ave Correct %: 66.8

Weighted Outcome = 98.8%*0.8+66.8%*0.2 =92.4% > 60% (Criteria Met)
H. Recognition of the need for and an ability to engage in continuing professional development.

223 LANGS, DOCS
323 LANGS
471 DEPR
481 DOCS

Exit Survey LANGS, DOCS

Result:

Course PI:
Satisfactory: 51.8%
Developing: 25%
Unsatisfactory: 23.2%
Sat + Dev = 76.8%

Survey PI: Ave Correct %: 60%

Weighted Outcome = 76.8%*0.8+60%*0.2 =73.4% > 60% (Criteria Met)
The Results - 9

I. Ability to use current techniques, skills, and tools necessary for computing practice.
120 IDE
121 IDE
254 CMD, CASE
332 CASE

Result:
Satisfactory: 65.2%
Developing: 21.3%
Unsatisfactory: 13.5%

Sat + Dev = 86.5% > 60% (Criteria Met)
J. Ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrate comprehension of the tradeoffs involved in design choices.

131 BIGO
335 BIGO, ALG, PNP
351 CRIT
440 CRIT
481 ALG, PNP
Satisfactory: 46.2%
Developing: 24.4%
Unsatisfactory: 29.4%
Sat + Dev = 70.6% > 60% (Criteria Met)
K. Ability to apply design and development principles in the construction of software systems of varying complexity.

121 OOP
131 DS
223 FP OOP
335 DSGN
351 DSGN
362 DSGN

Result:
Satisfactory: 49.5%
Developing: 32.8%
Unsatisfactory: 17.7%
Sat + Dev = 82.3% > 60% (Criteria Met)
EPP Passing Rate:

Spring 2019 mid-semester: 78%
Spring 2019: 38%
Fall 2018 mid-semester: 71%
Fall 2018: 49%
Summer 2018: 52%
Spring 2018: 54%
Fall 2017 mid-semester: 52%
Fall 2017: 46%
Summer 2017: 69%
Spring 2017: 40%
Conclusion

• **Strengths:**

The results show that BS program of Computer Science Department is academically robust. The program performance is eminently satisfactory since all six ABET Student Outcomes exceed the thresholds set for compliance.

• **Areas for Improvement:**

- There are areas of softness as we analyze the outcomes at course or component levels. For example, consistency is an issue in some cases. The EPP passing rate is lower than the PI performance in 120, 121, and 131.

- Certain PI’s, particularly in some upper level courses for SO J and K, are relatively low, also appear to need attention.
1. Refinement of Assessment and Evaluation Process

2. The ABET Website

3. 2017-2018 Assessment Report

4. Continuous Improvement – what we have done

5. The Plan
Improvement Initiatives

- Refinement of ABET process
- Program Improvements
ABET Process Refinement

- Consolidation of PIs based on changes in SOs and CS curriculum topics
- Refinement of ABET Handbook
- Automation of data submission and analysis
- ABET workshop each semester (focusing on adjunct faculty)
- One Stop ABET Site
Program Improvement Initiatives Examples

- **Standardization of lower level CS courses (120, 121, and 131)**
  - High PIs numbers but low EPP passing rate
  - Has shown some improvement: Fall 2018 and Spring 2019’s EPP passing rates is higher.

- **Improvement of CS curriculum**
  - New courses: CPSC 375 (Introduction to data science and Big Data analytics), CPSC 411 (Mobile device application programming), CPSC 458 (Malware analysis), CPSC 459 (Blockchain technologies), CPSC 474 (Parallel and distributed computing)
  - Retired outdated courses: CPSC 223H (visual basic programming), CPSC 476 (Java enterprise application development), CPSC 303 (Multimedia concepts), CPSC 322L (intro to computer-aided design), CPSC 376 (Client/server systems with Java), CPSC 459 (Micro-computer software systems), CPSC 477 (Introduction to grid computing), CPSC 491T (Variable topics in computer science)
  - Courses with major changes: CPSC 473 (Web programming and data management) changed to two courses CPSC 473 (Web front-end engineering) and CPSC 476 (web back-end engineering), CPSC 483 (Data mining and pattern recognition) changed to (Introduction to machine learning),
Program Improvement Initiatives Examples

- **Standardization of lower level CS courses (120, 121, and 131)**
  - High PIs numbers but low EPP passing rate
  - Has shown some improvement: Fall 2018 and Spring 2019’s EPP passing rates is higher.

- **Improvement of CS curriculum**
  - New courses: CPSC 375 (Introduction to data science and Big Data analytics), CPSC 411 (Mobile device application programming), CPSC 458 (Malware analysis), CPSC 459 (Blockchain technologies), CPSC 474 (Parallel and distributed computing)
  - Retired outdated courses: CPSC 223H (visual basic programming), CPSC 476 (Java enterprise application development), CPSC 303 (Multimedia concepts), CPSC 322L (intro to computer-aided design), CPSC 376 (Client/server systems with Java), CPSC 459 (Micro-computer software systems), CPSC 477 (Introduction to grid computing), CPSC 491T (Variable topics in computer science)
  - Courses with major changes: CPSC 473 (Web programming and data management) changed to two courses CPSC 473 (Web front-end engineering) and CPSC 476 (web back-end engineering), CPSC 483 (Data mining and pattern recognition) changed to (Introduction to machine learning),
What we heard in last brainstorming session

Some Improvement Examples:
- Improvement of CS curriculum (continue)
  - CPSC 362 and CPSC 481 are designated as a capstone project course.
  - CPSC 254 (Open source systems) becomes an elective
  - Future changes (approved in spring 2019): CPSC 353 (Intro. to security) will be a core course. CPSC 311 and 440 (Computer architecture) will be changed from cores to electives, new courses to be proposed in Fall 2019 are CPSC 496 (undergraduate seminar) and 497 (senior capstone project) covering capstone project experience and upper-division writing.

Recent Textbook changes: CPSC 120, 121, 131, 353, 349, 449.
- Course Ownership Initiative: Specialty groups for CS courses are created to manage course change and ensure consistence.

- Cloud Computing Environment and etc.
- Individual Course Improvement

There is an improvement culture, faculty has done a lot to improve his/her courses: textbook changes, adopting tools/package, refinement in projects/assignment, updating course materials, and etc.
What we heard in last brainstorming session

Improvement Suggestions -1:

- hold tutoring sessions
- Build infrastructure for students to access Cloud (AWS or Azure)
- Involve students in internship and course independent projects.
- Use a common syllabus for all sections.
- Instructors teaching same class to collaborate to develop a consistent set of criteria (assignments).
- For 311. In addition to technical writing, emphasize verbal communications and presenting.
- Consider appointing at least one adjunct faculty to the Assessment Committee; Adjust the content of CPSC 311 to include topics in reading per revised UPS.
- Explain where data are stored; give examples on how to submit; the deadline to submit.
- Add active learning strategies to address larger class sessions
What we heard in last brainstorming session

**Improvement Suggestions - 2:**

- Suggestions to add more topics, 253 – discuss IP (open source); 311 and 315 – discuss IP; 481 – need to include ethics based on the new 2018 ACM Code of Ethics; 315: needs new unit on security, change or refine assignments; add online forum in Titanium to enhance communication; Add team evaluation as formal part of collaborative project; add pairs as option for kick-off to add small group activity: paper and presentation; 362: add security/ethics as part of large semester long project.
- Add Ethics PI to 311
- Allow adjunct faculty to provide inputs on course materials, textbooks, and etc.; It is hard to find the course objectives, make CS catalog more accessible.
- Embed more Cloud technology into the program, such as DevOps, Solution Architects, and Cloud Practitioner. For Cloud technology, provide AWS, Azure, and GCP.
Improvement Suggestions -3:

• Use shared Google drive to create a consistent course syllabus and materials for all 120 and 121 instructors; Need a grader!
• Create a shared ABET google drive with all info.
• Institutionalize ABET process, course assignment includes the current course PIs.
• Ensure consistence in 120, 121, and 131.
• Standardization of 253 is needed, as Linux and Open Source are important topics.
• Instructors need to improve teaching.
• Need a course to cover Test Driven Development, refactoring, dealing with legacy codebase, etc.
• Integrate REST into database class.
Outline

1. Refinement of Assessment and Evaluation Process
2. The ABET Website
3. 2017-2018 Assessment Report
4. Continuous Improvement – what we have done
5. The Plan
Plan

- **Fall 2019**
  - Release ABET website
  - Analyze Fall 2018 – Spring 2019 data
  - Collect artifacts (resume, syllabus, and etc)
  - Complete the Handbook
  - Gap analysis and fixing
  - Collect data
  - Initial draft of the Self-Study Report (SSR)

- **Spring 2020**
  - Continue fixing gaps identified
  - Continue to track improvement effort
  - Collect artifacts needed
  - Complete the 1st comprehensive SSR

- **Summer 2020**
  - Finalize the baseline version of SSR

- **Fall 2020**
  - Prepare for onsite visit
  - Data analysis for Fall 2018 – Spring 2020
  - Onsite visit
THANK YOU FOR YOUR ATTENTION!
Specialty Groups for Core Courses

- 120: Kevin, Michael, Doina, Christopher
- 121: Paul, Kevin, Mikhail, Christopher
- 131: Anand, Kevin, Mikhail, Christopher
- 223: Floyd, Mikhail, Christopher
- 240: Shawn, Mikhail
- 311: Doina
- 315: Doina, Christopher
- 323: James, Doina
- 332: Shawn, Christopher
- 335: Kevin, Bin
- 351: Mikhail, Wenlin, Yun
- 362: Jo, James, Bin
- 440: Ning, Mikhail
- 471: Yun, Mikhail
- 481: Wenlin, Paul, Anand, Mikhail, Christopher
Brainstorm Session

• Do you have any suggestions to improve ABET website?

• What improvement actions you have taken? What are the results? How do you sustain the improvement?

• Do you have any suggestions to improve the courses you are teaching? Or things we can do to improve our program to achieve program educational objectives?