SOFTWARE ENGINEERING BS

The BS degree in software engineering is accredited by the Engineering Accreditation Commission (EAC) of ABET, **http://www.abet.org/.**

Enrollment and Certification/Graduation Data

Fall 2018 Enrollment – 557
2017-2018 Graduates – 84 BS degrees conferred

Objectives

The program educational objectives are our expectations for what graduates of the program will accomplish within a few years of graduation. Within that timeframe, graduates of the BS in software engineering program are expected to have:

1. Demonstrated the development of software systems of varying size and complexity through the use of fundamental software engineering knowledge, work at various abstraction levels, and with appropriate design principles, tools, and technologies; (*PEO1: Software System Designers*)
2. Demonstrated the incorporation and management of domain-appropriate software processes, methodologies, and quality practices in software product development; (*PEO2: Software Product Developers*)
3. Demonstrated effectiveness as both an individual contributor and a member of a development team with professional, ethical, and social responsibilities; (*PEO3: Engineering Professionals*)
4. Demonstrated the improvement and expansion of professional skills through formal education, participation in professional activities, acquisition of new technical proficiencies, or development of managerial and leadership abilities. (*PEO4: Life-Long Learners*)

Outcomes

By graduation, students in the bachelor of science in software engineering program will have demonstrated:

a. An ability to apply knowledge of mathematics, science, and engineering
b. An ability to design and conduct experiments, as well as to 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 multidisciplinary 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

Updated January 2019 | source: RIT Institutional Research and Policy Studies

j. A knowledge of contemporary issues
k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

The following list provides a basis for these student outcomes in the context of software engineering activities:

1. Use of formal methods, use of finite state machines, software quality measurement and analysis, applications of CS theory, requirements gathering, software system design.
2. Design and analysis of usability tests, system debugging, testing and analysis, performance testing and analysis, project metric tracking.
3. Design to constraints and requirements; scope management for time constraints; addressing non-functional "ilities", such as, maintainability, testability, or evolvability; software security.
4. Working on teams, working in project domain areas where you have little or no experience.
5. Identifying project constraints and solutions, problem decomposition, requirements elicitation, design trade-off analysis.
6. Working with assignments in critical domains requiring consideration of safety and security, consideration of professional ethics and behavior, discussion of code of ethics, being a member of a team and participating in a professional manner, professional responsibility to provide fair and honest peer evaluations.
7. Preparation of documentation and presentations; making presentations; participating in team meetings, brainstorming session, code reviews and walkthroughs, or artifact reviews; customer or project sponsor interactions; use of discussion forums, such as, online discussions, blog, wikis.
8. Feasibility analysis, environmental impact analysis, support for different languages, and usability analysis for people with impairments.
9. Using a new development tool, software framework, or programming language; discussion of careers and professional activity; working on problems requiring outside of class research.
10. Discussions of the profession, use of new technology or methodologies, effects of new methods to collaborate and communicate.
11. Using process methodologies, design methodologies, development tools, software frameworks, programming languages, middleware.

While we believe these program educational objectives and student learning outcomes are appropriate, we realize that they must be continuously reevaluated in terms of feedback from our primary constituencies: students and employers. In the case of students, cooperative education reports provide immediate but short-term evaluation of the curriculum. The program's Industrial Advisory Board (IAB) supplies a more strategic perspective - the IAB meets once a year to review the current status and to both initiate and evaluate changes to the curriculum.