About Software Engineering at RIT

We offer a challenging undergraduate program that prepares students for the demands and challenges of the software industry. The undergraduate degree consists of both core and elective courses that focus on the Software Engineering lifecycle. In addition, each student must complete a senior project on a team of four or five students.

Our software engineering program has four key elements: engineering design, software product development, teamwork, and communication. The curriculum ensures that the student's coursework balances between software design principles and software process practices. In every course, teamwork is emphasized with a significant part of the final grade being based on team project activities. By the time our students start their senior project, they will have worked on 20 to 25 different student teams in their software engineering courses. Software engineering students also develop their communication skills. In every course, they will be preparing engineering documentation, such as: requirements documents, design documents, project plans, burndown charts, and software test plans. Also, at the end of each project cycle the students make oral presentations on their work and receive critique from the instructor and other students in the class.

The RIT Software Engineering program focuses on developing skills to:

1. Model and analyze proposed and existing software systems, especially through the use of discrete mathematics and statistics.
2. Apply quality principles to the definition of software systems and processes.
3. Analyze and design complex software systems using contemporary principles such as cohesion and coupling, abstraction and encapsulation, design patterns, frameworks and architectural styles.
4. Apply contemporary software engineering methods to planning, management, and development of software systems.
5. Accurately communicate technical material related to all phases of the software life cycle via concise and correct documents, graphics and oral presentations.
6. Work in small teams to develop a software system. This includes the ability to assume distinct operational roles (e.g., configuration management, quality assurance) in addition to design and implementation.
7. Assess the social environmental, and cultural factors arising from existing software systems as well as potential risks of proposed systems with a clear understanding of the ethical and professional responsibilities necessary for different software product lines.

8. Relate principles of software engineering to at least one application domain where those principles can be applied.

9. Explore new topics in software engineering or related application domains with limited oversight and input from faculty or mentors.

10. Rapidly learn, assess, and adapt to new languages, environments, and paradigms for software development.

With the skills obtained in our program, software engineering students will be able to design and build quality software solutions that meet the customer's requirements, are delivered on time, without defects, and within budget.