

Students pursuing the M.S. in Computer Science and Software Engineering degree are expected to have a background in software engineering concepts and practices for admission to the degree program, and in particular prior to enrolling in core and elective courses such as *CSS 555 Evaluating Software Design* and *CSS 553 Software Architecture*.

As a student, you will need to understand and use terminology, concepts, practices, and artifacts from across the software lifecycle such as defined by the *Software Engineering Body of Knowledge* (SWEBOK) [<http://www.swebok.org>]. You should have knowledge of multiple methodologies and approaches and be able to select and justify practices based on organizational contexts and project requirements. Prior job experience may be helpful but likely needs to have been augmented by the study of software engineering to be fully prepared for class discussion and project work..

You will be expected to be able to have detailed knowledge about plan-driven (prescriptive), agile, and “lean” approaches to software development. You should be able to discuss solutions and approaches from software engineering, computer science, and systems engineering perspectives. You should understand a variety of software development lifecycle (SDLC) models including:

- “Waterfall” model and variants
- Iterative / incremental variants
- Spiral model
- Unified Process (UP) and variants
- Scrum and variants (“scrummy” and “scrumfall” approaches)
- Test-Driven Development (TDD) and related variants (FDD, ADD)

You should use be able to use an understanding of different SDLCs along with analysis, design, and construction concepts and practices with an organization and team context to identify, implement, and manage software development efforts.

In addition, you should understand concepts and apply practices appropriate across the software lifecycle including:

- Requirements Engineering
- Analysis to Design methodologies and approaches
- Project Management
- Risk Management
- Configuration Management (CM) / Software Configuration Management (SCM)
- Software Quality Assurance (SQA) including all aspects of Validation and Verification (V&V) from inception through transition and support:
  - Testing approaches such as “blackbox” and “whitebox”
  - Testing across the software lifecycle (acceptance, integration, and unit testing)
  - Testing for functional and non-functional requirements
  - Inspections and reviews
- Quality Management (CMMI, TSP, and PSP)

Documentation and diagrams are important to communicate about different aspects of software projects. You will be expected to read, create, and discuss software lifecycle artifacts such as:

- Business proposals and reports to justify software projects and software engineering practices
- Formal Software Requirement Specifications (SRS)
- Project and Sprint Backlogs
- User Stories
- Unified Modeling Language (UML) and related object-oriented artifacts
  - Use Cases including persona, scenario, and conversational forms
  - Use Case Diagrams
  - Robustness Diagrams
  - Static Structure Diagrams such as domain diagrams and class diagrams
  - Package Diagrams
  - Sequence Diagrams
  - State Diagrams
  - Component Diagrams
- Data Flow Diagrams (DFDs)
- Cyber security artifacts such as Misuse Cases and Threat Models

You should understand the importance of software engineering to industry and society. What is expected in the role as a software engineering professional including business, ethical, and societal responsibilities. You should be able to communicate about all aspects of software projects and software lifecycle practices to both technical and non-technical stakeholders. You should understand common failures of software projects and use methods to avoid and mitigate such possibilities. You should understand software as intellectual property that has legal protections and employment contractual obligations. You should be prepared to address the increasing importance of information assurance and cyber security in all aspects of the software lifecycle.

If you do not feel prepared with respect to software engineering knowledge, as described above, you should contact the CSS Advising Office for more information including supplemental programs that may be able to provide you with this knowledge, such as the [Graduate Certificate in Software Design & Development](#).