



## CET305

### SOFTWARE ENGINEERING

#### GENERAL OBJECTIVES

Many software systems are still being produced that are unreliable, over budget, poorly documented and not well suited to the user. A well engineered software system must be reliable, understandable, and maintainable.

A disciplined software engineering approach is absolutely critical to minimize the common problems with software. Most organizations follow a structured and disciplined approach to software development. They use different tools, techniques and methodologies but for the most part follow the System Development Life Cycle (SDLC).

In this course we will follow a systematic approach to software development and software maintenance. The student will have a working knowledge of the basic concepts of software engineering by using current tools and techniques.

The student will gain a thorough understanding of the SDLC through the preparation of deliverables (documents, discussions, coding) at each stage.

Communication is the key to success in software development and therefore documentation, oral, written and interpersonal communication skills will be the main focus of this course.

#### TEXTBOOK:

1. "TOOLS AND TECHNIQUES FOR STRUCTURED SYSTEMS ANALYSIS AND DESIGN" BY WILLIAM S. DAVIS.
2. HAND OUT MATERIAL BY INSTRUCTOR

#### ASSESSMENT:

Tests and Quizzes	40%
One Major Term Project	60%

Some minor modifications to the above percentages may be necessary. The instructor reserves the right to adjust the mark up or down 5% based on attendance, participation, leadership, creativity and whether there is an improving trend.

- \* All Assignments must be completed satisfactorily to complete the course. Late hand in penalties will be 5% per day. Assignments will not be accepted past one week late unless there are extenuating and legitimate circumstances.

## SPECIFIC OBJECTIVES

### BLOCK 1 INTRODUCTION TO SOFTWARE ENGINEERING

At the end of this block, the student shall be able to:

1. Explain the historical evolution of Software Engineering.
2. Define the terms pertinent to Software Engineering.
3. Identify and describe several tools available that provide discipline and structure to Software Development.
4. Describe the quality, effectiveness, productivity and political aspects of Software Projects.
5. Overview the general system requirements for the term project.
6. Produce the Problem Statement.
7. Document the Project plan for Management Backlog Of Work.
8. Describe all the stages in SDLC.

### BLOCK 2 PLANNING A SOFTWARE PROJECT (PROJECT MANAGEMENT)

At the end of this block, the student shall be able to:

1. Understand the concept of planning and its relevance.
2. Define project goals and requirements.
3. Discuss the relationship of planning with respect to project size.
4. Discuss the project planning development process including:
  - a) Project Phases
  - b) Milestones, Documents, Reviews
  - c) The cost aspects of each phase of the project
  - d) Prototyping
  - e) Successive versions
5. Discuss the project planning organizational structures including:
  - a) Project Format
  - b) Project team structure
  - c) Project quality assurance
  - d) Project verification and validation
6. Produce the project feasibility study. (also known as engineering study)

### BLOCK 3: SOFTWARE REQUIREMENT SPECIFICATION

At the end of this block, the student shall be able to:

1. Define the characteristics contained in a software requirement specification.
2. USE the JAD mechanism to define the problems and issues.
3. Define the project scope.
4. Produce a Requirement Specification Document.
5. Make a management presentation on system requirements, issues and recommendations.

### BLOCK 4: SYSTEMS ANALYSIS AND DESIGN

At the end of this block the student shall be able to:

1. Demonstrate communication skills through meetings and user interviews.
2. Produce a Functional Specification Document.
3. Make a management overview presentation of the system prior to the programming stage.
4. Obtain management approval to proceed.
5. Discuss design analysis tools such as
  - a) milestones
  - b) walkthroughs
  - c) inspections
  - d) Data Flow Diagrams
6. Discuss design guidelines, policies and procedures.

### BLOCK 5: COMPUTER SYSTEM DESIGN AND PROGRAMMING

At the end of this block the student shall be able to:

1. Discuss the technical issues of converting the functional specification to a computer system.
2. Define the file sizes and tool limitations.
3. Describe structured coding techniques and style.
4. Describe programming guidelines.
5. Describe documentation guidelines.
6. Prepare program specifications.
6. Document and schedule the term software project critical design review.

## BLOCK 6: VERIFICATION AND VALIDATION TECHNIQUES

At the end of this block the student shall be able to:

1. Discuss the purpose of software quality assurance.
2. demonstrate the usefulness of walkthroughs and inspections throughout the software life cycle.
3. Demonstrate unit testing and debugging.
4. Describe software system testings such as:
  - a) integration testing
  - b) acceptance testing
5. Produce a system overview document.
6. Post implementation managerial presentation.

## BLOCK 7: SOFTWARE MAINTENANCE

At the end of this block the student shall be able to:

1. Define the maintenance phase of the software life cycle.
2. Describe the activities that enhance maintainability during development.
3. Discuss the managerial aspects of software maintenance.
4. Discuss automated tools for software maintenance.

## GRADING SCHEME

### 1. TESTS

Written tests will be conducted as deemed necessary; generally at the end of each block of work. They will be announced about one week in advance. Quizzes may be conducted without advance warning.

### 2. ASSIGNMENTS

Assignments not completed by the assigned due-date will be penalized by 5% per day late. All assignments must be completed satisfactorily to complete the course.

### 3. GRADING SCHEME

A+	90	-	100%	Outstanding achievement
A	80	-	89%	Excellent achievement
B	70	-	79%	Average Achievement
C	55	-	69%	Satisfactory Achievement

U Incomplete: Course work not complete at Mid-term. Only used at mid-term.

R Repeat

X A temporary grade that is limited to instances where special circumstances have prevented the student from completing objectives by the end of the semester. An X grade must be authorized by the Chairman. It reverts to an R if not upgraded in an agreed-upon time, less than 120 days.

### 4. UPGRADING OF INCOMPLETE

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when the student's performance warrants it. Attendance and assignment completion will have a bearing on whether upgrading will be allowed. A failing grade on all tests will remove the option of any upgrading and an R grade will result. The highest grade on re-written tests or assignments will be 56%.

Where a student's overall performance has been consistently unsatisfactory, an R grade may be assigned without the option of make-up work.

The method of upgrading is at the discretion of the teacher and may consist of one or more of the following options: assigned make-up work, re-doing assignments, re-writing of tests, or writing a comprehensive supplemental examination.