

IT3205: Fundamentals of Software Engineering (Compulsory)

INTRODUCTION

This course is designed to provide the students with the basic competencies required to identify requirements, document the system design and maintain a developed system. It presumes a general understanding of computers and programming which are covered in the first and second semester of the degree.

Software engineering is a subject that emerged recently as a result of the need to manage software projects that are rising in demand day by day. Software is developed in diverse areas and the fact that a systematic approach is required to manage their development spawns this interesting subject of study.

CREDITS: 03

LEARNING OUTCOMES

After successfully completing this module the student will be able to explain the software engineering principles and techniques that are used in developing quality software products.

MINOR MODIFICATIONS

When minor modifications are made to this syllabus, those will be reflected in the Virtual Learning Environment (VLE) and the latest version can be downloaded from the relevant course page of VLE. Please inform your suggestions and comments through the VLE. <http://vle.bit.lk>

ONLINE LEARNING MATERIALS AND ACTIVITIES

You can access all learning materials and this syllabus in the VLE: <http://vle.bit.lk/>, if you are a registered student of BIT degree program.

ONLINE ASSIGNMENTS

The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). Maximum mark for a question is 10, minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering 40% of assignment quiz 1 and 60% of assignment quiz 2. Pass mark for the online assignments in a course is 50. You are advised to do online assignments before the final exam of the course. It is compulsory to pass all online assignments to partially qualify to obtain year 2 certificate.

FINAL EXAMINATION

Final exam of the course will be held at the end of the semester. Each course in the semester 3 is evaluated using a two hour question paper which consists of 20-25 MCQs and 3-4 structured questions.

OUTLINE OF SYLLABUS

Topic	Hours
1. Introduction	04
2. Software Development Process Models	08
3. Requirements Analysis & Specification	08
4. Design	08
5. Coding	03
6. Software Testing and Quality Assurance	08
7. Software Maintenance	03
8. Managing Software Projects	03
Total for the Subject	45

REQUIRED MATERIALS**Main Reading**

Ref. 1: Software Engineering by Ian Sommerville, 7th edition, Addison-Wesley, 2006.

Ref. 2: Software Engineering: A practitioner's approach by Roger S. Pressman, 6th edition, McGraw-Hill International edition, 2005.

Ref. 3: <http://www.softwareengineering-9.com/>

DETAILED SYLLABUS**Section 1: Introduction (04 hrs.)****Learning Objectives**

- Describe what software is, different types of software and software quality attributes
- Describe with the problems associated with software and software development
- Define what software engineering is and explain why it is important
- State some professional issues related to software development

Sub Topics

- 1.1. Software
 - 1.1.1. What is software? [Ref.1: pg. 27-28]
 - 1.1.2. Types of software [Ref.2: pg. 40-41]
 - 1.1.3. Characteristics of Software [Ref.2: pg. 36-39]
 - 1.1.4. Attributes of good software [Ref.1: pg. 34-35]
- 1.2. Software Engineering
 - 1.2.1. What is software engineering? [Ref.1: pg. 29]
 - 1.2.2. Software engineering costs [Ref.1: pg. 31-33]
 - 1.2.3. What are the key challenges facing software engineering?
[Ref.1: pg. 35-36]
 - 1.2.4. Systems engineering & software Engineering
[Ref.1: pg. 29-30; Ref. 2: pg. 154-155]
 - 1.2.5. Professional Practice [Ref.1: pg. 36-39; Ref.2: pg. 128-149]

Section 2: Software Development Process Models (08 hrs.)**Learning Objectives**

- Describe different process models used for software development
- Identify the most appropriate software process model for a given problem
- Identify how CASE tools can be used to support software process activities

Sub Topics

- 2.1. What is a software process? [Ref.1: pg. 30]
- 2.2. What is a software process model? [Ref.1: pg. 30-31, 87-88]
- 2.2.1. The waterfall model [Ref.1: pg. 88-90; Ref.2: pg. 79-80]
 - 2.2.2. Evolutionary development [Ref.1: pg. 90-91; Ref.2: pg. 83-85]
 - 2.2.3. Component-Based Software Engineering (CBSE) [Ref.1: pg. 91-93]
- 2.3. Process Iteration [Ref.1: pg. 93]
- 2.3.1. Incremental delivery [Ref.1: pg. 93-95; Ref.2: pg. 80-81]
 - 2.3.2. Spiral development [Ref.1: pg. 95-96; Ref.2: pg. 86-88]
- 2.3. Rapid software development
- 2.4.1. Agile methods [Ref.1: pg. 418-420]
 - 2.4.2. Extreme programming [Ref.1: pg. 420-427]
 - 2.4.3. Rapid application development (RAD) [Ref.1: pg. 427-431; Ref. 2: pg. 81-83]
 - 2.4.4. Software prototyping [Ref.1: pg. 431-434]
- 2.4. Rational Unified Process (RUP) [Ref.1: pg. 104-107; Ref.2: pg. 94-99]
- 2.5. Computer Aided Software Engineering (CASE)
- 2.6.1. Overview of CASE approach [Ref.1: pg. 107-108]
 - 2.6.2. Classification of CASE tools [Ref.1: pg. 108-111]

Section 3: Software Requirement Analysis and Specification (08hrs.)

Learning Objectives

- Identify the types of requirements, which should be captured in a software project
- Describe and apply different requirement analysis and specification techniques
- Prepare a software requirement specification (SRS) for a given software problem

Sub Topics

3.1.	System and software requirements	[Ref. 2: pg. 155-157]
3.2.	Types of software requirements	
3.2.1.	Functional and non-functional requirements	[Ref.1: pg. 141-147]
3.2.2.	Domain requirements	[Ref. 1: pg. 147-149]
3.2.3.	User requirements	[Ref. 1: pg. 149-151]
3.3.	Elicitation and analysis of requirements	
3.3.1.	Overview of techniques	[Ref. 1: pg. 168-171]
3.3.2.	Viewpoints	[Ref. 1: pg. 171-174]
3.3.3.	Interviewing	[Ref. 1: pg. 174-175]
3.3.4.	Scenarios	[Ref. 1: pg. 175-176]
3.3.5.	Use-cases	[Ref. 1: pg. 176-178]
3.3.6.	Ethnography	[Ref. 1: pg. 179-180]
3.4.	Requirement validation	[Ref. 1: pg. 180-182]
3.5.	Requirement specification	[Ref.1: pg. 158-161]
3.6.	Feasibility	[Ref. 1: pg. 166-168]

Section 4: Software Design (08hrs.)

Learning Objectives

- Describe the important software design issues and concepts
- Compare different approaches to software design
- Identify suitable design approaches for a problem

Sub Topics

- | | | |
|------|--|--|
| 4.1. | Design concepts | |
| | 4.1.1. | Abstraction [Ref. 2: pg. 265] |
| | 4.1.2. | Architecture [Ref. 2: pg. 265-266] |
| | 4.1.3. | Patterns [Ref. 2: pg. 266] |
| | 4.1.4. | Modularity [Ref. 2: pg. 267-268] |
| | | 4.1.4.1. Cohesion [Ref. 2: pg: 335-336] |
| | | 4.1.4.2. Coupling [Ref. 2: pg: 337-338] |
| | 4.1.5. | Information hiding [Ref. 2: pg. 268] |
| | 4.1.6. | Functional independence [Ref. 2: pg. 268-269] |
| | 4.1.7. | Refinement [Ref. 2: pg. 269] |
| 4.2. | Architectural design | |
| | 4.2.1. | Repository model [Ref. 1: pg: 269-271] |
| | 4.2.2. | Client-server model [Ref. 1: pg. 271-272, 292-297] |
| | 4.2.3. | Layered model [Ref. 1: pg. 272-274] |
| | 4.2.4. | Modular decomposition [Ref. 1: pg. 274-275] |
| 4.3. | Procedural design using structured methods | [Ref. 2: pg. 347-348] |
| 4.4. | User Interface design | |
| | 4.4.1. | Human-computer interaction [Ref. 1: pg. 389-392] |
| | 4.4.2. | Information presentation [Ref. 1: pg. 392-398] |
| | 4.4.3. | Interface evaluation [Ref. 1: pg. 405-407] |
| 4.5. | Design notations | [Ref. 1: pg. 347-353] |

Section 5: Coding (03hrs.)

Learning Objectives

- Select appropriate programming language and development tools for a given problem
- Identify the features of a good program, good programming practices and program documentation

Sub Topics

- 5.1. Programming languages and development tools
- 5.2. Selecting languages and tools
- 5.3. Good programming practices [Ref. 2: pg. 144-146]

Section 6: Software Testing and Quality Assurance (08hrs.)

Learning Objectives

- State the software testing process and required documentation
- Explain the different software testing techniques and integration strategies
- Design test cases and write test programs for a given simple software problem
- Describe the code verification techniques
- Describe the importance of software quality
- Distinguish the difference between product quality and process quality
- Describe some important quality standards with respect to software

Sub Topics

- 6.1. Verification and validation [Ref. 1: pg. 538-541]
- 6.2. Techniques of testing
 - 6.2.2. Black-box and White-box testing [Ref. 2: pg. 423-441]
 - 6.2.3. Inspections [Ref. 1: pg. 543-549]
- 6.3. Levels of testing
 - 6.3.1. Unit testing [Ref. 1: pg. 569-571]
 - 6.3.2. Integration Testing [Ref. 1: pg. 563-565; Ref.2: pg. 397-401]

6.3.3.	Interface testing	[Ref. 1: pg. 571-573]
6.3.4.	System testing	[Ref. 2: pg. 408-411]
6.3.5.	Alpha and beta testing	[Ref. 2: pg. 406-407]
6.3.6.	Regression testing	[Ref. 2: pg. 401]
6.3.7.	Back-to-back testing and Thread testing	
6.3.8.	Statistical Software Testing	
6.3.9.	Object Oriented Testing	
6.4.	Design of test cases	[Ref. 1: pg. 573-585]
6.5.	Quality management activities	[Ref. 1: pg. 664-666]
6.6.	Product and process quality	[Ref. 1: pg. 666-667]
6.7.	Standards	[Ref. 1: pg. 667-674]
6.7.1.	ISO9000	[Ref. 1: pg. 670-671]
6.7.2.	Capability Maturity Model (CMM)	[Ref. 2: pg. 59-63]

Section 7: Software Maintenance (03hrs.)

Learning Objectives

- Describe the types of software maintenance
- Describe the software maintenance process
- Describe activities of configuration management

Sub Topics

7.1.	Evolving nature of software	[Ref. 1: pg. 512-514]
7.1.1.	Different types of maintenance	[Ref. 1: pg. 514-518]
7.1.1.1.	Fault repair	[Ref. 1: pg. 514-518]
7.1.1.2.	Software adaptation	[Ref. 1: pg. 514-518]
7.1.1.3.	Functionality addition or modification	[Ref. 1: pg. 514-518]
7.1.2.	Maintenance prediction	[Ref. 1: pg. 518-520]
7.1.3.	Re-engineering	[Ref. 1: pg. 523-526]

- 7.2. Configuration Management (CM)
 - 7.2.1. Importance of CM [Ref. 1: pg. 712-714]
 - 7.2.2. Configuration items [Ref. 1: pg. 715-716]
 - 7.2.3. Versioning [Ref. 1: pg. 720-727]

Section 8: Software Project Management (03hrs.)

Learning Objectives

- State the requirement of managerial control of the development process
- Describe the main phases of software project management
- Describe project planning and project scheduling activities in detail

Sub Topics

- 8.1. Need for the proper management of software projects [Ref. 1: pg. 115-116]
- 8.2. Management activities
 - 8.2.1. Project planning [Ref.1: pg. 118-121]
 - 8.2.2. Estimating costs [Ref.1: pg. 635-636, 660]
 - 8.2.3. Project scheduling [Ref. 1: pg. 121-126]
 - 8.2.4. Risk management [Ref. 1: pg. 126-133]
 - 8.2.5. Managing people [Ref. 1: pg. 614-631]