www.kamaldnp.com

KAMALDNP
kamaldnp.com
PDF created by Raj Kamal using AI

Software Engineering — Complete Syllabus

BTEUP Diploma in Information Technology • 3rd Semester • Session 2026–27
Subject 3.3 (Practicum) • Total 100 Marks • 4 Credits • L 3 – P 2

About this document Software Engineering is subject 3.3 of the 3rd semester of the Diploma in Information Technology. It is a practicum (3 lecture and 2 practical periods a week) WITH a 60-mark theory external paper: 40 marks internal theory and 60 marks external theory (3 hours). This document gives the complete unit-wise syllabus, the examination scheme, the practical exercises, the course outcomes and the reference books from the official curriculum.

Examination Scheme

Code	Paper	Type	L – P	Credits	Internal	External	Total
3.3	Software Engineering	Practicum	3 – 2	4	Theory 40	Theory 60 (3 hrs)	100
Software Engineering Total				**4**	**40**	**60**	**100**

What This Subject Aims To Do

The system analysis and design is the backbone of application software development. After studying the subject the students will be able to develop and design the system according to given requirements. It involves various steps in analysis and design of the system. It includes the knowledge of preparing project systematically. It is important to know about various aspects of the system analysis and design so that the students will be able to understand the responsibilities while designing and implementing the project.

www.kamaldnp.com

Unit-Wise Complete Topic List

Unit	Topic	Periods	Marks (%)
1	Introduction to Software Engineering	8	20
2	Software Life Cycle Models	8	20
3	Software Planning, Requirement Analysis and Specification	10	20
4	Software Design and Implementation	8	20
5	Software Testing and Maintenance	8	20

Total: 42 periods, 100%. The percentages are the suggested distribution of marks in the official syllabus, out of the 60-mark external theory paper. All five units carry the same 20%, so no unit can be skipped.

Unit 1 — Introduction to Software Engineering (8 Periods, 20%)

Topics: System concept, types of systems (open, closed, static and dynamic systems). Basics of software engineering, need for software engineering — definition, programs vs software products, software characteristics.

In simple words: What software engineering is and why it exists — the kinds of systems software works in, and how a software product differs from a single program.

Unit 2 — Software Life Cycle Models (8 Periods, 20%)

Topics: Requirement of life cycle model, classical waterfall model, prototyping model, evolutionary model, spiral model, introduction to agile methodology.

Practical exercise: Perform comparative analysis of various models.

In simple words: The different ways a software project can be organised from start to finish — waterfall, prototyping, evolutionary, spiral — and an introduction to agile, with their strengths and weaknesses compared.

KAMALDNP

www.kamaldnp.com

Unit 3 — Software Planning, Requirement Analysis and Specification (10 Periods, 20%)

Topics: Requirement gathering and analysis, Software Requirement Specifications (SRS), characteristics of good SRS. Metrics for project size estimation — LOC (Lines of Code), Function Point Metric. Project estimation techniques — using COCOMO model. Responsibilities of Software Project Manager.

Practical exercise: Develop an SRS on a given topic/project/problem.

In simple words: How a project is planned before a line of code is written — gathering requirements, writing them down as an SRS, and estimating size and effort with LOC, function points and COCOMO.

Unit 4 — Software Design and Implementation (8 Periods, 20%)

Topics: Characteristics and features of good software design, cohesion and coupling. Software design approaches — function-oriented design (data flow diagrams, data dictionary, decision tree and tables), object-oriented design. Structured coding techniques, coding styles and documentation.

Practical exercise: Develop DFD model (level 0 and level 1 DFD) of the problem.

In simple words: How a design is drawn before it is coded — data flow diagrams, cohesion and coupling, function-oriented vs object-oriented design, and the coding styles that keep code readable.

Unit 5 — Software Testing and Maintenance (8 Periods, 20%)

Topics: Concept of testing, unit testing, black box testing, white box testing, alpha, beta testing, integration testing, system testing, testing types, cycle (V-Model), verification vs validation, configuration management, overview of test cases.

Practical exercise: Use of testing tools such as JMeter, Canoo WebTest. Write test cases for any known application.

In simple words: How software is checked before it is delivered — unit, integration and system testing, black box vs white box, the V-Model, and writing test cases; plus configuration management, which keeps track of every version.

Course Outcomes

- CO1: Understand the basic concepts of software development.
- CO2: Classify life cycle models and their comparison.
- CO3: Will be able to write the SRS for the given problem and project planning estimation techniques.
- CO4: Explain various software design strategies.

www.kamaldnp.com

- CO5: Explain about different testing methods, need for testing.

KAMALDNP • kamaldnp.com • support@kamaldnp.com
Source: BTEUP IRDT Kanpur NEP-2020 curriculum for Diploma in Information Technology, effective session 2026-27 (urise.up.gov.in). Topic lists follow the official syllabus; short explanations are the author's own plain-English summaries, not syllabus text. Please confirm with your own polytechnic for any local changes.

