SYLLABUS
CY3103PC: SOFTWARE ENGINEERING

II-II:CSE(DS)								
Course Code	**Category**	**Hours/Weak**			**Credits**	**Max Marks**		
		L	T	P	C	CIE	SEE	Total
CY3103PC	**Core**	**3**	**0**	**0**	**3**	**30**	**70**	**100**
Contact Classes:45	**Tutorial classes:15**	**Practical classes: Nill**				**Total Classes:60**		
Prerequisites: None								

Course Objectives:

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of larges oft ware development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes

- Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
- Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
- Will have experience and / or awareness of testing problems and will be able to develop a simple testing report

UNIT-I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths .**A Generic view of process:** Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models. **Process models:** The waterfall model, incremental process models, evolutionary process models, the unified process.

UNIT-II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. **Requirements engineering process:** Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. **System models:** Context models, behavioral models, data models, object models, structured methods.

UNIT-III

Design Engineering: Design process and design quality, design concepts, the design model. **Creating anarchitecturaldesign:**softwarearchitecture,datadesign,architecturalstylesandpatterns,architecturaldesign,conceptualmodelofUML,basicstructuralmodeling,classdiagrams,sequencediagrams,collaborationdiagrams,usecasediagrams,componentdiagrams.

UNIT-IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT-V

Metrics for Process and Products: Software measurement, metrics for software quality. **Risk management:** Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan. **Quality Management:** Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXTBOOKS:

1. Software Engineering, Apractitioner's Approach-RogerS.Pressman, 6th edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7thedition, PearsonEducation.
3. The unified modeling language user guide Grady Booch, James Rambaugh, IvarJacobson, Pearson Education.

REFERENCEBOOKS:

1. Software Engineering, an Engineering approach-James F.Peters, Witold Pedrycz, JohnWiley.
2. Software Engineering principles and practice-Waman S Jawadekar, The Mc Graw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.