CS9254 SOFTWARE ENGINEERING

L	T	P	C
3	0	0	3

AIM:
The course is intended to give Software Engineering principles in classical sense.

OBJECTIVES:

- To be aware of generic models to structure the software development process.
- To understand fundamental concepts of requirements engineering and requirements specification.
- To understand different notion of complexity at both the module and system level
- To be aware of some widely known design methods.
- To understand the role and contents of testing activities in different life cycle phases.

UNIT I SOFTWARE PROCESS MODELS 9

The Evolving role of Software – Software – The changing Nature of Software – Legacy software —A generic view of process– A layered Technology – A Process Framework – The Capability Maturity Model Integration (CMMI) – Process Assessment – Personal and Team Process Models. Product and Process. Process Models – The Waterfall Model – Incremental Process Models – Incremental Model – The RAD Model – Evolutionary Process Models – Prototyping – The Spiral Model – The Concurrent Development Model – Specialized Process Models – the Unified Process.

UNIT II REQUIREMENTS ENGINEERING 9

Software Engineering Practice – communication Practice – Planning practice Modeling practice– Construction Practice –Deployment. Requirements Engineering - Requirements Engineering tasks – Initiating the requirements Engineering Process- Eliciting Requirements – Developing Use cases – Building the Analysis Models – Elements of the Analysis Model – Analysis pattern – Negotiating Requirements – Validating Requirements.

UNIT III MODELING 9

Requirements Analysis – Analysis Modeling approaches – data modeling concepts – Object oriented Analysis – Scenario based modeling – Flow oriented Modeling – Class based modeling – creating a behaviour model.

UNIT IV SOFTWARE DESIGN 9

Design Engineering – Design process -Design Quality-Design model-User interface Design – Testing strategies- strategies Issues for conventional and object oriented software-validation testing –system testing –Art of debugging – Project management

UNIT V SOFTWARE MEASUREMENT 9

Software evolution - Verification and Validation -Critical Systems Validation – Metrics for Process, Project and Product-Quality Management -Process Improvement –Risk Management- Configuration Management

TOTAL= 45 PERIODS

TEXT BOOKS:

1. Roger S.Pressman, "Software Engineering: A Practitioner's Approach", McGraw Hill International edition, Sixth edition, 2005.
2. Ian Sommerville, "Software Engineering", 8th Edition, Pearson Education, 2008(UNIT V)

REFERENCES:

1. Stephan Schach, "Software Engineering", Tata McGraw Hill, 2007.
2. Pfleeger and Lawrence, "Software Engineering: Theory and Practice", Pearson Education, second edition, 2001