RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Scheme of

POSTGRADUATE DEGREE COURSE

M.Tech. I to IV Semester

Software Engineering

(Effective from academic session: 2020-21)

Rajasthan Technical University, Kota
Akelgarh, Rawatbhata Road, Kota-324010

28.06.2020

Office of Dean Academic Affairs
Rajasthan Technical University, Kota

Teaching & Examination Scheme
M.Tech.: Software Engineering
1st Year –I Semester

THEORY												
SN	Category	Course		Contact hrs/week			Marks				Cr
		Code	Title	L	T	P	Exm Hrs	IA	ETE	Total	
1	PCC	1MSW1-01	Statistical Methods in Computer Science	3	0	0	3	30	70	100	3
2	PCC	1MSW1-02	Software Architecture	3	0	0	3	30	70	100	3
3	PCC	1MSW1-03	Software Metrics	3	0	0	3	30	70	100	3
4	PEC		**Elective-I:**	3	0	0	3	30	70	100	3
		1MSW2-11	Software Project Management								
		1MSW2-12	Software Testing								
		1MSW2-13	Requirements Engineering								
5	MCC	1MCC3-21	Research Methodology and IPR	2	0	0	2	30	70	100	2
		Sub Total						**150**	**350**	**500**	**14**
PRACTICAL & SESSIONAL											
6	PCC	1MSW1-06	Software Design Lab	0	0	4	4	60	40	100	2
		1MSW1-07	Software Testing Lab	0	0	4	4	60	40	100	2
7	SODECA	1MSW5-00	Social Outreach, Discipline & Extracurricular Activities							100	2
		Sub- Total						**120**	**80**	**300**	**6**
		TOTAL OF I SEMESTER						**270**	**430**	**800**	**20**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*

Office of Dean Academic Affairs
Rajasthan Technical University, Kota

Teaching & Examination Scheme
M.Tech.: Software Engineering
1st Year –II Semester

S N	Category	Course		Contact hrs/week			Marks				Cr
		Code	Title	L	T	P	Exm Hrs	IA	ETE	Total	
THEORY											
1	PCC	2MSW1-01	Service-Oriented Architecture	3	0	0	3	30	70	100	3
2	PCC	2MSW1-02	Software Quality & Assurance	3	0	0	3	30	70	100	3
3	PCC	2MSW1-03	Secure Software Design and Enterprise Computing	3	0	0	3	30	70	100	3
4	PEC		**Elective-II:**	3	0	0	3	30	70	100	3
		2MSW2-11	a. Agile Software Development								
		2MSW2-12	b. Software Reliability								
		2MSW2-13	*c.* Graph Theory and Network Algorithms								
5	MCC	2MCC3-XX	Audit Course-I	2	0	0					
		Sub Total						**120**	**280**	**400**	**12**
PRACTICAL & SESSIONAL											
6	PCC	2MSW1-06	Service-Oriented Architecture Lab	0	0	4	4	60	40	100	2
7	PCC	2MSW1-07	Software Architecture Lab	0	0	4	4	60	40	100	2
8	REW	2MSW4-50	Mini Project with Seminar	0	0	4	4	60	40	100	2
9	SODECA	2MSW5-00	Social Outreach, Discipline & Extracurricular Activities							100	2
		Sub- Total						**180**	**120**	**400**	**8**
		TOTAL OF II SEMESTER						**300**	**400**	**800**	**20**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*

Office of Dean Academic Affairs
Rajasthan Technical University, Kota

Teaching & Examination Scheme
M.Tech.: Software Engineering
2nd Year –III Semester

THEORY											
SN	Category	Course		Contact hrs/week			Marks				Cr
		Code	Title	L	T	P	Exm Hrs	IA	ETE	Total	
1	PEC		**Elective-III:**	3	0	0	3	30	70	100	3
		3MSW2-11	a. Security Analysis of Software Systems								
		3MSW2-12	b. Dependable Software								
		3MSW2-13	c. Data Visualisation								
2	MCC	3MCC3-XX	Open Elective	3	0	0	3	30	70	100	3
3	MCC	3MCC3-XX	Audit Course-II	2	0	0					
								60	**140**	**200**	**6**
PRACTICAL & SESSIONAL											
4	PCC	3MSW4-60	Dissertation-I / Industrial Project	0	0	20		240	160	400	10
			Sub- Total					**240**	**160**	**400**	**10**
			TOTAL OF III SEMESTER					**300**	**300**	**600**	**16**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*

Office of Dean Academic Affairs
Rajasthan Technical University, Kota

Teaching & Examination Scheme
M.Tech.: Software Engineering
2nd Year –IV Semester

PRACTICAL & SESSIONAL											
SN	Category	Course		Contact hrs/week			Marks				Cr
		Code	Title	L	T	P	Exm Hrs	IA	ETE	Total	
1	REW	3MSW4-70	Dissertation-II	0	0	32		360	240	600	16
		Sub- Total								**600**	**16**
		TOTAL OF IV SEMESTER						**360**	**240**	**600**	**16**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*

Office of Dean Academic Affairs
Rajasthan Technical University, Kota

1MSW1-01: Statistical Methods in Computer Science

Course Code	
Course Name	Statistical Methods in Computer Science
Credits	3
Pre-Requisites	Discrete Mathematics

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction:** Probability mass, density, and cumulative distribution functions, Parametric families of distributions, Expected value, variance, conditional expectation, Applications of the univariate and multivariate Central Limit Theorem, Probabilistic inequalities, Markov chains	6
Unit 2 **Sampling:** Random samples, sampling distributions of estimators**,** Methods of Moments and Maximum Likelihood**,**	4
Unit 3 **Introduction to Multivariate Statistical Models:** Statistical inference**,** Introduction to multivariate statistical models: regression and classification problems, principal components analysis**,** The problem of overfitting model assessment.	5
Unit 4 **Graph Theory:** Isomorphism, Planar graphs, graph colouring, Hamilton circuits and Euler cycles. Permutations and Combinations with and without repetition. Specialized techniques to solve combinatorial enumeration problems.	10
Unit 5 **Computer Science and Engineering Applications:** Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems, Bioinformatics, Machine learning.	10
Unit 6 **Recent Trends:** Recent Trends in various distribution functions in mathematical field of computerscience for varying fields like bio informatics, soft computing, and computervision.	5

References:

1. John Vince, Foundation Mathematics for Computer Science, Springer.
2. K. Trivedi Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley.
3. M. Mitzenmacher and E. Upfal.Probability and Computing: Randomized Algorithms and Probabilistic Analysis.
4. Alan Tucker, Applied Combinatorics, Wiley

1MSW1-02: Software Architecture

Course Code	
Course Name	Software Architecture
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction to Software Architecture:** The Architectural Concept, Architectural Patterns, Reference Models and Reference, Importance of Software Architecture, Architectural Structures and Views. **Architectural Styles:** Architectural Styles, Other Familiar Architectures.	5
Unit 2 **Software Architecture – Case Studies:** Key Word in Context, Instrumentation Software, Mobile Robotics, Cruise Control, Three Vignettes in Mixed Style, Real Time Applications and Distributed Applications. **Architectural Quality Attributes:** Functionality and Architecture, Architecture and Quality Attributes, System Quality Attributes, Quality Attributes Scenario in Practice, Other System Quality Attributes.	5
Unit 3 **Achieving Quality:** Introduction, Tactics, Relationship of Tactics to Architectural Patterns. **Architectural Patterns –1:** Architectural Pattern, From Mud to Structure- Layers, Pipes and filters, Blackboard. **Architectural Patterns –2:** Distributed Systems-Broker architecture, Interactive Systems- Model-View-Controller (MVC), Presentation-Abstraction-Control (PAC). **Architectural Patterns – 3:** Adaptable Systems- Microkernel, Reflection.	9
Unit 4 **Important Design Patterns:** Design Patterns, Structural Decomposition, Organization of Work, Access Control. **Architectural Design Guidance:** User Interface Architecture, The Quantified Design Space, Architectural Design Space Formalism.	5
Unit 5 **Formal Models and Specifications:** Z-Notation, Formalizing an Architectural Style, Formalizing an Architectural Design Space. **Linguistic Issues:** Architectural Description Language, First Class Connectors, Adding Implicit Invocation to Traditional Programming Languages.	6

Unit 6 **Tools for Software Architecture:** CASE Tools, Analysis and Design tools, Software Development Tools, Software Tools for Architecture Design, Excel as an Architecture Tool, Exploiting Style in Architectural Design, Quality-Driven Software Architecture Design. **Designing and Documenting Software Architecture:** Forming a Team Structure, Creating a Skeleton System, Uses of Architectural Documentation, Rules for Documentation, Views, Documenting a View.	10

References:

1. Len Bass, Paul Clements, and Rick Kazman, Software Architecture in Practice, 2nd ed, Addison-Wesley, 2003.
2. Eric Braude, Software Design: From Programming to Architecture, Wiley, 2004.

1MSW1-03: Software Metrics

Course Code	
Course Name	Software Metrics
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Basics of Measurement:** Scope of software metrics, representational theory of measurement, measurement and models, measurement scales, meaningfulness in measurement, goal-based framework for software, measurement.	6
Unit 2 **Software-metrics Data Collection and Analysis:** What is good data, how to define the data, how to collect the data, how to store and extract data, analyzing software-measurement data, frequency distributions, various statistical techniques	7
Unit 3 **Measuring internal product attributes:** Modelling software quality, measuring aspects of software quality, software reliability, basics of software reliability, software reliability problem, parametric reliability growth models, predictive accuracy, recalibration of software-reliability growth predictions, importance of operational environment, wider aspects of software reliability.	10
Unit 4 **Metrics for object-oriented systems:** The intent of object-oriented metrics, distinguishing characteristics of object-oriented metrics, various object-oriented metric suites – LK suite, CK suite and MOOD metrics.	8
Unit 5 **Metrics for component-based systems:** The intent of component-based metrics, distinguishing characteristics of component-based metrics, various component-based metrics.	5
Unit 6 **Resource measurement:** Measuring productivity, teams, tools, and methods.	4

References:

1. N.E. Fenton and S.L. Pfleeger; Software Metrics – A Rigorous and Practical Approach, Thomson Asia Pte., Ltd, Singapore.
2. S.H. Kan; Metrics and Models in Software Quality Engineering, Addison Wesley, New York.
3. K. H. Möller and D. J. Paulish; Software Metrics - A Practitioner's Guide to Improved Product Development, Chapman and Hall, London.
4. M. Lorenz and J. Kidd; Object-Oriented Software Metrics, Prentice Hall, New York.

1MSW2-11: Software Project Management

Course Code	
Course Name	Software Project Management
Credits	3
Pre-Requisites	Software engineering

COURSE CONTENT	CONTACT HOURS
Unit 1 **Basics of project management:** Introduction to S/W project management, S/W project management competencies, responsibilities of a software project manager.	6
Unit 2 **Software process:** S/W process models, project planning, organization of project team.	4
Unit 3 **S/W size estimation:** Estimation of effort & duration, Halstead's software Science, models, dependency & scheduling, staffing, Organizing a software engineering project.	7
Unit 4 **S/W configuration management:** S/W configuration management, monitoring & controlling S/W projects, developing requirements.	5
Unit 5 **Risk management:** Project tracking & control, communication & negotiating, S/W quality, S/W quality engineering, defining quality requirements, quality standards, practices & conventions.	8
Unit 6 **Software Quality Standard:** ISO 9000, ISO 9001, S/W quality matrices, managerial and organization issues, defect prevention, reviews & audits, SEI capability maturity model, PSP, six sigma. Special topics in process and quality management.	10

References:

1. B. Hughes, M. Cotterell, Software Project Management, McGraw Hill, 4th ed, 2005.
2. R. Walker, Software Project Management, Pearson, 2003.
3. R. H. Thayer, Software Engineering Project management, IEEE CS Press, 2nd Ed, 1988.
4. R. Pressman, Software Engineering A Practitioner's approach, McGraw Hill, 4th Ed, 2005.

1MSW2-12: Software Testing

Course Code	
Course Name	Software Testing
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction:** Introduction, Basic concepts, discrete mathematics for testers, Graph theory for testers.	5
Unit 2 **Black Box Testing:** Boundary value testing, Equivalence class testing, etc.	4
Unit 3 **White Box Testing:** Statement coverage, Branch coverage, condition coverage, path coverage, Mc Cabe's cyclomatic complexity; Decision Table based testing, Data flow based testing, Integration testing.	8
Unit 4 **System Testing:** Interaction testing, Performance testing, Mutation testing, Regression testingand error seeding.	5
Unit 5 **Object-Oriented Testing:** Issues in object oriented testing, Test case design by object oriented software, Fault based testing, test cases and class hierarchy, Scenario based Test design, Testing surface structure and deep structure.	8
Unit 6 **Class Testing:** Random testing for object oriented classes, Partition testing at the class level; Inter class test case design: multiple class testing, tests derived from behaviour models, Test case generation using UML diagrams, GUI testing, object oriented system testing. Special topics in software testing.	10

References:

1. C. J. Paul, Software testing: A craftsmen's approach, CRC Press, 2nd Ed, 2002.
2. R. Gopalswamy, Software testing, Pearson, 2005.
3. G. J. Myers, The art of software testing, Wiley Interscience New York, 2005.
4. R. S. Pressman, Software Engineering A Practitioner's approach, McGraw Hill, 4th Ed, 1982.
5. R. Mall, Fundamentals of Software Engineering, Prentice Hall of India, 2nd Ed, 2003.

1MSW2-13: Requirements Engineering

Course Code	
Course Name	Requirements Engineering
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Basics of Requirements Engineering:** Definition of requirements engineering, importance of requirements engineering, place of, requirements engineering in development process, types of requirements: functional requirements, non-functional requirements, quality attributes, main requirements engineering activities, documents and processes.	7
Unit 2 **Requirements Inception and Elicitation:** Product vision and project scope, traditional elicitation approaches (interviews, stakeholders study, workshops, ...), scenario/use case approaches, prototyping, requirements negotiation and risk management.	5
Unit 3 **Requirements Analysis and Specification - Modelling Techniques:** inception vs. specification, techniques for writing high-quality requirements, documentation standards (e.g., IEEE 830-1998), goal-oriented modelling, Structured analysis and other techniques, UML v2 and URN notations, external qualities management, contract specification.	8
Unit 4 **Requirements Verification, and Validation:** Detection of conflicts and inconsistencies, completeness, techniques for inspection, verification and validation, feature interaction analysis and resolution.	7
Unit 5 **Requirements Management:** Traceability, priorities, changes, baselines, tool support (e.g., DOORS).	3
Unit 6 **Examples of Requirements Approaches in Typical Development Processes:** Requirements for various types of systems: embedded systems, consumer systems, web-based, systems, business systems, systems for scientists and other engineers, requirements engineering in RUP, requirements engineering in agile methods.	10

References:

1. Ian K. Bray, An Introduction to Requirements Engineering, Addison Wesley, 2002.
2. Ian F. Alexander, Richard Stevens, Writing better requirements, Addison-Wesley, 2002 - for the topic of how to write requirements.
3. Karl E. Wiegers, Software Requirements, Microsoft Press, 2003.
4. Axel van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley, 2009.
5. Ian Alexander and Ljerka Beus-Dukic, Discovering Requirements: How to Specify Products and Services, Wiley, 2010.

1MSW1-06: Software Design Lab

1. **Software Requirement Specification:**
   Prepare the IEEE format SRS document of given case studies. You should identify the appropriate requirements for each problem; Draw the Use Case diagrams, Domain Models, Class Diagrams, and all behavioural diagrams using Modelling tool (Rational Rose, Magic Draw, IBM RSA, MS Visio Software, StarUML etc.).
1. Develop a software for
   a. Automation of the dispensary of university.
   b. Activities of the Estate Office of the university.
   c. Word processing software with limited number of facilities
   d. Graphics editor software package.
   e. Library Information System of University.
2. Write a program that calculate the cyclomatic complexity and generate independent test scenarios with CFG graph of module.
3. Write a program that calculate cyclomatic complexity and find out DD paths using Data flow Graph.
4. Perform Estimation of effort using FP Estimation for chosen system.
5. To Prepare time line chart/Gantt Chart/PERT Chart for selected software project.
6. Understanding of Halstead software science metrics like , token count, program length, Vocabulary of the program, Volume, estimated program length of the program, potential volume V* etc. Calculation of Halstead software science metrics.

1MSW1-06: Software Testing Lab

1. Write the test cases and compare with expected results using Junit and Elaborate JaButi tool using standard case study.
2. Generate test scenarios using selenium tools of given web application.
3. Calculate mutation score using Jumble testing tool of standard case study.
4. Calculate performance testing using Jmeter testing tool of standard web application.
5. Develop a translator for generating automatic test scenarios using UML diagrams.
6. Generate and execute test scenarios of android application using selendroid testing tool.

To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.

2MSW1-01: Service-Oriented Architecture

Course Code	
Course Name	Service-Oriented Architecture
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction of SOA, Evolution of SOA:** Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits of SOA;An SOA timeline (from XML to Web services to SOA); The continuing evolution of SOA (Standards organizations and Contributing vendors); The roots of SOA (comparing SOA to Past architectures).	6
Unit 2 **Web Services and Primitive SOA:** The Web services framework; Services (as Web services); Service descriptions (with WSDL); Messaging (with SOAP). Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography, Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing.	8
Unit 3 **Principles of Service – Orientation:** Services-orientation and the enterprise; Anatomy of a service-oriented architecture; Common Principles of Service- orientation; How service orientation principles inter-relate; Service- orientation and object-orientation; Native Web service support for service- orientation principles.	10
Unit 4 **Service Layers:** Service-orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios.	7
Unit 5 **Business Process Design:** WS-BPEL language basics; WS-Coordination overview; Service-oriented business process design; WS-addressing language basics; WS-Reliable Messaging language basics	5
Unit 6 **SOA Platforms:** SOA platform basics; SOA support in J2EE; SOA support in .NET; Integration considerations	4

References:

1. Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Pearson Education, 2005.
2. Dan Woods and Thomas Mattern, Enterprise SOA Designing IT for Business Innovation, First Editioned. O'REILLY, 2006.
3. Kai Qian, Xiang Fu, Lixin Tao, Chong-Wei Xu, and Jorge L. Diaz-Herrera, Software Architecture and Design Illuminated.: Jones and Bartlett, 2010.
4. Len Bass, Paul Clements, and Rick Kazman, Software Architecture in Practice, 2nd ed.: Pearson Education.
5. Newcomer and Lomow , Understanding SOA with Web Services.: Pearson Education, 2005.

2MSW1-02: Software Quality and Assurance

Course Code	
Course Name	Software Quality and Assurance
Credits	3
Pre-Requisites	Software engineering

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction to Software Quality and Assurance:** The software quality challenge, Software quality, Software quality factors Management and its role in software quality assurance.	6
Unit 2 **Components of SQA:** The components of the software quality assurance system – overview Pre-project Software Quality Components Contract review, Development and quality plans.	8
Unit 3 **SQA Components in the Project Life Cycle and Strategies:** Integrating quality activities in the project life cycle, Reviews, Software testing – strategies.	7
Unit 4 **Software Testing – Implementation:** Software Quality Implementation, Assuring the quality of software maintenance components, Assuring the quality of external participants' contributions, CASE tools and their effect on software quality.	10
Unit 5 **Software Quality Infrastructure Components:** Procedures and work instructions, Staff training and certification,Corrective and preventive actions, Documentation control.	6
Unit 6 **Software Quality Metrics:** Software Quality metrics, Cost of Quality.	3

References:

1. Daniel Galin, "Software Quality Assurance", Pearson Publication, 2009.
2. R. Walker, Software Project Management, Pearson, 2003.
3. KshirsagarNaik and PriyadarshiTripathy, Software Testing & Quality AssuranceTheory and Practice, Wiley Student edition.
4. William E. Perry, Effective Methods for Software Testing, WILLEY, . 3rd Edition.
5. Alan C. Gillies, "Software Quality: Theory and Management", International Thomson Computer Press, 1997.
6. M G Limaye, Software Testing, Tata McGraw-Hill Education, 2009.

2MSW1-03: Secure Software Design and Enterprise Computing

Course Code	
Course Name	Secure Software Design and Enterprise Computing
Credits	3
Pre-Requisites	Computer Programming, Software Engineering

COURSE CONTENT	CONTACT HOURS
Unit 1 **Secure Software Design :** Identify software vulnerabilities and perform software security analysis, Master security programming practices, Master fundamental software security design concepts, Perform security testing and quality assurance.	6
Unit 2 **Enterprise Application Development :** Describe the nature and scope of enterprise software applications, Design distributed N-tier software application, Research technologies available for the presentation, business and data tiers of an enterprise software application, Design and build a database using an enterprise database system, Develop components at the different tiers in an enterprise system, Design and develop a multi-tier solution to a problem using technologies used in enterprise system, Present software solution.	10
Unit 3 **Enterprise Systems Administration :** Design, implement and maintain a directory-based server infrastructure in a heterogeneous systems environment, Monitor server resource utilization for system reliability and availability, Install and administer network services (DNS/DHCP/Terminal Services/Clustering/Web/Email).	7
Unit 4 **Manage and Troubleshoot a Network :** Obtain the ability to manage and troubleshoot a network running multiple services, Understand the requirements of an enterprise network and how to go about managing them.	7
Unit 5 **Insecure Exception and Mommand/SQL Injections:** Handle insecure exceptions and command/SQL injection, Defend web and mobile applications against attackers, software containing minimum vulnerabilities and flaws.	6
Unit 6: **Case Study** Case study of DNS server, DHCP configuration and SQL injection attack.	4

References:

1. Theodor Richardson, Charles N Thies, Secure Software Design, Jones & Bartlett.
2. Kenneth R. van Wyk, Mark G. Graff, Dan S. Peters, Diana L. Burley, Enterprise SoftwareSecurity, Addison Wesley.

2MSW2-11: Agile Software Development

Course Code	
Course Name	Agile Software Development
Credits	3
Pre-Requisites	Software Engineering

COURSE CONTENT	CONTACT HOURS
Unit 1 **Agile Programming:** Introduction, Flavours of Agile Development, Agile Manifesto, Refactoring Techniques, Limitations of The Agile Process.	6
Unit 2 **Extreme Programming (XP):** Introduction, XP Equation, XP Values, Assuming Sufficiency- Sufficient time and resources, Constant change of cost, Developer effectiveness, Freedom to experiment. Extreme Programming Practices- Introduction, Coding Practices, Developer Practices, Business Practices.	10
Unit 3 **XP Events:** Introduction, Iteration Planning- Stories and tasks, Estimates and schedules, First iteration, Iteration, Releasing. Extreme Programming Practices- Introduction, Story Cards, Task Cards, Bullpens.	6
Unit 4 **Roles in Extreme Programming:** Introduction, Customer's Roles, Developer's Roles, Supplementary Roles. Coding XP Style- Introduction, Balance Functionality with Simplicity, Implement only the needed Features, Eliminate Repetition. Adopting XP- Introduction, Before Commencing XP, Eliminating Fear and Working Together, Starting Feedback, Including Managers and Customers.	10
Unit 5 **Agile Modelling with XP- Introduction, Agile Modelling:** Principles, Comparing XP and Agile Modelling, Scrum Methodology- The roles of Scrum, Advantages of Scrum. Dynamic Systems Development Methodology- Introduction, Overview of DSDM, the Principles of DSDM, Phases of DSDM, Core Techniques Used in DSDM.	8

References:

1. Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, Prentice Hall.
2. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson.
3. R. S. Pressman, Software Engineering A Practitioner's Approach, McGraw Hill Publications, 2006.
4. Lisa Crispin, Janet Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison Wesley.
5. Alistair Cockburn, Agile Software Development: The Cooperative Game, Addison Wesley.
6. Mike Cohn, User Stories Applied: For Agile Software, Addison Wesley.

2MSW2-12: Software Reliability

Course Code	
Course Name	Software Reliability
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction:** Basic Ideas of Software Reliability, Hardware reliability vs. Software reliability, Reliability metrics, Failure and Faults – Prevention, Removal, Tolerance, Forecast, Dependability Concept – Failure Behaviour, Characteristics, Maintenance Policy, Reliability and Availability Modelling, Reliability Evaluation Testing methods, Limits, Starvation, Coverage, Filtering, Microscopic Model of Software Risk.	9
Unit 2 **Computation of Software Reliability:** Functional and Operational Profile, Operational Profiles – Difficulties, Customer Type, User Type, System Mode, Test Selection - Selecting Operations, Regression Test.	6
Unit 3 **Software Reliability Models:** Classes of software reliability Models, Time Dependent Software Reliability Models: Time between failure reliability Models, Fault Counting Reliability Models. Time Independent Software Reliability Models: Fault injection model of Software Reliability, Input Domain Reliability Model, Orthogonal defect classification, Software availability Models. Software Reliability Modelling: A general procedure for reliability modelling.	9
Unit 4 **Short and Long Term Prediction:** Model Accuracy, Analysing Predictive Accuracy – Outcomes, PLR, U and Y Plot, Errors and Inaccuracy.	6
Unit 5 **Recalibration:** Detecting Bias, Different Techniques, Power of Recalibration, Limitations in Present Techniques, Improvements.	4
Unit 6 **Class Testing:** Random testing for object oriented classes, Partition testing at the class level; Inter class test case design: multiple class testing, tests derived from behaviour models, Test case generation using UML diagrams, GUI testing, object oriented system testing. Special topics in software testing.	6

References:

1. J.D. Musa, Software Reliability Engineering, McGraw Hill, New York , 2004.
2. H. Pham, Software Reliability, Springer Verlag, New York , 2000.
3. Patric D. T.O Connor, Practical Reliability Engineering, 4th Edition, John Wesley & Sons , 2003.
4. D. Reled, Software Reliability Methods, Springer Verlag, New York , 2001.

2MSW2-13: Graph Theory and Network Algorithms

Course Code	
Course Name	Graph Theory and Network Algorithms
Credits	3
Pre-Requisites	

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction:** Graphs, Isomorphism, Walks, Paths, Circuits, Trees, Properties of Trees, Co-trees and Fundamental Circuits, Cut Sets.	4
Unit 2 **Fundamental Cut Sets and Cut Vertices:** Planar and Dual Graphs, Metric Representation of Graphs, Coloring and covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, matching, covering, four color problem, Directed graphs, some type of directed graphs, Directed paths, and connectedness, Euler digraphs, trees with directed edges, fundamental circuits in digraph, matrices A, B and C of digraphs adjacency matrix of a digraph,, enumeration, types of enumeration, counting of labelled and unlabelled trees, polya's theorem, graph enumeration with polya's theorem.	12
Unit 3 **Graph Algorithms:** Elementary Graph Algorithms, Representations of graphs, Breadth-first search, Depth-first search, Topological sort, strongly connected components.	4
Unit 4 **Minimum Spanning Trees:** Growing a minimum spanning tree, The algorithms of Kruskal and Prim, Single-Source Shortest Paths: Shortest paths and relaxation, Dijkstra's algorithm, The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Difference constraints and shortest paths, All-Pairs.	8
Unit 5 **Shortest Paths:** Shortest paths and matrix multiplication, The Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs, and A general framework for solving path problems in directed graphs.	5
Unit 6 **Maximum Flow:** Flow networks, The Ford-Fulkerson method, Maximum bipartite matching, Preflow-push algorithms, The lift-to-front algorithm. Special topics in graph theory and network algorithms.	7

References:

1. T. H. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to Algorithms, Prentice Hall of India, 3rd ed,2006.
2. N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall of India, 2004.
3. D. B. West, Introduction to Graph Theory, 2nd Ed, Prentice Hall of India, 2007.
4. R. Diestel, Advanced Graph Theory, Springer Verlag Heidelberg, New York, 2005.
5. M. T. Goodrich and R. Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Wiley,1st ed, 2001.

2MSW1-06: Service-Oriented Architecture Lab

1. Uses and implementation of SOAP-web services and REST-web services. Using Simple Object Access Protocol – User Interface (SOAP-UI) tool, Description of the SOAPUI tool (using SOAP, REST, Web services, WSDL, client server architecture, etc.)
2. Creating BPEL project using Open-ESB tool, Create WSDL document for services such as add and square of numbers. Create services and implementing them on server.
3. Make web services for Quadratic Equation (using Square, multiply, subtraction, square-root, calculating root etc.), Linear Simultaneous equation (with one dimensional array, two dimensional arrays, Cramer's rule). Database connectivity in OpenESB. Use HeildiSQL tool to create database, Add MySql and java connector.
4. Create account on Bluemix, create services, add with database and deployment of projects on cloud by using IBM Bluemix Server.

2MSW1-07: Software Architecture Lab

1. Develop object-oriented applications based on various UML diagrams using Rational Rose tools on case study i.e., Bank consortium by Rambaugh build its Use case diagram Activity diagram, State chart diagram, Sequence diagram, Collaboration diagram, class diagram, component diagram, deployment diagram etc.
2. Implement of various Architectures Styles (Peer to Peer, Client Server, Blackboard, mobile code, Pipe and Filter, Publish-subscribe, event based) using programming language (Java, C++ etc.).
3. Design and Implement ATM System using Rapide Koala, Darwin and ACME studio ArchStudio (Archipelago, ArchEdit) Tools in ADL (Architecture Description Languages).
4. Design and Vaildate ATM System Using Formal Specifications in Z notation(Alloy, Fastest, Petrinet tools etc.).

3MSW2-11: Security Analysis of Software System

Course Code	
Course Name	Security Analysis of Software System
Credits	3
Pre-Requisites	Risk analysis, Software Engineering

COURSE CONTENT	CONTACT HOURS
Unit 1 **Introduction:** Security protocols, Security properties, Public-key certificates and infrastructures, Cryptographic hash functions, Digital signatures, Security protocol vulnerabilities	8
Unit 2 **Security Protocols:** Needham- Schroeder public-key protocol and its security analysis, Protocols for anonymity, Anonymity and MIX networks, Fairness and contract signing, Fair exchange and contract signing protocols, Game-based verification of contract signing protocols. Yahalom protocol: Secrecy, Authentication, Non-repudiation, Anonymity; Dolev-Yao threat model.	10
Unit 3 **Analysis Tools:** Protocol analysis tools	3
Unit 4 **Content Security Policy:** Basic building blocks, Parallel operators, Process behaviour, Modelling security protocols in CSP - Trustworthy processes, Modelling an intruder, protocol goals.	5
Unit 5 **Transformations:** Transformations on protocols, Safe simplifying transformations, Structural transformations. Formal analysis: Formal definitions of security for symmetric ciphers, Formal model for secure key exchange. Theorem proving - Rank functions, Secrecy of shared key, Authentication.	9
Unit 6 **Case Study:** Case study of DNS server, DHCP configuration and SQL injection attack.	5

References:

1. Theodor Richardson, Charles N Thies, Secure Software Design, Jones & Bartlett.
2. Peter Ryan, Steve Schneider, Michael Goldsmith, Gavin Lowe, Bill Roscoe: Modelling & Analysis of Security Protocols, Addison Wesley.
3. Stephen W. Mancini: Automating Security Protocol Analysis, Storming Media.

3MSW2-12: Dependable Software

Course Code	
Course Name	Dependable Software
Credits	3
Pre-Requisites	Fault Tolerance, Software Engineering

COURSE CONTENT	**CONTACT HOURS**
Unit 1 **Dependability Concepts:** Dependable system, techniques for achieving dependability, dependability measures, fault, error, failure, faults and their manifestation, classification of faults and failures. **Fault Tolerant Strategies:** Fault detection, masking, containment, location, reconfiguration, and forward recovery, backward recovery.	6
Unit 2 **Fault Tolerant Design Techniques:** Hardware redundancy, software redundancy, time redundancy, and information redundancy. **Testing and Design for Testability:** Self-checking and fail-safe circuits.	5
Unit 3 **Types of Redundancy for Software Fault Tolerance:** Information or Data Redundancy , Temporal Redundancy, coding techniques, error detection and correction codes, burst error detection and correction, unidirectional codes. **Fault Tolerance in Distributed Systems:** Byzantine General problem, consensus protocols, check pointing and recovery, stable storage and RAID architectures, and data replication and resiliency.	8
Unit 4 **Structuring Redundancy for Software Fault Tolerance:** Robust Software, Design Diversity, Case Studies and Experiments in Design Diversity, Levels of Diversity and Fault Tolerance Application, Data Diversity, Structure for Development of Diverse Software.	5
Unit 5 **Design Diverse Software Fault Tolerance Techniques:** Recovery Blocks, Recovery Block Operation, Recovery Block Example, Recovery Block Issues and Discussion, N-Version Programming, N-Version Programming Operation, N-Version Programming Example, N-Version Programming Issues and Discussion, Distributed Recovery Blocks, Distributed Recovery Block Operation, Distributed Recovery Block Example, N Self-Checking Programming, N Self-Checking Programming Operation, N Self-Checking Programming Example, Consensus Recovery Block, Consensus Recovery Block Operation, Consensus Recovery Block Example.	11
Unit 6 **Dependable Communication:** Dependable channels, survivable networks, fault-tolerant routing. Case studies of fault tolerant multiprocessor and distributed systems.	5

References:

1. Laura L. Pullum, software Fault Tolerance Techniques and implementations. British Library Cataloguing in Publication Data Pullum, Laura , ISBN 1-58053-470-8, 2001.

3MSW2-13: Data Visualisation

Course Code	
Course Name	Data Visualisation
Credits	3
Pre-Requisites	Computer Graphics, Image Processing

LECTURE WITH BREAKUP	CONTACT HOURS
Unit 1 **Introduction:** Introduction of visual perception, visual representation of data, Gestalt principles, information overloads.	5
Unit 2 **Data Visualization Models:** Creating visual representations, visualization reference model, visual mapping, visual analytics, Design of visualization applications.	5
Unit 3 **Data Visualization Classification:** Classification of visualization systems, Interaction and visualization techniques misleading, Visualization of one, two and multi-dimensional data, text and text documents.	8
Unit 4 **Data Visualization Techniques:** Visualization of groups, trees, graphs, clusters, networks, software, Metaphorical visualization	7
Unit 5 **Data Visualization Applications:** Visualization of volumetric data, vector fields, processes and simulations, Visualization of maps, geographic information, GIS systems, collaborative visualizations, Evaluating visualizations	10
Unit 6 **Recent Trends:** Various perception techniques, various visualization techniques, data structures used in data visualization.	5

References:

1. WARD, GRINSTEIN, KEIM, Interactive Data Visualization: Foundations, Techniques, and Applications. Natick : A K Peters, Ltd.
2. E. Tufte, The Visual Display of Quantitative Information, Graphics Press.