VELAGAPUDI RAMAKRISHNA

SIDDHARTHA ENGINEERING COLLEGE::VIJAYAWADA

(AUTONOMOUS)

DEPARTMENT OF IT

MICRO LEVEL SYLLABUS

Class	**B.Tech**	**Regulation**	**VR20**
Subject Code	**20IT6205 A**	**Year & Semester**	**III/IV , 6th SEMESTER**
Title of the Subject	**Agile Software Development**		

Unit No	**Content/Topics Covered (mention Sub Topics as found in books)**	**Text Book**	**Chapter/ Section No.**	**Page Number**
Unit I	Introduction To Agile Software Development - Overview	[T1]	1.1	1
	Three Perspectives on Software Engineering	[T1]	1.4	3
	The Agile Manifesto	[T1]	1.5	4
	Individuals and Interactions over Processes and Tools	[T1]	1.5.1	5
	Working Software over Comprehensive Documentation	[T1]	1.5.2	6
	Customer Collaboration over Contract Negotiation	[T1]	1.5.3	7
	Responding to Change over Following a Plan	[T1]	1.5.4	7
	Definition of Scrum	[T2]	-	3
	Uses of Scrum	[T2]	-	4
	Scrum Theory- Transparency, Inspection, Adaptation	[T2]	-	4
	Scrum Values	[T2]	-	5
	The Scrum Team- The Product Owner, The Development Team, The Scrum Master	[T2]	-	6
	Scrum Events- The Sprint, Sprint Planning, Daily Scrum, Daily Scrum, Sprint Retrospective	[T2]	-	9
	Scrum Artifacts- Product Backlog, Sprint Backlog, Increment	[T2]	-	14
	Introduction to Kanban: Agile Software Development approach-Kanban	https://kruschecompany.com/kanban-method-agile-software-development/		
	Teamwork-Overview	[T1]	2.1	25
	Objectives	[T1]	2.2	26
	A Role Scheme in Agile Teams	[T1]	2.4	27
	Remarks on the Implementation of the Role Scheme	[T1]	2.4.1	31
	Human Perspective on the Role Scheme	[T1]	2.4.2	32
	Using the Role Scheme to Scale Agile Projects	[T1]	2.4.3	34
	Dilemmas in Teamwork	[T1]	2.5	34
	Teamwork in Learning Environments	[T1]	2.6	36
	Teaching and Learning Principles	[T1]	2.6.1	36
	Role Activities	[T1]	2.6.2	37
	Student Evaluation	[T1]	2.6.3	40
Unit II	**Agile estimation techniques**: T-Shirt Sizing, Sprint Poker, Three-Point Method, Affinity Estimation, Relative Mass Evaluation, Dot voting, Maximum allowable size (MAS), Big, Uncertain, Small.	https://www.zstream.io/blog/top-7-most-popular-agile-estimation-techniques		
	Customers And Users-Overview	[T1]	3.1	45
	Objectives	[T1]	3.2	47
	The Customer	[T1]	3.4	48

	Customer Role	[T1]	3.4.1	48
	Customer Collaboration	[T1]	3.4.2	54
	The User	[T1]	3.5	55
	Combining UCD with Agile Development	[T1]	3.5.1	57
	Customers and Users in Learning Environments	[T1]	3.6	61
	Teaching and Learning Principles	[T1]	3.6.1	61
	Customer Stories	[T1]	3.6.2	62
	Case Studies of Metaphor Use	[T1]	3.6.3	62
	Time-Overview	[T1]	4.1	71
	Objectives	[T1]	4.2	72
	Time-Related Problems in Software Projects	[T1]	4.4	73
	List of Time-Related Problems of Software Projects	[T1]	4.4.1	74
	Case Study 4.1. Software Organizational Survey from the Time Perspective	[T1]	4.4.2	75
	Tightness of Software Development Methods	[T1]	4.5	77
	Sustainable Pace	[T1]	4.6	79
	Case Study 4.2. An Iteration Timetable of an Agile Team	[T1]	4.6.1	80
	Time Management of Agile Projects	[T1]	4.7	81
	Time Measurements	[T1]	4.7.1	81
	Prioritizing Development Tasks	[T1]	4.7.2	83
	Time in Learning Environments	[T1]	4.8	86
	The Planning Activity	[T1]	4.8.1	86
	Teaching and Learning Principles	[T1]	4.8.2	88
	Students’ Reflections on Time-Related Issues	[T1]	4.8.3	89
	The Academic Coach’s Perspective	[T1]	4.8.4	89
Unit III	MEASURES-Overview	[T1]	5.1	93
	Objectives	[T1]	5.2	95
	Why Are Measures Needed	[T1]	5.4	95
	Who Decides What Is Measured?	[T1]	5.5	96
	What Should Be Measured?	[T1]	5.6	97
	When Are Measures Taken?	[T1]	5.7	98
	How Are Measures Taken?	[T1]	5.8	98
	Who Takes the Measures?	[T1]	5.9	99
	How Are Measures Used?	[T1]	5.10	99
	Case Study- Monitoring a Large-Scale Project by Measures	[T1]	5.11	100
	Measure Definition	[T1]	5.11.1	100
	Measure Illustration	[T1]	5.11.2	102
	Measures in Learning Environments	[T1]	5.12	108
	Teaching and Learning Principles	[T1]	5.12.1	108
	Measurement Activities	[T1]	5.12.2	109
	Case Study 5.2. Role-Related Measures	[T1]	5.12.3	111
	Quality-Overview	[T1]	6.1	115
	Objectives	[T1]	6.2	116
	The Agile Approach to Quality Assurance	[T1]	6.4	117
	Process Quality	[T1]	6.4.1	119
	Product Quality	[T1]	6.4.2	120
	Test-Driven Development	[T1]	6.5	121
	How Does TDD Help Overcome Some of the Problems Inherent in Testing?	[T1]	6.5.1	122
	Case Study 6.1. TDD Steps	[T1]	6.5.2	124
	Case Study 6.2. Reflection on TDD	[T1]	6.5.3	125

	Measured TDD	[T1]	6.6	127
	Quality in Learning Environments	[T1]	6.7	128
	Case Study 6.3. Size and Complexity Measures	[T1]	6.7.1	128
	Case Study 6.4. Illustrating Measured TDD	[T1]	6.7.2	130
	Teaching and Learning Principles-The Case of Quality	[T1]	6.7.3	136
Unit IV	Learning-Overview	[T1]	7.1	139
	Objectives	[T1]	7.2	140
	How Does Agile Software Development Support Learning Processes	[T1]	7.4	141
	Agile Software Development from the Constructivist Perspective	[T1]	7.4.1	141
	The Role of Short Releases and Iterations in Learning Processes	[T1]	7.4.2	142
	Learning in Learning Environments	[T1]	7.5	144
	Gradual Learning Process of Agile Software Engineering	[T1]	7.5.1	145
	Learning and Teaching	[T1]	7.5.2	146
	The Studio Meeting-End of the First Iteration	[T1]	7.5.3	147
	Intermediate Course Review and Reflection	[T1]	7.5.4	147
	Abstraction-Overview	[T1]	8.1	155
	Objectives	[T1]	8.2	156
	Abstraction Levels in Agile Software Development	[T1]	8.4	158
	Roles in Agile Teams	[T1]	8.4.1	158
	Case Study 8.1. Abstraction During Iteration Planning	[T1]	8.4.2	159
	The Stand-Up Meeting	[T1]	8.4.3	161
	Design and Refactoring	[T1]	8.4.4	162
	Abstraction in Learning Environments	[T1]	8.5	164
	Teaching and Learning Principles	[T1]	8.5.1	165
	Case Study 8.2. Refactoring Activity	[T1]	8.5.2	166
	Trust-Overview	[T1]	9.1	171
	Objectives	[T1]	9.2	172
	Software Intangibility and Process Transparency	[T1]	9.4	173
	Game Theory Perspective in Software Development	[T1]	9.5	175
	Ethics in Agile Teams	[T1]	9.6	179
	Diversity	[T1]	9.7	183
	Trust in Learning Environments	[T1]	9.8	186
	Teaching and Learning Principle	[T1]	9.8.1	186