

Contents

1. Introduction to Agile Software Development	1
1.1 Overview	1
1.2 Objectives	2
1.3 Study Questions	2
1.4 Three Perspectives on Software Engineering	3
1.5 The Agile Manifesto	4
1.5.1 Individuals and Interactions over Processes and Tools . .	5
1.5.2 Working Software over Comprehensive Documentation	6
1.5.3 Customer Collaboration over Contract Negotiation . . .	7
1.5.4 Responding to Change over Following a Plan	7
1.6 Application of Agile Software Development	8
1.7 Data About Agile Software Development	13
1.8 Agile Software Development in Learning Environments	15
1.8.1 University Course Structure	15
1.8.2 Teaching and Learning Principles	15
1.8.3 The Studio Environment	17
1.8.4 The Academic Coach Role	18
1.8.5 Overview of the Studio Meetings	19
1.8.6 Launching the Project Development in the Studio	20
1.9 Summary and Reflective Questions	23
1.10 Summary	24
References	24
2. Teamwork	25
2.1 Overview	25
2.2 Objectives	26
2.3 Study Questions	26
2.4 A Role Scheme in Agile Teams	27
2.4.1 Remarks on the Implementation of the Role Scheme . . .	31
2.4.2 Human Perspective on the Role Scheme	32
2.4.3 Using the Role Scheme to Scale Agile Projects	34
2.5 Dilemmas in Teamwork	34
2.6 Teamwork in Learning Environments	36

2.6.1	Teaching and Learning Principles	36
2.6.2	Role Activities	37
2.6.3	Student Evaluation	40
2.7	Concluding Reflective Questions	42
2.8	Summary	42
	References	42
3.	Customers and Users	45
3.1	Overview	45
3.2	Objectives	47
3.3	Study Questions	47
3.4	The Customer	48
3.4.1	Customer Role	48
3.4.2	Customer Collaboration	54
3.5	The User	55
3.5.1	Combining UCD with Agile Development	57
3.6	Customers and Users in Learning Environments	61
3.6.1	Teaching and Learning Principles	61
3.6.2	Customer Stories	62
3.6.3	Case Studies of Metaphor Use	62
3.7	Summary and Reflective Questions	67
3.8	Summary	68
	References	68
4.	Time	71
4.1	Overview	71
4.2	Objectives	72
4.3	Study Questions	72
4.4	Time-Related Problems in Software Projects	73
4.4.1	List of Time-Related Problems of Software Projects . . .	74
4.4.2	Case Study 4.1. Software Organizational Survey from the Time Perspective	75
4.5	Tightness of Software Development Methods	77
4.6	Sustainable Pace	79
4.6.1	Case Study 4.2. An Iteration Timetable of an Agile Team	80
4.7	Time Management of Agile Projects	81
4.7.1	Time Measurements	81
4.7.2	Prioritizing Development Tasks	83
4.8	Time in Learning Environments	86
4.8.1	The Planning Activity	86
4.8.2	Teaching and Learning Principles	88
4.8.3	Students' Reflections on Time-Related Issues	89
4.8.4	The Academic Coach's Perspective	89

4.9	Summary and Reflective Questions	90
4.10	Summary.	91
	References	91
5.	Measures	93
5.1	Overview.	93
5.2	Objectives	95
5.3	Study Questions	95
5.4	Why Are Measures Needed?	95
5.5	Who Decides What Is Measured?	96
5.6	What Should Be Measured?	97
5.7	When Are Measures Taken?	98
5.8	How Are Measures Taken?	98
5.9	Who Takes the Measures?	99
5.10	How Are Measures Used?	99
5.11	Case Study 5.1. Monitoring a Large-Scale Project by Measures	100
	5.11.1 Measure Definition	100
	5.11.2 Measure Illustration.	102
5.12	Measures in Learning Environments	108
	5.12.1 Teaching and Learning Principles.	108
	5.12.2 Measurement Activities.	109
	5.12.3 Case Study 5.2. Role-Related Measures	111
5.13	Summary and Reflective Questions	114
5.14	Summary.	114
	References	114
6.	Quality	115
6.1	Overview	115
6.2	Objectives	116
6.3	Study Questions	117
6.4	The Agile Approach to Quality Assurance.	117
	6.4.1 Process Quality	119
	6.4.2 Product Quality.	120
6.5	Test-Driven Development.	121
	6.5.1 How Does TDD Help Overcome Some of the Problems Inherent in Testing?	122
	6.5.2 Case Study 6.1. TDD Steps.	124
	6.5.3 Case Study 6.2. Reflection on TDD	125
6.6	Measured TDD	127
6.7	Quality in Learning Environments.	128
	6.7.1 Case Study 6.3. Size and Complexity Measures	128

6.7.2	Case Study 6.4. Illustrating Measured TDD.	130
6.7.3	Teaching and Learning Principles—The Case of Quality.	136
6.8	Summary and Reflective Questions	137
6.9	Summary.	137
	References	138
7.	Learning.	139
7.1	Overview.	139
7.2	Objectives	140
7.3	Study Questions	140
7.4	How Does Agile Software Development Support Learning Processes?	141
7.4.1	Agile Software Development from the Constructivist Perspective.	141
7.4.2	The Role of Short Releases and Iterations in Learning Processes	142
7.5	Learning in Learning Environments.	144
7.5.1	Gradual Learning Process of Agile Software Engineering	145
7.5.2	Learning and Teaching Principle	146
7.5.3	The Studio Meeting—End of the First Iteration	147
7.5.4	Intermediate Course Review and Reflection	147
7.6	Summary and Reflective Questions	152
7.7	Summary.	152
	References	152
8.	Abstraction	155
8.1	Overview	155
8.2	Objectives	156
8.3	Study Questions	157
8.4	Abstraction Levels in Agile Software Development	158
8.4.1	Roles in Agile Teams	158
8.4.2	Case Study 8.1. Abstraction During Iteration Planning	159
8.4.3	The Stand-Up Meeting	161
8.4.4	Design and Refactoring	162
8.5	Abstraction in Learning Environments	164
8.5.1	Teaching and Learning Principles.	165
8.5.2	Case Study 8.2. RefactoringActivity.	166
8.6	Summary and Reflective Questions	169
8.7	Summary.	170
	References	170

9. Trust	171
9.1 Overview	171
9.2 Objectives	172
9.3 Study Questions	172
9.4 Software Intangibility and Process Transparency	173
9.5 Game Theory Perspective in Software Development	175
9.6 Ethics in Agile Teams	179
9.7 Diversity	183
9.8 Trust in Learning Environments	186
9.8.1 Teaching and Learning Principle	186
9.9 Summary and Reflective Questions	187
9.10 Summary	188
References	188
10. Globalization	189
10.1 Overview	190
10.2 Objectives	190
10.3 Study Questions	191
10.4 The Agile Approach in Global Software Development	191
10.4.1 Communication in Distributed Agile Teams	192
10.4.2 Planning in Distributed Agile Projects	193
10.4.3 Case Study 10.1. Tracking Agile Distributed Projects	193
10.4.4 Reflective Processes in Agile Distributed Teams	194
10.4.5 Organizational Culture and Agile Distributed Teams	195
10.5 Application of Agile Principles in Non-Software Projects	196
10.5.1 Case Study 10.2. Book Writing	196
10.6 Globalization in Learning Environments	197
10.6.1 Teaching and Learning Principles	197
10.6.2 An Agile Perspective on the Book/Course Structure	198
10.6.3 Case Study 10.3. Follow-the-Sun with Agile Development	199
10.7 Summary and Reflective Questions	201
10.8 Summary	202
References	202
11. Reflection	205
11.1 Overview	205
11.2 Objectives	206
11.3 Study Questions	206
11.4 Case Study 11.1. Reflection on Learning in Agile Software Development	207

11.5	Reflective Practitioner Perspective	208
11.6	Retrospective	210
11.6.1	The Retrospective Facilitator	211
11.6.2	Case Study 11.2. Guidelines for a Retrospective Session	212
11.6.3	Application of Agile Practices in Retrospective Sessions	213
11.6.4	End of the Release Retrospective	215
11.7	Reflection in Learning Environments	219
11.8	Summary and Reflective Questions	219
11.9	Summary	220
	References	220
12.	Change	223
12.1	Overview	223
12.2	Objectives	224
12.3	Study Questions	225
12.4	A Conceptual Framework for Change Introduction	225
12.4.1	Changes in Software Requirements	227
12.4.2	Organizational Changes	230
12.5	Transition to an Agile Software Development Environment	234
12.5.1	Organizational Survey	235
12.5.2	Case Study 12.1. A Report of an Organizational Survey	237
12.5.3	Case Study 12.2. Applying an Agile Process to a Transition Process	241
12.6	Change in Learning Environments	244
12.6.1	Introducing the Teaching of Agile Software Development	244
12.6.2	Two-Day Workshop	245
12.6.3	Two-Day Workshop Format for a Team of Academic Coaches	250
12.7	Summary and Reflective Questions	251
12.8	Summary	252
	References	252
13.	Leadership	253
13.1	Overview	253
13.2	Objectives	255
13.3	Study Questions	255
13.4	Leaders	256
13.4.1	Leadership Styles	257
13.4.2	Case Study 13.1. The Agile Change Leader	258
13.5	Coaches	264

13.6	Leadership in Learning Environments	264
13.6.1	Teaching and Learning Principles	265
13.6.2	Case Study 13.2. A Coaching Framework	265
13.7	Summary and Reflective Questions	273
13.8	Summary	273
	References	273
14.	Delivery and Cyclicity	275
14.1	Overview	275
14.2	Objectives	276
14.3	Study Questions	276
14.4	Delivery	277
14.4.1	Towards the End of the Release	277
14.4.2	Release Celebration	278
14.4.3	Reflective Session Between Releases	280
14.5	Cyclicity	287
14.6	Delivery and Cyclicity in Learning Environments	288
14.6.1	The Delivery in the Studio	288
14.6.2	Teaching and Learning Principles	290
14.7	Summary and Reflective Questions	291
14.8	Summary	291
	References	292
	Epilogue	293
	Index	295



<http://www.springer.com/978-1-84800-198-5>

Agile Software Engineering
Hazzan, O.; Dubinsky, Y.
2008, VIII, 296 p., Softcover
ISBN: 978-1-84800-198-5