

Contents

Part I Motivation

1	Fundamentals of Agile Distributed Software Development	3
	Darja Šmite, Nils Brede Moe, and Pär J. Ågerfalk	
1.1	Introduction	3
1.1.1	Distributed Software Development	3
1.1.2	Agile Software Development	4
1.2	Merging Agility with Distribution	4
1.2.1	Potential Issues	5
1.2.2	All or Nothing versus À la carte	6
1.3	Current Practice	6
1.4	Conclusions	7
	References	7

Part II Transition

2	Implementing Extreme Programming in Distributed Software	
	Project Teams: Strategies and Challenges	11
	Likoebe M. Maruping	
2.1	Introduction	11
2.2	Implementing XP Practices: Where Is an Organization to Start? . .	12
2.2.1	The Promise of XP	12
2.2.2	Understanding How Your Software Project Team Is Structured and Why It Matters	13
2.3	Case Overview	14
2.4	XP in Distributed Software Project Teams: Implementation Strategies and Pitfalls to Avoid	16
2.4.1	The Planning Game	16
2.4.2	Collective Ownership	18
2.4.3	Coding Standards	19
2.4.4	Use of a Metaphor	20

2.4.5	Simplicity of Design	21
2.4.6	Sustainable Pacing	22
2.4.7	Pair Programming	23
2.4.8	Continuous Integration and Unit Testing	24
2.4.9	Refactoring	25
2.4.10	Customer Involvement	26
2.4.11	Small Functional Releases	27
2.5	Conclusions	28
	References	29
	Further Reading	30
3	Transitioning from Distributed and Traditional to Distributed and Agile: An Experience Report	31
	Daniel Wildt and Rafael Prikladnicki	
3.1	Introduction	31
3.2	Case Overview	32
3.3	Transitioning to Agile in a Distributed Environment	34
3.3.1	Don't Tell What Agile Is and Be Successful	35
3.3.2	A Fully Cultural Transition from Traditional to Agile Development	37
3.3.3	Benefits of Using Agile Methods in Distributed Environment	40
3.4	Practical Recommendations	41
3.5	Conclusions	45
	References	45
4	Tailoring Agility: Promiscuous Pair Story Authoring and Value Calculation	47
	Steve Tendon	
4.1	Introduction	47
4.2	The Case	48
4.2.1	Background	48
4.2.2	Management Support and Sponsorship	50
4.2.3	The Pilot Project	51
4.2.4	The Journey of Implementing Agility	53
4.2.5	The Final: Project Approval	61
4.3	Benefits from Implementing Agility over Traditional Approaches	62
4.3.1	More Commonality	62
4.3.2	Smaller Scope	63
4.3.3	ROI Anticipation	63
4.3.4	Smaller Country-Specific Dependencies	63
4.3.5	Avoiding Waste Upfront	63
4.4	Why Agile Succeeded?	64
4.4.1	Induction	64
4.4.2	Co-location and Alternating On- and Off-Site Activities	65
4.4.3	Promiscuous Pair Story Authoring	66

4.4.4	Economic Value of Story Points	67
4.5	Conclusions	68
	References	69
5	Scrum and Global Delivery: Pitfalls and Lessons Learned	71
	Cristiano Sadun	
5.1	Introduction	71
5.2	Cases Overview	72
5.2.1	Background	72
5.2.2	Project NOR1	74
5.2.3	Project NOR2	76
5.3	The Experiences	77
5.3.1	Signing Agreements	77
5.3.2	Establishing Remote Access	79
5.3.3	Overcoming Communication Barriers	80
5.3.4	Actively Managing Distributed Agile Projects	82
5.3.5	Dealing with Idle Time	84
5.3.6	Achieving Motivation and Peer Feeling	86
5.3.7	Adapting Governance and Steering	87
5.4	Conclusions	88
	References	89
6	Onshore and Offshore Outsourcing with Agility: Lessons Learned	91
	Clifton Kussmaul	
6.1	Introduction	91
6.2	Case Overview	92
6.2.1	Background	92
6.2.2	Project Organization	93
6.2.3	Introduction of Agility	95
6.2.4	Overview of Project Activities	95
6.2.5	Cross-border Relationship Dynamics	97
6.3	Lessons Learned	98
6.3.1	People	99
6.3.2	Processes	100
6.3.3	Coordination	102
6.4	Conclusions	104
	References	105
	Further Reading	105
7	Contribution of Agility to Successful Distributed Software Development	107
	Saonee Sarker, Charles L. Munson, Suprateek Sarker, and Suranjan Chakraborty	
7.1	Introduction	107
7.2	Distributed Project Success	108
7.3	Types of Agility	109

7.4	Study Background	109
7.5	Contribution of Agility to Distributed Project Success	112
7.6	Conclusions	114
	References	116
8	Preparing your Offshore Organization for Agility: Experiences in India	117
	Jayakanth Srinivasan	
8.1	Introduction	117
8.2	Distributed Agile Software Development in India	118
8.3	Experiences from AgileCo	119
	8.3.1 Case Overview	119
	8.3.2 Personnel Selection and Training	120
	8.3.3 Teaching and Mentoring	122
	8.3.4 Managing Customer Expectations	123
8.4	Experience from BankCo	124
	8.4.1 Case Overview	124
	8.4.2 Impact of Senior Leadership Vision	125
	8.4.3 Heterogeneous Process Environment	126
	8.4.4 Agile Coaching	127
8.5	Conclusions	127
	References	129
Part III Management		
9	Improving Global Development Using Agile	133
	Alberto Avritzer, Francois Bronsard, and Gilberto Matos	
9.1	Introduction	133
9.2	The Projects	134
9.3	Deploying Agile Techniques in Global Projects	136
	9.3.1 Organizational Issues	136
	9.3.2 Communication Issues	137
	9.3.3 Process Issues	139
	9.3.4 Tools and Technical Issues	141
9.4	Improving Global Projects Using Agile Processes	143
9.5	Conclusions	147
	References	147
10	Turning Time from Enemy into an Ally Using the Pomodoro Technique	149
	Xiaofeng Wang, Federico Gobbo, and Michael Lane	
10.1	Introduction	149
10.2	Time Is an Enemy?	151
10.3	The Pomodoro Technique	152
	10.3.1 Pomodoro as Time-box	153
	10.3.2 Pomodoro as Unit of Effort	153

10.4	The Application of the Pomodoro Technique in Sourcesense	
	Milan Team	154
10.4.1	Background of Sourcesense Milan Team	154
10.4.2	The Development Process of Sourcesense Milan Team	155
10.4.3	Pomodoro as Time-box	156
10.4.4	Pomodoro as a Unit of Effort	159
10.4.5	Addressing Remote Collaboration with Teams That Do Not Employ the Pomodoro Technique	161
10.5	Turning Time into an Ally	161
10.5.1	Shared Pomodoro	162
10.5.2	Collective Breaks	162
10.5.3	Estimation and Tracking	163
10.5.4	One Pomodoro Rules All Sites?	163
10.6	Conclusions	164
	References	165
11	MBTA: Management By Timeshifting Around	167
	Erran Carmel	
11.1	Management by Wandering and Flying Around	167
11.2	Enter Timeshifting	168
11.3	Conclusions	170
	References	170
12	The Dilemma of High Level Planning in Distributed Agile Software Projects: An Action Research Study in a Danish Bank	171
	Per Svejvig and Ann-Dorte Fladkjær Nielsen	
12.1	Introduction	171
12.2	Research Methodology	173
12.2.1	Action Research	173
12.2.2	Research Settings	173
12.3	The Action Research Cycle	174
12.3.1	Diagnosing the Problem and the Underlying Causes	174
12.3.2	Action Planning	175
12.3.3	Action Taking	175
12.3.4	Evaluating and Learning	180
12.4	Conclusions	181
12.4.1	Applying a Holistic Approach to High Level Planning	181
12.4.2	Using Action Research to Software Process Improvement	182
12.4.3	Summary	182
	References	182
13	Tools for Supporting Distributed Agile Project Planning	183
	Xin Wang, Frank Maurer, Robert Morgan, and Josyleuda Oliveira	
13.1	Introduction	183
13.2	Distributed Planning Tool Requirements	185
13.2.1	Agile Planning Requirements	186

13.2.2	Requirements for Collaborative Interactions	187
13.3	Tool Review	188
13.3.1	Wikis	188
13.3.2	Web Form-Based Applications	189
13.3.3	Card-Based Planning Systems	190
13.3.4	Plugin for Integrated Development Environment	190
13.3.5	Synchronous Project Planning Tool	191
13.3.6	Digital Tabletop-Based Agile Planning Tool	193
13.4	Tool Evaluation	193
13.5	Practical Advice	195
13.5.1	Advice for Agile Planning Tool User	195
13.5.2	Advice for Designers of Distributed Agile Planning Tools	196
13.6	Conclusions	198
	References	199
14	Combining Agile and Traditional: Customer Communication in Distributed Environment	201
	Mikko Korkala, Minna Pikkariainen, and Kieran Conboy	
14.1	Introduction	201
14.2	Customer Communication in Distributed Agile Development	202
14.2.1	Issues Hindering the Customer Communication in Distributed Agile Development	204
14.3	Findings	205
14.3.1	Case Context	205
14.3.2	The Use of Agile Methodologies in the Case Project	207
14.3.3	The Use of Customer Communication Media	208
14.3.4	Identified Customer Communication Challenges	211
14.4	Discussion and Lessons Learned	214
	References	216
15	Coordination Between Global Agile Teams: From Process to Architecture	217
	Jan Bosch and Petra Bosch-Sijtsema	
15.1	Introduction	217
15.2	Large-Scale Software Development	220
15.3	Case Study Companies	221
15.3.1	Case Company GLOembed	221
15.3.2	Case Company GLOtelcom	222
15.3.3	Case Company GLOsoftware	222
15.4	Coordination and Integration Inter-team Challenges	224
15.4.1	Top-Down Approach Challenges	224
15.4.2	Interaction Problems	225
15.5	Coordination Through Architecture	226
15.5.1	Road Mapping	227
15.5.2	Requirements	228
15.5.3	Architecture	229

15.5.4	Development	229
15.5.5	Integration or Composition	230
15.5.6	Architecture-Centric Software Engineering	231
15.6	Conclusions	232
	References	233
16	Considering Subcontractors in Distributed Scrum Teams	235
	Jakub Rudzki, Imed Hammouda, Tuomas Mikkola, Karri Mustonen, and Tarja Systä	
16.1	Introduction	235
16.1.1	Company Context	236
16.1.2	Methodology	236
16.1.3	Main Results	237
16.2	Subcontractors in an SSP Company	238
16.2.1	Why Subcontractors?	239
16.2.2	Distributed Development Stakeholders	239
16.2.3	Subcontractor Selection Process	240
16.3	Subcontractors in Scrum Teams	242
16.3.1	Scrum	242
16.3.2	Communication	243
16.3.3	Planning and Progress Tracking	244
16.3.4	Code Sharing and Development Feedback	245
16.3.5	Knowledge Sharing	246
16.3.6	Team Spirit	246
16.4	Subcontractors and Project Phases	247
16.4.1	Preparation	247
16.4.2	Development	248
16.4.3	Release	251
16.5	Conclusions	251
16.5.1	Practical Implications	252
16.5.2	Research Implications	252
16.5.3	Summary	253
	Appendix	253
	References	255
	Further Reading	255

Part IV Teams

17	Using Scrum Practices in GSD Projects	259
	Maria Paasivaara and Casper Lassenius	
17.1	Introduction	259
17.2	Research Methodology	260
17.3	Distributed Daily Scrums	260
17.3.1	Application of Daily Scrums to Distributed Projects	262
17.3.2	Benefits of Daily Scrums	263

17.3.3	Challenges of Daily Scrums	263
17.4	Scrum-of-Scrums Meetings	264
17.4.1	Application of Scrum-of-Scrums to Distributed Projects . .	265
17.4.2	Benefits of Scrums-of-Scrums	265
17.4.3	Challenges of Scrums-of-Scrums	266
17.5	Sprints	266
17.5.1	Application of Sprints to Distributed Projects	267
17.5.2	Benefits of Sprints	267
17.5.3	Challenges of Sprints	268
17.6	Sprint Planning Meetings	268
17.6.1	Application of Sprint Planning Meetings to Distributed Projects	268
17.6.2	Benefits of Sprint Planning Meetings	269
17.6.3	Challenges of Sprint Planning Meetings	270
17.7	Sprint Demos	270
17.7.1	Application of Sprint Demos to Distributed Projects	270
17.7.2	Benefits of Sprint Demos	271
17.7.3	Challenges of Sprint Demos	271
17.8	Retrospective Meetings	271
17.8.1	Application of Retrospective Meetings to Distributed Projects	271
17.8.2	Benefits of Retrospective Meetings	272
17.8.3	Challenges of Retrospective Meetings	272
17.9	Backlogs	272
17.9.1	Application of Backlogs to Distributed Projects	273
17.9.2	Benefits of Backlogs	273
17.9.3	Challenges of Backlogs	273
17.10	Frequent Visits	273
17.10.1	First Visit	274
17.10.2	Further Visits	274
17.10.3	Benefits of Frequent Visits	275
17.10.4	Challenges of Frequent Visits	275
17.11	Multiple Communication Modes	276
17.11.1	Benefits of Multiple Communication Modes	276
17.11.2	Challenges of Multiple Communication Modes	277
17.12	Conclusions	277
	References	277
18	Feature Teams—Distributed and Dispersed	279
	Jutta Eckstein	
18.1	Introduction	279
18.2	Context	280
18.3	Historical Structures of Distributed Teams	280
18.3.1	Consequences	281
18.4	Building Agile Teams	281

18.4.1	Feature Teams—Co-located or Dispersed	282
18.4.2	Creating Proximity for Dispersed Feature Teams	284
18.5	Technical Service Team Ensures Conceptual Integrity	285
18.5.1	Starting Team as Role Model	286
18.6	Conclusions	286
	References	287
	Further Reading	287
19	Roles and Responsibilities in Feature Teams	289
	Jutta Eckstein	
19.1	Introduction	289
19.2	Context	290
19.3	Configuration of a Feature Team	291
19.4	Product Owner	292
19.4.1	Team of Product Owners	292
19.4.2	Lead Product Owner	293
19.4.3	Collaborating with Both: Customers and Feature Team	293
19.5	Coach—Also Known as Scrum-Master	294
19.6	Architect and Architecture	295
19.6.1	Chief Architect	296
19.7	Project Manager	297
19.8	Key Roles Support Their Teams Directly	297
19.9	Conclusions	298
	References	299
	Further Reading	299
20	Getting Communication Right: The Difference Between Distributed Bliss or Miss	301
	Jan-Erik Sandberg and Lars Arne Skaar	
20.1	Introduction	301
20.2	Background Overview	302
20.2.1	Background	302
20.3	Starting a Distributed Agile Project	303
20.4	Low-cost and Effective Communication	304
20.5	Empower the Team	306
20.6	Common Architecture Across Locations	307
20.7	On “Proxies”	308
20.8	Conclusions	309
	References	309
21	A Task-Driven Approach on Agile Knowledge Transfer	311
	Jörn Koch and Joachim Sauer	
21.1	Introduction	311
21.2	Case Overview	312
21.3	Hands-On Approach (Task-Driven Approach)	315

21.3.1	Joint Task Planning	316
21.3.2	Question-Driven Task Scheduling	316
21.3.3	Adequate Task Design	317
21.3.4	Scrupulous Task Sign-Off	318
21.4	Conclusion	318
	References	319
22	Architecture-Centric Development in Globally Distributed Projects .	321
	Joachim Sauer	
22.1	Introduction	321
22.2	Case Overview	322
22.3	Software Architecture and Architecture-Centric Development . . .	323
22.3.1	Software Architecture	323
22.3.2	Architecture-Centric Development in General	324
22.3.3	Architecture-Centric Development in Agile Distributed Settings	324
22.4	Distributed Continuous Integration and Collective Ownership . . .	325
22.5	Practical Advice for Software Architects	326
22.6	Conclusions	328
	References	328
Part V	Epilogue	
23	Agility Across Time and Space: Summing up and Planning for the Future	333
	Darja Šmite, Nils Brede Moe, and Pär J. Ågerfalk	
23.1	The Beginning of the End	333
23.2	Current Themes	334
23.3	Practical Advice	334
23.4	Areas for Improvement and Future Research	336
23.5	The End of The End	337
Index	339

Agility Across Time and Space

Implementing Agile Methods in Global Software Projects

Šmite, D.; Moe, N.B.; Ågerfalk, P.J. (Eds.)

2010, XXXVI, 341 p., Hardcover

ISBN: 978-3-642-12441-9