## Contents

<table>
<thead>
<tr>
<th>Preface</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description and assessment</td>
<td>vii</td>
</tr>
<tr>
<td>Keeping a cool head</td>
<td>viii</td>
</tr>
<tr>
<td>Previous attempts</td>
<td>ix</td>
</tr>
<tr>
<td>Structure of the book</td>
<td>ix</td>
</tr>
<tr>
<td>Perspective and scope</td>
<td>x</td>
</tr>
<tr>
<td>Analysis: instinctive, experiential, logical or empirical?</td>
<td>xi</td>
</tr>
<tr>
<td>Free critical inquiry</td>
<td>xii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xiii</td>
</tr>
</tbody>
</table>

## Contents xv

<table>
<thead>
<tr>
<th>1 OVERVIEW</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 VALUES</td>
<td>2</td>
</tr>
<tr>
<td>1.2 PRINCIPLES</td>
<td>4</td>
</tr>
<tr>
<td>- Organizational principles</td>
<td>5</td>
</tr>
<tr>
<td>- Technical principles</td>
<td>6</td>
</tr>
<tr>
<td>1.3 ROLES</td>
<td>7</td>
</tr>
<tr>
<td>1.4 PRACTICES</td>
<td>8</td>
</tr>
<tr>
<td>- Organizational practices</td>
<td>8</td>
</tr>
<tr>
<td>- Technical practices</td>
<td>9</td>
</tr>
<tr>
<td>1.5 ARTIFACTS</td>
<td>10</td>
</tr>
<tr>
<td>- Virtual artifacts</td>
<td>10</td>
</tr>
<tr>
<td>- Material artifacts</td>
<td>11</td>
</tr>
<tr>
<td>1.6 A FIRST ASSESSMENT</td>
<td>12</td>
</tr>
<tr>
<td>- Not new and not good</td>
<td>12</td>
</tr>
<tr>
<td>- New and not good</td>
<td>13</td>
</tr>
<tr>
<td>- Not new but good</td>
<td>14</td>
</tr>
<tr>
<td>- New and good!</td>
<td>14</td>
</tr>
</tbody>
</table>
2 DECONSTRUCTING AGILE TEXTS
   2.1 THE PLIGHT OF THE TRAVELING SEMINARIST 17
      Proof by anecdote 18
      When writing beats speaking 19
      Discovering the gems 20
      Agile texts: reader beware! 21
   2.2 THE TOP SEVEN RHETORICAL TRAPS 22
      Proof by anecdote 22
      Slander by association 23
      Intimidation 23
      Catastrophism 26
      All-or-nothing 27
      Cover-your-behind 27
      Unverifiable claims 28
      Postscript: you have been ill-served by the software industry! 30
3 THE ENEMY: BIG UPFRONT ANYTHING 31
   3.1 PREDICTIVE IS NOT WATERFALL 31
   3.2 REQUIREMENTS ENGINEERING 32
      Requirements engineering techniques 32
      Agile criticism of upfront requirements 32
      The waste criticism 33
      The change criticism 35
      The domain and the machine 36
   3.3 ARCHITECTURE AND DESIGN 37
      Is design separate from implementation? 37
      Agile methods and design 39
   3.4 LIFECYCLE MODELS 41
   3.5 RATIONAL UNIFIED PROCESS 42
   3.6 MATURITY MODELS 43
      CMMI in plain English 44
      The Personal Software Process 46
      CMMI/PSP and agile methods 46
      An agile maturity scale 47
4 AGILE PRINCIPLES 49
   4.1 WHAT IS A PRINCIPLE? 49
   4.2 THE OFFICIAL PRINCIPLES 50
   4.3 A USABLE LIST 51
   4.4 ORGANIZATIONAL PRINCIPLES 51
Contents

1 INTRODUCTION 43
1.1 INTRODUCTION 43
1.2 WHAT IS AGILE? 44
1.3 THE AGILE MANIFESTO 45

2 AGILE PRINCIPLES 47
2.1 PEOPLE AND PROCESS OVER TOOLS 47
2.2 WORK OVER FEAR OF CHANGE 48
2.3 COLLABORATION OVER AUTONOMY 51
2.4 DELIVER VALUE DAILY 52

3 OVERVIEW OF THE AGILE PROCESS 53
3.1 THE AGILE PROCESS 53
3.2 THE AGILE CIRCLE 57

4 AGILE PROCESS 59
4.1 THE SPRINT 59
4.2 CLEAR COMMUNICATION 60
4.3 AUGMENTED NAVIGATION 62
4.4 AUGMENTED ACCESS 63

4.5 TECHNICAL PRINCIPLES 70
4.5.1 DEVELOP ITERATIVELY 70
4.5.2 TREAT TESTS AS A KEY RESOURCE 75
4.5.3 DO NOT START ANY NEW DEVELOPMENT UNTIL ALL TESTS PASS 76
4.5.4 TEST FIRST 77
4.5.5 EXPRESS REQUIREMENTS THROUGH SCENARIOS 77

5 AGILE ROLES 79
5.1 MANAGER 79
5.2 PRODUCT OWNER 80
5.3 TEAM
5.3.1 SELF-ORGANIZING 80
5.3.2 CROSS-FUNCTIONAL 81
5.4 MEMBERS AND OBSERVERS 82
5.5 CUSTOMER 82
5.6 COACH, SCRM MASTER 84
5.7 SEPARATING ROLES 86

6 AGILE PRACTICES: MANAGERIAL 89
6.1 SPRINT 89
6.1.1 SPRINT BASICS 89
6.1.2 THE CLOSED-WINDOW RULE 90
6.1.3 SPRINT: AN ASSESSMENT 91
6.2 DAILY MEETING 91
6.3 PLANNING GAME 93
6.4 PLANNING POKER 94
6.5 ONSITE CUSTOMER 96
6.6 OPEN SPACE 96
6.7 PROCESS MINIATURE 97
6.8 ITERATION PLANNING 98
6.9 REVIEW MEETING 99
6.10 RETROSPECTIVE 99
6.11 SCRUM OF SCRUMS 99
6.12 COLLECTIVE CODE OWNERSHIP 100
The code ownership debate 100
Collective ownership and cross-functionality 102

7 AGILE PRACTICES: TECHNICAL 103
   7.1 DAILY BUILD AND CONTINUOUS INTEGRATION 103
   7.2 PAIR PROGRAMMING 105
      Pair programming concepts 106
      Pair programming versus mentoring 107
      Mob programming 107
      Pair programming: an assessment 107
   7.3 CODING STANDARDS 109
   7.4 REFACTORING 109
      The refactoring concept 109
      Benefits and limits of refactoring 110
      Incidental and essential changes 112
      Combining a priori and a posteriori approaches 113
   7.5 TEST-FIRST AND TEST-DRIVEN DEVELOPMENT 113
      The TDD method of software development 113
      An assessment of TFD and TDD 115

8 AGILE ARTIFACTS 117
   8.1 CODE 117
   8.2 TESTS 117
   8.3 USER STORIES 119
   8.4 STORY POINTS 121
   8.5 VELOCITY 123
   8.6 DEFINITION OF DONE 125
   8.7 WORKING SPACE 125
   8.8 PRODUCT BACKLOG, ITERATION BACKLOG 126
   8.9 STORY CARD, TASK CARD 127
   8.10 TASK AND STORY BOARDS 127
   8.11 BURNDOWN AND BURNUP CHARTS 128
   8.12 IMPEDIMENT 129
   8.13 WASTE, TECHNICAL DEBT, DEPENDENCY, DEPENDENCY CHARTS 129

9 AGILE METHODS 133
   9.1 METHODS AND METHODOLOGY 133
      Terminology 133
      The fox and the hedgehog 133
   9.2 LEAN SOFTWARE AND KANBAN 134
CONTENTS

Lean Software’s Big Idea 134
Lean Software’s principles 134
Lean Software: an assessment 135
Kanban 136

9.3 EXTREME PROGRAMMING 137
XP’s Big Idea 137
XP: the unadulterated source 137
Key XP techniques 138
Extreme Programming: an assessment 139

9.4 SCRUM 139
Scrum’s Big Idea 139
Key Scrum practices 140
Scrum: an assessment 140

9.5 CRYSTAL 141
Crystal’s Big Idea 141
Crystal principles 141
Crystal: an assessment 142

10 DEALING WITH AGILE TEAMS 145
10.1 GRAVITY STILL HOLDS 145
10.2 THE EITHER-WHAT-OR-WHEN FALLACY 146

11 THE UGLY, THE HYPE AND THE GOOD:
AN ASSESSMENT OF THE AGILE APPROACH 149
11.1 THE BAD AND THE UGLY 149
Deprecation of upfront tasks 149
User stories as a basis for requirements 150
Feature-based development and ignorance of dependencies 150
Rejection of dependency tracking tools 150
Rejection of traditional manager tasks 150
Rejection of upfront generalization 151
Embedded customer 151
Coach as a separate role 151
Test-driven development 151
Deprecation of documents 151

11.2 THE HYPED 152

11.3 THE GOOD 153

11.4 THE BRILLIANT 154

Bibliography 155
Index 163
Agile!
The Good, the Hype and the Ugly
Meyer, B.
2014, XIX, 170 p. 15 illus. in color., Softcover
ISBN: 978-3-319-05154-3