## 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

#### 1 Overview 1

1.1 Values 2

1.2 Principles 4

- Organizational principles 5
- Technical principles 6

1.3 Roles 7

1.4 Practices 8

- Organizational practices 8
- Technical practices 9

1.5 Artifacts 10

- Virtual artifacts 10
- Material artifacts 11

1.6 A First Assessment 12

- Not new and not good 12
- New and not good 13
- Not new but good 14
- New and good! 14
## 2 Deconstructing Agile Texts

### 2.1 The Plight of the Traveling Seminarist
- Proof by anecdote
- When writing beats speaking
- Discovering the gems
- Agile texts: reader beware!

### 2.2 The Top Seven Rhetorical Traps
- Proof by anecdote
- Slander by association
- Intimidation
- Catastrophism
- All-or-nothing
- Cover-your-behind
- Unverifiable claims

Postscript: you have been ill-served by the software industry!

## 3 The Enemy: Big Upfront Anything

### 3.1 Predictive is Not Waterfall

### 3.2 Requirements Engineering
- Requirements engineering techniques
- Agile criticism of upfront requirements
- The waste criticism
- The change criticism
- The domain and the machine

### 3.3 Architecture and Design
- Is design separate from implementation?
- Agile methods and design

### 3.4 Lifecycle Models

### 3.5 Rational Unified Process

### 3.6 Maturity Models
- CMMI in plain English
- The Personal Software Process
- CMMI/PSP and agile methods
- An agile maturity scale

## 4 Agile Principles

### 4.1 What is a Principle?

### 4.2 The Official Principles

### 4.3 A Usable List

### 4.4 Organizational Principles
CONTENTS

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
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