Agile Software Development
Agile software development is the most important paradigm that has swept the software development world over the last decade. Even if it does not represent the most popular software development approach in actual use, it has certainly become of the most talked about. Its vocabulary and prime ideas have already started spilling over into other fields, project management in particular.

Agile software development also carries the unique distinction of being the source of continued debate since its inception. The controversy surrounding it simply doesn’t want to die out. Few other movements have pitted detractors and advocates against each other so passionately, so religiously.

But why? Shouldn’t a decade be enough to settle at least the fundamental arguments? Those arguments should be getting pretty old by now if it were not for two reasons that still fuel the debate.

The first reason is the inherent fuzziness of the topic. Agile software development is multi-faceted and poorly delimited. As such the paradigm doesn’t lend itself to a crisp definition, let alone straightforward study. Is agility a general development philosophy? Is it about dealing with change and uncertainty? Is it a project management philosophy? A way of working with software development teams? A way of working inside software development teams? A way of thinking about software? A set of technical practices that target better quality and higher productivity? A set of collaboration practices that cater to the needs of customers and end users? A set of principles and values of professional conduct? A way of life? A rebellion? A religion? A cultural revolution of the software intellectuals? Well, it’s all of the above and none of the above at the same time. That the scope of agile software development is nebulous and dependent on personal and contextual interpretation makes it a hard nut to crack in systematic means. Misconceptions both by detractors and advocates find fertile ground to take hold, amplify and multiply in an unproductive cycle. When a topic is that fuzzy, understanding the boundaries and what lies inside those boundaries become almost as important as understanding the intricacies of the individual constituents.

The second reason is poor dissemination and insufficient synthesis of fragmented research results. It’s not that our knowledge of the different facets of agile software development has not expanded significantly over the past years. It has, thanks to the still ongoing research efforts that have undertaken the difficult task of dissecting the elastic anatomy of agile software development. Alas, the scattered results of these efforts are neither well publicized nor readily available to the questioning reader. The world simply doesn’t know what we collectively know about agile software development. Worse it doesn’t know what we still don’t know about it.

This book contributes to the agile debate by addressing both sources of the agile confusion: fuzzy, multi-faceted scope and poor, unconsolidated dissemination of efforts representing the collective understanding of an expert community. The
book represents a comprehensive snapshot of the knowledge accumulated over many years of research by those working closely with the industry, collecting data, observing practitioners in the field, synthesizing insights, devising theories, trying new methods to investigate core issues, and gathering clues to overcome outstanding challenges. It’s your one-stop resource to agile software development research with contributions by the best people in the community, by people who know what they’re talking about. Enjoy it. Digest it. Use it.

Ottawa, March 2010

Hakan Erdogmus, Kalemun Research Inc.
Preface

Principles of agile software development have a large impact on how software is developed. Some have characterized the change towards agile development as a paradigm shift, leading the focus to topics that have not been addressed or understood in traditional development.

It is therefore important to address what defines and characterizes agile development, what are the historical roots? How do the different principles, processes and methods work in practice, how does agile development affect various groups who are participating in software development? What new challenges arise when using agile development, and what challenges will the methods be unable to solve?

The transition to agile software development has been driven by practitioners, more or less informed by research results, mainly from fields not traditionally focusing on software development. However, researchers focusing on agile software development have a role in developing an understanding of how agile development methods work. Further, why they do or do not work, and in which situations or environment they work better or worse.

This book seeks to show the current state of research on agile software development through an introduction and ten invited contributions on some of the main research fields and by some of the main researchers. The chapters both show the main results in each subfield, and in addition explain what these results mean to practitioners as well as for future research in the field.

The book is aimed at reflective practitioners and researchers, and we hope the book also can serve for graduate courses at universities.

We are very grateful to the chapter authors who have contributed with important overview articles in their own research areas, and also are presenting their chapters at the 11th International Conference on Agile Software Development (XP2010). The editing of this book was supported by the EVISOFT project, which is partially funded by the Research Council of Norway under Grant 174390/I40.

Trondheim, March 2010

Torgeir Dingsøyr
Tore Dybå
Nils Brede Moe
# Contents

1 Agile Software Development: An Introduction and Overview ........................1
   1.1 Introduction ..................................................................................1
   1.2 What is Agile Development? ..........................................................2
   1.3 Research on Agile Software Development .................................2
   1.4 Main Themes in this Book .............................................................7
   1.5 Conclusion ....................................................................................11
   References ..........................................................................................11
   Author Biographies ............................................................................12

2 Towards an Understanding of the Conceptual Underpinnings of Agile Development Methodologies .................................................................15
   2.1 Introduction ..................................................................................15
   2.2 The Challenges of Contemporary Software Development ................16
   2.3 What’s New About Agile Development? .........................................19
   2.4 Principles of Sociotechnical Systems ............................................21
   2.5 ADM and the System of Systems Methodologies (SOSM) ...............23
   2.6 Implications for Research and Practice .........................................25
   2.7 Conclusion ....................................................................................26
   References ..........................................................................................27
   Author Biographies ............................................................................29

3 Agile Software Development Methods: A Comparative Review ..............31
   3.1 Introduction ..................................................................................31
   3.2 An overview of agile methods .......................................................33
   3.3 Comparative review of the existing agile methods .........................36
   3.4 Discussion ....................................................................................49
   3.5 Conclusion ....................................................................................52
   Acknowledgements ............................................................................53
   References ..........................................................................................53
   Author Biographies ............................................................................58

4 Three ‘C’s of Agile Practice: Collaboration, Co-ordination and Communication ..........................................................61
   4.1 Introduction ..................................................................................61
   4.2 Fieldwork ....................................................................................62
   4.3 Results ..........................................................................................69
   4.4 Discussion ....................................................................................79
   4.5 Conclusion ....................................................................................82
   Acknowledgements ............................................................................82
   References ..........................................................................................83
   Author Biographies ............................................................................85
5 From Exotic to Mainstream: A 10-year Odyssey from Internet Speed to Boundary Spanning with Scrum

5.1 Introduction
5.2 Research Methodology
5.3 Study One Results: Racing the E-bomb
5.4 Study Two Results: A New Software Development Process
5.5 Study Three Results: Balancing Speed and Quality
5.6 Study Four Results: Boundary Spanning with Scrum
5.7 Discussion and Conclusion

References
Author Biographies

6 An Ideal Customer: A Grounded Theory of Requirements Elicitation, Communication and Acceptance on Agile Projects

6.1 Introduction
6.2 Research Method
6.3 What is the Experience of the Customer?
6.4 Is the Customer a Single Person or a Team?
6.5 What practices enhance the effectiveness of the customer?
6.6 Discussion and Conclusion

References
Author Biographies

7 Pair Programming: Issues and Challenges

7.1 Introduction
7.2 Horse Trading Problem: Understanding Pair vs. Solo
7.3 Pair Programming Studies
7.4 Repeat Programming
7.5 Conclusion

References
Author Biographies

8 Architected Agile Solutions for Software-Reliant Systems

8.1 Introduction
8.2 Key Principles
8.3 How Much Architecting is Enough?
8.4 Early Architected-Agile Success Cases
8.5 Implications for Practice and Future Research
8.6 Conclusion

References
Author Biographies
9 Agile Interaction Design and Test-Driven Development of User Interfaces – A Literature Review ........................................... 185
  9.1 Introduction ........................................................................... 185
  9.2 Agile Interaction Design .......................................................... 186
  9.3 User Interface Test-Driven Development ............................... 191
  9.4 Conclusion ............................................................................. 199
  References ..................................................................................... 199
  Author Biographies ........................................................................ 201

10 Organizational Culture and the Deployment of Agile Methods: The Competing Values Model View ........................................ 203
  10.1 Introduction ............................................................................ 203
  10.2 Organizational Culture and the Competing Values Model ........ 204
  10.3 Organizational Culture and the Deployment of Agile Methods ...... 208
  10.4 Conclusion ............................................................................. 217
  References ..................................................................................... 218
  Author Biographies ........................................................................ 222

11 Future Research in Agile Systems Development: Applying Open Innovation Principles Within the Agile Organisation .................... 223
  11.1 Introduction ............................................................................ 223
  11.2 Innovation in Agile Development – The Current State of Research .... 224
  11.3 Contemporary Thinking on Innovation – Open Innovation .......... 225
  11.4 Project Management in An Open Agile Environment ................ 227
  11.5 Conceptual Framework to Guide Future Research ..................... 227
  11.6 Conclusion and Implications for Future Research ...................... 230
  Acknowledgements ........................................................................ 232
  References ..................................................................................... 232
  Author Biographies ........................................................................ 234

Index .................................................................................................. 237
List of Contributors

Pekka Abrahamsson
Department of Computer Science
University of Helsinki
PO Box 68
FI-00014 University of Helsinki, Finland
pekka.abrahamsson@cs.helsinki.fi

VenuGopal Balijepally
Department of Accounting, Finance & MIS
Prairie View A&M University
Prairie View, Texas 77446-0519, USA
vebalijepally@pvamu.edu

Kyle Atikus Barnes
Seattle Software Solutions
1969 SW Hillcrest Road
Seattle, WA 98166, USA
kyle@seattle-softwaresolutions.com

Richard Baskerville
Department of Computer Information Systems
Georgia State University
University Plaza, Atlanta
Georgia 30302, USA
baskerville@acm.org

Robert Biddle
School of Computer Science
Carleton University
214C, Social Sciences Research Building
Ottawa, ON K1S 5B6, Canada
robert_biddle@carleton.ca

Barry Boehm
Center for Systems and Software Engineering
University of Southern California
941 W. 37th Place, SAL Room 328
Los Angeles, CA 90089-0781, USA
boehm@usc.edu
Philip L. Bond  
Department of Information Systems and Operations Management  
University of Texas at Arlington  
Arlington, TX 76019, USA  
philip.bond@mavs.uta.edu

Alan Cannon  
Department of Information Systems and Operations Management  
University of Texas at Arlington  
Arlington, TX 76019, USA  
acannon@uta.edu

Keith C.C.Chan  
Department of Computing  
The Hong Kong Polytechnic University  
Hung Hom, Kowloon, Hong Kong  
cskcchan@comp.polyu.edu.hk

Kieran Conboy  
Department of Accounting and Finance  
National University of Ireland, Galway,  
Galway, Ireland  
kieran.conboy@nuigalway.ie

Torgeir Dingsøyr  
SINTEF  
NO-7465 Trondheim, Norway  
torgeir.dingsoyr@sintef.no

Tore Dybå  
SINTEF  
NO-7465 Trondheim, Norway  
tore.dyba@sintef.no

Theodore D. Hellmann  
Department of Computer Science  
University of Calgary  
ICT 602, 2500 University Drive NW  
Calgary, Alberta, T2N 1N4, Canada  
tdhellma@ucalgary.ca
Ali Hosseini-Khayat  
Department of Computer Science  
University of Calgary  
CT 602, 2500 University Drive NW  
Calgary, Alberta, T2N 1N4, Canada  
hosseisa@ucalgary.ca

Juhani Iivari  
Department of Information Processing Sciences  
University of Oulu  
P.O. Box 3000, 90014 Oulun yliopisto, Finland  
juhani.iivari@oulu.fi

Netta Iivari  
Department of Information Processing Sciences  
University of Oulu  
P.O. Box 3000, 90014 Oulun yliopisto, Finland  
netta.iivari@oulu.fi

Supannika Koolmanojwong  
Center for Systems and Software Engineering  
University of Southern California  
941 W. 37th Place, SAL Room 328  
Los Angeles, CA 90089-0781, USA  
koolmano@usc.edu

Jo Ann Lane  
Center for Systems and Software Engineering  
University of Southern California  
941 W. 37th Place, SAL Room 328  
Los Angeles, CA 90089-0781, USA  
jolane@usc.edu

Kim Man Lui  
Department of Computing  
The Hong Kong Polytechnic University  
 Hunghom, Kowloon, Hong Kong  
cskmlui@comp.polyu.edu.hk
List of Contributors

Sabine Madsen
Department of Communication, Business and IT
Roskilde University
Universitetsvej 1
DK-4000 Roskilde, Denmark
sabinem@ruc.dk

Angela Martin
Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, 3240, New Zealand
angela@cs.waikato.ac.nz

Frank Maurer
Department of Computer Science
University of Calgary
2500 University Drive NW
Calgary, Alberta, T2N 1N4, Canada
fmaurer@ucalgary.ca

Nils Brede Moe
SINTEF
NO-7465 Trondheim, Norway
nils.b.moe@sintef.no

Lorraine Morgan
LERO - The Irish Software Engineering Research Centre
University of Limerick
Limerick, Ireland
lorraine.morgan@ul.ie

Sridhar Nerur
Department of Information Systems and Operations Management
University of Texas at Arlington
Arlington, TX 76019, USA
snerur@uta.edu

James Noble
School of Engineering and Computer Science
Victoria University
PO Box 600
Wellington 6140, New Zealand
kjm@ecs.vuw.ac.nz
Nilay Oza
VTT Technical Research Centre of Finland
PO Box 1000
FI-02044 VTT, Finland
nilay.oza@vtt.fi

Jan Pries-Heje
Department of Communication, Business and IT
Roskilde University
Universitetsvej 1
DK- 4000 Roskilde, Denmark
janph@ruc.dk

Hugh Robinson
Mathematics, Computing and Technology Faculty
The Open University
Walton Hall
Milton Keynes, MK7 6AA, UK
h.m.robinson@open.ac.uk

Helen Sharp
Mathematics, Computing and Technology Faculty
The Open University
Walton Hall
Milton Keynes, MK7 6AA, UK
h.c.sharp@open.ac.uk

Mikko Siponen
Department of Information Processing Science
University of Oulu
PO Box 3000
FI-90014 University of Oulu, Finland
mikko.siponen@oulu.fi

Richard Turner
Stevens Institute of Technology
Hoboken, NJ 07030, USA
rturner@stevens.edu
1 Agile Software Development: An Introduction and Overview

Torgeir Dingsøyr, Tore Dybå, Nils Brede Moe

Abstract: Agile software development is an important topic in software engineering and information systems. This chapter provides a characterization and definition of agile software development, an overview of research through a summary of existing overview studies, an analysis of the research literature so far, and an introduction to the main themes of this book. The first part of the book provides foundations and background of agile development. The second part describes findings from studies of agile methods in practice. The third part identifies principal challenges and discusses new frontiers that agile development methods will meet in the future.

1.1 Introduction

Agile software development has had a major influence on how software development is conducted. It has become an umbrella term for a number of changes in how software developers plan and coordinate their work, how they communicate with customers and external stakeholders, and how software development is organized in small, medium-sized and large companies from the telecom and healthcare sectors to games and interactive media.

We see the agile development methods as a reaction to plan-based or traditional methods, which emphasize “a rationalized, engineering-based approach” (Dybå 2000) incorporating extensive planning, codified processes, and rigorous reuse (Boehm 2002). By contrast, agile methods address the challenge of an unpredictable world by recognizing the value competent people and their relationships bring to software development (Nerur and Balijepally 2007).

In this chapter, we will first define what we see as agile software development and define other central terms that will be used throughout the book. Further, we give a broad overview of research conducted in this field, and describe the main themes of the book: foundations and background of agile development, agile methods in practice and principal challenges and new frontiers. Finally, we state what we see as some of the main challenges and main future directions for research on agile software development.
1.2 What is Agile Development?

In an introduction to the special issue on agile methods in IEEE Computer in 2003, Williams and Cockburn (2003) state that agile software development “is about feedback and change”, and they emphasize that software development is an empirical or nonlinear process, where short feedback-loops are necessary to achieve a desirable, predictable outcome. Ericksson et al. (2005) further underline the importance of lightweight processes in agile development, defining agility as to “strip away as much of the heaviness, commonly associated with the traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like.” (p. 89).

In an article discussing the concept of agility and leaness in software development, Conboy (2009) argues that agile methods must contribute to one of more of the following: creation of change, proaction in advance of change, reaction to change or learning from change. Further, an agile method must contribute to and not detract from: perceived economy, perceived quality, and perceived simplicity. A third requirement is to be continually ready to prepare the component for use.

Thus, agile software development has been characterized differently than plan-based or traditional development methods, mainly with the focus adapting to change and delivering products of high quality through simple work-processes. Nerur & Balijepally (2005) state that agile and traditional methods diverge on a number of aspects, including their fundamental assumptions, approach to control, management style, knowledge management, role assignment, role of the customer, project cycle, development model and desired organizational structure.

1.3 Research on Agile Software Development

In this section we first give an overview of prior research on agile software development, and then characterize the status of current research through examining the volume of scientific studies on the topic.

1.3.1 An Overview of Prior Research

Introductions to and overviews of agile development are given by Abrahamsson et al. (2002), Cohen et al. (2004), Erickson et al. (2005) and Dybå and Dingsøyr (2008). These four reports describe the state of the art and state of the practice in terms of characteristics of the various agile methods and lessons learned from applying such methods in industry.
The first review of the literature on agile software development was done in a technical report published by Abrahamsson et al. at VTT (2002). The report discusses the concept of agile development, presents processes, roles, practices, and experience with 10 agile development methods, and compares the methods with respect to the phases that they support and the level of competence that they require. Only DSDM and the Rational Unified Process were found to give full coverage to all phases of development, while Scrum mainly covers aspects related to project management. Abrahamsson et al. found anecdotal evidence that agile methods are “effective and suitable for many situations and environments”, but state that very few empirically validated studies support these claims. Chapter 3 in this book updates this review.

Cohen et al. (2004) published a review that emphasized the history of agile development, showing some of the roots to other disciplines, and, in particular, discussed relations between agile development and the Capability Maturity Model. The authors believed that agile methods would be consolidated in the future, just as object-oriented methods were consolidated. Further, they did not believe that agile methods would rule out traditional methods. Rather, they believe that agile and traditional methods will have a symbiotic relationship, in which factors such as the number of people working on a project, application domain, criticality, and innovativeness will determine which process to select.

Erickson et al. (2005) described the state of research on extreme programming (XP), agile software development, and agile modelling. With respect to XP, they found a small number of case studies and experience reports that promote the success of XP. The practice of pair programming is supported by a more well-established stream of research, and that there are also some studies on iterative development. Erickson et al. recommend that the other core practices in XP be studied separately in order to identify what practices are working. Further, they saw challenges with matching agile software development methods with standards such as ISO, and they argued that this is an area that needs further research.

What is currently known about the benefits and limitations of agile software development, the strength of the evidence in support of these findings, and the implications of these studies for the software industry and the research community were the focus areas for Dybå and Dingsøyr’s (2008) systematic review. The studies fell into four thematic groups: introduction and adoption, human and social factors, perceptions of agile methods, and comparative studies. The authors identified a number of reported benefits and limitations of agile development within each of these themes:

Regarding introduction, XP was difficult to introduce in complex organizations, yet seemingly easy in other types of organizations. Most studies reported that agile development practices are easy to adopt and work well. Benefits were reported in the following areas: customer collaboration, work processes for handling defects, learning in pair programming, thinking ahead for management, focusing on current work for engineers, and estimation.
A recurring theme in studies on agile development was human and social factors. A benefit of XP was that it thrived in radically different environments; in organizations that varied from having a hierarchical structure to little or no central control. Further, conversation, standardization, and tracking progress have been studied and are described as mechanisms for creating awareness within teams and organizations.

Many studies sought to identify how agile methods are perceived by different groups. Customers are satisfied with opportunities for feedback and response to change. However, the role of on-site customer can be stressful and cannot be sustained for a long period. Companies that use XP have more satisfied employees. There were mixed findings regarding the effectiveness of pair programming. University students perceive agile methods as providing them with relevant training.

The last theme was comparative studies. Some studies suggest benefits in projects that use agile methods because changes are incorporated more easily and business value is demonstrated more efficiently. It is possible to combine agile project management with overall traditional principles, such as the stage-gate project management model. With respect to the productivity of agile and traditional teams, three of the four comparative studies that address this issue found that using XP results in increased productivity.

![Fig. 1.1 Publications on agile software development, total number (top), conference papers (middle) and journal articles (bottom). Note that the numbers for 2009 do not include all publications from that year](image-url)
1.3.2 An Analysis of Literature on Agile Development

To describe the status of research on agile software development, we conducted a literature search in the ISI Web of Science. We found 719 scientific publications regarding agile software development, published between 1997 and 2009. Of these, 627 were papers in conference proceedings (87%), and 92 were journal articles (13%, including review articles). Figure 1.1 illustrates the trend in publications for conference papers, journal articles and the total. There seems to be a steady growth in the number of publications in total (until 2009).

Table 1.1 The 20 countries with most publications on agile software development

<table>
<thead>
<tr>
<th>Rank</th>
<th>Country</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USA</td>
<td>153</td>
</tr>
<tr>
<td>2</td>
<td>Germany</td>
<td>52</td>
</tr>
<tr>
<td>3</td>
<td>Finland</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>Canada</td>
<td>46</td>
</tr>
<tr>
<td>5</td>
<td>Australia</td>
<td>41</td>
</tr>
<tr>
<td>6</td>
<td>England</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>Ireland</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>Sweden</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>Spain</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>Italy</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>Israel</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>Austria</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>Denmark</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Norway</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>India</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>Chile</td>
<td>11</td>
</tr>
<tr>
<td>17</td>
<td>Iran</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>New Zealand</td>
<td>10</td>
</tr>
<tr>
<td>19</td>
<td>Brazil</td>
<td>9</td>
</tr>
<tr>
<td>20</td>
<td>Peoples Republic of China</td>
<td>9</td>
</tr>
</tbody>
</table>

Note that all publications from 2009 have not been indexed at the time of the search. For example, papers from the Agile 2009 conference were not indexed, and this conference alone had 28 papers in the search results from 2008.

---

1 Search conducted on 22 February 2010, using the term “agile AND software AND development” as topics for subject areas computer science: software engineering, theory and methods and information systems. Document types proceedings paper, review or article. Databases: SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH.
These figures seem to indicate that there is a substantial interest in agile software development amongst research environments.

An examination of the country of origin of the publications, gives the list as in Table 1.1, which states that authors from the USA are by far most frequent in the author lists of publications, followed by Germany and Finland. The total number of articles with authors from Europe among the top 20 countries is 301, about twice that of the USA. Thus, it seems that most agile research is authored in Europe, followed by North America, Oceania and Asia.

Further, we present the institutions that are more frequently occurring in the search, in Table 1.2. The VTT Technical Research Centre of Finland has the highest number of publications, followed by the University of Calgary in Canada, SINTEF in Norway and Technion Israel Institute of Technology. Europe has eleven institutions among the top 20, while North America has four, Oceania three, South America and Asia one each.

Another interesting bibliographic finding in the search is what papers on agile development are most cited, and how much cited they are. In Table 1.3 we list the 20 most cited papers, and we see that the top three have 61, 46 and 33 citations respectively. Further we see that only two of the highly cited articles are published at conferences, papers 2 and 17, which were published at the International Conference on Software Engineering. Of the remaining, nine were published in magazines (IEEE Software, IEEE Computer and Communications of the ACM) and nine were published in primary journals like IEEE Transactions on Software Engineering and the European Journal of Information Systems. 70% of the articles were published in venues we can categorize as mainly software engineering, and 30% we can arguably categorize as mainly information systems.

This analysis shows that journal publications are most important in the field of agile software development, and that the field gathers interest mainly from the software engineering community, but also from the information systems community.

---

2 Some institutions were registered with several names in ISI Web of Science. For Table 1.2, we aggregated numbers for VTT and SINTEF, which were the only institutions among the top 20 with several names.
Table 1.2 The 20 institutions with most publications on agile software development

<table>
<thead>
<tr>
<th>Rank</th>
<th>Institution</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VTT Technical Research Centre of Finland</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>University of Calgary, Canada</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>SINTEF, Norway</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Technion Israel Institute of Technology</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Dundalk Institute of Technology, Ireland</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>University of Technology Sydney, Australia</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>North Carolina State University, USA</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>Blekinge Institute of Technology, Sweden</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>Carnegie Mellon University, USA</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>Helsinki University of Technology, Finland</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>Pontificia Universidad Catolica Valparaiso, Chile</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>Sharif University of Technology, Iran</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>Technical University of Munich, Germany</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>University of Alabama, USA</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>University of Wollongong, Australia</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>Georgia State University, USA</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>University of Aalborg, Denmark</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>University of Cagliari, Italy</td>
<td>5</td>
</tr>
<tr>
<td>19</td>
<td>University of Limerick, Ireland</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>University of Oulu, Finland</td>
<td>5</td>
</tr>
</tbody>
</table>

1.4 Main Themes in this Book

The remaining chapters of the book are organized into three parts: Foundations and Background of Agile Development, Agile Methods in Practice and Principal Challenges and New Frontiers.

1.4.1 Foundations and Background of Agile Development

In addition to this chapter, this part includes two chapters: Chapter 2, “Towards an understanding of the Conceptual Underpinnings of Agile Development Methodologies” by Nerur, Cannon, Balijepally and Bond, and Chapter 3, “Agile Software Development Methods: A Comparative Review” by Abrahamsson, Oza and Sipponen.
<table>
<thead>
<tr>
<th>Rank</th>
<th>Reference</th>
<th>Citations</th>
</tr>
</thead>
</table>
The intellectual foundation of agile methods is explored in Chapter 2, to understand underlying premises for this new paradigm in software development. This understanding will be valuable both in assessing current practice as well as to advance the field.

An overview and analysis of the agile methods that exist is presented in Chapter 3, to make sense of the approaches. Agile methods are described in an analytical framework covering: project management support, life-cycle coverage, type of practical guidance, adaptability in actual use, type of research objectives and existence of empirical evidence for the method. The analysis shows that agile methods support different phases of development, and most methods lack support for project management.

### 1.4.2 Agile Methods in Practice


What is crucial in supporting team collaboration, co-ordination and communication is described in Chapter 4, based over on ten eight years of studies of agile development teams. Story cards and walls are crucial to support collaboration and co-ordination related to project progress. To share information on functional dependencies, agile teams rely on communication and social practices.

The evolution of methods is the topic of Chapter 5, deriving from four empirical studies of software development over a ten-year period. From “internet-speed development”, methods were affected by dramatic changes in the market for software development, which caused disruption of established practices, experimentation and process adaptations. This was followed by consolidations of lessons learned into more mature software development processes.

The customer has a very important role in agile development projects, and this role is under investigation in Chapter 6. Described as a critical, complex and demanding role, the chapter outlines practices that ensure that the customer role works effectively and sustainably in a way that involve the whole development team.

Pair programming is perhaps still the most known practice in agile development, made popular mainly through extreme programming. But even though pair programming has been one of the most researched topics in agile development, Chapter 7 describe the practice as controversial, despite its growth in popularity both among practitioners and academics. One concern has been the productivity of
solo and pair programming, which is still not fully understood. However, the chapter also seeks to explain how and why pair programming can be made productive.

### 1.4.3 Principal Challenges and New Frontiers


Balancing architecture and agility has been a concern of many, including advocates of plan-driven software development. Chapter 8 describes key principles when scaling up agile development projects that require stronger architectural support. Further, the chapter provides guidance to the key principles, and illustrates these with case studies.

Usability of software solutions is a challenge in any development project. Chapter 9 gives an overview of previous studies on usability engineering which has approaches shown to improve usability and reduce probability of revision. Further, the chapter gives an overview of test-driven development of user interfaces, and explains how these two styles of development can be combined to produce testable GUI-based applications by agile teams.

Agile development seem to be exceptionally well-received by practitioners, what is the explanation for this success? This is the central question in Chapter 10, which describe organizational culture as a factor that affects the deployment of agile software development methods. Among the hypothesis posed is that agile methods are incompatible with a hierarchical organizational culture, although agile development is more disciplined than ad hoc development.

Innovation is the topic of Chapter 11, which argues that the idea of a single customer representative must be abandoned because it leads to a too narrow focus, and to a lack of involvement of important stakeholders. Agile development should adopt current thinking on open innovation, which leads to a much broader perspective, involving other business units, customers and partners.
1.5 Conclusion

In this chapter, we have motivated why agile development is an important topic, and further, we have defined and characterized agile software development. In addition, we have given an overview of research on agile software development in two parts:

First, an overview of prior research in the area through a summary of four overview articles, focusing on describing the agile methods, relations to the capability maturity model, status of research on extreme programming and presenting what is known about the benefits and limitations of agile methods. These articles indicate that the field is still in a nascent phase, and that we need more studies of high quality.

Second, an examination of studies of agile development through a literature analysis. We presented an overview of publications by year for conference papers and journal articles, the 20 countries with most publications, the 20 institutions with most publications, and the 20 most cited articles. The main findings from the literature analysis is that there seems to be a substantial interest in agile software development amongst research environments. Further, we find publications from all over the world: primarily Europe and North America, but also in Oceania, Asia and South America. When we rank the institutions after publications, they follow almost the same pattern. From the presentation of the most cited papers, we see that most highly cited articles are published in journals. Further, we see that the field gathers most interest from the software engineering community, but also from the information systems community.

The overview articles that exist in the field summarize research until 2005. The literature analysis identifies more than 500 articles published on agile software development since then, which justifies the need for a contemporary overview of research in the field, which you will find in the remaining chapters of this book.

References