
A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZÜRICH
(Dr. sc. ETH Zürich)

presented by

DENNIZ DÖNMEZ

Dipl.-Ing. Univ. Techische Universität München
BSc London School of Economics and Political Science

born on 20 September 1982

citizen of
Germany and Turkey

accepted on the recommendation of
Prof. Dr. Gudela Grote, examiner
Prof. Dr. Stefano Brusoni, co-examiner

2016
Preface

I could not have accomplished this work without the extraordinary support of many people to whom I am deeply indebted. I am grateful to my thesis advisor, Gudela Grote, above all, for creating a life-changing opportunity, and for bestowing freedom and inexhaustible trust upon me. Encouraging and fostering my work in a field novel to me, she has not only paved the way for my exploration of worlds that had lied beyond my imagination, but also for many discoveries that will continue to shape my professional and personal development. She has deeply influenced me, and I look back on an exciting, exceptional and variegating experience. Also, I have profited immensely from the wisdom, guidance and inspiration of my co-advisor, Stefano Brusoni. Through our discussions I experienced the most comforting acknowledgement, encouraging intellectual challenges, and opportunities to grow. His excitement for science is pleasantly contagious. Last but not least, I wish to extend my special gratitude to Matthew Kerry, who came into my life not only as a collaborator and mentor, but also as a friend and loyal companion in ambitious ventures within and without the academic context.

My work benefited tremendously from the advice, encouragement, as well as critical and constructive reflections of my present and former colleagues, who have displayed extreme dedication and diligence to support my endeavour. I am deeply grateful to them not only for investing immense time and ef-
fort, but more importantly for sharing invaluable experience and
guidance with me. Further, I profited from the support of my
students and student assistants, industry partners, and collabor-
rators who come from different industries and diverse academic
backgrounds. Through their interest in diverse facets of my work
they have helped me understand its value from very different per-
spectives. Many have spent countless hours discussing my ideas
and reviewing my drafts, for which I am extremely grateful. Ev-
ery single person was a teacher to me in the process of learning
important lessons not only with regard to my work, but also in a
much broader sense; with regard to choosing a dedication in life.
Through them I have learnt much about myself and understood
much better my values and my path. Many of them have become
very dear friends who impress and inspire me, and it is with true
excitement that I look forward to continuing our explorations of
ideas and activities in the future.

I also wish to express my gratitude to many who were already
dear to me before my ventures into the social sciences, including
my friends, my family, and Marina, on whose unconditional love
and support I was always able to count. I consider myself blessed
and lucky for having so many wonderful people in my life.

Zürich, February 2016

Denniz Dönmez
Babam için.
Contents

Table of Contents x
List of Tables x
List of Figures xii
Abstract / Résumé / Zusammenfassung xv

Part One 3

1 Introduction 3
    Organisation of the Thesis 6

2 Theoretical Concepts 7
    2.1 Managing Uncertainty 7
    2.2 Stability and Flexibility 10
    2.3 Agile Software Development 13
    2.4 Organisational Routines 18

3 Research Questions 21

4 Methodological Approaches 25
    4.1 Qualitative Study Design 26
# Contents

4.2 Research Context and Sample .................................. 28
4.3 Qualitative Analysis .............................................. 33
4.4 Quantitative Study Design ....................................... 33
4.5 Quantitative Analysis .............................................. 36

5 Summaries of the Scientific Papers 37
5.1 Paper No. 1 ....................................................... 38
5.2 Paper No. 2 ....................................................... 41
5.3 Paper No. 3 ....................................................... 44

6 Discussion of the Findings and Contribution 49
6.1 Implications for Practice ......................................... 52
6.2 Limitations and Future Research ............................... 55

References .......................................................... 59

Part Two: Scientific Papers I, II and III 77

I ................................................................. 77
I.1 Introduction ...................................................... 80
I.2 Theoretical Background ......................................... 82
  I.2.1 Defining Uncertainty ....................................... 82
  I.2.2 Types of Uncertainty ....................................... 84
  I.2.3 Approaches to Uncertainty Management ................ 86
  I.2.4 Agile Uncertainty Management ........................... 87
I.3 Research Approach ............................................... 89
  I.3.1 Sample and Data Collection .............................. 89
  I.3.2 Data Analysis ................................................ 95
I.4 Strategies for the Management of Uncertainty ............... 96
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.4.1 Uncertainty Anticipation</td>
<td>98</td>
</tr>
<tr>
<td>I.4.2 Information Accrual</td>
<td>101</td>
</tr>
<tr>
<td>I.4.3 Solution Inspection</td>
<td>106</td>
</tr>
<tr>
<td>I.4.4 Role-based Coordination</td>
<td>109</td>
</tr>
<tr>
<td>I.5 Discussion</td>
<td>114</td>
</tr>
<tr>
<td>I.5.1 Limitations of the Study</td>
<td>118</td>
</tr>
<tr>
<td>I.5.2 Implications for Industry</td>
<td>119</td>
</tr>
<tr>
<td>I.5.3 Implications for Research</td>
<td>121</td>
</tr>
<tr>
<td>I.6 Conclusion</td>
<td>122</td>
</tr>
<tr>
<td>References</td>
<td>123</td>
</tr>
</tbody>
</table>

## II

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>II.1 Introduction</td>
<td>133</td>
</tr>
<tr>
<td>II.2 Conceptual Background</td>
<td>136</td>
</tr>
<tr>
<td>II.2.1 Organizational Routines</td>
<td>136</td>
</tr>
<tr>
<td>II.2.2 Stability and Flexibility</td>
<td>137</td>
</tr>
<tr>
<td>II.3 Research Setting and Approach</td>
<td>139</td>
</tr>
<tr>
<td>II.3.1 Agile Software Development</td>
<td>140</td>
</tr>
<tr>
<td>II.3.2 Data Collection</td>
<td>143</td>
</tr>
<tr>
<td>II.3.3 Sample</td>
<td>144</td>
</tr>
<tr>
<td>II.3.4 Data Analysis</td>
<td>146</td>
</tr>
<tr>
<td>II.4 Findings</td>
<td>147</td>
</tr>
<tr>
<td>II.4.1 Agile Routines</td>
<td>153</td>
</tr>
<tr>
<td>II.4.2 Establishing Stability while Increasing Flexibility</td>
<td>158</td>
</tr>
<tr>
<td>II.5 A Model of Interdependent Routines for Balancing Stability and Flexibility</td>
<td>169</td>
</tr>
<tr>
<td>II.6 Discussion</td>
<td>176</td>
</tr>
</tbody>
</table>
List of Tables

1  Overview of Agile Software Development Methods 16

2  Overview of qualitative data collected, including companies, teams and project characteristics . . . 29

3  Teams and projects analysed in paper 1 . . . . . . 90

4  Practices for the management of uncertainty . . . 97

5  Profiles of agile software development routines including their characteristics . . . . . . . . . . . . . . . 148

6  Examples for the creation of stability and flexibility via effects of different, interdependent routines . . 151

7  Meta-analytically obtained results for main effects of stability and flexibility on performance . . . . . . 207

8  Meta-analytically obtained results for stability – performance relation . . . . . . . . . . . . . . . . . . . . . 209

9  Meta-analytically obtained results for flexibility – performance relation . . . . . . . . . . . . . . . . . . . . 211

10 List of studies included in the meta analysis including operationalizations of stability and flexibility . 234
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Variable Codebook for Stability – Flexibility Conceptualizations used in the meta-analysis</td>
<td>247</td>
</tr>
<tr>
<td>12</td>
<td>Variable Codebook for Criterion Level used in the meta-analysis</td>
<td>248</td>
</tr>
</tbody>
</table>
## List of Figures

1. A model of interdependent routines showing how they are connected through routine links, which transmit triggers and information feeds. Under pressures for stability and flexibility they support the balancing of both regarding the work process and outcome. .......................... 169

2. Relationship between stability, flexibility and routine protection analysed in paper no. 2 ............ 183

3. Flow chart of the study selection process and criteria for exclusion of studies in the meta-analysis. . 233

4. Meta-analytically obtained results for stability—flexibility bivariate correlations. ....................... 249

5. Results for effects of stability on performance. . . . 250

6. Results for effects of flexibility on performance. . . 251
Abstract

Innovation is a key success factor for organisations. To innovate, organisations must successfully manage the uncertainty that stems from complex and dynamic environments. Toward this goal, the ability to react quickly and adequately to unexpected events is crucial, and requires considerable flexibility. At the same time, stability is a necessary condition to achieve the efficiency required to remain competitive. The quest for a balance between stability and flexibility represents a long standing puzzle in organisation theory.

In this dissertation, I conduct empirical research to study how stability and flexibility are balanced for the management of uncertainty in product development teams. Surrounded by pronounced market dynamics as well as fast changing technologies, development teams face especially high levels of uncertainty. Based on qualitative as well as quantitative analyses, I investigate how teams increase and balance stability and flexibility for the purpose of mitigating threats and seizing opportunities that stem from uncertainties. I present three studies that offer answers to the questions of how development teams manage different uncertainties, how they balance stability and flexibility to do so, and how stability and flexibility are conceptualised in product development. The results offer insight regarding the management of projects in innovative organisations.

In the first study, based on qualitative research in the con-
text of agile software development teams, I elicit practices that support teams in successfully managing distinct types of uncertainty, which prevail in product development projects. Findings suggest that different team practices are guided by several overarching principles. Specifically, teams aim to anticipate uncertainties, and mitigate threats by incrementally accruing information. They also strive to remain open to opportunities that stem from uncertainty, and frequently evaluate intermediary output of their work. Further, a system of dynamic functional team member roles supports development teams to coordinate their efforts.

In the second study, I integrate organizational routine theory, and proceed to explore how the practices employed in software development projects can simultaneously increase flexibility and stability. The study demonstrates how several interdependent routines (defined as patterns of collaborative action) are orchestrated, and receive varying levels of protection against deviations in order to increase stability or flexibility. Both stability and flexibility occur as distinct phenomena; with regard to processes, as well as products of the work. On both levels teams strive to establish stability that is regularly and purposefully interrupted to allow for the necessary flexibility to react to dynamic environments. Against the background of distinguishing product- and process-related phenomena, the study shows empirically how stability and flexibility are mutually reinforcing.

Despite the diverse conceptualizations available on stability and flexibility, few studies have comprehensively studied the interrelationships between these constructs, which are usually researched under the assumption of beneficial effects on performance. The third study provides quantitative support for the effects of stability and flexibility on performance within the wider context of product development. Based on a meta-analysis of the product development literature, the relationship between stability, flexibility and performance is established. The analyses also revealed differences in the employment of the concepts, and offers important clarifications that may guide future research. Based
on the results of these empirical studies, I discuss implications for practitioners and academics that may guide teams and team research with regard to uncertainty management in fields beyond product development.
Résumé

L’innovation est un facteur déterminant pour le succès des organisations. Afin d’innover, les organisations doivent gérer avec succès l’incertitude qui découle des environnements complexes et dynamiques. La capacité de réagir rapidement et adéquatement à des événements inattendus est crucial et requiert une grande flexibilité. Parallèlement, une certaine stabilité est nécessaire afin d’atteindre l’efficacité nécessaire pour rester compétitif. La recherche d’un équilibre entre la stabilité et la flexibilité représente un défi de longue date dans la théorie des organisations.

Dans cette dissertation, je mène des études empiriques afin d’étudier la façon dont la stabilité et la flexibilité sont équilibrées pour la gestion des facteurs aléatoires au sein des équipes de développement de produits. De par leur immersion dans l’environnement dynamique et fluctuant du marché et des technologies qui y sont utilisées, les équipes de développement de produits font souvent face à des facteurs hautement aléatoires. Par le biais d’analyses autant qualitatives que quantitatives, j’étudie la façon dont les équipes procèdent pour trouver un équilibre entre stabilité et flexibilité leur permettant minimiser les risques tout en saisissant les opportunités provenant de facteurs aléatoires. Je présente trois études qui offrent des réponses sur la façon dont les équipes de développement de produits gèrent différentes incertitudes, comment elles équilibrèrent la stabilité et la flexibilité.
de le faire, et comment la stabilité et la flexibilité sont conceptualisées dans le développement de produits. Les réponses à ces problématiques nous fournissent un regard approfondi sur la conception des organisations innovatrices.

Au cours de la première étude, je me penche sur les équipes dynamiques de développement de logiciels et sur les pratiques qui leur permettent de gérer avec succès différents types de facteurs aléatoires. Les résultats suggèrent que les différentes stratégies utilisées par les équipes sont souvent guidées par un certain nombre de principes généraux. De manière plus précise, il s’agit, pour les équipes, d’accumuler graduellement de l’information leur permettant d’anticiper et de palier aux aléas, ainsi qu’aux menaces. De la même manière, les équipes doivent rester ouvertes aux opportunités qui résultent de facteurs aléatoires, et évaluer régulièrement les résultats intermédiaires de leur travail. En outre, une gestion dynamique des rôles de chaque membre de l’équipe permet une coordination plus organique et réactive des efforts fournis par l’équipe.

Dans la deuxième étude, j’intègre la théorie de la routine organisationnelle et je me penche sur la façon dont les stratégies utilisées au sein d’équipes de développement de logiciels permettent d’accroître à la fois la flexibilité et la stabilité. Mon analyse démontre comment plusieurs routines interdépendantes (définies comme modèles d’actions collaboratives) sont orchestrées, et reçoivent différents niveaux de protection contre les déviations afin de maximiser la stabilité ou la flexibilité. La flexibilité et la stabilité se présentent comme deux phénomènes distincts, autant du point de vue des procédés que celui du résultat du travail. Qu’il s’agisse de l’un ou de l’autre, les équipes fonctionnent de telle manière à pouvoir établir une stabilité qui est interrompue régulièrement et intentionnellement, permettant la flexibilité nécessaire pour réagir à l’environnement dynamique dans lequel elles évoluent. L’étude vise à démontrer empiriquement comment la flexibilité et la stabilité se renforcent l’un l’autre.
Malgré la multitude de façons de conceptualiser la flexibilité et la stabilité, très peu d’études ont mis en évidence l’interconnexion entre ces deux principes, qui sont pourtant respectivement étudiés dans le but d’atteindre un rendement optimal. La troisième étude, quantitative, s’intéresse aux effets de la flexibilité et de la stabilité sur le rendement dans un contexte plus large de développement de produits. Me basant sur une approche méta-analytique de la littérature en lien au développement de produits, j’établis le lien entre flexibilité, stabilité et performance. L’étude révèle aussi les différents usages de ces concepts, et apporte des clarifications qui peuvent guider de futures recherches. Ainsi, le résultat de ces recherches empiriques informe et guide les chercheurs et les équipes sur la gestion des aléas au delà des étapes de conception de produits.
Zusammenfassung


In dieser Dissertation präsentiere ich empirische Studien, die sich damit beschäftigen, wie eine Balance zwischen Stabilität und Flexibilität erreicht werden kann, um Unsicherheiten in Produktentwicklungsteams zu managen. Solche Teams sind von ausgeprägten Marktdynamiken und schnellem technologischen Wandel geprägt, welche zu sehr großen Unsicherheiten führen. Basierend auf qualitativen und quantitativen Analysen untersuche ich, wie Stabilität und Flexibilität erhöht und ausgeglichen werden können, um negative Konsequenzen (Bedrohungen) dieser Unsicherheiten abzuwenden und sich präsentierende positive Konsequenzen (Chancen) nutzbar gemacht werden können. Hierzu präsentiere ich drei Studien, die Antworten darauf geben, wie Produktentwicklungsteams mit verschiedenen Unsicherheiten umge-


Zusammenfassung

Part One
Chapter 1

Introduction

Organisations lie at the heart of the endeavours for coordination in all societies. They are shaped by their reactions to the increasing speed of globalisation. One of their fundamental tasks is to react to the enormous pressures that persist in increasingly complex and dynamic market environments, which render innovation and agility vital (Shuen, Feiler, and Teece, 2014). To survive, organisations must not only innovate once but do so repeatedly (Winby and Worley, 2014). Especially technology-oriented firms, which are exposed to fierce competition and high levels of uncertainty, face additional pressures to innovate rapidly and repeatedly, or face failure otherwise (Jelinek and Schoonhoven, 1991).

According to Schumpeterian thought (Schumpeter, 1942), innovation is important beyond individual organisations. It has brought humanity not just the iPhone and Facebook (and, of course, the internet). It is also the driver behind technological progress underlying rising standards of living based on, for example, technologies for the production of safe drinking water and access to financial services. While critical reflections on technological progress and the conditions for its application to the benefit of societies are justified, they cannot be covered by the scope of
Chapter 1. Introduction

this thesis. I shall side with prevailing economic thought (Solow, 1970; Nelson, 2000) and regard innovation as vital and generally desirable in the studies I present.

The studies I conducted address the creation of innovation in product development settings, where teams play an increasingly important role (Edmondson, 2013). New product development (NPD) teams, which are by definition required to innovate (O’Sullivan and Dooley, 2008), need high levels of creativity to explore solutions, technologies, and create novelty to be successful. This context might spark the imagination of colourful workplaces where inventors with wild hair tinker at random. However, organisations tend to reap efficiencies in innovation processes that are much less wild and rather highly structured even in the most creative companies (Hargadon and Sutton, 1997). For example, in a study of a research lab, Sele and Grand (2010) show how teams embed creativity in highly routinised processes to support the formalisation of team coordination and the navigation of solution spaces to create innovation. To do so efficiently, teams must strive to establish stability and control in environments marked by high levels of uncertainty that originate, for example, from rapid changes in market environments, customer preferences, or production technologies (Song and Montoya-Weiss, 2001; Montoya-Weiss and Calantone, 1994). As a result, NPD teams need considerable flexibility to react to such changes and repeatedly produce high quality outputs at the same time. In other words, they have to innovate persistently in ever faster changing, unpredictable environments (Chen, Reilly, and Lynn, 2012).

In my empirical investigations, I turn to the study of software development teams, which are exposed to especially swift and potentially dramatic market changes (Fagerholm et al., 2015). I explore different aspects of their capabilities to innovate under uncertainty, for example, by drawing on the concept of organisational routines—defined as collaborative patterns of action (Becker, 2004)—which have lend themselves to the study of team capabilities (Salvato and Rerup, 2011). Sgourev (2013) has ar-
argued that such capabilities stem from several different aspects. One is the structure of opportunities that present themselves. Another is a teams’ ability to mobilise and align resources in a cooperative effort, for which team coordination is crucial (Burke et al., 2006). In this dissertation, I examine coordination practices that product development teams employ to manage uncertainty, and elicit overarching principles that guide teams to reap beneficial outcomes of uncertainty while avoiding negative consequences in the creation of innovative products.

Managing the demand for continuity in uncertain and highly dynamic environments places the endeavour of product development teams at the heart of organisation research, which is concerned with the “relation between the exploration of new possibilities and the exploitation of old certainties” (March, 1991, p. 71). On the one hand, there is the need for flexibility to react quickly and adequately to uncertainty stemming from changing environments and other sources of dynamism. On the other hand, stability is required for continuity and efficiency, for example, in the form of standardised procedures or decision rules that provide guidance. Whereas the necessity for organisations to manoeuvre between stability and flexibility in order to successfully manage uncertainty is widely acknowledged, many details concerning how and when this is possible remain to be investigated. For example, Eisenhardt, Furr, and Bingham (2010) point to a lack of clarity with regard to different dimensions of uncertainty, as well as to the role of organisational structures in balancing stability and flexibility.

To advance research, this dissertation, in essence, explores coordination practices employed in product development teams that strive to increase flexibility and stability in order to manage and profit from the uncertainties they encounter. The different studies shed light on the questions of how stability and flexibility are simultaneously increased and kept in balance. My studies also investigate how stability and flexibility are conceptualised in the product development context, which is important to de-
rive meaningful implications. The theoretical foundations I draw upon, such as the concept of organisational routines, uncertainty and stability and flexibility, which lie at the heart of innovation, are presented in detail subsequent to this chapter.

Organisation of the Thesis

The dissertation is cumulative and includes three studies prepared for publication in scientific journals. The thesis consists of two parts. Subsequent to the presentation of the conceptual background of the research, part one provides an outline and discussion of the research findings, which are further detailed and presented in full manuscript form in part two.

Part one is organised into six chapters. I provide an introduction to the thesis in chapter 1. In chapter 2 I introduce and connect the central concepts this dissertation draws upon. Next, in chapter 3, I offer an overview of the main research questions addressed by the three empirical studies, which the thesis entails. In chapter 4 I describe the methodological research approach and contextual setting in which I collected data, and explain the employed research designs and samples characteristics. In chapter 5 I provide a summary of each study along with an overview of the main research findings. I discuss these findings in chapter 6, where I also offer reflections on limitations and possible future directions to continue this line of research.

Part two contains the full manuscripts that were developed for publication in scientific journals. It offers detailed analyses and discussions omitted in the overview provided in part one. Note that tables and figures are numbered consecutively throughout the entire thesis.
Chapter 2

Theoretical Concepts

Different approaches have been employed to study how organisations manage uncertainty (Grote, 2009). The consideration of contradictory challenges, such as the demands for stability and flexibility, has been suggested as a promising avenue for research (e.g., Eisenhardt, Furr, and Bingham, 2010). I studied the creation of stability and flexibility in part by employing the lens of organisational routines to look at practices for uncertainty management in agile software development teams. To this end, I highlight the importance of investigating different types of uncertainty, and considering the interdependence of different routines to establish and balance stability and flexibility. This chapter introduces these theoretical concepts in order to prepare the presentation of my contributions.

2.1 Managing Uncertainty

Uncertainty plays a pivotal role in organisations. It permeates almost all aspects of organising, and can be broadly defined as incomplete information. Argote concluded that “there are almost as many definitions of uncertainty as there are treatments of the
subject” (1982, p. 420). Uncertainty is undoubtedly a broad concept. It has attracted the attention of scholars from diverse disciplines including sub-domains of psychology, economics, and philosophy (Hofstede, 1997; Kahneman and Tversky, 1982; Bradley and Drechsler, 2013), to name a few. Many have sought to explore its sources and consequences from a number of different perspectives. In organisation science, uncertainty plays an important role for the design of organisational systems because it affects the outcome of every work process (Perminova, Gustafsson, and Wikström, 2008). However, the conceptualisation of uncertainty is, thus far, imprecise. Whereas the “manifold nature of uncertainty” (Lipshitz and Strauss, 1997) covers aspects as distinct as risk, ambiguity, and equivocality, scholars of uncertainty are often interested in the inability it imposes to predict the future, or more precisely, firm performance (Martin, Gözübüyük, and Becerra, 2013).

In an attempt to define uncertainty, a number of approaches identify incomplete information, or “not knowing for sure” (Argote, 1982; Grote, 2009) at the core of uncertainty. Approaching uncertainty by way of excluding what it is not, it has been contrasted to related but distinct concepts such as risk and ambiguity. Uncertainty entails ambiguity, which is associated with confusion over several possible interpretations (Madsen and Pries-Heje, 2012). While uncertainty is often associated with risk (Ward and Chapman, 2003), a fundamental difference is widely acknowledged that can be traced back to Knight (1921); in his seminal work “Risk, Uncertainty and Profit,” he put forward the idea that risk can be estimated by assigning it a probability, whereas uncertainty refers to immeasurable properties. For instance, organisations in the oil drilling industry can estimate the risk of an earthquake striking at an offshore drilling site by assigning the event a probability. It is close to impossible, however, to account for uncertainties that can manifest as any kind of event affecting the drilling site from the outbreak of epidemic diseases to the strike of a meteorite. Organisations learn only gradually about
uncertainty, as events that occur repeatedly develop into risks. Hence, organisations may not be able to manage uncertainty and risk by the same means (De Bakker, Boonstra, and Wortmann, 2010). Holt (2004, p. 253) summarises this argument as follows:

“uncertainty pertains where probability has no grip […]. Only estimates can be made – things have propensities but no probabilistic direction, and can be followed only creatively, or imaginatively […]. Risk management is focused upon achieving strategic control in order to improve certainty in operation and confidence […].”

Risk is often defined as the product of an event’s probability and its impact (Bannerman, 2008), which in theory is neither positive nor negative in its consequence. Because risk can be assigned expected values, Knight (1921) argued that it invites the possibility for insurance and, thus, convert it into “effective certainties,” whereas this is not possible for uncertainty. Since all participants in the market face uncertainty that cannot be estimated, those managing it best are able to achieve the largest profit. However the consequences of uncertainty can also be negative (Knight, 1921). An organisation’s welfare, therefore, results from exploiting the potential of uncertainty without exposing itself overly to its negative effects (Holt, 2004).

Several researchers argue that uncertainty should be approached as a multi-faceted concept to improve our understanding of their role in relation to a balance of stability and flexibility (e. g., Eisenhardt, Furr, and Bingham, 2010; Tatikonda and Montoya-Weiss, 2001). While various classification schemes of uncertainties exist, some of which are specific to particular industries, the recognition of differences is important to successfully manage uncertainties, and to reap positive outcomes (Jalonen, 2011; Tatikonda and Montoya-Weiss, 2001; Marinho, Sampaio, and Moura, 2014). For example, requirements uncertainty, i. e.,
uncertainty regarding the product features that need to be developed, is a common type of uncertainty in new product development (Nidumolu, 1996). At the same time, other types of uncertainty exist, such as regarding required versus available resource or technical difficulties (Na et al., 2004; MacCormack and Verganti, 2003), which may require different approaches to be managed successfully.

Addressing the question of how to exploit the beneficial aspects of uncertainty while avoiding exposure to its negative consequences, it has been argued that organisations need to carefully manage uncertainty instead of striving for its elimination (Grote, 2004; Ibrahim et al., 2009; Dönmez and Grote, 2015). Successfully doing so, however, requires considerable flexibility, which in turn has implications for organisations with regard to elements of stability to complement or counterbalance it (Grote, 2009).

### 2.2 Stability and Flexibility

Two concepts that lie at the heart of organisation studies are stability and flexibility. Both are complementary, highly desired, and concurrently essential for organisations but difficult to obtain simultaneously (e.g. Manz and Stewart, 1997; Leana and Barry, 2000). Stability enhances predictability and control, whereas flexibility is required for quick and adequate reactions to dynamic environments. Excess stability at the neglect of flexibility hinders innovation (Leonard-Barton, 1992), whereas excess flexibility is likely to cause inefficiencies (Leana and Barry, 2000). While both stand in seemingly direct opposition, they impose contrasting goals upon organisations.

The quest for a solution to contrasting goals has been referred to as the “central paradox of administration” (Thompson, 1967, p. 150). Greve (2007) locates this paradox in the simultaneous demand for dual constructs, which compete for the same resources.
2.2. Stability and Flexibility

within organisations. This implies serious strategic challenges in the form of a need to allocate time and attention, capital, and other production factors between them. Many researchers agree that teams and the organisations they belong to are required to strive for distinct and contrasting qualities, such as innovation and maintenance, speed and accuracy, or reliability and adaptiveness, which lead to organisational tensions (Nosella, Cantarello, and Filippini, 2012). It is widely acknowledged that tensions can only be resolved by a balance between dual concepts (March, 1991). Further, scholars including Smith, Binns, and Tushman (2010) argue that dual concepts must be attended to simultaneously.

Challenges stem from both the detrimental effects of an imbalance, as well as the difficulty to integrate dual concepts independently of each other. In a study of product innovation, for example, Greve (2007) discusses the concepts of exploration and exploitation as constituents of an important incompatibility. Rooted in the organisational learning and strategy literature, exploration and exploitation (March, 1991) are akin to stability and flexibility in constituting two opposed imperatives. Whereas exploration is linked to the search for novelty and flexibility to change, exploitation is vital to ensure the efficiency of operations and stability within organisations.

Several other related streams of literature have proposed different solutions to reconciling incompatibilities, and have put forward a number of conceptualisations. For example, ambidexterity scholars have pointed to the need for structural separation of organisational forces directed at contrasting goals, which can be achieved by different organisational units or even different persons within a team (Benner and Tushman, 2003; Tushman and O’Reilly, 1996). Others have suggested to temporally cycle between phases of stability and flexibility (cf. Gupta, Smith, and Shalley, 2006). Further solutions have been proposed, including semi-structures (Brown and Eisenhardt, 1997), enabling bureaucracy (Adler and Borys, 1996) and flexible routines (Feldman and Pentland, 2003), which are concerned with the design of organisa-
tional structures and are mainly located on macro-organisational levels. They are discussed in the context of the scientific papers I provide below. The papers also elaborate on the on-going debate regarding the possibility to simultaneously achieve stability and flexibility (Farjoun, 2010), as well as the application of different conceptualisations to team level constructs related to coordination practices (Faraj and Xiao, 2006; Klein et al., 2006).

While considerable progress has been made toward solving the challenges of organisational tensions, there is also agreement that much research remains to be done (Benner and Tushman, 2015), for example with regard to the antecedents of balancing forces (Zimmermann, Raisch, and Birkinshaw, 2015; Laureiro-Martínez et al., 2014), as well as the repeated or continued pursuit of such a balance especially in the face of increasing dynamics in which organisations operate (Eisenhardt, Furr, and Bingham, 2010).

In this thesis, I contribute to this endeavour by studying agile software development methods, which were designed for employment in highly dynamic environments. They proclaim flexibility but at the same time cultivate stability. While studies in diverse settings have attributed organisations with the tendency to emphasise stability over flexibility (Benner and Tushman, 2003; Gupta, Smith, and Shalley, 2006; Levinthal and March, 1993), agile software developers tend to prioritise flexibility over stability (Boehm and Turner, 2003). More importantly, they permit my research to delineate different aspects of stability and flexibility. In particular, I draw on Farjoun (2010) to conceptualise stability and flexibility as characteristics of either a process or an outcome of the software development work. As agile software development allows for variation in these characteristics with regard to both processes and outcomes, they have much to offer for the study of organisational tensions imposed by stability and flexibility demands.
2.3 Agile Software Development

In response to the demands for uncertainty management, as well as for the increase of stability and flexibility, organisations have developed diverse mechanisms that include routines and non-routine activities. Depending on institutional capabilities, responses are usually industry-specific. In product development, they are often embedded in project management approaches, which carry suggestions for the management of risk and uncertainty (Moran, 2014). At the same time, project management guidelines shape opportunities to increase stability or flexibility. Calls for more flexibility to mitigate unpredictable events have long existed. Several decades ago, the early project management literature began to voice the increasing importance of flexibility in product development (Takeuchi and Nonaka, 1986). One significant innovation was the spiral model of software development (Boehm, 1988). It marked an important milestone in the rise of “iterative and incremental development techniques,” which emerged in software engineering (Larman and Basili, 2003) and later spilled into other industries and disciplines such as manufacturing (Womack, Jones, and Roos, 1991) and entrepreneurship (Ries, 2011). Perhaps due to advantageous production characteristics, such as a relatively low dependency on capital (in an economic sense), the software industry has been characterised by progressive work arrangements including management innovations. These innovations permitted the industry to repeatedly initiate improvements in project management in response to the need for flexibility in increasingly dynamic environments.

To this day, few industries are as dynamic as the software development industry, where product cycles are very short and further decreasing (consider the frequent software updates our computers, cell phones, and other devices download often automatically). Possibilities to deliver products cheaply and quickly, along with advantages for fast movers, provide strong incentives for software developing companies to engage in races against time.
Arora, Caulkins, and Telang (2006) describe how, as a result, companies routinely sell insufficiently tested products that are not free of software bugs and deliver bug-fixes only later (when security relevant, they are aptly called patches).

The enormous pressures for speed and flexibility in the race for innovation have far-reaching impacts on the entire industry. They are leading an increasing number of organisations to approach the ambidextrous task of creating simultaneous stability and flexibility by the introduction of self-organised teams (Hoda, Noble, and Marshall, 2011). The teams are bestowed with high levels of autonomy to be directed at the establishment of stability via structures for control and team coordination (Barker, 1993). They follow a set of guidelines that were put forward in 2001 by some of the leading figures in software development management, which are known as the Agile Manifesto (cf. Moran, 2015, p. 235). These guidelines incorporate the core values of several approaches to project management commonly referred to as agile software development (ASD), a concept developed in an attempt to respond to the specific demands for stability and flexibility in software development projects (Ramesh, Mohan, and Cao, 2012). Explicitly incorporating both, the Manifesto\(^1\) stresses the production of “working software” (stability), while appraising “responding to change” (flexibility) at the core of its values.

Combining stability and flexibility, ASD has much to offer for researchers as well as industry practitioners, which have extensively adopted agile methods (Melo et al., 2013). While the latter are primarily interested in details regarding the implementation of agile methods (Moran, 2015) and scaling them from team to organisational levels (Dingsøyr and Moe, 2014), research has increasingly focused on empirical questions related to concepts such as team collaboration (Sharp and Robinson, 2008), empowerment (Tessem, 2014), decision making (Moe, Aurum, and Dybå, 2012) and organisational learning (Yanzer Cabral, Ribeiro, and Noll, 

\(^1\)see also http://agilemanifesto.org
2.3. Agile Software Development

2014), as well as ways to exploit agile methods to impact organisational functioning (Vidgen and Wang, 2009; Moe, Dingsoyr, and Dyba, 2010).

Agility is, however, not a straightforward and uniformly understood concept (Fontana et al., 2014). Several distinct agile software development methods have been put forward (an overview is given in Table 1), including eXtreme Programming (XP) (Beck, 2000) and Scrum (Schwaber and Beedle, 2002), which enjoy the highest popularity among them (Fitzgerald, Hartnett, and Conboy, 2006). The methods differ, for example, in their emphasis on technical or managerial aspects of the software development process. Their commonality is a set of “agile principles”, such as “deliver[ing] working software frequently,” or “reflect[ing] on how to become more effective […] and adjust[ing] behaviour accordingly” (cf. Moran, 2015, p. 235), yet Conboy (2009) notes that there is no single agreed-on definition of the agility concept. In their daily development work, pragmatists usually borrow from different agile software development methods, and freely combine their different elements. In the empirical studies I conducted, all teams followed the Scrum method, while some teams complemented it by integrating development techniques from Kanban and XP, such as work-flow limitation (to constrain focus on a small number of tasks), pair programming (where two programmers collaborate by simultaneously sharing one computer), and test-driven development (where quality criteria for the software code are defined prior to writing the code).
### Table 1: An overview of different Agile Software Development Methods (based on Tarhan and Yılmaz, 2014; Dyba and Dingsoyr, 2008)

<table>
<thead>
<tr>
<th>ASD Method</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal</td>
<td>A family of methods emphasizing practices for incremental software development and communication, such as, reflective improvement, access to expert users, and frequent delivery.</td>
<td>(Cockburn, 2004)</td>
</tr>
<tr>
<td>DSDM</td>
<td>A collection of principles, including user involvement, frequent delivery, testing at the end of every project stage, changeability of requirements, and iterative design.</td>
<td>(Stapleton, 1997)</td>
</tr>
<tr>
<td>eXtreme Program-</td>
<td>A collection of engineering practices such as small releases, continuous integration and testing, collective code ownership, and pair programming.</td>
<td>(Beck, 2000)</td>
</tr>
<tr>
<td>(XP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature-Driven</td>
<td>A model-driven approach (i.e. affecting the structure of product specifications) emphasizing the division of work features, and their iterative design and development.</td>
<td>(Palmer and Felsing, 2002)</td>
</tr>
<tr>
<td>Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kanban</td>
<td>A system for scheduling work tasks based on the Toyota just-in-time mechanism, using workflow visualizations and limitations to the number of tasks simultaneously in a workflow.</td>
<td>(Brechner, 2015)</td>
</tr>
<tr>
<td>Lean Software</td>
<td>An adaptation of principles from lean production to software development. Consists of principles: eliminating waste, amplifying learning, deciding as late as possible, delivering as fast as possible, empowering the team, building integrity, and seeing the whole.</td>
<td></td>
</tr>
<tr>
<td>Development</td>
<td></td>
<td>(Poppendieck and Poppendieck, 2003)</td>
</tr>
</tbody>
</table>
Scrum stands out in its focus on project management aspects (Dingsoyr, Dyba, and Abrahamsson, 2008). For example, teams are advised to follow the method’s recommended temporal structure for conducting meetings to plan and reflect on their work and on team collaboration processes. Meetings are prescheduled around development iterations that last no longer than four weeks, and have to be conducted within a stable temporal pattern. Scrum also creates several team member roles and artefacts that facilitate communication and development process visualisation, which are intended to increase workflow transparency. Further emphasised values centre on information sharing within teams and also among team-internal and external project participants. They address fundamental questions of the approach to developing products, including the stance on when, how and by whom to conduct requirements elicitation, software design activities, software code implementation as well as product validation and testing (Estler et al., 2014).

All of these aspects affect team collaboration (Yu and Petter, 2014) and, in some cases, organisational design decisions such as related to team structures (Dingsoyr, Dyba, and Abrahamsson, 2008). The resulting product development processes aim to produce agility (flexibility) but also efficiency (stability) in dynamic environments. While different approaches and theoretical lenses have been applied to investigate the creation and balancing of both (Grote, Kolbe, and Waller, 2012), several scholars have noted the advantage of modelling and analysing product development processes using the lens of organisational routines to derive
important insight (D’Adderio, 2003; Vidgen and Wang, 2009).

2.4 Organisational Routines

Organisational routines, which are defined as “repetitive, recognizable patterns of interdependent actions, carried out by multiple actors” (Feldman and Pentland, 2003), are fundamental to organisations (Grodal, Nelson, and Siino, 2015). They have provided a valuable lens to study work processes in a myriad of collaborative settings, such as the maintenance of organisational core functions (Feldman and Pentland, 2003; Howard-Grenville, 2005), delivery of services (Turner and Rindova, 2012), and knowledge creation and innovation (Hargadon and Sutton, 1997; Salvato, 2009). As such, routines are not only researched in highly standardised contexts, but increasingly so in dynamic environments such as research and development (Sele, forthcoming). Further, they have proved suitable to study team processes to address paradoxical challenges in software development settings, for example, by allowing to capture and analyse the properties of short, collaborative development cycles (D’Adderio, 2003; Vidgen and Wang, 2009).

Routines have been used to describe the dynamic capabilities (Teece and Pisano, 1994) necessary to establish stability and flexibility in organisations (Parmigiani and Howard-Grenville, 2011) and are discussed as embodiments of a balance between stability and flexibility (Tsoukas and Chia, 2002; Turner and Rindova, 2012). Dynamic capabilities are defined as “organizational and strategic routines by which firms achieve new resource configurations” (Eisenhardt and Martin, 2000, p. 1107). While they are seen as higher-level constructs that entail changes to routines (Zollo and Winter, 2002), the capabilities literature “does not explain how behaviours that are clearly patterned and persistent can facilitate the creation of novelty and human creativity” (Salvato and Rerup, 2011, p. 473), as is necessary for innovation.
2.4. Organisational Routines

Hence, to gain a thorough understanding of organisational functioning in this regard, one must turn to the microfoundations and key mechanisms provided by routines and explore their internal dynamics by looking into the “black boxes” used to represent routines (Parmigiani and Howard-Grenville, 2011; Salvato and Rerup, 2011; Spee, Jarzabkowski, and Smets, 2011). This line of research has started to refine the traditional view of routines according to which they are path dependent and inert, i.e. entailing foreseeable and stable patterns (Pentland and Hærem, 2015).

Toward this goal, seminal work by Feldman and Pentland (2003) conceptualised routines as dynamic systems that may change due to exogenous or endogenous causes. Their tendency to change or remain the same continues to occupy researchers (Pentland and Hærem, 2015). Recent literature has shed light on the inner workings of routines by investigating the routine dynamics that highlight tensions between the need for efficient reproduction and the need to change innovatively (D’Adderio, 2014). Routine dynamics have been addressed through conceptualising routines as compositions of performative and ostensive parts, which represent cognitive versus behavioural aspects (Feldman, 2000; Feldman and Pentland, 2003); whereas the ostensive aspect refers to the abstract idea of the routine, the performative aspect refers to the enactment of the routine in specific situations. Reminiscent of structuration theory (Giddens, 1984), where structure and agency are seen as mutually constitutive, Feldman (2000; 2003) borrows terminology from anthropology (Latour, 1986) to argue that the ostensive aspect a) guides the actions of routine participants, and b) is in turn maintained or modified by the performative aspect, which builds the basis for stability and flexibility. To shed light on the details of this process, routine scholars have increasingly delineated how routines operate. For example, Howard-Grenville (2005) explains how routines that are performed flexibly can persist over time, while Zbaracki and Bergen (2010) show how conflicts that are inherent in routines can trigger change. Similarly, Turner and Rindova (2012) show how competing pressures are
balanced by employing multiple ostensive patterns of routines. Recently, scholars have increasingly paid attention to the context in which routines operate, for example, by studying the interactions of multiple routines (Kremser and Schreyögg, forthcoming). In this thesis, I continue the work on routine interactions to derive important insight on the potential of routines to establish and balance stability and flexibility in the process of managing uncertainty.
Chapter 3

Research Questions

Central to this dissertation are questions regarding the management of uncertainty in innovation settings. Based on the underlying assumption that a balance between stability and flexibility is required for successful uncertainty management, I address several related questions. First, I explore what constitutes uncertainty in software development, and elicit practices and principles guiding their management. Second, I investigate how stability and flexibility are simultaneously enhanced toward this goal. Third, I broaden the research scope to the wider context of product development in a meta-analytic quantification of the effects of stability and flexibility on performance.

In the first study, I investigate the creation and balancing of stability and flexibility to manage uncertainty in agile software development teams, where very high levels of uncertainty persist, and teams lack formal guidance to their management (Moran, 2014). Several scholars have stressed the consideration of distinct types of uncertainty (MacCormack and Verganti, 2003; Jalonen, 2011; Carson, Wu, and Moore, 2012), as well as the importance to distinguish between their consequences, which can be either detrimental or beneficial (Bannerman, 2008). Development teams
Chapter 3. Research Questions

need to innovate and, thus, depend on beneficial outcomes to create competitive advantages for their companies (Tatikonda and Montoya-Weiss, 2001). The following questions are addressed in paper no. 1:

**RQ 1a:** What types of uncertainty govern software development projects?

**RQ 1b:** What are the mechanisms employed by agile software development teams to manage different types of uncertainty and their positive versus negative consequences?

In addressing uncertainties, agile software development teams display several characteristic endowments, such as the autonomy to influence team structures. These characteristics are interesting for studying stability and flexibility because they may reveal important insight to the functioning of effectively balancing stability and flexibility. Especially interesting is the software development teams’ ability to alter the processes of innovation in order to optimise productivity. Organisational structures may be modified in response to the demand for stability and flexibility with regard to different qualities, such as the development process or the software product. Software development teams employ a number of organisational routines, which have provided a valuable lens to study stability and flexibility in the past (Parmigiani and Howard-Grenville, 2011). However, research has paid less attention to their contribution for the simultaneous creation of different types of stability and flexibility, in particular in the highly dynamic environments of product development, where pressures to manage uncertainty are high. Moreover, the aspect of possible collective dynamics of several routines has been largely ignored by researchers, yet may be important as it reflects organisational complexities more adequately. In addressing and combining the theoretical concepts introduced above, I investigate the following research questions in paper no. 2:
**RQ 2a:** How are organisational routines employed for the purpose of increasing and balancing stability and flexibility in teams operating in dynamic environments?

**RQ 2b:** How can different routines support the creation of distinct types of stability and flexibility, such as regarding processes versus products?

Further, I am interested in the concepts of stability and flexibility in the wider context of product development. While it is generally agreed that both concepts are vital to the development of novelty, their investigation has taken different forms. For example, researchers have operationalised stability and flexibility at different levels of analyses, within various industrial settings, and in reference to diverse properties, such as team and organisational structures, cognitive factors, or behaviours (e.g., Leana and Barry, 2000; Eisenhardt, Furr, and Bingham, 2010; Laureiro-Martínez et al., 2014; Burke et al., 2006; Bechky and Okhuysen, 2011; Klein et al., 2006). As a result, it is difficult to derive clear implications and advice for practitioners in product development. To consolidate the literature, I systematically review studies to shed light on the contingencies of effects for different conceptualisations, and aggregate quantifications of the impact on performance. The following questions are addressed in paper no. 3:

**RQ 3a:** How are stability and flexibility conceptualised in product development settings?

**RQ 3b:** What are the differences regarding their benefits for team and organisational performance within the product development domain?
Chapter 4

Methodological Approaches

This thesis integrates three studies I conducted to develop theory and enhance understanding regarding the balancing of stability and flexibility as well as the management of uncertainty in product development teams. I applied both qualitative and quantitative methods to understand different aspects regarding how teams achieve and employ stability and flexibility, and their effects on performance.

In the first two studies, I conducted analyses of rich qualitative data from multiple cases. The studies report on practices used to manage uncertainty in the context of agile software development, and on organisational routines employed to produce stability and flexibility. The third study aims to broaden the research scope by a quantitative, meta-analytical investigation and establishes the effects of stability and flexibility in wider product development settings.
4.1 Qualitative Study Design

I chose a qualitative approach to investigate agile software development team settings, because of the richness in opportunities that qualitative research methods provide for the creation of a thorough understanding of underlying concepts (Gioia, Corley, and Hamilton, 2013). The objective of the qualitative researcher differs from that of the quantitative researcher; the latter aims to obtain standardised categories of answers to improve comparability among respondents, and is rather interested in the quantification of a phenomenon and concerned with its generalisability, whereas the former seeks to understand the meaning of certain behaviours (Lareau and Shultz, 1996) in order to “discover what really goes on in particular (technical) communities and [to reveal] subtle but important aspects of work practices” (Easterbrook et al., 2008, p. 301). To shed light on the functioning of collaborative software development, and agile software development in particular, empirical research has received increasing popularity during the last decade (Dingsoyr et al., 2012), as qualitative researchers started to explore collaboration in corporate agile team settings (e.g., Cherry and Robillard, 2008; Sharp and Robinson, 2008; Coman et al., 2014). I continued this line of work by following an ethnographically informed approach (Robinson, Segal, and Sharp, 2007) using field observations and semi-structured, open-ended expert interviews (Miles, Huberman, and Saldaña, 2013) with professional software developers.

My research started with a familiarisation stage to better understand agile software development using different strategies. First, I conducted non-participant observations in two agile software development teams from different companies, with which I had acquired contact through industry partners of the research group. I visited the teams several times per week for a period of three months. Second, I participated in professional training through which software developers become certified as Professional Scrum Masters, a job title for managerial positions in
software developing companies. Third, I spent two months with a third team as a participant observer on the basis of an internship, going to work with the team and undergoing the technical on-boarding process along with two other new team members. I worked with the team on three to four days per week for either a half or a full day, delivering small programming tasks and attending all formal and many informal team meetings. Further, I participated in company-wide knowledge sharing programmes, joint lunches and coffee breaks, and seized the opportunity to freely approach employees on all hierarchical levels within the company. Fourth, I regularly participated in networking and knowledge-sharing events organised by the local Swiss-based agile development community to acquaint myself with currently debated themes. Through workshops and talks, as well as through my exposure to countless discussions, I gained insight to the issues that concern practitioners. Fifth, I subscribed to a number of online services, such as discussion forums and mailing lists, through which I could verify and contrast locally raised issues and concerns with contents produced by a globally active community.

These activities permitted me to gain valuable credibility for my later research, and in particular affected the quality of in-depth interviews and non-participant observations in other teams and companies. Especially during the subsequent research phases involving several large and less informal company settings, I often enjoyed an atmosphere of openness between the study participants and myself during data collection. This was perhaps due to high levels of trust fostered by the expertise I had developed regarding IT development and agile methods as well as our similar educational backgrounds. As a result, I was able to elicit a broad range of information during interviews, as participants shared their opinions and experiences very openly with me. Several even went as far as to disclose private and confidential information such as their intentions of quitting their jobs, which they had not yet shared with their companies.
4.2 Research Context and Sample

Through my exposure to different companies and the agile community during the familiarisation stage, I acquired access to several research sites for the subsequent data collection. None of the initial three teams from the familiarisation stage were part of the later data collection process, however, I maintained infrequent contact with the teams and repeatedly met different team members to discuss the research progress. I also maintained similarly informal contact throughout and beyond the data collection phase with all participating teams and had informal meetings with almost all of them. These meetings helped me to keep track of the projects I had visited and cautioned me to draw unjustified conclusions about their development or performance. They also served as opportunities to discuss and validate intermediary research findings.

During my time in the field, i.e. mainly starting in the summer of 2010, but especially between 2011 and 2014, I was exposed to many activities and events organised by the agile community in Switzerland. This exposure enabled me to obtain the participation of different companies in the research studies. I used purposeful sampling (Adolph, Hall, and Kruchten, 2008) to include companies with different sizes because I expected to find differences in the way they employ agile development methods. Within these companies, I approached teams following a theoretical sampling strategy (Miles, Huberman, and Saldaña, 2013), i.e., I conducted observations and interviews until I had exhausted the teams’ potential to reveal further insight regarding my research questions.

I started data collection based on observations and semi-structured in-depth interviews with a sample of eight teams (83 individuals) acquired from five different companies in Switzerland. The sample comprised of teams that were experienced with agile software development techniques and had been working together
prior to the data collection. Most teams had worked together for several years at turn-over rates typical for the IT industry. As data collection progressed, I could secure three further teams for participation in my research and increase the sample size to 11 teams (103 individuals). I conducted interviews with 40% of the project members, including developers, agile consultants, team leaders and other managers. An overview of the teams and corresponding project characteristics is given in Table 2. Due to the fact that I had already commenced data collection when I realised that the lens of organisational routines provided an excellent fit for my research questions, I extended my interviews to include additional questions, so that interviews could be analysed separately for the different research questions addressed in the studies. To increase reliability, a student researcher with experience in agile software development was involved in part of the data collection and coding process for the first study.

**Table 2:** An overview of companies, teams, project characteristics and data collected (Int.=interviews; Obs.=observations; Doc.=documents and/or further artefacts; Team tenure=time the team was formed before data collection started)

<table>
<thead>
<tr>
<th>Company, team and industry</th>
<th>Project characteristics</th>
<th>Data obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A Team 1</td>
<td>Team members: 12 Team tenure: 2 years Project objective: developing a document archiving system for the bank’s internal use.</td>
<td>Int., Obs.</td>
</tr>
<tr>
<td>Company A Team 2</td>
<td>Team members: 12 Team tenure: 7 years Project objective: development and maintenance of communication technologies internal to the company.</td>
<td>Int., Obs.</td>
</tr>
<tr>
<td>Team</td>
<td>Project characteristics</td>
<td>Data</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| Team 3 | Team members: 10  
Team tenure: 5 years  
Project objective: on an annual basis, releasing updates for a software to test business rule compliance. Team members collaborated closely with their clients internal to the bank. | Int., Obs., Doc. |
| Team 4 | Team members: 12  
Team tenure: 3 months  
Project objective: innovating on existing tools to provide financial services. | Int., Obs., Doc. |
| Team 5 | Team members: 19  
Team tenure: 2 years  
Project objective: development of a product for internal use in the sales department. It was organised into two sub-teams and composed of interdisciplinary team members with diverse types of expertise including several external developers who were contractors with special knowledge about functionality provided by suppliers. | Int., Obs., Doc. |
| Team 6 | Team members: 12  
Team tenure: 1.5 years  
Project objective: development of a software application that consolidates information from several customer relationship management products for internal company use. | Int., Obs., Doc. |
### 4.2. Research Context and Sample

<table>
<thead>
<tr>
<th>Team</th>
<th>Project characteristics</th>
<th>Data</th>
</tr>
</thead>
</table>
| Team 7 | Team members: 6  
Team tenure: almost 1 year  
Project objective: developing a mobile application for end users. The project also served as a pilot venture to create knowledge about new technologies for the company. | Int., Obs. |
| Company D  
A software development and IT services consulting company (less than 500 employees) | Team 8 | Team members: 6  
Team tenure: 2 years  
Project objective: developing control applications that integrate different hardware components for a client company via a mobile interface. | Int., Obs. |
| Company E  
A small company supplying software to the car industry (less than 50 employees in IT) | Team 9 | Team members: 4  
Team tenure: 2 years  
Project objective: developing mobile applications for end users with a focus on navigation. | Int., Doc. |
| Company F  
A small software development company supplying end users (less than 50 employees) | Team 10 | Team members: 3  
Team tenure: 4 years  
Project objective: developing web and mobile consumer applications. The development team works closely together with a larger team of the company that develops a specialised service for the back-end system. | Int. |
<table>
<thead>
<tr>
<th>Team</th>
<th>Project characteristics</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 11</td>
<td><strong>Company G</strong>&lt;br&gt;A small software development company specialising in customised software, mainly for client companies (less than 250 employees)**&lt;br&gt;Team members: 7&lt;br&gt;Team tenure: 5 years&lt;br&gt;Project objective: developing embedded systems (hardware and software) in close collaboration with a partner company (the client) from the building services engineering industry. The project is one of the company’s most stable long-term collaborations.</td>
<td>Int., Obs.</td>
</tr>
<tr>
<td>Team 12</td>
<td>Team members: 9&lt;br&gt;Team tenure: 1.5 years&lt;br&gt;Project objective: development of mobile applications and exploration of new technology for the company.</td>
<td>Int., Participant obs., Doc./Pilot study; data not used</td>
</tr>
<tr>
<td>Team 13</td>
<td>Team members: 11&lt;br&gt;Team tenure: 4 years&lt;br&gt;Project objective: development and customization of security software for about ten client companies.</td>
<td>Int., Obs./Pilot study; data not used</td>
</tr>
<tr>
<td>Team 14</td>
<td><strong>Company H</strong>&lt;br&gt;A medical software development company (less than 250 employees)**&lt;br&gt;Team members: 15&lt;br&gt;Team tenure: 3 months (on this project)&lt;br&gt;Project objective: development of software for clients from the medical engineering industry.</td>
<td>Int., Obs., Doc./Pilot study; data not used</td>
</tr>
</tbody>
</table>
4.3 Qualitative Analysis

As is common for qualitative research, the phases of data collection and analysis overlapped, while focus shifted from spending much time with the software teams to increasingly withdrawing from the field as I started the writing process. During data analyses, I worked with verbatim transcripts of the interviews. Interview coding included reading through the transcripts with the research questions in mind and noting ideas and overarching concepts in a separate spreadsheet. Subsequently, I approached the transcripts separately for each study by assigning relevant codes where appropriate, and extending the code library when necessary. Moving from deductive to inductive coding (Miles, Huberman, and Saldaña, 2013), the initial set of codes captured the concepts I wanted to explore and was extended whenever new concepts emerged during the analysis, at which point I re-coded previously analysed data with the newer, complete list again. In order to make sense of the data, I also referred to field notes and documents, such as project plans, diagrams, photos of planning boards and other artefacts, performance reports, etc., which I had collected during company visits. This process of data triangulation (Adolph, Hall, and Kruchten, 2008) enriched my understanding of the organisational and team processes and structures as well as the projects’ contextual settings. Whenever I felt confused about a particular idea or finding, I strove to exploit the relationships I maintained with the teams to meet repeatedly with the company contacts for clarifications, which also enabled me to verify preliminary analyses and receive updates regarding the projects’ developments.

4.4 Quantitative Study Design

Departing from the insight that stability and flexibility are not uniformly used and might not be understood in the same way
within the broader product development context, i.e. beyond software development, the third study aims to consolidate research and clarify conceptualisations across the literature. To collate findings from a larger body of literature, I applied a quantitative study design with the objective to capture the effects of stability and flexibility on performance in a meta-analysis. Meta-analyses offer the possibility to systematically collect and synthesise evidence from a large body of studies (Card, 2014). Their advantage over performing a narrative review lies in the statistical comparison of effect sizes across a large sample, which leverages the strength of quantitative analyses, such as high reliability and broad generalisability (Borenstein et al., 2011). At the cost of excluding studies (however, systematically) according to pre-defined criteria, meta-analyses provide the additional benefit of focusing the research endeavour on a specific sample (e.g., those studies reporting correlations between performance and stability and flexibility, in my case). Thus, meta-analyses support the summary of a range of studies by a highly systematic approach that yields thorough aggregation and higher objectivity than a narrative review (Hodgkinson and Ford, 2014). For the purpose of conducting a the meta-analysis, I systematically searched eleven major academic research databases (see the methods section of paper 3 on p. 202 for a list) for studies that examine the impact of stability and flexibility on organisational performance. Keyword searches included combinations of the terms “flexibility,” “stability,” and “product development.” In total, 3,557 studies were retrieved and their eligibility assessed according to the following criteria:

1. Studies had to focus on product development settings.

2. Studies had to report on stability and flexibility with regard to organisational or team properties (as opposed to, for example, product or material characteristics.)

3. Studies had to report correlations between stability, flexibility, and performance. Defining the latter, measures were
borrowed from the project management literature. Stability and flexibility were defined in accordance with their introduction above.

The first two criteria led to the exclusion of about two thirds of studies in the initial sample. Details of the data collection process are depicted in Figure 3 (see page 233). To determine eligibility for inclusion in the final sample, I manually screened the remaining studies by reading the titles and abstracts of the studies with the support of a student assistant. Following the independent screening of a sub-sample of 50 studies, Cohen’s kappa for inter-rater agreement was determined at 1, as we identified the exact same studies for inclusion in the final sample. We retained 94 studies for closer inspection and data coding. The discarded studies contained 25% duplicates as well as 58% studies that did not match the focus of investigation or contained no relevant statistical data. I contacted authors via email to acquire studies for 9% of the sample, which were inaccessible via our library subscriptions.

Subsequently, together with the second author of the study, I extracted independent and dependent variable data from the retained 94 studies. We coded part of the studies independently and resolved differences through discussion. Coding was performed according to criteria that we had previously documented in a codebook. The codebook was updated several times to add more precision to our definitions as we developed better formalisations of our initially implicit understanding of the concepts we wished to investigate. The codebook was also used for the training of the student assistant who had supported me in the study elicitation process. In total data was found in 55 of the 94 studies retained for coding. A list of the final sample can be found in Table 10 on p. 234. In addition to extracting data regarding measures of stability, flexibility, and performance, we also recorded a variety of study characteristics, including sample sizes, industry context, study design, and the year and location of data collection.
4.5 Quantitative Analysis

To enhance the study quality, the meta-analytic research process was oriented along the check-list of preferred reporting items for systematic reviews and meta-analyses (PRISMA) published by Moher et al. (2009). With support from the study’s second author, statistical operations were performed using Comprehensive Meta-Analysis (version 3; Borenstein et al., 2014), a software tool, for analyses of correlational data extracted from the literature. To establish the strength of the relationship between the investigated variables, we computed main effect sizes using a random effects model (Card, 2014), which acknowledges variability among true effect sizes across different studies. Further, we performed subgroup analyses under the same assumption to investigate potential moderating effects. Although assumption of random effects produces greater uncertainty (e.g., confidence intervals are wider), it increases generalisability of the obtained results (Rosenthal and DiMatteo, 2001).

Subgroup analyses were performed for different outcome measures in terms of DV unit of analysis and IV conceptualisation. Specifically, three units of analyses of the dependent variable were distinguished; product performance, project performance, and firm performance. Different IV conceptualisations included structural, behavioural, and cognitive aspects of stability and flexibility. Results of the meta-analysis are summarised below and included in full detail, as well as in tabular form, in the third scientific paper (see page 206).
Chapter 5

Summaries of the Scientific Papers

This dissertation builds on several studies, which I have crafted as the first author. I had the lead in the development of three manuscripts for submission to scientific journals, not only with regard to the study design, data collection and analysis, but also in terms of paper design and manuscript writing. In these endeavours, I received support from several co-authors and valuable feedback in the form of discussions with and early reviews from a number of colleagues. All scientific papers were co-authored by Gudela Grote. In addition, the second paper was co-authored by Stefano Brusoni, and the third paper was co-authored by Matthew Kerry. In producing each manuscript, I profited immensely from their collaboration and advice. I also received valuable comments from colleagues and students at the Department of Management, Technology, and Economics, and at international conferences, where I presented intermediary results. Some of these intermediary results are also published in conference proceedings or books (see publication list in the curriculum vitae on page 253). While the scientific journal papers are included in the second part of this thesis, this chapter provides a summary of each to admit a sub-
sequent discussion of the main research findings.

5.1 Paper No. 1

Reference: Dönmez, D., and Grote, G. Two Sides of The Same Coin–How Agile Software Development Teams Manage Uncertainty as Threats and Opportunities. Manuscript invited for Revision and Resubmission to Information and Software Technology

Based on qualitative analyses of data collected in agile software development teams, this paper empirically investigates how teams manage uncertainty, i.e., situations in which they lack important information. It addresses questions regarding the occurrence of uncertainty in different forms in product development, and details practices employed to manage uncertainties based on the assumption that successful uncertainty management requires teams to balance both stability and flexibility.

In the light of high failure rates in software development projects, uncertainty is often associated with the negative outcomes of risk and threats. Its relevance as enabler of learning and innovation tends to be disregarded. As a result, the literature is skewed toward a focus on threats and, consequently, favours strategies for uncertainty minimisation over the management by adequate coping strategies (Bannerman, 2008). The current study addresses this imbalance by discerning uncertainty as entailing potentially negative versus positive consequences, neither of which can be excluded without collecting sufficient and relevant information.

Instead of categorically eliminating uncertainty, as seems to be the objective of many risk management frameworks, this paper presents arguments in favour of exploring coping mechanisms and strategies that are simultaneously flexible and efficient. One
important contribution is the discernment of different types of uncertainty. Three types were derived from prior research and confirmed by a review of the literature; (i) task uncertainty, which refers to incomplete information regarding the ways to solve design and programming problems; (ii) resource uncertainty, which refers to changes in production factors such as the availability of staff, and (iii) requirements uncertainty, which refers to demanded changes in product functionality.

The results of our study indicate that different types of uncertainty demand different mechanisms to cope with threats or to seize opportunities that may arise. The failure to distinguish between different types of uncertainty can lead to their inadequate management, for example, by eliminating potential opportunities. Our data also show that both threat and opportunity are outcomes of ambiguous situations that bear the potential to turn into either. Requirements uncertainty, for example, can transform into difficult technical problems that draw unexpected amounts of resources and may cause costly project delays. Contrary to this, requirements uncertainty can also turn into a highly demanded product feature that is marketable to a larger than envisioned customer base.

The lack of complete information increases the need for team members to share information as it becomes available. Our results show how, through the collective action of several actors, structures for the creation and exchange of information are established that serve to reduce, transfer, or exploit uncertainties. The practices employed to these ends follow underlying principles, which guide teams to refrain from premature action until a classification of uncertainties into either beneficial or harmful becomes possible. When employed purposefully, several practices can serve as powerful tools to successfully manage different types of uncertainty and profit from the opportunities that present themselves.

The paper enhances the empirical body of evidence on strategies for uncertainty management in software development projects
by analysing different practices that are summarised by four overarching principles to address uncertainties through carefully eroding threats while remaining open to arising opportunities; 1. Teams follow the principle of anticipating uncertainty by incorporating it into project plans and remain vigilant to opportunities. Due to the fragility of plans, the process of planning is valued more than the possession of a plan. 2. Through the step-wise accrual of information, uncertainty is managed via the incremental collection of feedback, as well as team-based task analyses and knowledge sharing. 3. The inspection of technical solutions serves as an important test in uncertain market environments, which is backed by prototyping and the parallel creation of alternative solutions. 4. Following the principle of role-based coordination, teams create dynamic, functional roles, integrate stakeholders into the product development process, and engage in task switching to establish both stability and flexibility.

Key findings:

- Uncertainty is mostly regarded as something negative (threat), while ignoring its positive potential (opportunity for innovation).

- Profiting from uncertainty while staving off negative outcomes requires effective uncertainty management based on appropriate levels of stability as well as flexibility.

- To manage uncertainty, teams follow a number of practices, which are associated with four general principles, namely 1) uncertainty anticipation, 2) information accrual, 3) solution inspection, and 4) role-based coordination.
5.2 Paper No. 2


This paper introduces the concepts of stability and flexibility as two distinct, desirable attainments, and investigates how teams in product development settings can increase both at the same time. Using qualitative research methods, the study empirically addresses the question of how teams innovate by drawing on routines in support of the increase and balancing of stability and flexibility. Following Farjounian thought, the idea is adopted that stability and flexibility can fall into two categories; they can be products, outcomes and objectives on the one hand, or they can be processes on the other hand (Farjoun, 2010). Departing from this view, and in combination with the focus of the investigation (stability and flexibility), the paper analyses four resulting categories, namely (i) stability concerning software development processes; (ii) stability concerning the software development outcome or product; (iii) flexibility concerning the software development process; and (iv) flexibility concerning the software development outcome or product.

These categories are investigated through the theoretical lens of organisational routines, which are defined as “repetitive, recognizable patterns of interdependent actions, carried out by multiple actors” (Feldman and Pentland, 2003, p. 95). Specifically, several interdependent routines are used as a vehicle to examine how stability and flexibility are created and increased individually and simultaneously.

Past organisational routines research has used analyses of discrete routines to explain how routines are established, how they change, or how they exert influence on organisational outcomes
Chapter 5. Summaries of the Scientific Papers

(e.g., Feldman, 2000; Howard-Grenville, 2005). Less attention has been paid to the interplay between several routines that co-exist at a given point in time. Also, most research on routines has been carried out in teams where the desired outcome of the work is known, such as in the service sector where customers expect reliability and homogeneity from the routine participants serving meals (Leidner, 1993) or collecting garbage (Turner and Rindova, 2012). In similar settings, researchers were able to identify factors that support the delivery of stable output and maintenance of flexibility by the routines’ participants. One problem, however, arises when the outcome of an endeavour is neither clearly defined nor certain, such as faced by research and development (R&D) and new product development (NPD) teams, which are tasked with the creation of novelty. These teams need to reliably deliver innovative products, oftentimes with project goals changing dramatically over time as they reflect market dynamics.

To expand insight from past routine research, this study explicitly examines the interplay of routines, as well as their role in innovation settings. The results show that, through simultaneously performing different routines, stability and flexibility can be increased independently. In reaction to dynamic project environments, teams can choose among a number of available routines with specific effects on stability and flexibility to balance both. Specifically, this is achieved by protecting the performances of routines in order to establish process stability, or allowing for process flexibility by refraining from protecting the performances. Similarly, outcome stability is achieved by protecting the content of certain routines (i.e., with regard to product features such as through setting development goals), while outcome flexibility stems from the absence of such a protection.

Routines and their varying levels of received protection, in this sense, represent options for actions that routine participants can choose to engage in, or otherwise ignore, depending on project circumstances. The results also suggest that stability and flexibility are not mutually exclusive, but can be increased simultaneously
by a set of interdependent routines. The routines are linked via trigger signals and information flows, and exert mutual impacts regarding their performances. Further, the study explains how routines may be purposefully designed for regular disruption of and reflection on the product development process, permitting opportunities for change. The paper contributes to the literature on routines by an empirical investigation of how stability and flexibility can be increased simultaneously in order to lay the foundation for product innovation in development teams.

Key findings:

- Product development teams use several routines that are interdependent to concurrently create and balance stability and flexibility. The routines are linked through triggers and information flows.

- Interdependent routines are performed upon being triggered either by a pre-defined schedule or based on events. They exchange information, for example via artefacts, which permits performances independently of synchrony.

- Routines are employed to create stability regarding processes and development outcomes, but also to enhance flexibility regarding both, processes and outcomes. Stability and flexibility are achieved by varying the levels of protection with regard to performances and content of the routines.
5.3 Paper No. 3


The third paper complements the previously conducted qualitative studies by a quantitative analysis of the examined concepts of stability and flexibility in a broader context. It aims to shed light on both concepts beyond the previously applied focus on agile software development. The objective of the study is a clarification of the conceptualisation of stability and flexibility in the product development literature, as well as to quantify their relationship and their effects on performance in product development. The prior qualitative analyses were conducted to investigate how stability and flexibility are created and balanced in support of effective uncertainty management. In line with the general tenor in organisational science, beneficial impacts on team and organisational performance has been attributed to both, often implicitly so (Manz and Stewart, 1997). Several studies have measured these benefits (e. g., Tatikonda and Rosenthal, 2000; Worren, Moore, and Cardona, 2002; Eisenhardt and Tabrizi, 1995), however, with mixed and partly contradictory results in terms of both magnitude and direction (positive versus negative) of the effects of stability and flexibility.

The contribution of this study is twofold. First, this study leverages meta-analytical methods to aggregate findings from the product development literature and to quantify the effects of stability and flexibility on performance. Second, the results from the systematic search are seized to produce an overview of diverse conceptualisations employed in the product development literature. The latter bears potential to enable much more systematic and consistent future research.
To our knowledge, this is the first study to perform a comprehensive review of stability and flexibility in product development based on a systematic analysis of the literature using scientific databases to retrieve original research. Major academic databases, including Web of Science, Scopus, Google Scholar, Science Direct, and others (see paper 3 on page 202 for a complete list), were searched using combinations of keywords based on the terms stability, flexibility and product development. The study elicitation process produced $k = 55$ studies ($N = 11,666$). The process is depicted on page 233.

Clarifying the relationship between stability, flexibility, and performance, our analyses suggest that their mutual effects are positive. Specifically, there is a positive relationship between stability ($r = .233; p < .001$) and flexibility ($r = .245; p < .001$) with performance. The results also suggest a positive relationship between stability and flexibility ($r = .18; p < .001$), which was established based on $k = 51$ studies that reported a bivariate correlation. Complete results are presented in the full manuscript in part two of this thesis, including in tabular form in the appendix of the third scientific paper.

Based on qualitative analyses of the study sample, stark differences were observed among the conceptualisations of stability and flexibility. This may in part be due to blurry definitions of stability and flexibility used in the literature. Several researchers offer distinctions within concepts. For example, Larso, Doolen, and Hacker (2009) investigate eight types of flexibility, such as product modification flexibility, production volume flexibility, or labour flexibility (which is defined in terms of task variety and task switching capacity).

Looking at the various conceptualisations researchers employ for measuring stability, flexibility, and performance, it is possible to note systematic differences. Qualitative comparison suggests several overarching types of conceptualisations. The majority of studies refer to stability and flexibility as structural properties,
such as temporal (process) structures or team structures. Other studies define stability and flexibility in terms of behaviours, such as performing quick design changes, or reacting to demand variability. A third category (although the smallest in our sample) characterises stability and flexibility with regard to cognitive properties, such as openness (flexibility) or emotional stability.

Meta-analytic subgroup analyses indicate differences in the effect sizes of stability versus flexibility on performance when employing different operationalisations. Although analyses of a relatively small number of studies ($k < 30$) did not provide significant evidence for differences between effect sizes regarding performance, studies report larger significant effects of stability on project performance, and of flexibility on firm performance. Stronger significance was obtained for effect differences regarding structural versus behavioural aspects of stability and flexibility on performance. There is evidence that behavioural aspects of stability and flexibility influence performance more than structural aspects. This has important implications for product development settings, where teams need to be equipped with both, supportive structures, as well as autonomy to exert beneficial behaviours. Analyses of cognitive aspects indicated the smallest effects, but effect size estimations did not yield significance. Further subgroup analyses, such as according to industry types, were compromised by inconsistent reporting practices in the studies. Details are reported in Table 10 on p. 234.

This study not only contributes to the product development literature by a quantitative aggregation of the effects of stability and flexibility on performance. It also offers an important qualitative contribution in its clarification of different conceptualisations of stability and flexibility. The results may assist researchers in adding precision to their future studies on stability and flexibility in product development and other fields. Implications and limitations of the study are discussed in further detail in the full manuscript.
Key findings:

• Across product development studies, the overall relationship between stability, flexibility and performance is positive.

• There are differences in the relative importance of conceptualisational characteristics of stability and flexibility. Behavioural aspects have a larger impact on performance than structural aspects.

• Considerable heterogeneity among studies was observed in the product development literature, which bears potential to compromise precision and blur research findings obtained with regard to stability and flexibility.
Chapter 6

Discussion of the Findings and Contribution

In this dissertation, I studied how innovation teams balance stability and flexibility to manage uncertainty. With the support of my collaborators and co-authors, I produced several empirical studies detailing how teams approach distinct uncertainties and how they employ multiple organisational routines in this process, as well as how the increase of stability and flexibility affects performance. I addressed these aspects by an investigation in the context of agile product development teams, which provide an ideal study focus because they strive for high levels of flexibility while they simultaneously employ various measures that produce stability. The studies’ findings specify several principles and delineate practices for the management of uncertainty. They also portray a system of interdependent routines for the simultaneous creation of stability and flexibility. Further, findings of a meta-analysis—conducted to systematically review the literature on stability and flexibility in product development—aggregate effects that permit generalisability to wider product development settings. These
studies collectively answer several important questions and suggest implications for research and industry.

*Stability and flexibility of what?* Stability and flexibility have received much interest from researchers. The systematic search in scientific databases revealed several thousand studies—which is problematic at the same time. The vast amount of retrieved literature is likely to be an artefact of ill-defined conceptualisations of stability and flexibility. Indeed, several distinct terms are used in the literature, such as stability and change (Turner and Rindova, 2012), or efficiency and flexibility (Adler, Goldoftas, and Levine, 1999; Eisenhardt, Furr, and Bingham, 2010). Addressing the issue of imprecise and inexplicit operationalisations, primarily in the third scientific paper, but also in the second, I offer clarifications and derive suggesting for researchers to consider what exactly becomes stable or efficient, what is flexible, and what it is that change manifests in. While the second paper departs from Farjoun’s (2010) duality proposition\(^1\), I discuss the explicit classification of stability and flexibility into structural, behavioural and cognitive factors in the third paper. Similar to the need for a delineation of distinct uncertainty-types, researchers need to apply more care when conceptualising stability and flexibility in order to advance our field, and produce reliable precision regarding the conditions and limitations of balancing the underlying paradoxical challenges. The empirical evidence I present speaks favourably to Farjoun’s (2010) consideration of the conceptual possibility to establish stability and flexibility simultaneously. It is based upon the demonstration that stability and flexibility can be seen as complementary when considering their compatible properties, e. g., stable team structures that permit

---

\(^1\)Farjoun (2010) suggested that stability and change constitute a duality rather than a dualism. Regarding two concepts as duality maintains their distinction without committing to a strict antagonism or separation; both can coexist and complement each other. Regarding two concepts as dualism, by contrast, denies them complementary properties and places them at opposed ends of a spectrum. In this line of argument, Farjoun argued that stability and change form a dualism and can be mutually enhancing.
(and are possible reinforced by) flexible behaviours. I am confident that future research will rapidly continue to bring us closer to fully understand the mutual relationship of stability and flexibility, and the various similar dual concepts that impose tensions on organisations.

Acknowledging complexity. Similarly, following the argument that uncertainty is multi-faceted, the acknowledgement of increased levels of complexity is needed to better understand risk and uncertainty (Wallace, Keil, and Rai, 2004). I underline previous calls for applying more precision to distinguish multiple types of uncertainty and argue for the consideration of more detail in uncertainty research. To produce better theory, we need to conduct research with much more attention to the complexities that govern uncertainty (Carson, Wu, and Moore, 2012). The importance of distinguishing multiple types of uncertainty and differences in their effects on performance has often been noted (e.g., MacCormack and Verganti, 2003), yet, it is rarely considered in product development research (Bannerman, 2008). Several literatures, ranging from product development to management, which are occupied with and to some extent confused by the concept of uncertainty, invite the possibility to profit from a single agreed-on conceptual definition as a common ground for future research. As I show in the first scientific paper, teams require different approaches to successfully manage not only different types of uncertainty, but also their dramatically different consequences in the form of opportunities and threats. These differences are not only important in innovation teams, but should be considered by uncertainty scholars in general. More precision and commitment to developing a common language will yield more comparable results and swifter progress to uncertainty scholarship.

Routine-based innovation. Becker and Zirpoli (2009) observe that the concept of routines is generally regarded as opposed to innovation in much of the literature. This is especially true in the product development literature, which tends to highlight issues related to inertia (Barrales-Molina, Montes, and Gutierrez-
Gutierrez, 2015). However, this may be unjustifiably so. Rather, these concerns should caution researchers to consider boundary conditions as well as possible detrimental effects of routinisation on flexibility. While the past decade has produced considerable progress in advancing our knowledge about how routines function, the specification of their relationship with innovation has largely been neglected (Becker and Zirpoli, 2009). Only recently, an increasing number of researchers has started to study this relationship (Sele and Grand, 2010), in part with an explicit focus on the creation and balancing of stability and flexibility (D’Adderio, 2014; Sele and Grand, 2010; Turner and Rindova, 2012). As the body of high quality articles on routines is growing steadily, we may soon be able to explain important details, for example regarding the impact of firm or industry characteristics on the potential to establish routines for innovation (Becker and Zirpoli, 2009). Understanding the way in which several routines might interact to jointly support the balancing of stability and flexibility could provide an important piece in solving the puzzle not only for teams with highly dynamic project goals (Spee, Jarzabkowski, and Smets, forthcoming). While I address this aspect in the second scientific paper, much further research is required to fully understand the potential of routines in innovation settings. There are many unanswered questions remaining, for which routine theory provides a promising avenue to pursue.

6.1 Implications for Practice

Numerous organisations striving for greater agility seek remedy in the adoption of agile development practices. This is reflected in the plentiful trainings and certifications (for example, Professional Scrum Master, Professional Product Owner, etc.) offered by agile software development associations such as the Agile Alliance and Scrum.org as well as their promotion by an increasing number of professional agility consultants. Agile software devel-
6.1. Implications for Practice

Development conferences that attract representatives from research and industry have been growing steadily in the last decade (Dingsøyr et al., 2012). The research I present has important implications for their endeavour.

In line with the confused definitions of agile software development, there is some cloudiness about how to reap the benefits of agile and the implied costs for teams (for example, in terms of commitment, change in communication practices, styles of collaboration, etc.) In particular, I find that merely officially following agile methods does not contribute sufficiently to team functioning if the underlying organisational principles are misunderstood, or worse, if processes are installed and followed blindly. Instead, influences on team performance can be attributed to mechanisms that are independent of agile methods when these methods are uncovered as collections of leadership and coordination practices. For example, role-based coordination and shared leadership, which are discussed in the presented papers, can be detected at the heart of agile methods. While much of the practitioner literature seems ignorant of explanations for collaborative practices or unable to include them, readers are left without clear understanding. As a result, agile software development methods lack clear guidance regarding several important issues that target the management of uncertainty, especially so with regard to opportunity management (Moran, 2014; Dönmez and Grote, 2013).

The results I present suggest that organisational and team structures need to be complemented by adequate behaviours in order to seize their full potential for the creation of stability and flexibility. This carries implications for organisations, which will have to consider the design of employee training and development schemes that reach beyond methodological training provided by agile certifications. The results also imply that practitioners should pay attention to delineate different types of uncertainty within their projects, and map them to possible response options available to their teams, such as routines that aim to increase stability or flexibility. While projects differ substantially, and
universal advice is difficult to establish, the frameworks I present in this dissertation may support practitioners in thinking about underlying mechanisms and overarching objectives. For example, the kind of innovation an organisation strives to accomplish could present an important influence. The focus of my studies was on product development. To successfully develop innovative products, teams must design effective organisational processes that permit flexibility (Hammer and Champy, 2003), for example by adapting routines according to their requirements (Salvato, 2009). From the perspective of evolutionary economics (Nelson and Winter, 1982), the design of routines reflects the organisational environment. In a stable environment (i.e., different from new product development) product requirements are more likely to be stable, and companies will have more incentives to focus on process innovation. This may lead some companies to substitute some employees with technology (or capital, in economic terms) and other, less skilled, cheaper employees, which increases the specificity of the routines (for example, they will be tailored to function efficiently for producing a smaller range of outputs). This has implications for flexibility, but also for stability. A set of more specific, less flexible routines will be less adequate for dealing with drastic environmental changes, at the benefit of being more adequate for specific tasks. Thus, practitioners from diverse organisational settings face different demands for the design of routines, and should consider not only the benefits, but also the costs of maintaining flexibility or stability.

In this sense, my hope is that practitioners across industries—including ones that may be very different from software development—use my findings in ways that may benefit their endeavours. For example, there is much to be learned from the way in which agile teams use high levels of autonomy to create flexibility without necessarily compromising organisational control. The scientific papers partly address issues of leadership and coordination, as well as psychological safety—defined as the shared belief that it is safe to engage in well-intended inter-personal risks
6.2. Limitations and Future Research

(Edmondson and Bohmer, 2001)—which are implicitly addressed by many agile practices and carry beneficial consequences for innovation and other high-performing teams.

6.2 Limitations and Future Research

In this dissertation, I conducted research using different methods to investigate uncertainty management in multiple sites. My approach was empirical and guided by the attempt to leverage the strengths of qualitative as well as quantitative research designs. While I took great care in employing different methods, there remain a number of limitations, based on which I recommend readers to interpret the findings with the appropriate caution.

First, the findings from the two qualitative studies of agile software development teams may be specific to this setting. While agile software developers share several commonalities with product development teams in different industries, I can only speculate about the generalisability of my findings. Instead, I recommend future research to validate if similar mechanisms to manage uncertainty can be discovered in product development in general, and possibly beyond product development teams. For example, service teams such as in high-risk settings must operate quickly and adequately in dynamic environments, and oftentimes employ creativity to solve complex problems. While such teams differ substantially from agile software development teams (Dönmez, Kolbe, and Grote, 2013), they might be able to employ similar practices to some extent as are employed in agile software development to manage uncertainty. Heeager and Rose (2014) note that agile development methods were designed for the development of new software products, for which they are well suited. Looking at different activities, such as software maintenance, the authors note a number of challenges that need to be addressed by tailoring agile methods. For example, more frequent interruptions
and the increased need for documentation have to be accommodated for when tailoring agile methods to software maintenance. A systematic review of the literature by Campanelli and Parreiras (2015) reveals that tailoring agile methods in order to improve the fit between a project’s internal characteristics and external environment is common among software developers. In line with this research, the developers I studied were prone to modify agile development methods and tailoring them to their needs. In conclusion, the possibility to tailor agile methods for purposes that differ considerably from their original intended field raises optimism for the possibility to adopt them also in projects outside the software domain. The conditions and limits of such possibilities remain to be detailed further by future research.

Second, on a similar note, qualitative data were collected in particular settings that share important characteristics. Specifically, I studied small teams that work in a collocated fashion, i.e. most team members were situated in the same work space, which has effects on the quality of their communication and other abilities to coordinate (Hinds and Bailey, 2003). While this limits the generalisability of the findings, much research is being conducted with regard to distributed or virtual software development teams (Ramesh, Mohan, and Cao, 2012), which may shed light on the transferability of practices between both settings.

Third, I have looked at teams facing uncertainty without measuring possible differences in the amounts of uncertainty they faced. While I discussed the pressures to innovate with several teams, which yielded little variation, I have not mapped specific differences in performance to different levels of uncertainty in projects. The relationship between stability, flexibility and performance is explored in the third paper. Yet, details regarding the relationship between innovation and uncertainty as well as between uncertainty and performance, which have been investigated in different settings (e.g., Austin, Devin, and Sullivan, 2012; Hoque, 2004), should further and more systematically be studied with regard to agile software development, and product
6.2. Limitations and Future Research

In conclusion, this dissertation can serve as a starting point for promising future work that continues to unbundle the relationship between stability and flexibility and their combined effects on innovative performance based on the successful management of uncertainty. A number of steps in this direction have been pointed out, including the consideration of different types of uncertainties with regard to project characteristics and the investigation of collaboration practices for their management (paper no. 1). Also, structural team and project properties that foster innovation have been discussed, which are supported by the employment of a set of interdependent routines in order to manage uncertainties by balancing stability and flexibility (paper no. 2). Lastly, the importance of specifying precise conceptualisations of stability and flexibility to reach relevant conclusions was stressed and implications for effects on performance detailed (paper no. 3). Further details will be presented in the full manuscripts contained in part two.
References


Borenstein, M et al. (2014). Comprehensive Meta-Analysis Version 3 [computer software].

References


Kremser, W and G Schreyögg. „The Dynamics of Interrelated Routines: Introducing the Cluster Level (forthcoming)“. In: *Organization Science*.


Moher, David et al. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement.


Yanzer Cabral, Anderson R, Marcelo Blois Ribeiro, and Rodrigo Perozzo Noll (2014). „Knowledge Management in Agile Soft-


Part Two
Two Sides of The Same Coin – How Agile Software Development Teams Manage Uncertainty as Threats and Opportunities

Manuscript invited for revise and resubmit at Information and Software Technology

AUTHORS
Denniz Dönmez, denniz@ethz.ch
Gudela Grote, ggrote@ethz.ch

AFFILIATION & ADDRESS
ETH Zurich
Department of Management, Technology and Economics
Weinbergstr. 56-58, CH-8092 Zurich, Switzerland
Highlights

- Uncertainty (states of incomplete information) is inherent in product development.
- Mostly regarded as threat, uncertainty also carries potential opportunities.
- Successfully managing uncertainty requires collaborative team practices.
- To distinguish opportunities from threats, teams combine different practices.
- Our analysis of agile dev. teams details ten practices and four general principles.

Keywords


Abstract

Context: Uncertainty affects software development projects in many ways, especially so in the fast changing environments that surround high-technology organizations. Capabilities to manage uncertainty can determine the success or failure of entire projects, and even companies. Against the background of high failure rates of software development projects, uncertainty is often seen in conjunction with the negative outcomes of uncontrolled threats. Its relevance as enabler of learning and innovation tends to be disregarded by scholars and practitioners.
Objective: This study extends research that empirically investigates the management of different types of uncertainties. Acknowledging both the possibility for negative and positive consequences of uncertainty, we aim to expand existing knowledge and contribute to improving practices in the field of uncertainty management in software development.

Method: A multiple-case study including field observations and in-depth interviews with 42 professional software developers was performed with a sample of 11 agile software development teams from different companies in Switzerland.

Results: We detail different uncertainty management practices employed to mitigate threats while simultaneously allowing teams to remain open to opportunities. Our contribution is three-fold. First, we establish a taxonomical classification of the uncertainty in software development, which extends beyond often-addressed requirements uncertainty to additionally include uncertainty related to resource availability and to task specifications. Second, against the background of different uncertainty types, we discuss distinct practices and four overarching principles for the management of uncertainty related to strategies for threat avoidance versus opportunity realization. Third, based on these practices and principles, we derive recommendations for software development teams.

Conclusion: We argue that high costs can be incurred when teams fail to distinguish between different types of uncertainty and manage them inadequately as a result. We discuss the implications of our findings for organizational decision makers as well as product development teams.
I.1 Introduction

Uncertainty is one of the fundamental facts of life. It is as ineradicable from business decisions as from those in any other field. (F. H. Knight, 1921, p. 347)

The work of software development teams is pervaded by uncertainty, which affects a vast number of decisions and is highly critical for project performance (Nidumolu, 1996a; Na et al., 2004; Islam, Mouratidis, and Weippl, 2014). Ignoring uncertainty management can lead to increased development costs, low product quality, delayed product delivery and dissatisfied customers (Ibrahim et al., 2009). The criticality of uncertainty management is further increased by the fact that uncertainty is difficult to manage because it can stem from different sources and affect many different aspects of software development (Stamelos and Angelis, 2001). Whereas uncertainty affects decisions that can cause projects and even entire companies to fail (Drury, Conboy, and Power, 2012; Flyvbjerg and Budzier, 2011), uncertainty is also a vital predecessor of innovation (Jalonen, 2011) and no economic profit exists without it (Penrose, 1959). Hence, uncertainty management occupies a critical role especially in innovation projects, such as often seen in software product development, where levels of uncertainty are especially high (Jeschke et al., 2011; Paixao and Souza, 2015). Yet, only a small number of authors acknowledge a “dual nature” of uncertainty (Perminova, Gustafsson, and Wikström, 2008) that bears the possibility for negative as well as positive effects.

Much of the uncertainty management literature has ignored beneficial aspects and has instead argued for the reduction of uncertainty (Kolltveit, Karlsen, and Gronhaug, 2004). For example, authors have discussed the need to reduce uncertainty arising as a consequence of changing requirements (Nidumolu, 1996a; Na et al., 2004; Bannerman, 2008) because of their potentially large
impacts on development projects (Turner, 1990). This approach ignores potential opportunities arising from such changes (Ban-nerman, 2008). Innovation research points to the benefits of developing capabilities to incorporate design changes even at late stages in the project (MacCormack, Verganti, and Iansiti, 2001; Austin, Devin, and Sullivan, 2012). While the avoidance of negative consequences from uncertainty is increasingly addressed in the literature (Shrivastava and Rathod, 2015), there is scant guidance and insufficient insight for innovation teams regarding the question of how to reap positive effects of uncertainty through appropriate uncertainty management. Specifically, little is known about if and how the same collaborative practices employed by teams can bring about positive and negative consequences of uncertainty.

Acknowledging the beneficial effects of collaborative development approaches (Barney et al., 2009), software-developing companies are increasingly trying to become more agile in order to be better equipped in their response to uncertainty and the changes caused by it (Moe, Aurum, and Dybå, 2012). The current trend to increasingly employ agile software development techniques acknowledges the need to manage – as opposed to reduce – uncertainty for the development of innovative software products (Andriopoulos and Lewis, 2009; Moe, Aurum, and Dybå, 2012; Fontana et al., 2015). Yet, uncertainty is not explicitly addressed by agile software development methods, nor has research explored collaboration practices targeted specifically at different uncertainty types (Moran, 2014). As a result, uncertainty management remains one of the largest unresolved challenges for practitioners and organizations (Gregory et al., 2015; Valkenhoef et al., 2011).

To investigate strategies and practices for the management of uncertainty when an agile approach to software development is taken, we conducted an empirical study of teams that explicitly acknowledges both threats and opportunities embedded in uncertainty. We draw on the uncertainty management literature to derive different types of uncertainty and to subsequently investigate
the mechanisms teams use for their management. Our data enable us to examine the relevance of a number of uncertainty management practices for distinct types of uncertainty, and we derive implications for researchers and practitioners from fully acknowledging the two faces of uncertainty. In the following sections, we first explain how uncertainty has been conceptualized and distinguished from risk or threat. We delineate different perspectives on uncertainty and present an investigation of how distinct types of uncertainty are approached in agile software development projects. The contribution of our study is based on the empirical investigation of strategies for uncertainty management, and the solidification of our understanding of uncertainty management as a team-based process of addressing both threat and opportunity embedded in uncertainty. Our findings are structured according to a taxonomy of uncertainty that we link to different strategies for their management. Our results also permit conclusions regarding the fit of agile software development methods for uncertainty management.

I.2 Theoretical Background

I.2.1 Defining Uncertainty

The concept of uncertainty centers on missing information (Kolltveit, Karlsen, and Gronhaug, 2004). However, Argote notes that “there are almost as many definitions of uncertainty as there are treatments of the subject” Argote (1982, p. 420). Their commonality is incomplete information, or “not knowing for sure” (Grote, 2009). Yet, the “manifold nature of uncertainty” (Lipshtitz and Strauss, 1997) covers a range of aspects including risk, ambiguity, and equivocality. Hence, we need to ascertain further qualities of these concepts in order to grasp uncertainty empirically. While it may be difficult to determine exactly what uncertainty can be manifested in, we can move the attempt of a
I.2. Theoretical Background

definition forward by contrasting it with similar concepts. For example, uncertainty entails ambiguity, which is associated with confusion over several possible interpretations (Madsen and Pries-Heje, 2012). Furthermore, uncertainty is often associated with risk (Ward and Chapman, 2003). Scholars acknowledge a fundamental difference between risk and uncertainty, however, that can be traced back to Knight (1921), who put forward the idea that risk can be assigned a probability, whereas uncertainty refers to something that is immeasurable. As a result, uncertainty may not be manageable by the same means as risk (De Bakker, Boonstra, and Wortmann, 2010). Holt (2004, p. 253) summarizes that “uncertainty pertains where probability has no grip”. Following product development researchers, we rely on a conceptualization of uncertainty that encompasses “anything that matters” (Ward and Chapman, 2003, p. 98), defining it as the lack of certainty that leads to a situation in which “potential outcomes and causal forces are not fully understood” (Miller and Lessard, 2001, p. 438).

In divorcing risk from uncertainty, Knight established uncertainty as the underlying reason for corporate profit (based on the argument that risk can be assigned expected values, therefore inviting the possibility for insurance), and he warned that the consequences of uncertainty can also be negative (1921). An organization’s welfare, therefore, results from exploiting the potential of uncertainty without exposing itself overly to its negative effects (Holt, 2004).

A focus on uncertainty rather than risk has been suggested to enhance project management by “providing an important difference in perspective, including, but not limited to, an enhanced focus on opportunity management” (Ward and Chapman, 2003, p. 97). Although neither risk nor uncertainty are necessarily harmful, both are usually associated with negative outcomes (Ward and Chapman, 2003). In order not to give rise to confusion, we avoid the term ‘risk’ in this study, and use the terms ‘threat’ and ‘opportunity’ when we refer to any negative or positive outcomes.
of uncertainty, respectively. By definition, opportunity refers to “a set of circumstances that makes it possible to do something” (New Oxford American Dictionary) and creates or adds value for the customer or end user (Singh, 2001). We use the term opportunity for any positive outcome that is not necessarily envisioned originally but rather realized during the course of a project, such as an unexpectedly discovered business opportunity. In contrast, we use the term threat to refer to any negative outcome – such as project delays – encountered as a potential, i.e., without certainty.

I.2.2 Types of Uncertainty

Recognizing differences within the conceptualization of uncertainty is important for their successful management (Marinho, Sampaio, and Moura, 2014). Various strategies exist to cope with distinct types of uncertainty (Lipshitz and Strauss, 1997). Stemming from diverse sources, different types of uncertainty can hold distinct implications for development projects (Tatikonda and Montoya-Weiss, 2001). At the same time, a number of classification schemes exist. On a general level, one can differentiate between external versus internal sources of uncertainty (Kolltveit, Karlsen, and Gronhaug, 2004). In a recent review of the innovation literature, Jalonen refines this dualism to 18 distinct sources of uncertainty that span eight categories; technological, market, regulatoryinstitutional, socialpolitical, acceptancelegitimacy, managerial, timing, and consequence uncertainty (2011). The authors find that the vast majority of studies focus their investigation on one specific type of uncertainty. In contrast, our study explicitly acknowledges and distinguishes three different types previously discussed elsewhere (Dönmez and Grote, 2015) that are relevant for software projects. In line with the innovation literature, we operationalise uncertainty to include three distinct types that feature prominently in the software development literature and are deemed vital for software development teams. They concern requirements, tasks or activities, and resources.
I.2. Theoretical Background

Requirements uncertainty is the projection of distinct external sources of uncertainty stemming from changes in the market, as well as from regulatory, social, or political changes (Jalonen, 2011). Requirements uncertainty usually funnels into requirements changes, which are communicated through customers or their representatives. Studies of uncertainty that capture demand for product functionality feature prominently in the IS literature (Moynihan, 2000). They conform to conceptualizing requirements uncertainty as incomplete information or ambiguity regarding product functionality (Nidumolu, 1996a).

Task uncertainty comprises incomplete information regarding the best way to perform a task at hand. It can, for example, stem from inexperience or insufficient knowledge concerning the employed technologies (Tatikonda and Montoya-Weiss, 2001). Software developers commonly experience task uncertainty with regard to optimal solutions to novel problems. This is often conceptualized as incomplete information regarding the capabilities and knowledge needed to accomplish a project. In the IS literature, task uncertainty may refer to either inexperience concerning the employed technology or to an insufficient understanding of a problem in order to complete a task (McLeod and Smith, 1996).

Resource uncertainty refers to incalculable means of production that are necessary to complete the work (Ropponen and Lyytinen, 2000). The resources that are necessary to accomplish software development projects, which are subject to uncertainty, include the fundamental production factors of human (Drury, Conboy, and Power, 2012) and financial capital (Souder and Moenaert, 1992). They span the unpredictable or unreliable availability of software development capabilities and necessary infrastructure, such as software licenses, test user accounts, and other means of production.

The software development literature refers frequently to these three types of uncertainty, and considers them to be vital to project success (Nidumolu, 1996b; Maruping, Venkatesh, and
Agarwal, 2009; Faraj and Yan, 2009; Tatikonda and Montoya-Weiss, 2001; Ropponen and Lyytinen, 2000). Threats and opportunities may arise in diverse ways for the different kinds of uncertainties. Software development teams need to be wary of these differences and develop a portfolio of adequate approaches for their exploitation versus mitigation.

I.2.3 Approaches to Uncertainty Management

The research literature has a long history of recommendations for the reduction of uncertainty (Maruping, Venkatesh, and Agarwal, 2009), which is accompanied by attempts to increase control over development processes. While some authors see this as critical for development projects, as uncertainty is “inherent to every aspect of software development” (Ibrahim et al., 2009), no widely accepted general framework has emerged for the management of uncertainty.

More recently, scholars started to emphasize the importance of flexibility in order to manage uncertainty (McBride, 2008). Arguing for the need to react quickly to unforeseen events, this stream of literature acknowledges levels of complexity that deem any attempt for control utopian and, hence, favours agile development approaches.

These two different practices can be attributed to two general strategies to manage uncertainty; coping versus minimizing uncertainty (Grote, 2009). Minimization is aimed at the increase of control and based on strategies to eliminate as much uncertainty as possible. While this seems desirable at first sight, it may be neither possible (e.g., when the source of uncertainty lies beyond the influence of the uncertainty manager), nor most beneficial in important situations (e.g., when an uncertainty turns into an opportunity). Coping strategies, in contrast, do not aim to eliminate uncertainty, but manage it flexibly. This, however, requires increased effort and follows no standardisable procedure.
I.2. Theoretical Background

For example, coping mechanisms may require deferring problems (Bannerman, 2008) while additional information is collected, or decision makers might rely on assumptions or intuition (Lipshitz and Strauss, 1997). To cope with uncertainty means to acknowledge that some uncertainties cannot be eliminated, and furthermore, to recognize that the elimination of certain uncertainties leads to a deprivation of opportunities for innovation (Austin, Devin, and Sullivan, 2012; Sgourev, 2013).

Successful uncertainty management reaches beyond preempting the consequences of threats. It is the attempt to exploit the benefits of uncertainty while avoiding exposure to its negative aspects. Thus, software developers are charged with managing uncertainty in a way that aims at controlling threats while simultaneously remaining receptive for opportunities that may arise from the same sources.

I.2.4 Agile Uncertainty Management

In software development, requirements uncertainty dominate the perception of uncertainty (Nidumolu, 1996b; Ropponen and Lyytinen, 2000). Diverse measures for requirements uncertainty were proposed in order to mitigate resulting threats to product performance (Larman and Basili, 2003; Jun, Qiuzhen, and Qingguo, 2011). For example, the improvement of communication and planning practices are prominent among the propositions for improved robustness of requirements (Maier, Doenmez, and Heppnerle, 2011). However, unsuccessful attempts to protect requirements from changing have spawned alternative approaches that structure development around cyclical phases of step-by-step refinement (Larman and Basili, 2003), which allow development teams to meet unforeseen situations with the possibility to react flexibly when changes occur.

In an attempt to cope with rather than minimize uncertainty, large numbers of software development teams have adopted ag-
ile methods in the last decade (Marchenko and Abrahamsson, 2008). The objective of agile methods is rather overarching in its aim to achieve greater levels of flexibility (Golfarelli, Rizzi, and Turricchia, 2013), which has “become an imperative, not an option” (Lee and Xia, 2010). Although these approaches were initially proposed with rather general aspects of project management in mind and without taking into account the management of uncertainty explicitly, their characteristics fit into some of the broader needs of uncertainty management. Most prominently, the iterative development approaches that agile software development methods draw upon are well suited to address requirements uncertainty (Larman and Basili, 2003; Maruping, Venkatesh, and Agarwal, 2009).

By introducing flexible techniques to react to changes, agile development methods offer an approach that differs starkly from traditional project management (Moe, Dingsoyr, and Dyba, 2010; Magdaleno, Lima Werner, and Araujo, 2012). They explicitly propose mechanisms for team coordination such as iterative planning and empowering developers to self-management in support of coping with uncertainty (Valkenhoef et al., 2011; Maruping, Venkatesh, and Agarwal, 2009; Moe, Aurum, and Dybå, 2012). Of these methods, especially XP (Beck, 2000) and Scrum (Schwaber and Beedle, 2002) are most widely adopted by agile software developers and have been studied predominantly (Dingsoyr et al., 2012). Scrum, for example, takes a managerial rather than technical approach to project management, and emphasizes team autonomy. Investigating relationships among project control, team autonomy, and project performance Maruping, Venkatesh, and Agarwal (2009) found that agile methodology use is most effective in environments with high levels of uncertainty.

Agile software development methods addend to the need for uncertainty management more than classical approaches to software development (Golfarelli, Rizzi, and Turricchia, 2013), however, without providing explicit guidance on how to manage un-
I.3. Research Approach

certainty (Moran, 2014). More importantly, their potential for the management of opportunities arising from uncertainty has largely remained under-studied. On the one hand, reaping this potential requires increased understanding for how decision makers distinguish threat from opportunity. On the other hand, it is important to investigate if and how threat and opportunity can be managed using the same mechanisms for uncertainty management. These questions are important as they affect the core activities of development projects. To better understand the mechanisms for uncertainty management both in terms of avoiding threats and capitalizing on opportunities, we designed a qualitative study of agile software development teams outlined in the following section.

I.3  Research Approach

In order to capture how teams manage different types of uncertainty, we performed a multiple-case study (Yin, 2013) based on observations and semi-structured interviews with professional software developers across different research sites. Multiple-case studies allow to identify patterns in organizational practices (Eisenhardt and Graebner, 2007). They are well suited to examine contemporary corporate phenomena, such as agile development techniques, using multiple data sources (Yin, 2013) and triangulate information gathered in their natural context (Runeson and Hoest, 2009).

I.3.1  Sample and Data Collection

Our data comprise rich field observations and 42 face-to-face interviews with professionals from 11 agile software development teams. The teams belong to five Swiss software-developing companies from different industries that operate globally. Details are
summarized in Table 1. Interviews ranged from 20 to 90 minutes in length. They were audio-recorded and transcribed verbatim as soon as possible after they had been conducted. During data collection, we visited each team at several points in time over the course of their projects, and had the opportunity to meet most of the team members informally. We also had the possibility to talk with different internal and external stakeholders of the software development projects to discuss the research progress, verify insights and clarify our understanding where necessary. Data collection was stopped when theoretical saturation was reached (Miles, Huberman, and Saldaña, 2013) and no additional insights were gained.

Table 3: Overview of teams and project profiles

<table>
<thead>
<tr>
<th>Team</th>
<th>Industry</th>
<th>Project</th>
<th>Team Size</th>
<th>Interview participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Software Development</td>
<td>The project is run in close collaboration with a client in the ventilating and air-conditioning industry for whom an embedded system of smart devices is constantly enhanced. The project is one of company A’s most stable collaboration and has been ongoing for more than 5 years with different team members working on it</td>
<td>7</td>
<td>2 interviews: 1 team manager (Scrum Master) and 2 developer.</td>
</tr>
</tbody>
</table>
### Table 3 (continued): Overview of teams and project profiles

<table>
<thead>
<tr>
<th>Team</th>
<th>Industry</th>
<th>Project</th>
<th>Team Size</th>
<th>Interview participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Telecom-</td>
<td>The project had been set up two years before data collection in order to develop an application for customer order management of future product offerings. The team consists of several external developers who are contractors with special knowledge about functionality provided by suppliers. The client is internal to the company.</td>
<td>19 team members</td>
<td>5 interviews: 1 team manager (Scrum Master) and 4 developers, one of which later became a second Scrum Master.</td>
</tr>
<tr>
<td>C</td>
<td>Finance</td>
<td>This project was started two years before data collection with the objective to develop an archiving system for documents that need to be stored and retrieved digitally across the company’s various locations. The client is internal to the financial institution.</td>
<td>12 team members</td>
<td>4 interviews: 1 team manager (Product Owner) and 3 developers.</td>
</tr>
<tr>
<td>D</td>
<td>Finance</td>
<td>The project serves the development and continuous improvement of new company communication technologies. It started seven years prior to data collection, with two product release cycles per year.</td>
<td>12 team members</td>
<td>5 interviews: 2 team managers (Scrum Master and Product Owner) and 3 developers.</td>
</tr>
</tbody>
</table>
Table 3 (continued): Overview of teams and project profiles

<table>
<thead>
<tr>
<th>Team</th>
<th>Industry</th>
<th>Project</th>
<th>Team Size</th>
<th>Interview participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Insurance</td>
<td>The project was started 1.5 years prior to data collection with the aim to develop a software application that consolidates information from several customer relationship management products for internal company use.</td>
<td>12 members</td>
<td>5 interviews: 1 team manager (Product Owner) and 3 developers. We also interviewed 1 person responsible for IT development processes in the company.</td>
</tr>
<tr>
<td>F</td>
<td>Insurance</td>
<td>This project had two objectives. One was the development of a mobile application for end users, and the second was the creation of knowledge about new technologies that the company aims to exploit in the future. The project was started less than one year before data collection.</td>
<td>5 members and 1 Scrum coach</td>
<td>4 interviews: 1 team manager (Product Owner), 2 developers and the Scrum coach.</td>
</tr>
<tr>
<td>G</td>
<td>Finance</td>
<td>For the previous five years, this project annually released improved versions of a software that is used by other software products of the bank to test their compliance with business rules. Team members regularly work in close collaboration with their clients, who are from different departments of the company.</td>
<td>10 members</td>
<td>4 interviews: 2 team managers (Scrum Master and Product Owner) and 2 developers.</td>
</tr>
</tbody>
</table>
Table 3 (continued): Overview of teams and project profiles

<table>
<thead>
<tr>
<th>Team</th>
<th>Industry</th>
<th>Project</th>
<th>Team Size</th>
<th>Interview participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Finance</td>
<td>The project has the objective of implementing enhancements to existing financial tools in the bank. It had been ongoing for only 3 months at the time of interviewing, and the team was still in the process of setting up agile development practices.</td>
<td>12</td>
<td>3 interviews: 1 team manager (Scrum Master), 1 external stakeholder (acting as Scrum coach) and 1 developer.</td>
</tr>
<tr>
<td>I</td>
<td>Software Dev. and IT Consulting</td>
<td>This project was set up two years prior to data collection to develop an application as ordered by the client, which integrates control over 54 different hardware products via a mobile interface.</td>
<td>6</td>
<td>4 interviews: 1 team manager (Product Owner) and 3 developers.</td>
</tr>
<tr>
<td>J</td>
<td>Software Development</td>
<td>This project involves the development, deployment and operation of a server back-end for a web and mobile consumer application, which integrates several services. It has been ongoing for two years. The development team works closely together with a larger team of the company that develops a specialized service for the back-end system.</td>
<td>3</td>
<td>3 interviews: 1 team manager (Scrum Master) and 2 developers.</td>
</tr>
<tr>
<td>K</td>
<td>Automotive</td>
<td>The objective of this project is to create smartphone applications and end-user software for integrated systems. The focus of the interviewed team was on navigation and routing software.</td>
<td>4</td>
<td>3 interviews: 1 team manager (Scrum Master) and 2 developers.</td>
</tr>
</tbody>
</table>
We strove to conduct interviews with individuals holding as many different functions as possible, following a self-imposed policy to always include at least one person with managerial tasks and two team members with distinct software development tasks. In each team, the first author conducted formal interviews with software developers and team leaders. The collection of several interviews per team allowed to triangulate situations from the perspectives of different team members, and increased the quality of our data (Merriam, 2014).

To guide our interview questions, we adopted a theoretical framework consistent with the literature on uncertainty management that suggests that multiple uncertainty types govern project work. We used a semi-structured interview guideline to cover several types of uncertainty in the interviews; however, during the interviews the first author also followed interesting clues and probed for details whenever he felt this would contribute to a more thorough understanding. Team members were asked how their teams had dealt with uncertainty in different situations in the past and were encouraged to share examples for incidents that had either positive or negative outcomes. This approach was based on probing for critical incidents (Flanagan, 1954), but allowed for flexibility when we felt this was appropriate and encouraged participants to freely talk about situations their teams had experienced. Interview participants were informed about our goal to investigate the management of various types of uncertainties that are faced in software development projects, and were subsequently asked about a number of situations in which they or their teams experienced resolved or unresolved problems or difficulties of any kind. We also asked for specific examples of experiences that had occurred a) recently and b) during the course of the entire current project. In each case, we probed for further information in order to obtain a better understanding of the situation, including descriptions of relevant information such as preceding events or reactions by stakeholders. In some cases, we were able to exploit this knowledge in subsequent interviews in
I.3. Research Approach

order to capture different views of issues that the teams had experienced. This allowed us to construct and analyse situations based on descriptions from different perspectives.

Apart from general questions related to uncertainty management, the exploration of how the teams dealt with different uncertainties, namely regarding requirements, tasks and resources, guided our data collection. Following general questions (e.g., regarding the status and progress of the project), we would ask specific uncertainty-related questions such as, “Has there ever been a situation in which your team was unsure about the best way to complete a task?” or “Do you remember a situation in which the availability of team members was uncertain?” In most cases, we had to clarify the concepts we wanted to study by examples or, otherwise, use vocabulary from a software development context in order to correctly comprehend our questions. We further inquired about the consequences and subsequent actions taken.

As we commenced to systematically analyse the data, we increasingly let clues that we developed influence subsequent data collection by extending our interview guideline and including further interesting aspects as far as this did not compromise the time to cover our initial set of questions regarding the management of uncertainty. This allowed us to understand the teamwork context more fully and helped us to develop interesting insight regarding the potential and boundary of uncertainty management.

I.3.2 Data Analysis

We performed data analysis in an iterative fashion, oscillating between data collection and analysis as we repeatedly returned to the field for additional observations and interviews. In order to make sense of our data, we began by reading through the field notes and interview transcripts that were produced after the field visits, mindful to our initial questions: To what extent are the different types of uncertainty faced by software developers, and how
do teams deal with these uncertainties? How is information collected, analysed and used in order to manage uncertainties? Are uncertainties viewed and managed as threats and/or as opportunities? We concluded that we had reached theoretical saturation when the answers to our questions were largely consistent across interview partners, and additional interviews ceased to provide new information. Whenever we felt a lack of understanding of our data, we strove to revisit the teams for clarification.

In support of our systematic analysis, we relied on HyperResearch, a software tool for qualitative data analysis, as well as spreadsheets in which we recorded interim results. We used these tools to develop a coding scheme for our analysis. To increase the reliability of the coding process, a student researcher with experience in agile software development was involved in part of the data collection and data analysis. Only aware of our research questions, he coded part of the interview data independently from the authors. The aim of this collaboration was to produce a coding scheme that captured uncertainty management practices in the development teams. Differences were resolved in clarifying discussions with the first author until a common coding scheme displayed high congruity. The coding scheme was extended when we gained new insight, and was subsequently used to revisit any previously coded transcript again with the complete list of codes.

I.4 Strategies for the Management of Uncertainty

We identified different uncertainty management practices applied by agile software development teams, and grouped them according to four general principles. The principles were developed to structure the observed practices based on their similarities in terms of objectives or involved activities. An overview is provided in Table I.4. Some of the practices reflected the fact that teams organized their work in line with agile software development methods,
I.4. Strategies for the Management of Uncertainty

which centre on iterative and incremental development. As these methods address the management of uncertainty only implicitly and do not provide suggestions for managing opportunities, teams had to complement and adapt the suggested practices in order to meet specific demands of uncertainty management. All teams in our study faced several situations in which uncertainty emerged with regard to requirements, tasks, or resources. Sometimes these turned into opportunities, and sometimes they produced threats. Teams employed different practices depending on the specific situation, which we detail below.

Table 4: Practices for the management of uncertainty

<table>
<thead>
<tr>
<th>Principle</th>
<th>Practice</th>
<th>Description of practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncertainty</td>
<td>1. Incorporating uncertainty in plans</td>
<td>Acknowledging that changes will occur, teams try to anticipate and focus on product requirements least expected to change, yet remain flexible to adapt their plans.</td>
</tr>
<tr>
<td></td>
<td>2. Developing vigilance</td>
<td>Team members strive to remain alert to opportunities that present themselves</td>
</tr>
<tr>
<td>Information actual</td>
<td>3. Incremental feedback</td>
<td>Team members frequently collect feedback from colleagues and external project stakeholders.</td>
</tr>
<tr>
<td></td>
<td>4. Team based task analysis</td>
<td>Team members collectively analyze requirements before they create and plan tasks to fulfill them.</td>
</tr>
<tr>
<td></td>
<td>5. Knowledge sharing</td>
<td>Knowledge management structures are established to manage resource uncertainty;</td>
</tr>
<tr>
<td>Solution inspection</td>
<td>6. Prototyping</td>
<td>Preliminary versions of the final product foster discussions on functionality.</td>
</tr>
</tbody>
</table>
Table 4 (continued): Practices for the management of uncertainty

<table>
<thead>
<tr>
<th>Principle</th>
<th>Practice</th>
<th>Description of practice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7. Creating alternatives</td>
<td>Developers strive to explore several solutions in parallel.</td>
</tr>
<tr>
<td>Role-based coordination</td>
<td>8. Creating functional roles</td>
<td>Team member roles are created temporally in order to handle unexpected events efficiently.</td>
</tr>
<tr>
<td></td>
<td>9. Stakeholder integration</td>
<td>Teams value close collaboration with suppliers and clients and design decision structures accordingly.</td>
</tr>
<tr>
<td></td>
<td>10. Task Switching</td>
<td>Teams aim to create structures that permit developers to flexibly distribute tasks to freed resources.</td>
</tr>
</tbody>
</table>

I.4.1 Uncertainty Anticipation

Planning with Uncertainty

The teams we studied engaged in frequent planning activities to produce just enough short-term clarity and guidance regarding necessary steps toward a development goal, and tried to accommodate for events that require a change of the produced plans as much as this was possible. Whereas such events could be impedimental or opportunistic, re-planning was required in both cases. Re-planning was, thus, a real option over which teams seemed not surprised; they almost anticipated the possibility for produced plans to be short-lived: “[We don’t] specify everything from the beginning and [...] sometimes that business idea translates into something at the end that you wouldn’t expect in the beginning, and it’s often for the better”, (Product Owner, Team I).
In our data, the discovery of opportunities was most often related to newly arising business possibilities based on added functionality that was not included in the initial list of product features. For example, some teams realized opportunities based on change requests that became profitable only when cross-sold to other customers, but would not have been worth pursuing otherwise. Teams had to rely on their assessment of the market value for requested product features. This happened in Team G, which extended their product’s functionality against specifications for the given software release after receiving a request for additional functionality by one of many customers. This altered the features initially intended for the subsequent product version, which were allocated to development iterations on an annual basis. In this case, betting on the change proved profitable. Yet it incorporated large uncertainties because the accuracy of effort and market value estimations was not known. In other words, it was unknown in the beginning if any potentially developed product feature, including one discovered to provide new a business opportunity, turned out to be a threat to available resources, requiring more time or delivering less market value than expected.

Teams often faced the pressure to react flexibly and adjust continuously. This also meant remaining open to alternatives that could either become necessary or more attractive. In one reported incident, a new customer approached the team manager of Team C with a feature request, which changed the market values of the planned features and reshuffled the priorities of the requirements list. Although unexpected, the customer’s demand increased the value of a feature that was low on the list of priorities and not expected to be completed within the year’s software release. Developing of this feature was discovered to add value for other existing customers, and caused an increase in the project’s budget that the team or project leader had not expected initially. This example elucidates that it is important and desirable to be able to suspend decisions until more information becomes available. The pursuit of this emerging opportunity would not have
existed if the product functionality (i.e., the project scope) had been specified unchangeably at the outset of the development.

**Developing Vigilance**

Vigilance, a state of alertness to potential scenarios, emerged in our data as one of the most valuable assets in turning uncertainties into opportunities. Development teams need to remain alert to all types of potential uncertainties and their outcomes, as it is not clear in early stages of a project what a specific uncertainty may develop into. This was summarized by developers as follows: “...throughout the year, there are, I would say, incidents, for example licensing problems [...] or positive events, they also occur. Like a project that has a change request, an add-on [...] which also means, they bring money”, (Product Owner, Team G).

In developing a mindset that is alert to technical and business opportunities, team members were vigilant to any shortcuts to the project goal. Sometimes, this required that the project goal be modified into a more profitable one. Oftentimes, this included substituting originally planned product functionality by similar functionality. For example, one team decided to make changes to the design of the project setup; the original product was split in two parts upon realizing that it would be easier to work separately on different parts. Tasks were distributed between two sub-teams, who then realized that the separate product parts they worked on needed some communication interface. While the separation of the project brought a number of uncertainties regarding requirements and resources, the team estimated it produced less task uncertainty. More importantly, this was turned into a business opportunity that was discovered by the software developers: “when we split our projects up into different projects internally in the company, and we had to have an API [application programming interface] from one to the other, and so we found “Well if we do it like this, we can sell the API, just the bare bones technical
thing as a second product.” The developers thought of something and it turned into a product right away and we found a customer [for it]” (developer, Team J).

Here, an opportunity was realized when the separation of tasks led to the idea to offer added functionality not only internally, but also to offer it as added product value to existing and new customers.

The vigilance team members developed is important for the pursuit of opportunities, but it is also important because it helped teams to manage threats proactively. For example, one developer described how she remained vigilant for threats when dealing with customer service requests: “When someone says: The server is down, then I can have a look myself and check if the server is really down and [...] see if I can find out why it is down. Because this shouldn’t be the case. Theses sort of things. But then it could be a user comes and says he doesn’t have permission for something, but according to security I see all necessary rights, then this sounds rather like a bug to me, so I would take a look at that, what could cause that. But before I spend more than half an hour on that I say: OK, I can’t fix it myself, so I pass it on” (developer, Team M). Based on her experience, the developer gauged the possibility for an error to be caused by a software bug that requires further attention. Her statement describes the importance to remain vigilant in order to distinguish minor problems from threats that will draw further resources.

I.4.2 Information Accrual

Incremental Feedback

Strongly influenced by the project management framework Scrum (Schwaber and Beedle, 2002), teams followed a number of principles that are geared toward the management of requirements uncertainty. Frequent feedback from within the team, as well as
from external project stakeholders on incremental development outcomes, was seen as very important. Teams developed specific communication practices for the giving and receiving of feedback, many of which relied on informal and ad-hoc conversations. In addition, teams structured much of their product specific feedback with the help of project tracking tools such as Jira, where progress was documented. Some teams invited their clients to access this documentation in an attempt to involve them closely in team-internal communication. Clients were linked to the team either through team leaders, or connected directly with developers who needed clients’ input. They were mostly absent from the “shop floor” where developers worked but required to be present at meetings several times per month, and development teams demanded their daily availability via email and telephone. Developers sought feedback from their team members on a daily basis. They clarified technical details with co-workers and discussed requirements with team leaders. One developer commented: “you get a lot [of] earlier feedback [so] that you can react and […] adapt and discuss and change” (developer, Team K).

Development iterations served as larger time frames within which developers could collect feedback on incremental improvements of the software product from external project stakeholders. As more information regarding requirements specifications and possible solutions accumulated over time, uncertainty remained latent as either threat or opportunity, and was subsequently subjected to strategic action: “...we stop time and again and look together with the client: what did we do? What else do we need to do? Are we on the right track? Then we continue, step by step” (developer, Team A).

Teams seemed to invite rather than disapprove of customer change requests, which were explicitly encouraged as long as they were brought forward in the time between two development iterations, or otherwise buffered through product backlogs, which were maintained to permit work without interruptions during development iterations. Using short development iterations was viewed
as beneficial as it allowed team members to collect feedback more frequently, and thus enabled them to accrue additional information to reduce uncertainty: “At some point we had like a change in the requirements, which was pretty disruptive and we had to recalculate our requirements, [and repeat] the Sprint [iteration]. This shouldn’t happen of course. [...] having the shorter iterations helps you with doing that” (developer, Team J).

**Team-based Task Analysis**

Software developers often faced situations in which they were not sure about how to solve a problem in the best possible way. Underlying reasons for this included ill-defined problems or a developer lacking experience with regard to the used technology. We found that teams faced many situations in which it was unclear whether a particular envisioned solution would be feasible. Teams performed collective meetings to further investigate the task at hand, in order to acquire the required insight for a meaningful decision-making process about how to proceed. Oftentimes two or more team members were delegated to these task investigations, depending on their knowledge and experience. They jointly investigated for a fixed duration of time, “[...] to create more knowledge, so that we can estimate [the task size] more comfortably. It’s important that it is time-boxed. For example, that we take four hours, and then we stop. Because with these investigations you can burn days. But that doesn’t get you anywhere. You have to time-box them; that’s an important point.” (Scrum coach, Team F).

This pooling of resources in order to investigate possible solutions had the effect that the teams would eventually come to know not only how to proceed, but also how much effort a certain task might require. Estimations of task sizes, which were almost always performed collectively by the entire teams, were mentioned as the single most important practice regarding the
reduction of task uncertainty. Some teams held special meetings
dedicated to the estimation of batches of tasks, and usually pooled
knowledge from several developers in an attempt to reach more
accurate estimates. Teams performed estimation meetings not
only at the start of each iteration, but they also spent much time
on corrections of task size estimations whenever they acquired
new information. This accrual of information in an iterative fash-
on enabled teams to apply changes to the development project
on a continuous basis.

While the accuracy of estimations benefited from the involve-
ment of several team members, the downside was that more time
needed to be spent on estimations. Although teams tried to limit
the duration of estimation meetings, developers were troubled
when regularly exceeding these limits: “[the estimation meeting]
should take one hour [but] everything is so unclear, it’s gonna
take two hours, two and a half hours, and that always annoys
me” (developer, Team J).

Conservative estimates were used to buffer uncertainty, as it
became clear that task sizes could turn out to differ starkly from
original estimations. One developer explained: “most of the time
we try to be a bit more generous in terms of how much time we es-
timate so that we are sure that we’re not going to need more time”
(developer, Team C). As a result, actual teams aimed at situa-
tions in which actual effort turned out to be less than estimated.
Sometimes this effort dropped by magnitudes when developers
found elegant solutions to problems that were initially expected
to be much more complex. In these cases, conservative estima-
tions yielded unexpected benefits in the form of saved resources.
However, the opposite can occur as well when the complexity of a
task is underestimated. Team A experienced a case where the ini-
tial task size estimation had to be corrected upwards repeatedly
as unexpected dependencies were discovered. Eventually the task
size grew three times larger than its original estimate, requiring
the attention of a sub-team of four developers instead of a single
person. In this case, task uncertainty developed into a real threat
that imposed vast unexpected costs. To manage task uncertainty, teams learned to keep task sizes as small as possible in attempts to contain negative consequences. In addition, splitting up tasks to smaller ones increased the precision of estimations and reduced the necessity for revising them.

**Knowledge Sharing**

Structures for knowledge sharing were employed in all teams. They supported the management of resource uncertainty and made workflows more robust. Teams with broad ranges of tasks and high degrees of specialization felt that their diversely distributed expertise hampered collaboration practices such as task analysis and effort estimation. To counteract this, they strove to foster the systematic sharing of knowledge regarding different aspects such as employed technologies or decisions about solutions to problems. On the one hand, this included knowledge about task specifications: “we discussed [the tasks] and tried to formulate them as precisely as possible so that everybody had the same understanding” (developer, Team F).

On the other hand, teams strove to maintain shared capabilities so that the work on a task could continue independently of individual developers: “what if, when I am sick, for example or I leave the company or I change roles. [We wanted to] separate the task from the person in order to create a pool with all the knowledge” (developer, Team H).

Some teams conducted regular knowledge exchange meetings, for instance in the form of mini seminars on current technology use. Pair programming techniques, i.e. the employment of two developers per computer who work in collaboration on a single task in order to learn from each other and increase code quality, was used extensively in one team. After the completion of a task, pairs were regularly broken up and recomposed with different team members so that task specific knowledge would spread
through the entire team.

I.4.3 Solution Inspection

Prototyping

Prototyping is much valued, because it helps to focus on important product features and reassures developers. Addressing the threat of inefficient resource deployment toward features deviating from the customers’ vision, the use of early prototyping in the teams studied supported to some extent the exploration of options while focusing discussions with customers on functionality. At the same time, prototyping was described to us as an important instrument for the exploration of opportunities, because it stimulates meaningful customer feedback. One team manager explained: “So from this meeting we were able to somehow come up with a prototype, just because we wanted to test the feasibility of some solutions [...] we wanted to actually test and see if some ideas might be realized or not. So this was the intent of the prototype, then we used it also to convince the sponsor. [...] people start to imagine how they can do business with that and then they’ll give you much more input, and much more concrete” (Product Owner, Team C).

However, there are also drawbacks of early prototyping, which are linked to the communication of functionality. A prototype is by definition a preliminary version that is inferior to the envisioned product feature. While prototypes provide customers with a better understanding for the development in progress, they trigger early feedback and steer decision-making. This worked well, when the development team and its customers had a shared vision of the final product. When the developers envisioned a technical solution that could not be fully expressed and represented in a prototype, the benefits of early feedback broke down. In one situation, team A experienced a situation in which developers saw future product flexibility threatened by implementing features they
I.4. Strategies for the Management of Uncertainty

had discussed with their customer. They argued for changes in the product architecture, which were costly, but would allow – in their view – improved opportunities for future development. They were not able to convince the customer, who did not see the benefit and did not approve of the extra cost. The developers then decided to push ahead their solution at their own cost, risking that the extra hours would not be paid eventually. After implementing a prototype of the suggested solution, the team was eventually able to convince the customer of the benefit, which had previously been impossible. A developer noted that one needs to have more time to experiment on one’s own, an activity which was not provided for during the short development iterations. Eventually, the team was successful in convincing the customer to cover their extra cost, which they had been prepared to accept at their own risk. Prototyping, in this example, favoured short-term customer orientation that was detrimental to product quality in the long run.

Creating Alternatives

The deliberate pursuit of solutions that are later abandoned can lead to significant advantage. Pursuing multiple solutions simultaneously served software developers to increase flexibility and offered possibilities to defer decisions regarding which option to exclude and which one to keep developing to later points in time: “we had in fact two strategies or two options in mind from a technical point of view, which we were able to go and the one we decided for was so far the right solution. […] It’s ... you’re more flexible, you can be more innovative” (developer, Team K). In this case, an opportunity could be realized when the option that turned out as inferior was abandoned at the cost of having started development of both in order to keep options open. However, this strategy, as well as the reliance on task backlogs to avoid developer idleness required redundancies of skills within the team to some extent. It also required the ability to transfer available tasks
to the next free developer, which must be preceded by adequate knowledge of production and product details despite high levels of complexity.

Developers, therefore, need high levels of freedom to explore as well as adequate tools allowing them to convince customers of deviations from original plans. This can be beneficial when different technical solutions are more feasible to realize or more readily accessible. Developers from Team A explained their frustration after failing to agree with a customer representative to a deviation from a technical solution. Several members of the team had grown convinced that an alternative solution would be superior, but they failed to convince their customer until they produced a prototype at their own cost and concluded “sometimes it’s difficult to convince the customer if you can’t show them something” (developer, Team A). The example highlights the importance of trust in ones abilities and the development of intuition in order to spot opportunities. Risk taking and failure tolerance, thus, play an important role for the exploration that permits opportunities. Teams not only need to tolerate uncertainty and its potential drawbacks, but also operate in environments that provide the psychological safety and team support necessary for effective uncertainty management. This becomes clear from the statement of a developer describing his trust in the team’s ability to solve any type of problem without necessarily knowing how in the beginning: “I’ve never been involved in something that failed so drastically that we didn’t find a solution to it […] there has never been a problem that has actually amounted to become something. It’s quite funny, because when I think about it there’s always, we always somehow find [a solution], you know, whether it’s a workaround, or it may just be fixing the symptom, but not the real root cause. We’ve still somehow always managed” (developer, team I).
I.4.4 Role-based Coordination

Creating Functional Roles

In processes that draw on consensus-based decision-making, agile software development teams create their tasks as well as their roles in a self-organizing manner. Teams created roles upon realizing the need for them, for example when recurring customer calls were handled by a new customer service role, the occupants of which rotated on a bi-weekly basis. Other roles were temporally limited, such as when an emergency task required the formation of a sub-team with an explicit leadership role, which was retired upon returning to non-emergency development work. Yet other roles were created not in reaction to events, but in anticipation of problems or aspiration of uninterrupted development work, e.g., through a rotating but permanent role for specific aspects of project documentation.

Some teams used practices to concentrate tasks of a similar uncertain nature to dedicated functional roles within the project. For example, Team G faced regular service requests from a large number of customers, which could mean anything between 20 minutes and several days of work depending on the reported issue. After observing the insufficient coordination with which the team was dealing with incoming customer requests, it was decided that one person would be responsible for all requests on a bi-weekly rotation scheme. As a consequence, participants reported that everybody else on the team could follow their work plan without interruptions, thus, not facing the uncertainty of having emergencies require the complete reorganization of their daily tasks.

As much as the freedom to design and re-design roles can increase efficiency, it also bears the potential for conflict. For example, development teams in larger companies were challenged to adopt team role structures inherent in agile development methods.
The guidelines of agile methods instruct teams to abandon job titles in an attempt to facilitate shared responsibility and flat hierarchies that support collaboration. However, bureaucratic structures in large companies preserved pre-existing job titles (such as ‘solution architect’ or ‘requirements engineer’) based on unique specializations of team members. Role creation became an exercise in combining both structures. As a result, many of the created roles were either tailored to specific types of team members or designed to exclude team members from taking a role in order to preserve scarce team capacities for critical development work.

In the face of uncertainties that required decision-making, teams referred to informal leadership structures based on the roles they had created. The occupation of leadership roles depended strongly on a team member’s tenure, as well as their expertise related to the project. In shared decision-making processes, the opinions of team members with specific knowledge, for example regarding specific market environments, customers, or technologies, received more weight than the opinions of less experienced team members. Senior team members were assigned leadership roles sometimes without seeking or even accepting them. One developer complained: “I have been with the team for a long time now. For me now the pressure increased, because I should take the role, even though I don’t know everything. I am not the same person [as the senior colleague who had just left]. So for me the pressure has increased, because many questions come to me, and I am expected to make more decisions, you know” (developer, Team B).

While the establishment of roles aims at enabling agility through decoupling work tasks from individuals, change can be undermined when the system becomes inert and roles do not lend themselves to temporality. This means that some roles, once created, are sticky, i.e. difficult to retire and decouple from individuals who have acquired specialized knowledge through filling them. Our data show that individuals regularly had difficulties resign-
ing from a role for various reasons such as specialization, trust, or conscientiousness. For example, one developer from team G who was no longer occupying the rotating customer service role explained that customers still approached her even months later in the role she no longer occupied, because she used to be most familiar with their issues and customers refused to approach the current role occupant: “[they] often come to me because they know me, but they could as well write to the mailbox. [...] I was just the one who happened to know them best and who also knew the problem best, so I still end up doing it even it’s not part of my job any longer” (developer, Team G).

High degrees of specialization and expert knowledge did impede team members from switching roles when the familiarization with a new technology or business context was required to do so. In several cases, team members reported that they continued performing the tasks of a role they formerly occupied for reasons of team and external stakeholder convenience. Personal bonds also made it difficult to discard a role, especially when trust was formed between external stakeholders, such as suppliers and customers and the contact persons within the team.

Another reason for the stickiness of roles was that team members felt responsible for their work and tended to maintain caring for its proper execution out of conscientiousness. For example, team members who had been formerly responsible for the facilitation of a project planning meeting kept attending subsequent planning meetings after they had officially handed over responsibility to their successors.

Given the implicit and sometimes uninvited assignment of roles, as well as the stickiness of roles, developers faced challenges regarding role related coordination activities. While role structures were created to support uncertainty management, these challenges threatened to inhibit flexibility and thus impede uncertainty management.
Stakeholder Integration

Project teams tried to manage requirements uncertainty especially with regard to changes or ambiguities threatening to cause more work through integrating diverse stakeholders closely into the software development process. Frequent communication with stakeholders was regarded as important because it was perceived to significantly improve decision-making. To some extent, this shifted responsibility, e.g. for product design decisions, away from the development team and toward clients. For instance, a developer noted: “in the old [non-agile development] style waterfall system, you get a screenshot, you implement it, you don’t understand it directly, you just do what’s on the screenshot, and afterwards the customer says ‘Well, that’s not what I like’” (developer, Team K). In contrast to this practice, some teams integrated clients to a degree where work would not continue as long as no decision was made to resolve requirements uncertainties, thus eliminating the threat of investing effort in the development of unapproved details: “we have a lot of external dependencies. There are many [tasks] we cannot continue because we just need to clarify. So I have to ask: is that correct that way? Or how is the status? And inquire a bit and […] communication, a lot of communication, that’s what I mean” (developer, Team B).

Teams desired direct contact between developers and external stakeholders, especially customers. At the same time, this created tensions that were described as unwelcome dependencies. Acknowledging the necessity for frequent contact with stakeholders, these dependencies were seen as necessary on the one hand, but detrimental to efficiency on the other, which tended to trouble teams. Attempts to resolve them resulted in a negotiation between adopting external parties’ schedules and working styles, and imposing the team’s schedules and working styles on stakeholders. One team, for example, persuaded several of their clients to adopt the planning board that plays a central role in their work in order to synchronize and discuss future tasks. Others shared
access to online collaboration tools with their clients. In our data, there was a lot of complaint that suppliers that were not “agile” did not understand the timing of events and the role of iteration cycles, which prescribed frequent deadlines. The importance of adhering to them could not always be communicated clearly and was ignored by suppliers, threatening to disrupt project schedules and the pace of development work. As a remedy, teams strove to establish very close integration of project stakeholders.

### Task Switching

Switching from a current task to another, or taking over a task from a colleague let teams uphold development work when interruptions occurred. Technical difficulties and work-flow disruptions were frequently experienced when input was not available from external project stakeholders, e.g., when deliveries of work from contractors or suppliers were late or were not of sufficient quality so that they needed clarification or rework. One developer noted the importance of a task backlog that holds additional items to be completed during a development iteration to which one can return in order to avoid idleness when waiting: “I mean that almost always occurs in any project, that you need to wait for something. I mean, otherwise everything is planned, I mean, that doesn’t happen that everything is planned out in a way that you don’t need to wait [...] it happens really all the time in the best projects. In the worst projects, it’s, it just happens a lot more and basically I think the trick for fixing that is just to have ... be open minded enough to work on something else and not get yourself ... be stopped in the process, I mean, ... But of course you need to have a backlog ... that’s actionable to some degree and makes sense” (developer, Team I). Another developer described how, if switching to another task while awaiting specific information is not possible, inefficiencies and re-work may result: “we couldn’t get information about the error handling, if there was a false input, how to react. So I had the uncertainty that
I have to... should I throw an exception and stop everything or should I return null values, I don’t know, and I couldn’t get the information from the customer. So I had to choose one, one way, and of course it was the wrong one (laughs) and I had to change it later” (developer, Team K).

The importance of managing task uncertainty and task interdependencies surfaced particularly in situations, when task switching was difficult. One team reported experiencing the threat that developers became idle because of the team’s dependency on database specialists within their company. The specialists repeatedly failed to deliver what the team ordered. In particular, lack of requested infrastructure threatened to delay the project several times. The problem was linked to the fact that there was not one individual in the database team responsible, but each time the next available employee processed incoming requests. The team leader negotiated a proposal with the database department according to which one database administrator would be integrated in his project team. This had the effect of creating of a novel business model; henceforth database specialists could be hired individually by development teams to process their entire requests. This supported more effective communication and the elimination of misunderstood requests, as one specialist processed all requests. It also allowed turning a threat into the opportunity of improving the production process by a newly created customized service.

I.5 Discussion

In this study, we emphasize that two important and desirable outcomes of uncertainty management are mitigating threats and fostering opportunities. Detailing different approaches available to these ends, we investigate software development team practices. While agile development teams do not explicitly receive guidance for the management of uncertainty (Moran, 2014), they employ
uncertainty management practices that are partly informed by agile software development methods. In particular, agile methods provide structures that support the ability to respond to uncertainty by fostering capabilities for collaborating with customers, iterative work, team learning, and improving the software development process (Vidgen and Wang, 2009).

Presenting examples for approaches to uncertainty management that assume both the possibilities for positive and negative outcomes, we document the benefits and challenges of practices employed by software development teams. Our results include ten practices (see Table 2) that span four overarching principles; 1) uncertainty anticipation, which relates to the attempt to prepare for different possible scenarios regarding the future development of uncertainties and remain vigilant to profit from opportunities; 2) information accrual in an incremental fashion, which lies at the heart of uncertainty management, as uncertainty gradually turns into opportunity or is unmasked as threat; 3) the frequent evaluation and inspection of solutions, which in our data increased the efficiency of discussions on multiple possible solutions and provided a basis for discussions among developers and customers, and 4) role-based coordination, which integrates team members and external project stakeholders on the basis of practices for establishing temporary, dynamic, and de-individualized roles to accomplish tasks.

Proactively managing different types of uncertainty. The identified principles function conjointly to provide teams with effective practices to manage different types of uncertainty. For example, task uncertainty was addressed by teams through the design of manageable work packages that can be subjected to team based task analyses. Accomplishing small tasks in an iterative fashion and integrating customer feedback on prototypes supported the management of requirement uncertainty, while distributing tasks across several roles often resulted in opportunities to create supplementary product features with better insight into technologies and market environments. Considering different types of uncer-
tainty and their potential development is therefore vital.

Ignoring the difference between uncertainties leading to threat versus uncertainties leading to opportunity decreases innovative potential and gives rise to a fixation on threat. While we do not intend to say that threats should be accepted, we extend a warning regarding the premature elimination of related uncertainties. Threat mitigation, for example, in the form of avoiding costly investments into later changing product requirements, is important. Yet, there are situations in which design changes even at late stages in the development cycle can carry benefits (Austin and Devin, 2009). We warn, therefore, that decreasing flexibility in favour of a focus on control (minimizing uncertainty) implies the cost of forgone opportunities.

Our analysis suggests that scepticism toward uncertainty dominates even in agile software development settings, where we would expect to see teams rely on the exploitation of opportunities to innovate. We find that some developers are alert to opportunities, but most are more alert to threats – above all the threat of failing to meet rapidly changing market demands.

Given that most developers we interviewed associated uncertainty with threats, while team and project leaders were generally more open to consider opportunities as outcome of uncertainty, our concern is that much innovative potential remains unrealized. This finding is in line with observations by Austin, Devin, and Sullivan (2012), which illustrate the cost-driven, as opposed to opportunity-seeking, behaviour of product developers. Whereas the types of behaviour we have witnessed may be influenced by individual preferences regarding uncertainty avoidance versus uncertainty tolerance, we see a need for teams to create structures that incentivise exploratory activities while shielding opportunity explorers from detrimental outcomes. To support this, it is advisable for teams to cultivate psychological safety, which is defined as a shared belief in the safety for team members to engage in behaviour that exposes them or their team to uncertainty without
I.5. Discussion

risking to be penalized (Edmondson, 1999). Creating a climate of psychological safety, in which it is safe to speak up with ideas, concerns, and mistakes is beneficial for learning and collaboration and critical to innovation in product development teams (Edmondson and Nembhard, 2009). Organizational leaders should not only aim to support team psychological safety, but actively encourage exploration by reinforcing vigilance of team members. Leadership researchers point to the effectiveness of rewards for innovation (Oke, Munshi, and Walumbwa, 2009), although this may be challenging in the face of hierarchical structures (or rather, their absence) observed in agile teams.

Threats and opportunities in self-managing teams. Agile software development teams rely on high levels of autonomy (Hoda, Noble, and Marshall, 2011), which they use in order to influence collaboration practices. Extending earlier accounts for effective coordination in teams that point to role-based coordination (Valentine and Edmondson, 2015), our findings document a dynamic system of de-individualized roles in software development. In the face of insufficient guidance regarding coordination, agile software developers create role-based coordination structures. Dynamic functional roles (i.e., changing informal within-team job descriptions) were used by scrutinizing the creation, modification and maintenance of team member roles in support of adaptation to varying and unstable project goals. There are several ways in which dynamic role allocation influences the management of uncertainty.

In contrast to work environments with standardized output, the goals of agile software development projects are neither known at the outset, nor stable; they emerge slowly and are subject to frequent variation. As a result, development teams continuously aim to forge their supporting organizational structures by tailoring functional roles to match the development project’s dynamic requirements. The dynamism of organizational structures results directly from the creation and retirement of roles for specific work tasks, which are fundamentally different from each other, for ex-
ample, in terms of managerial stint or temporal nature.

When teams are able to employ dynamic roles, they can increase team efficiency by intelligently allocating individual capacities. Our results contain several examples for situations in which teams reap the benefits of dynamic role-based coordination, however, we also note some of the drawbacks. Specifically, the stickiness of roles (i.e., the difficulty that occurs when switching or retiring a role) holds potentially detrimental implications for team coordination in situations where roles are intended to be dynamic, but this is undermined by established team structures. When roles are intended to be dynamic and flexible by design, but do not conform to this characteristic in practice, teams will be hindered in their productivity and use their resources inefficiently. Recognizing specific characteristics of such inefficiencies may equip teams with the potential to act upon them. For example, it seems that role-based coordination issues can go unnoticed because individuals are not aware of their systematic occurrence, while their mere discussion within teams would initiate possible solution strategies.

I.5.1 Limitations of the Study

While seizing the opportunities that qualitative methods offer, our research approach carries limitations that are connected to sample size and difficulties regarding the generalisability of findings. In particular, we draw on observations and interviews with a limited number of agile software development teams in the Swiss software industry. Agile software development teams are becoming increasingly popular in and beyond software-developing companies, yet, they operate under special conditions; for example with regard to decision latitude and team autonomy. We strive to mitigate the limitations by methodologically triangulating data from several diverse companies that operate globally. This strengthens our confidence that our sample is representative for research and development contexts spanning the wider
software industry. The management of uncertainty plays a vital role in projects across a variety of industries and organizational settings (Song and Montoya-Weiss, 2001; Bonabeau, Bodick, and Armstrong, 2008; Perminova, Gustafsson, and Wikström, 2008), and the implications of our results may be applicable beyond agile software development projects.

I.5.2 Implications for Industry

The importance of risk and uncertainty in software development projects is reflected by the large number of risk management frameworks that have been proposed. Yet, critics note that none of them has become a generally accepted standard (Jiang, 2014), nor have they been integrated well into development projects (Islam, Mouratidis, and Weippl, 2014). For example, the most widely adopted methods for agile software development, Scrum and XP, do not address the management of risk or uncertainty (Moran, 2014). A lack of formal guidance for practitioners results from this, which is compensated by experience-based practices (Coyle and Conboy, 2009).

In this study, we investigate principles and practices that may provide guidance by broadening the notion of risk management and shifting the focus toward uncertainty. We highlight several distinct types of uncertainty. Based on our empirical results, we suggest that practitioners should not only be aware of these differences but also consider diverse options to approach them. Systematic and continuous attention to uncertainties should best be structured along the different categories of uncertainty in order to enable teams to formulate adequate responses. This does not mean their early elimination, as we have stressed. Innovation depends on being open to unexpected events (Austin, Devin, and Sullivan, 2012), therefore uncertainty should be iteratively evaluated for its potential to produce opportunities.

Developing vigilance in spotting opportunities supports this.
To derive innovation regarding technical solutions as well as their commercialization, companies should seize the creative capital of each individual software development team member. Incentivising and rewarding vigilant practices could increase awareness among developers and produce and sustained an innovation culture. Our results highlight the necessary cross-disciplinary knowledge, and underlying psychological safety to foster individual team members to pursue creative solutions, which companies should aim to cultivate complementary to ‘agile values’ such as autonomy.

As the concept of uncertainty carries a negative connotation among practitioners, teams are at risk to dismiss hidden opportunities by attempts to eliminating instead of coping with uncertainties. At the same time, existing approaches to uncertainty or risk management may fail to uncover opportunities for similar reasons; many risk management frameworks are designed with the mitigation of negative events in mind. In order to profit from unexpected events, a change in perceptions that extends uncertainty to include opportunity as a possible outcome is strongly advised. We propose that communication practices in agile development be extended to include uncertainty management, which can be structured along the types of uncertainty we identified. Team members should aim to integrate the sharing of incrementally collected information in existing communication structures to uncover uncertainty’s positive or negative developments. A structured approach, as opposed to practices left to be carried out only by mindful team members, is likely to improve uncertainty management practices and help practitioners in identifying threats and opportunities earlier. Learning and information sharing, which are crucial for this, are already fostered by agile development methods. These need to be extended by a shift to exploratory activities beyond technological solutions, and complemented by the systematic search for opportunities. Whereas the literature documents failures to systematically approach risk and uncertainty management (Kutsch, Denyer, and Hall, 2012),
our analysis stresses the importance to collection information systematically and in combination with further practices that enable teams to reap benefits while preventing potentially detrimental developments.

I.5.3 Implications for Research

Different implications for scholars of uncertainty can be derived. Researchers should not only follow calls to look beyond risk and consider uncertainty management as important source of insight (Ward and Chapman, 2003), as well as acknowledge the existence of different types of uncertainty to increase investigations of beneficial uncertainties, which have often been ignored in the risk management literature. Considering that uncertainty affects projects in different ways (Bradley and Drechsler, 2013), it is advisable that future work addresses the conditions under which different types of uncertainty become manageable for development teams. For example, it is not clearly understood if the possibility for opportunities can be categorically excluded for some cases, which would allow different conditions for the mitigation of threats. Another important aspect to extend our understanding is the consideration of individual differences such as risk attitudes that influence the behavior of actors. These aspects should be addressed in future studies. Grote (2009, p. 48) suggests that the potential for uncertainty management in organizations is related to the “belief systems held by occupational groups and the belief systems developed in the companies themselves, that is, company culture”. While the complexity of the uncertainty concept has the potential to deter researchers and practitioners, it is a powerful lever of beneficial impacts on development projects and deserves increased attention.
I.6 Conclusion

Like the ancient Roman god Janus, uncertainty has two faces. One is beneficial whereas the other is malign. In this article, we empirically explore how agile software development teams manage uncertainty, i.e., in situations in which they have incomplete information, in order to reap opportunities and avoid threats resulting from it. We present work that explicitly continues to move the study of risk beyond the focus on threats by including potential benefits from uncertainty.

Our findings detail practices to manage different types of uncertainty (regarding resources, task, and requirements), which allow teams to mitigate threats while remaining alert to unanticipated opportunities. Many of these practices were performed simultaneously or in a periodically recurring fashion following overarching principles for uncertainty management. Incorporating the possibility for unexpected events led development teams to reckon with changes especially with regard to requirements and availabilities of resources. Through frequent probing and replanning, teams were able to elaborate most adequate tasks and mitigate many negative consequences, for example, with regard to avoiding ineffective work. At the same time, some developers displayed the vigilance to remain alert to potential opportunities stemming from the same sources of change, which often resulted in benefits for their teams.

We hope that these findings will guide teams in software development projects and in new product development projects in general, which face difficulty in the discernment of uncertainty as threat versus opportunity.
Acknowledgements

The authors declare no conflict of interest. We wish to thank the participants of our study for their time and effort, and are extremely grateful for their support and valuable comments. An earlier version of this paper was presented at the 16th International Conference on Agile Software Development (XP2015). We are grateful to the conference participants and in particular to Torgeir Dingsoyr and Nancy Van Schooenderwoert for their valuable feedback and suggestions for revisions.

References


References


Lee, G and W Xia (2010). „Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility“. In: Mis Quarterly 34.1, pp. 87–114.


Oke, A, N Munshi, and F O Walumbwa (2009). „The influence of leadership on innovation processes and activities“. In: *Organizational Dynamics*. 
References


Turner, Jon (1990). „A Comparison of the Process of Knowledge Elicitation with that of Information Requirements Determination“. In:


Responding to change over following a plan: How interdependent routines support stability and flexibility in innovation teams

Manuscript invited for revise and resubmit at
Information and Organization

AUTHORS
Denniz Dönmez, denniz@ethz.ch
Gudela Grote, ggrote@ethz.ch
Stefano Brusoni, sbrusoni@etz.ch

AFFILIATION & ADDRESS
ETH Zurich
Department of Management, Technology and Economics
Weinbergstr. 56-58, CH-8092 Zurich, Switzerland
Highlights

- Concurrent flexibility and stability are crucial but difficult to achieve in firms.

- We study how organizational routines can be employed to create and balance both.

- Our analysis of product development teams points to several interdependent routines.

- Interdependent routines are not only performed flexibly to produce stability.

- Also, interdependent routines purposefully disrupt stability to create flexibility.

Keywords

Organizational routines; Routine interdependence; Flexibility and stability balance; Product development.

Abstract

A growing body of research documents the implementation of organizational routines not only for the creation of stability, but also flexibility, both of which are essential to organizational functioning yet difficult to achieve simultaneously. In this study, we examine how both stability and flexibility are enhanced in team processes geared toward innovation. In such units, continuous delivery of innovative solutions is imperative in order to meet the requirements of highly dynamic environments. We analyze how product development teams employ different organizational routines to increase and balance stability and flexibility. Based
on a qualitative study of agile software development teams, our study sheds light on the interdependence of organizational routines in addressing contradictory requirements of stability and flexibility. We find that teams rely on a number of interdependent routines (e.g., for the planning of and collaboration on specific tasks) in order to enhance predictability and control while remaining capable of versatility and responsiveness. On the one hand, interdependent routines are performed flexibly to produce stability. On the other hand, interdependent routines may be purposefully designed for regular disruption of stability and reflection on the product development process, permitting opportunities for change. Organizations can benefit from increasing adaptability and efficiency through choosing the extent to which they protect the routines’ performances and content to balance stability and flexibility with regard to processes and outcomes. Our empirical results speak to a current debate in the literature by showing how stability does not necessarily come at the cost of flexibility. We contribute to routine theory by explaining how stability and flexibility can be increased simultaneously through interdependent routines that support innovation in product development.

II.1 Introduction

At 9:44 a.m. Ralph walks up to the whiteboard that has countless color-coded post-it notes attached to it, which are grouped into columns entitled ‘backlog’, ‘on hold’, ‘in progress’, ‘verified’, and ‘done’. One minute later he is surrounded by the six members of his product development team, none of whom are late for the daily stand-up meeting. Therefore, today no coins are put in the piggy bank set up to discourage tardiness, which regularly finances team dinners. The team forms a half circle around Ralph, and then each team member delivers a brief status report on their work, structured by three questions: 1) What did I do since yesterday that helped the team meet its goal, 2) What will I do until tomor-
row to help the team meet this goal, and 3) Do I see any impediment that prevents me or my team from meeting this goal. Most speak with the low voice that characterizes conversations in the office space shared with four other teams, where telephones ring rarely, and the clicking sound of typing on computer keyboards is omnipresent. As the standing team-members follow their protocol, colleagues occasionally ask clarifying question. While they speak, Ralph tracks the work progress, shifting post-its up or down a column, or attributing further information such as a red sticker for ‘software bug’. One team member will later copy the information into a spread sheet shared with management and clients, and document relevant details regarding impracticable and potential attempts to resolve the issue. Upon completing the round of status reports, Ralph closes the meeting at exactly 9:58 a.m., thereby triggering the next daily routine two minutes earlier than planned, which is an informal meeting with a varying set of participants; the coffee break.

Situations like these are experienced every day in many software-developing companies around the globe, where teams follow rigid protocols to pursue creativity and innovation. Innovation teams, including product development teams, are expected to deliver a steady flow of creative and novel solutions to complex problems under high levels of uncertainty (e.g., Brown and Eisenhardt, 1997). They operate in increasingly dynamic and complex environments (Boehm and Turner, 2003), and are therefore required to work in ways that are both reliable and adaptive. In other words, they need to engage in activities aimed at the simultaneous increase of stability and flexibility, both of which are important for organizational functioning for different reasons (Benner and Tushman, 2002), and must be balanced carefully (March, 1991). The question of how stability and flexibility are created has often been studied through the lens of organizational routines, which are defined as “repetitive, recognizable pattern[s] of interdependent actions, involving multiple actors” (Feldman and Pentland, 2003, p. 95). While the potential of
II.1. Introduction

routines to create stability had been the primary focus of early research (Cyert and March, 1963; Nelson and Winter, 1982), later work noted that routines can produce flexibility as well (Adler, Goldoftas, and Levine, 1999). An important question, however, is how their potential can be employed to simultaneously establish and balance both (Turner and Rindova, 2012; D’Adderio, 2014). Considerable progress has been made to address this question by empirical investigations in diverse settings, however, there still remain important unanswered aspects. In particular, the relationship between routines and innovation has remained almost completely neglected (Becker and Zirpoli, 2009), yet may provide a very promising field to advance routine theory (Sele, forthcoming). Further, much insight remains to be gained from studying the employment and functioning of several routines collectively (Salvato and Rerup, 2011), as the dynamics of several routines may be very different from the dynamics of individual routines (Kremser and Schreyögg, forthcoming).

In this paper, we investigate these important aspects by addressing the following questions: For the purpose of balancing stability and flexibility, how are organizational routines employed, maintained, changed, and disestablished with regard to different aspects of product development processes in innovation teams? To sharpen our analysis, we draw on a distinction of stability and flexibility regarding processes versus outcomes (Farjoun, 2010) to empirically investigate how both are established. More importantly, we specifically focus on the collective employment of several routines for this purpose. Whereas past research has been mainly interested in the analysis of how routines are established or changed to individually exert influence on organizational outcomes, our findings suggest that important insight can be derived from considering the interplay between several routines that coexist at a given point in time. Our contribution speaks to the current debate whether individual routines carry the potential to create both stability and flexibility, or whether different routines or bundles of routines are required to this end, and answers
calls to investigate routine interactions (Parmigiani and Howard-Grenville, 2011).

II.2 Conceptual Background

II.2.1 Organizational Routines

The traditional view of routines emphasizes their potential to increase efficiency through the standardization and stabilization of processes (March and Simon, 1958; Cyert and March, 1963). In this view, routines are employed for the purpose of capturing organizational knowledge, contributing to effective processes, and signaling accountability and reliability to customers (Leana and Barry, 2000; Gilson et al., 2005). Hence, routines function as important contributors to stability. A different view on routines aims to establish their potential to create flexibility and adaptation. This stream of research has pointed to meta-routines that are used to impact other routines and, thus, induce organizational change (Nelson and Winter, 1982; Adler, Goldoftas, and Levine, 1999). Interested in how routines are capable of producing such diverse outcomes, researchers have turned away from treating routines as “black boxes” (Parmigiani and Howard-Grenville, 2011) to explore their complexity and investigate their functioning through looking at underlying mechanisms. In their seminal paper, Feldman and Pentland (2003) developed the concepts of “performative” and “ostensive” parts of routines. The performative part refers to how a routine is executed by its participants, whereas the ostensive part is associated with the shared understanding that enables participants to refer to the routine. This differentiation revived the debate about the potential of routines to bring about change in order to meet the challenges of dynamic organizational environments (Parmigiani and Howard-Grenville, 2011). One major contribution was the insight that flexibility is not only produced by meta-routines (Nelson and Winter, 1982),
II.2. Conceptual Background

but also by allowing for differences in the performances of routines (Feldman and Pentland, 2003). It has been suggested that the degree to which routines are embedded in other organizational structures, such as coordination or technological structures, influences their flexible use by individual actors (Howard-Grenville, 2005). Using routines flexibly allows actors to achieve their goals without altering the routines permanently. Similarly, Turner and Rindova (2012) describe how participants in organizational routines rely on flexible performances in order to produce stable, pre-defined outcomes. They identified that organizations use different routines, the ostensive parts of which are directed at either flexibility or stability, in order to achieve a stable level of outcomes. In this study, we aim to extend the understanding of the functioning of routines in teams with dynamic outcomes, which requires an elaborate conceptualization of stability and flexibility as either a process or an outcome (Farjoun, 2010).

II.2.2 Stability and Flexibility

Stability is defined in the New Oxford American Dictionary as a “state of being stable,” “not likely to change or fail,” or “firmly established,” and refers to the absence of variation. In patterns of collaborative action, stability can be detected where there is a regularity or continuity (Feldman and Pentland, 2003). By contrast, following Qin and Nembhard (2010, p. 325), we conceptualize flexibility as “the ability for actively taking advantage of opportunities and positively countering risks.” In this regard, flexibility is a property that can be seen as an antecedent of the change needed in reaction to dynamics in the organizational or team environment.

Farjoun (2010) proposes that stability and flexibility may refer to either an outcome or to an underlying mechanism (i.e., a process). For our study, the distinction between these two types of stability and two types of flexibility (i.e., regarding either a
process or an outcome) is valuable for several reasons. First, we can bring our analysis to a more profound level by discriminating between stability and flexibility as either a characteristic of the product development process or a characteristic of the outcome of the work. For example, the delivery of software products can be stable (outcome stability), while the development process is highly flexible (process flexibility). By contrast, the opposite can be the case; a stable process can produce many variations of products (outcome flexibility). Second, this view allows us to examine organizational routines in terms of their potential to enhance specific aspects of stability, flexibility, or both, which has been overlooked by the literature. Third, this distinction allows us to scrutinize distinct aspects of the routines that are directed at different types of stability and flexibility (i.e., outcome- versus process-oriented).

High demands for the contradictory effects of stability and flexibility permeate product development, which stem from the need to manage high levels of uncertainty. By uncertainty, we mean unforeseeable variations in customer demands and organizational resources, or the discovery of unexpected difficulties that are external (e.g., involving collaborating partners, suppliers, etc.) or internal (e.g., technical) to teams (McConnell, 2010). Whereas it has been long established that the possibility to exploit uncertainty ultimately enables the realization of positive outcomes (Knight, 1921), organizations tend to aspire to eliminate uncertainties in order to benefit from stability in producing outcomes in a consistent and continuous manner (Benner and Tushman, 2003; Grote, 2009). Project management often strives for thorough up-front planning to support this.

Project plans contain rigorously detailed task definitions along with the schedule of their execution in order to ensure a project’s stability. Yet such plans contain countless interdependencies, and critics point out that uncertainties in the development process cannot be minimized by highly structured project management (Boehm, 1991). Early deviations from the stability-driven ap-
II.3. Research Setting and Approach

In order to answer our research question of how various routines support the balancing of stability and flexibility in innovation settings, we focus our attention on the development of new software products. Software development teams have much to offer for our study for several reasons. First, software developers are charged with the task of pursuing dynamic goals in creating novelty and innovation, which Austin and colleagues define as “original, beneficial outcomes” Austin, Devin, and Sullivan (2012, p. 1508). At the same time, software developers face the need to deliver a steady flux of such outcomes in environments that are characterized by high uncertainty and low continuity—circumstances that simultaneously require high degrees of flexibility to adapt, as well as stable development processes.

Second, the software industry is currently undergoing a shift in production techniques that acknowledges the increased need for flexibility (Larman and Basili, 2003). Software development teams enjoy the capability to modify or build their own production tools, and many have started to optimize production processes by abandoning control-driven approaches in favor of increased self-management (Vidgen and Wang, 2009). This is
reflected by a move toward light-weight development methods (Cohn, 2009), which place more emphasis on flexibility while maintaining sufficient stability to ensure productivity and efficiency. Such possibilities provide an ideal setting for our study of how teams strive to balance high levels of agility with sufficient stability by establishing routines that support the creation of innovative products. Third, due to clearly defined structural elements in the procedural framework of modern software development approaches, the boundaries of routines can be easily discerned despite the complexity of the development process. This allows us to trace changes in the routines, which has been pointed out as an important difficulty in routines research (Becker, 2004).

Although writing computer programs may not appear to be a domain where routines can easily be observed, we found that software development is most suited for our purpose because it is an inherently collaborative work that allows for a number of routines to be clearly distinguished. This is especially the case in agile software development, which is primarily suited for highly innovative projects (Highsmith and Cockburn, 2001). Agile software development routines are marked by observable characteristics, such as the time or duration of their occurrence, the number and roles of their participants, or the outcomes they produce, they are distinguishable not only against non-routinized activities, but also against other routines. These features are most common in so-called agile software development techniques, which display an increased focus on flexibility.

II.3.1 Agile Software Development

In environments characterized by fast changes, such as surrounding new product development, companies and the teams they consist of need high levels of flexibility in order to remain competitive. In an attempt to increase flexibility, different industries have brought forward a number of approaches focused on improved de-
II.3. Research Setting and Approach

Development and production processes suited for very dynamic challenges. Prominent manufacturing methods, such as lean production, six sigma, or total quality management (TQM), have been adapted by various product developing industries. A recent trend in software development was the introduction of agile software development (Schwaber and Beedle, 2002), which differs from more established (so-called plan-driven) methods in its impact on communication, knowledge management, and desired organizational structure (Dingsoyr, Dyba, and Abrahamsson, 2008), placing emphasis on short development iterations, feedback loops, continuous (as opposed to up-front) planning, self-organizing development teams, and high process transparency for involved stakeholders including developers, managers, and customers. To engage in explorative activities aimed at the creation of innovation, agile software development teams employ a number of routines (Vidgen and Wang, 2009). These are arranged in schemes or marked by special activities, such as feedback cycles or strategy meetings. Routines often span across the entire development team, and sometimes integrate customers and suppliers for on-site collaboration, which is common in highly innovative teams. In addition to supporting flexibility, agile software development methods also provide a number of elements that enhance stability, for example, by means of fixed meetings and clear responsibilities. Agile routines aim at creating a stable development environment by prescribing a set of rigid structures, (e.g., with regard to work phases, team sizes, and meeting lengths). At the same time, developers seize the flexibility that is allowed within these stabilized boundaries. Teams not only engage in a process of continuous planning and re-planning, but are also encouraged to alter the very structure used for project management in order to redesign and improve it. Perhaps because there is a strict separation of process-related and content-related flexibility, practitioners often perceive agile process frameworks as flexibility-enhancing, despite their structural rigidity. This becomes apparent when studying the agile development method Scrum (a term borrowed from the
game of rugby and intended to convey team cohesion), which enjoys great popularity among software development teams across industries to the point that it is becoming the de-facto standard agile method used in companies (Dingsoyr et al., 2012).

The basic Scrum framework centers around a routine that divides the development process into equal phases of several weeks’ length; the so-called Sprint iterations. Many other routines, such as daily meetings to synchronize the team members, are performed within the Sprint. In an attempt to deliberately weaken hierarchies, all team members are charged with equal responsibility for the product, which is expressed, for example, in their shared ownership of the software code. The use of a physical white-board for task planning is strongly advised, as is the collocation of the entire team. Prescribing strictly structured work phases, the method suggests meeting lengths that leave a team with five to nine members a maximum of 25 hours of planning activity each month—spread over pre-scheduled meeting slots—for planning and revising development iterations, which last for only two to four weeks. In an effort to control team performance, Scrum also suggests techniques to monitor work flows and improve estimations of unknown work packages. There is advice to abandon specific job titles within the team and to reduce hierarchical differences. Two exceptions with regard to the intended homogeneity of roles are granted in the form of the Product Owner and Scrum Master, which are installed in order to support the development team. Ideally filled by two separate but exclusively dedicated team leaders, these roles are assigned external communication and internal coordination activities, respectively. The Product Owner functions as a representative of the client who can make decisions regarding the priorities of requirements and functionality of features to be implemented. The Scrum Master’s responsibility is to provide support for the team, for example, by performing managerial tasks such as facilitating meetings, organizing the availability of infrastructure, and stressing the adherence to suggested Scrum rules, including the execution of identi-
II.3. Research Setting and Approach

cal routine performances without deviations. With the extensive corporate adoption of agile methods (Melo et al., 2013), software development practices have received increasing interest from the academic community, however, mainly with a focus on industry practices (Dingsoyr et al., 2012). Because agile software development combines a set of elements that either enhance stability or flexibility, we seek to exploit it as an example that has much to offer for researchers as well as practitioners interested in the functioning of organizations through an investigation of how routines are employed for the act of balancing flexibility and stability.

II.3.2 Data Collection

We have adopted qualitative research methods using a multiple-case study design based on field observations and semi-structured, open-ended expert interviews (Yin, 2013). In addition, we collected project artifacts, such as drawings, process visualizations, or team performance documentation, during company visits, and acquired complementary information during a number of informal interviews with team members and project stakeholders that were external to the teams. Using an ethnographically-informed approach (Robinson, Segal, and Sharp, 2007), the first author collected data during a one-year period with frequent and close contact with the team leaders and team members involved in the software development, as well as several non-developing project stakeholders. He visited each team several times and had the chance to observe formal meetings, as well as approach team members informally on many occasions. Field notes were produced during and after all formal and informal meetings. In addition, 33 semi-structured interviews were conducted that lasted between 20 and 70 minutes and amounted to over 20 hours of audio recordings, which were transcribed verbatim as soon as possible after each interview. To triangulate insight from different sources and capture various perspectives on internal procedures (Adolph, Hall, and Kruchten, 2008), interviews were conducted
with a representative number of team members in each team. Between four and five team members were interviewed in all but one team where two of its five team members were interviewed, always including at least one person with official leadership responsibility per team (i.e., Scrum Master or Product Owner role). The interviews were based on semi-structured interview questions and centered on probing for critical incidents (Flanagan, 1954) in representative situations covering several topics, which included uncertainties that impact the work, team work structures, and team members’ handling of situations that required either stability, flexibility, or both. Interviewees were asked to give examples for routinized task performances and to describe recurring activities and situations they encountered on a typical work day. Each interview began with general questions, such as the current status of the project or the interviewee’s role and current work activities, and then moved to examples of recently experienced situations. We explicitly aimed to collect negative as well as positive experiences and discovered that most informants talked freely and openly even about sources of frustration and topics such as their motivation to leave the company. Our objective was to gain a thorough understanding of the routines in the development process, and to uncover the mechanisms that are part of these routines. During the interviews, the pursuit of interesting cues was given priority over the strict adherence to our interview guideline when it appeared appropriate; interview partners were probed for details regarding our research questions, but also deliberately encouraged to wander freely in their answers in order to allow important aspects to emerge.

II.3.3 Sample

We collected data in eight product development teams situated in five different companies in Switzerland. They operated in different industries; five teams developed products within large IT departments in telecommunications and financial services, serv-
II.3. Research Setting and Approach

Involving mainly company-internal clients. Three teams were part of small software developing companies dedicated to producing customized software for external clients. We adopted a theoretical sampling approach (Miles, Huberman, and Saldaña, 2013) and strove to increase generalizability of our findings by including innovating teams from distinct organizational settings (e.g., with regard to company size or industry) in order to exclude biases that may be rooted in company structures. We gained access to agile software development teams that are regarded as highly innovative and show similarities regarding their objectives to produce novel solutions for dynamic markets. The teams are similar in size, geographical distribution, members’ educational background, and team tenure. All project teams were granted high amounts of autonomy within their companies (however, some only after initial management resistance) and had decided as a team to increase their innovative capacity by employing agile software development techniques. At the time of data collection, all of the teams had used agile software development methods already in other projects that preceded the ones we observed. Each of the teams we selected consisted of five or more members (average: 10.1) who had been working together for at least one year at the time of data collection. All participants were professional software developers with academic qualifications in computer science or related fields. Few of them had received explicit training with regard to agile development methods or had gained relevant certifications, such as through professional Scrum training. Most participants had several years of working experience with agile software development in addition to other project management methods in the software industry. Two of the teams had hired a dedicated Scrum coach who was not involved in the development, but provided guidance regarding the project management technique, whereas the other teams internally appointed a person in charge of process guidance who would refer to books and other self-education material, as well as local communities of practice. Apart from the development work with their team, some also
acted as mentors or trainers for other teams because they had acquired considerable knowledge and experience in agile methods, which they sought to share within their company.

II.3.4 Data Analysis

To make sense of our qualitative data, we adopted an iterative approach. Consistent with approaches combining theory building and theory elaboration (Bingham and Eisenhardt, 2011), we began by reading the interview transcripts mindful of general issues linked to collaborative patterns. With the objective to explore the creation of stability and flexibility in the context of routines, we subsequently coded the data using in vivo codes or short descriptions. To facilitate the systematic coding process, several tools were employed; they included HyperResearch, a computer-assisted qualitative data analysis tool, as well as Excel spreadsheets and digital notebooks. Four categories concerning implementations of routines were used, namely the creation of (1) stability regarding the software development process, (2) stability regarding the software development outcome, (3) flexibility regarding the software development process, and (4) flexibility regarding the software development outcome. Whereas our coding scheme was centered on these concepts, we extended it with the emergence of new insight, for example, when we realized that the participants in our study not only had to respond to scant flexibility, but also faced excess flexibility in some situations. From this initial analysis resulted a list of excerpts or short passages, which allowed us to identify themes, such as the use of communication structures or constraints resulting from resource-dependencies. As we cycled between data collection and analysis—assigning the codes where appropriate and extending our code library when necessary—we re-examined previously analyzed data with the newer, complete list of codes again. We incorporated field notes and other artifacts that had been produced or collected during company visits to enrich our understanding of
the underlying processes and structures, as well as the projects’ contextual settings. We noted ideas and key issues in a separate file, and created memos (Groenewald, 2008) in order to elicit explanations linked to the increase or decrease of stability and flexibility within different routines. In order to understand how the routines were employed for these purposes, we identified patterns across several teams. A table was created for each routine that contained triggers of stability or flexibility, the analysis of which led to the insight that the interplay or routines was a key element in the process of balancing stability and flexibility. In order to strengthen content-validity from empirical induction, we discussed and clarified our findings with participants from the companies. The output of these analyses informs a model of the interplay between routines employed to balance stability and flexibility.

II.4 Findings

In order to present our findings, we first outline in more detail the most important routines that we identified in agile software development practices. Subsequently, we will demonstrate how these routines are involved in a number of software development activities, such as prototyping and project planning, and detail the interplay of several routines in the software development process. Our findings inform a model of the interplay of routines and their collective effect on balancing stability and flexibility, which we subsequently developed. In our analysis, we relied on four different categories of stability and flexibility that were derived using theoretical insight from the literature. These categories are

1. outcome stability (which produces a continuous delivery of software in the form of updates and product improvements),

2. process stability (which ensures an uninterrupted workflow),
3. **outcome flexibility** (which permits the change of product features, for example, in reaction to changes in a customer’s requirements), and

4. **process flexibility** (which allows for changes in the work process, for example, regarding the allocation of developers to tasks or the sequence of tasks).

Using this distinction, we were able to analyze how different routines are orchestrated in order to enhance stability or flexibility in a number of activities that lie at the heart of the software development process. To demonstrate our findings, we selected four routines that are prominent in agile software development based on their representativeness in all of the teams we studied. A comparison of the routines and their characteristics is given in Table 5. Table 6 details different effects via which the routines collectively operate on the creation of stability and flexibility.

**Table 5:** Profiles of agile software development routines including their characteristics

<table>
<thead>
<tr>
<th>The sprint iteration routine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>A routine borrowed from the Scrum framework for agile software development, which introduces iterative and incremental phases to the development process. The goal of a sprint consists of the production of a preliminary, working software product, such as a prototype.</td>
</tr>
<tr>
<td><strong>Participants</strong></td>
</tr>
<tr>
<td>All software developers and both team leader roles (i.e. Scrum Master and Product Owner) for planning purposes. The developers must not be interrupted in their self-organized work.</td>
</tr>
<tr>
<td><strong>Duration</strong></td>
</tr>
<tr>
<td>According to the Scrum framework, every performance of the routine should last for the same amount of time and a maximum of four weeks.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
</tr>
<tr>
<td>The routine is scheduled and often synchronized with the organization-wide software release cycle.</td>
</tr>
</tbody>
</table>
Table 5 (continued): Profiles of agile software development routines including their characteristics

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apart from information that is acquired through other routines, the Sprint routine depends on the developers’ continuous coordination in the creating tasks from requirements formulations and their accomplishment, directed as the Sprint goal.</td>
<td>A working product with some of the final product features implemented, which serves as the basis for subsequent Sprint iterations of incremental improvements.</td>
</tr>
</tbody>
</table>

**Task size estimation routine**

**Description**
Short discussions of the effort required to accomplish a task are held in order to support the compilation of tasks for the accomplishment of larger work packages, such as the implementation of a product feature.

**Participants**
Knowledgeable developers who are involved in or responsible for the tasks that require estimation.

**Duration**
Depending on the complexity of the task, the routine performance lasts between a few minutes to half an hour. It may be adjourned in case further knowledge is required to produce an estimate.

**Trigger**
The routine is performed whenever there is a modification or creation of a task. No task of unknown effort is allowed to enter the work flow.

**Input**
Knowledge regarding special technical issues is required from all involved developers.

**Output**
Estimations of task sizes in terms of their relative effort (compared to other tasks) are produced.

**The task allocation routine**

**Description**
A determination of free resources to pending tasks that serves an uninterrupted work flow. The routine is facilitated by the use of artifacts, such as physical or virtual project management devices.

**Participants**
Developers that have accomplished a task, sometimes supported by the Scrum Master.

**Duration**
A few minutes.
Table 5 (continued): Profiles of agile software development routines including their characteristics

<table>
<thead>
<tr>
<th>Trigger</th>
<th>The routine is performed after the accomplishment of a task or the formation of new tasks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>A preview of developers’ abilities to match current and future pending tasks facilitates the work flow and helps to avoid the necessity to allocate tasks frequently.</td>
</tr>
<tr>
<td>Output</td>
<td>Task allocation assigns responsibilities to developers regarding one or several tasks.</td>
</tr>
</tbody>
</table>

**The daily stand-up routine**

<table>
<thead>
<tr>
<th>Description</th>
<th>A short, daily meeting in order to update all team members on the status and progress of the work. It is held standing, usually around a central project planning device or other visualizations of the work.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participants</td>
<td>All developers and the Scrum Master, who supervises the meeting. Team members that operated under flexible working arrangements, such as from home, may participate via telephone conference systems.</td>
</tr>
<tr>
<td>Duration</td>
<td>Up to fifteen minutes maximally.</td>
</tr>
<tr>
<td>Trigger</td>
<td>The routine is scheduled every 24 hours on work days.</td>
</tr>
<tr>
<td>Input</td>
<td>Every developer delivers a brief report regarding the progress of her work.</td>
</tr>
<tr>
<td>Output</td>
<td>General knowledge is created with regard to the status of the development work. This serves the Scrum Master as a measure to predict problems in reaching the Sprint goal and overall long-term development goals.</td>
</tr>
</tbody>
</table>
Table 6: Examples for the creation of stability and flexibility via effects of different, interdependent routines

<table>
<thead>
<tr>
<th>Representative quote</th>
<th>Routine</th>
<th>Theoretical Observation</th>
<th>Routine links</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Scrum has helped me in the way that [the next process step] was always clear to me, and I didn’t have to ask where we are. Instead, I always knew: we have achieved something, the next step is this.”</td>
<td>SIR (=Sprint Iteration Routine)</td>
<td>Sprint iterations support the stabilization of processes by clarifying the sequences of tasks.</td>
<td>Different routines support work-flow planning (SIR) and visualizing progress (DSR)</td>
<td>Process Stability</td>
</tr>
<tr>
<td>“We have so-called dailies, where we synchronize our teamwork. They last about 15 minutes, and we discuss the hot topics and see how the progress is.”</td>
<td>DRS (=Daily Stand-up Routine)</td>
<td>Attending to potential impediments and addressing work progress frequently enhances the stability of the process.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“So yes, with all the tools that are available [electronically] to describe the status, the updates, the steps that are being taken so that if for the next Sprint, someone else takes that task, he can take over from where I stopped.”</td>
<td>TAR (=Task Allocation Routine)</td>
<td>Work processes are kept flexible through tasks that are allocated to different developers who can take over.</td>
<td>Different routines (TAR, TSER) support allocating resources flexibly (in both directions; task to developer and vice versa)</td>
<td>Process Flexibility</td>
</tr>
<tr>
<td>“the ‘on-hold’ column [on the planning board] is for when you have something urgent, so you have to put things aside […] or when something is not clear, when the customer has to specify something.”</td>
<td>TSER (=Task Size Estimation Routine)</td>
<td>A flexible process involves putting tasks on hold in order to attend to something more urgent, or to acquire further information such as estimates that need customer input.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6 (continued): Interdependent routines affect stability and flexibility

<table>
<thead>
<tr>
<th>Representative quote</th>
<th>Routine</th>
<th>Theoretical Observation</th>
<th>Routine links</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>“During the iteration, when you have a four week iteration, for example, you define in the beginning what you want to do in the next four weeks.”</td>
<td>SIR</td>
<td>Outcome stability is produced by freezing the Sprint goal, which is defined anew for each iteration.</td>
<td>Different routines support setting goals (SIR), evaluating goals (TER), and task awareness (DSR)</td>
<td>Outcome Stability</td>
</tr>
<tr>
<td>We have estimated this [task] and have excluded it for now, because the client wouldn’t pay for more than a week for that, because there is no visible added value.</td>
<td>TSER</td>
<td>Producing information about the cost-benefit ratio, task estimations impact the software development outcome by focusing teams on value-creating output.</td>
<td>Closer collaboration with suppliers (here, the integration of a graphic designer into the daily stand-up routine) allows a team to stabilize iteration outcomes as input becomes less uncertain.</td>
<td></td>
</tr>
</tbody>
</table>
II.4. Findings

Table 6 (continued): Interdependent routines affect stability and flexibility

<table>
<thead>
<tr>
<th>Representative quote</th>
<th>Routine</th>
<th>Theoretical Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>“This is not [traditional], but iterative so to speak, that means we don’t estimate and then do everything like [in traditional software development], but rather during the process of this task we stop time and again and look together with the client, [and ask] what did we do, what else do we need to do, are we on the right track? Then we continue, step by step.”</td>
<td>Sprint Iteration Routine (SI)</td>
<td>Flexibility in terms of the software product results from the opportunity to change directions and use discussions to consider different solutions.</td>
</tr>
<tr>
<td>“That’s where we discuss each mandate. Here it’s going like this, there it’s going like that, there I have this problem, here I can help... So that you are not alone with your problem, that you can share it with your team, maybe somebody has a good idea.”</td>
<td>Daily Stand-up Routine (DS)</td>
<td>New ideas can emerge when different project stakeholders come together each day for mutual updates on their progress. They remain open to and discuss new ideas and different solutions.</td>
</tr>
</tbody>
</table>

II.4.1 Agile Routines

We found that several of the various routines that agile software developers perform (e.g., to plan and review their work, manage knowledge, or allocate resources) are simultaneously employed in order to create stability and flexibility in development projects. Whereas some of the (ostensive parts of the) routines are established by the teams through adapting guidelines from the Scrum method for project management, modifications to them and complementary routines emerge over time. Our data suggest that developers value the stabilization of the development processes,
which is protected as a result of limiting deviations in the performances of the routines. Participants follow the strict guiding principles of agile methodologies in establishing the routines according to agile software development templates. Modifications of the routines are rejected until a certain momentum is reached in the routine performances during later stages of a project:

In the beginning we were still following [the Scrum] rules meticulously. We had to. But later, I don’t think it would have made sense [...] With time you get pretty routinized with it all. (Scrum Master, Team C).

In the following, we describe the four most important routines that are part of our subsequent analysis of agile practices, drawing upon and combining them in order to balance stability and flexibility. Further examples from the data are given in Table 6.

**The Sprint Iteration Routine**

The most prominent routine of agile software development temporally divides the development process into iterations of equal length, which are called Sprints, to evoke the notion of a speedy move in a strategic direction. Most teams rely on two-week Sprints, but some use much larger time intervals of several months. The purpose of a Sprint is twofold. First, based on product requirements and their priorities, a Sprint goal is defined in collaboration with the Product Owner, who represents the customer. Defining a Sprint goal contributes to stability regarding the software development outcome by obligating the developers to deliver working software product features at the end of each iteration. Second, the responsibility for the accomplishment of the Sprint goal is transferred to the software developers under the exclusion of managers and with the bestowal of sufficient autonomy
to work uninterruptedly toward it. Hence, participants of the routine are primarily the developers, which collectively commit to a set of work packages:

What we do is we have some [Sprint goal] we want to achieve in the Sprint, and up to that we are comfortable that we can do that (Developer, Team F).

The developers are supported by the Scrum Master, who is charged with the elimination of impediments to their work. During the Sprint, all developers work in a self-managed fashion on pre-defined modules while the coordinative effort of project managers is limited. The Sprint routines’ ostensive aspects trigger meetings for planning and reviewing the work in the beginning and at the end of the routine, respectively. Sprints are performed sequentially without any interludes between them.

The Task Size Estimation Routine

Agile software development processes consist of sequences of tasks that can be either stable or flexible. The estimation of effort required for each task is crucial, and diverse estimation techniques exist outside of the purview of the Scrum framework. One common practice is to perform a task estimation routine that involves developers gathering prepared with a deck of cards. The cards are either numbered with the Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, 34, ...), or size labels (S, M, L, XL, etc.). They represent task sizes (also called story points), which refer to the effort that a developer estimates for the completion of the task at hand. Although the task sizes include no units, such as hours, most developers translate the estimates to hours or days. Instead of estimating an effort single-handedly, developers responsible for a task convene the entire team, as is emphasized by the teams:
It’s not like a particular person comes up with an estimation. The entire team thinks [in a collaborative effort] and the best possible estimation is [agreed on]. We use complexity points. We use the standard Scrum scale with the Fibonacci series to estimate the complexity, and we know how much we can do, more or less. We don’t just guess at it (Developer, Team F).

Using estimation cards facilitates a secret ballot, the advantage of which lies in the elimination of biases through the possibility for simultaneously uncovering votes in the absence of a bureaucratic or complicated process. Any discrepancy between several developers’ estimates must be resolved in repeated rounds, following discussions, until a single estimate is agreed upon:

The one with the smallest number and the one with the largest number explain why they estimated that, and we just keep estimating until everyone has the same number (Developer, Team C).

Routine performances are usually terminated within a few minutes, but are not protected with regard to duration, and ultimately depend on the complexity of a task.

**The Task Allocation Routine**

Allocating tasks to developers is a frequent practice in agile software development. It contributes to efficient processes when the most urgent tasks are assigned to available developers. The necessity to maintain a routine for it results from the high level of autonomy that teams are entrusted with; in opposition to a team leader assigning resources under traditional software development, agile developers carry out large parts of this strategic activity without managerial intervention. Participants of the routine are the relevant developers involved in the tasks that require
II.4. Findings

attention as they move to the top of the priorities list. Whenever resources become available after the completion of a task, developers assemble in order to update the pertinent project planning devices, such as whiteboards or spreadsheets, and match their availabilities and capabilities to work demands:

We never took our own tasks like ‘I will do this and I will do this.’ It was always in the [...] meeting that we all sat together, and we decided that (Developer, Team F).

The task allocation routine may be performed several times per day, and requires the entire team to be knowledgeable about the requirements of each task. One or several developers commit to a task, depending on its size, urgency, and importance. When pressure increases, developers tend to involve decision makers with authority over task priorities, such as the Scrum Master:

At the beginning of the Sprint, I have the possibility to choose [among] the available tasks. Of course, at the end of the Sprint where we have time restriction and we have to deliver and deploy then [the Scrum Master] assigns us tasks according to our availability and to the priorities we have (Developer, Team F).

The Daily Stand-up Routine

In order to synchronize short-term activities, the Scrum framework suggests to perform a brief status report meeting performed at the same time each work day. The common understanding of this routine decrees that all developers deliver status reports on their work progress. Following their brief accounts, developers have the opportunity to raise issues in case they detect any problems, difficulties, or potential impediments to the Sprint goal. In
practice, each report is structured by a simple scheme. Developers first state what task they are currently working on, usually referring to a whiteboard where tasks are tracked using information on Post-it notes. Next, they explain what they have accomplished since the last day regarding this task, and proceed by laying out what they plan to work on subsequently. They point out every encountered problem and anticipated difficulty, and give their colleagues the opportunity to extend questions or comments so that you are not alone with your problem, that you can share it with your team, maybe somebody has a good idea (Developer, Team A).

In order not to temporally inflate the daily stand-up routine, the ostensive part of it is timeboxed to a duration of 15 minutes. Timeboxing constitutes a form of protecting the routine performance and contributes to process stability by providing temporally fixed modules that can be rearranged without threatening stable sequences. Any issues that require further attention are referred to separate meetings, including the responsible developer and the particular colleagues who volunteer their support. In this way, new tasks and even entire work processes can originate from the daily stand-up (e.g., when special task forces are created to work on a problem that is discovered to be too large or complex for a single developer due to unexpected problems). Often when this is the case, additional information is required, and separate routines may be triggered. The interaction of several routines prevails in a number of practices described below.

II.4.2 Establishing Stability while Increasing Flexibility

Routines are not isolated from each other. Their performances have mutual influences. Teams rely on and choose among several interlinked routines to steer a project and react to changes in its
environment. In this sense, routines can be viewed as resources that provide options for actions. Their activation is subject to choices according to circumstances that may either require an increase of stability or flexibility depending on the objective of a specific software development activity.

**Planning is Better than Having a Plan**

Because all plans will eventually change, agile software development teams posit that the act of planning is better than the possession of a plan (which inevitably would become invalid after some time.) Operating in fast changing environments, teams are almost continuously engaging in planning and re-planning activities in order to remain agile. For example, while the Sprint routine has a pre-defined beginning and end which demands scheduled performances, its content is determined only when it commences. Contributing to outcome stability, a planning meeting is conducted with the purpose of defining a goal at the beginning of each Sprint. This Sprint goal will receive protection with regard to its content until Sprint is over. This way, developers are shielded against drastic plan changes for the duration of the Sprint routine performance; whatever goal will be defined for the current Sprint iteration cannot easily be changed, but has to be pursued until the next iteration commences.

For the planning meeting, developers rely on information about the tasks to be performed as well as the required effort. The former are informed by customer requirements, usually collected through the Product Owner. The latter is provided by the task size estimation routine performed for each task. The task estimation routine is triggered during a Sprint with every modification of an item in the backlog of tasks, such as the need to break down a large task, extend a task, or to create an additional task. Depending on estimated task sizes, the sequence of tasks may be reorganized and sorted according to their contribution to
the Sprint goal. This flexibility of the process is crucial, because the tasks’ size estimates can suddenly increase when unexpected difficulties are encountered. Due to the complexity of products, the corrected estimates can be substantially larger compared to the initial estimates, as was the case in one team:

it’s extreme [but] we were wrong by a factor of three (Developer, Team H).

Some situations may require the dissection of large tasks into smaller ones, spreading responsibility from one to several developers, who then form pairs of two or create sub-teams to address more complex endeavors. Therefore, an important part of the continuous planning process is the task allocation routine. It is not only triggered during the explicit planning meeting in each Sprint, but also whenever there is a change in the status of a task, for example, when it becomes necessary for developers to defer or break down a task, which can be at any point in time during a Sprint.

The performance of this routine is oftentimes triggered during the stand-up routine, for example, when team members approach the completion of a task and need to project subsequent priorities and availabilities. Unlike scheduled routines, its performance resembles the event-triggered nature of the task size estimation routine, which is invoked when information is needed. Similarly, task allocation updates may become necessary when other routines, such as the Sprint planning, cannot proceed without a task having an assigned responsibility. This uncovers how the flexibility of process content and stability of process execution are closely interwoven and mutually supportive.

Aside from the short-term planning of brief Sprints and daily activities, there remains a need to map out the direction of future development. This is not only done during the planning of the Sprint iterations, but also by using an entourage of routines that
II.4. Findings

are invoked when needed in order to support the accomplishment of Sprint goals. For example, daily stand-up and task allocation routines are employed to update plans as circumstances change. Both routines depend on the estimation of task sizes, which are required in order to inform the prioritization of product features and arrive at updated plans.

Because the universal validity of project plans is unrealistic and emergencies can cause interruptions at any time, the software development process needs to be very flexible. Both the task size estimation and task allocation routine support this flexibility because they are not triggered by a schedule, but rather by specific events. On the one hand, this facilitates the flexibility of the process based on the possibility to adjust and reallocate work packages as needed. On the other hand, it increases outcome stability, because concrete tasks are produced as a result, which contribute to the progress toward completing product features.

Changing the Game

Whereas agile software development entails a set of strict actions that are incorporated in the ostensive parts of the routines, the performative parts of the routines display variation that ranges from minor deviations to major modifications, and in some cases even to the complete elimination of a routine. For example, the Sprint iteration is an uninterrupted period of working toward the content-protected Sprint goal in agile development theory. Yet, several teams have reported stark deviations when performing this routine in the form of frequent interruptions:

It’s every day—at least—that you have disruptions [...] on one day you wanted to concentrate on some-thing, that is then just not possible (Developer, Team D).
We’re not in a world where the iteration content is frozen. I mean, people think that is the case, but it’s not the case, it’s never frozen because if the big boss comes down here and says, ‘Okay, we want people to be working on this specific task,’ that’s what’s going to happen (Developer, Team H).

Several interviews revealed a high number of interruptions also through clients calling in with emergencies in case they were unable to continue working with the product. To cope, teams rely on systems for the management of emergent tasks that allow them to dynamically allocate available resources quickly to the numerous tasks with different priorities:

You have to understand the problem and context and, if need be, say, ‘No, we won’t solve this now, we will open a ticket and solve this, say, next week, or when we have time. Or we say: no, it’s urgent, there are 20 applications affected by this, we need a solution immediately,’ and then you need to get the right people involved (Developer, Team D).

The protection of content, such as currently developed product features, as well as routine performances is abandoned in emergencies to allow for the required flexibility to react based on the activation of different routines. For example, supporting process flexibility, the task allocation routine is performed as a collaborative effort to solve emergent problems concerning the assignment of available resources to tasks with possibly changing specifications and priorities. Still, unexpected issues can cause task sizes to explode and threaten the achievement of the Sprint goal. When confronted with unexpected difficulties, teams may have to break down large tasks into smaller ones (thus, breaking up the content-protection of tasks). In order not to jeopardize the Sprint goal, teams have to decide which tasks are most
II.4. Findings

important, and which ones could be carried over into the next Sprint. In these situations, some teams stretched the routine performance and extended the duration of the Sprint against the advice of the Scrum framework. Although uncommon, performance modifications of essential routines occurred never without negotiation between developers and their supervisors, especially the Scrum Master. Afraid of altering the core framework that is protected through the Scrum Master, teams tried to avoid deviations from routine performances whenever possible, for example, by rearranging tasks to fit the temporal constraints of the Sprint routine. This was not always possible, as several participants explained how fluctuations in the workloads had led their teams to delay the end of a Sprint, describing their deviation as if they had exceeded a deadline. We collected several reports of Sprints that had been extended in order to finalize tasks that teams were unable to break down.

Whereas teams generally avoid deviations among routine performances, they are explicitly encouraged to consider modifications to the ostensive parts of the routines. During so-called retrospective meetings, team members reflect upon the potential benefits that may be gained from altering any aspect of the work process, including the discontinuation of entire routines. In one case, a participant claimed that his team had disestablished Sprint iterations, but went on to explain the same routine in a different form:

We don’t use [iterations] any more. New tasks are collected on this so-called fast track, [...] where we collect tasks and we sit together with our client every two weeks to estimate all these items and allocate them to a release, depending on how urgent they are and on the effort (Developer, Team H).

Despite severe modifications, the iterative spirit of the Sprint routine remained intact. Severe modifications included replacing
the Sprint backlog of tasks with what project management theory refers to as a Kanban workflow system, which limits the number of tasks that can occupy developers at any point in time. This allowed the team to accept new tasks into the development process more flexibly, while maintaining the Sprint planning and review in collaboration with the client in order to discuss completed and emerging tasks on a regular basis:

During the iteration, [...] you define in the beginning what you want to do [...]. And we don’t have that, but we have a stack of issues that are approved [by the client], and in the meantime, there are new issues coming in that come from the stack [which replaced the Sprint backlog] (Developer, Team H).

Although the participant views the routine as nonexistent because of the stark difference of the ostensive parts before and after the modification, it remained intact as a regular pattern of collective behavior with the purpose of dividing development into short phases of increased focus on outcome stability.

**Establishing Development Velocity**

Performance is measured by comparison of scheduled versus completed tasks. The teams track the number of completed story points during each Sprint, which is referred to as development velocity. In order to obtain realistic velocity measures, task estimation updates are frequently performed, especially because estimates often fail to deliver precise values:

Usually, when we estimate 20, it can easily be 30 (Developer, Team B).

For this process, determining which tasks to pursue is even more important than is the estimation of task sizes. The formulation of tasks involves the creation of work packages from
so-called user stories, which describe desired product functionality. Supported by Product Owners, the teams collect these during feedback sessions that integrate customers. In addition, support routines serve to determine the technical details of user requirements.

When a story was not clear, we first do an investigation, we call that a spike. The goal of the spike is to get more information, so we can estimate more comfortably. It’s important that it is timeboxed. We would, say, take four hours and then we stop (Scrum Coach, Team C).

High levels of ambiguity can pose a threat to both process and outcome stability and, therefore, detain a task from entering the workflow. Product Owners regularly invoke the task estimation routine during a Sprint, however, they may be unable to modify the current protected set of tasks to which developers agreed for the ongoing Sprint. Planning meetings, which take place once per Sprint and at the beginning of a software release cycle, are the only possibilities for non-developers, including Product Owners and other external stakeholders, to exert influence on the course of a project and to set the direction and content for future development tasks. Participants of planning meetings are invited to define the Sprint goal in order to integrate them periodically into routines by accepting their feedback on working product features.

For the teams, focusing on producing working software means that prototypes are typically presented to and discussed with clients before determining the goal of the subsequent Sprint, through which effective discussions about future directions become possible. Teams find discussions that are centered on a prototype to be more effective. One Product Owner compared the effects before and after the introduction of prototyping, and noted that:
the same instance of the project last year basically failed because of the requirements that were coming, coming, coming and never stopping, and you know, in [traditional software development methods] you first do requirements collection, then analysis, then design and so forth, and so the requirements phase was never ending. Instead, with the prototype [that we use now], you kind of stop discussions and then you can say, ‘Well, let’s focus on this part’ (Product Owner, Team F).

Whereas there are several benefits of prototyping, such as collaboration and knowledge integration, agile software developers especially value their stabilizing effects on processes and outcomes.

Switching between Tasks

Teams derive large benefits from combining flexible task allocation with shared expertise. They explicitly strive to establish redundancy across experts in order to be able to jump in, which can be a source of process stability. For example, Scrum Masters and Product Owners utilized their functional flexibility to avail their diverse skills to the developers:

I was not a manager, I was not a team lead, I was just one of them and [...] I was basically taking the worst of the jobs; whatever was really bad to do I was taking it, and I apply always the philosophy never ask [of others] what you are not willing to do yourself (Product Owner, Team F).

Their ability to jump in is, however, limited by the specific technical knowledge of some developers:
II.4. Findings

We are not in a world where everybody is interchangeable, we need to—you need a set of skills, experience and also an understanding of the business (Developer, Team H).

Jumping in can become crucial for any team member at times when unexpected events cause temporary scarcity of developers. For example, our data reveal one case where the corresponding stability proved valuable to a team that had just received a new team member when his supervisor was hospitalized:

He just joined, was there for a couple of days, and then I was out, had to go to the hospital, and he just had to do it [...] learning by doing so to say, because of an emergency (Developer and new team member’s supervisor, Team D).

What is described here as learning by doing is, however, not possible without training provided by the remaining colleagues who need to know the absent developer’s job sufficiently well and to be able to jump in due to the opportunity for task switching (i.e., the flexible allocation of tasks to developers). Without adequate establishment of process flexibility through the routines beforehand, the work flow could break down in a similar situation. Essential to the prevention of this is the sharing of knowledge, for which the daily stand-up routine provides an important template.

The knowledge sharing function of the daily stand-up routine is also crucial for new team members, as it offers a way to engage them within the operations of the team, providing an overview not only of what currently keeps each team member occupied, but also of different problems and obstacles that are reported. At the same time, new team members learn the routines by engaging in them and through their observation—a practice that fosters their reinforcement:
[The Scrum Master] introduced me to [the task management tool] and how we organize the tasks, how we log our time, how we’re going to use comments [for code change documentation], what the workflow for a task is […], and how we should coordinate that and how we can set priorities and how we can organize work based on that. […] Then apart from that, the first Sprint for me I was, I would say, just the watcher [in the routines] because I couldn’t contribute so much and wasn’t sure how I should contribute to the meetings, the planning meeting, and retrospective meeting and the code review (Developer, Team F).

There are, of course, cases when it is impossible to establish the same level of expertise in highly specialized software development jobs, and therefore alternatives must be sought. An insufficient overlap of development skills can sometimes be mitigated by establishing new routines. One team established a routine to avoid developer idleness by providing previews of upcoming work to a specialist.

We do this meeting, […] in Sprint 9 in order to define what he does next. A bit like a preview of Sprint 10 or something. So I sit together with the Product Owner... Then we did this meeting with the whole team to see how we do this, so he gets busy again (Scrum Master, Team B).

This illustrates that teams use their flexibility to exert influence upon the very development process in a permanent quest for its improvement. The employed routines support them not only as essential enablers of flexible task management and regular exchange of problems and ideas, but also increased transparency regarding work progress. Working on deeply interdependent parts of one product, teams rely on their team members’ familiarity
with each other’s problems. Through fostering the exchange of information, the stand-up routine increases shared knowledge and expertise and the extent to which mutual support is possible, which contributes to process stability as well as outcome flexibility when the solution space is enlarged.

II.5 A Model of Interdependent Routines for Balancing Stability and Flexibility

**Figure 1:** Interdependent routines are connected through routine links, which transmit triggers and information feeds. Under pressures for stability and flexibility they support the balancing of both regarding the work process and outcome.

Several links constitute the relationship between stability, flexibility, and routines. The influences we have identified are illustrated in Figure 1, which depicts a model explaining the balancing of stability and flexibility in the context of routines, as well as reciprocal relationships between flexibility, stability, and routines. It explains the interplay of routines in order to produce stability
and flexibility directed at the work process, as well as the balancing of stability and flexibility as an outcome of the software development work. While the ostensive parts of interdependent routines guide routine performances, their enactment manifests at the performative level of the routines through different triggers, which set up influencing activities that produce stability and flexibility, and can also impact the ostensive parts of the routines.

**Routine Links**

Interdependent performances of routines are based on links between the performative aspects of the routines, which permit a routine to trigger another routine, or being triggered by another routine (see Figure 1). Beside transmitting trigger signals, routine links support interactions via information flows. In contrast to control inputs from meta-routines (Nelson and Winter, 1982), routine links connect routines that are located on the same hierarchical level and can exert mutual influence. Most importantly, when information exchange between routines is triggered, both routines receive information that shapes the routine performances. For example, a routine for planning, such as the daily stand-up routine, can be halted for the estimation of a task size, which will deliver important information to continue the planning process. In turn, the task size estimation routine requires information regarding the task to be estimated, which shapes its performance (e.g., complex tasks require more information and also more time to be estimated). The flow of information is often supported by the aid of artifacts, such as the planning board, where task sizes are recorded on post-it notes. Similarly, task allocations are performed via receiving previously recorded information from the planning board, and feeding back updated information. Hence, the use of artifacts permits asynchronous information exchange between routines without requiring their simultaneous performance. This ensures that information is available when different routines are triggered either regularly, or upon
II.5. A Model of Interdependent Routines

request when needed. For example, when information must be collected based on the visualization of the project status in order to monitor and evaluate progress against a development goal, a planning routine can be performed based on the last available output of another routine – the latter does not necessarily have to be performed again unless updated information is required. Through the interplay of different routine performances, development teams become flexible with regard to shifting their focus between the planning of specific development tasks and the specific development work. As a result, frequent planning can be employed to support and complement, instead of interrupt, the development process. In support of this interplay, routines links, which are part of the performative aspects of the routines, are designed and become selectively used with regard to both the need for short-term efficiency (e.g., allocating resources quickly) and long-term effectiveness (e.g., setting a Sprint goal).

However, there are limits to this interplay in software development. In particular, some routines, like the daily stand-up and the Sprint iteration, are performed following a strict timeframe (which is noticeable in the vignette presented in the introduction). Since deviations between performances would compromise process stability, the performances receive protection (e.g., through the Scrum Master) in an attempt to produce stability with regard to processes (this is explained below). In contrast, the ostensive part can be altered deliberately upon team consensus outside of routine performances. Similar findings can be observed in different settings, where the deviations between the performative aspects of routines is less accepted than deviations between ostensive aspects. For example, in hospital teams, (Edmondson and Bohmer, 2001) find that change in the ostensive parts of routines is deliberately designed during debriefing meetings, whereas surgery teams cannot deviate from aspired performances during an operation. By contrast, process flexibility is obtained when a routine performance receives less protection. This permits modification to its ostensive part in a similar way to the overflowing effect by which
performance deviations change the ostensive parts of routines over time DAdderio:2008is. This could occur when teams regularly fail to achieve consistency in performing the Sprint iteration routine, and decide to extend or abandon it altogether.

Reciprocal Influences: Reinforcement and Redesign of the Ostensive Parts of Routines

While routines provide development teams with flexibility, they also offer a stabilizing element that reduces uncertainty. Trust in the routines’ effectiveness ensures their reinforcement through participants, a process in which agency and interaction play a major role (Howard-Grenville, 2005; Felin and Foss, 2011; Dionysio and Tsoukas, 2013). If stability of processes is established, the self-reinforcing power of routines manifests in the repetition of well-tried routine performances. Routines are learned by new team members through participating in their performances and through observing them, which again produces reinforcement. Some of the routines even span across team boundaries and include suppliers and clients. The daily stand-up meeting, for example, is often used for close collaboration with external project stakeholders in an attempt to keep their work synchronized with internal team practices.

The fact that agile software development contains a number of very short and very frequently performed routines bears potential for conflict; whereas high performance frequencies entail the opportunity for frequent change, a large number of performances tends to create inertia. Agile development teams strive to resolve these opposed forces by introducing the role of the Scrum Master to protect the performative aspect of many routines, including the Sprint iteration and the daily stand-up routine. The Scrum Master is responsible for the stability of routine performances, and aims to steer the team away from routine performance deviations. The ostensive part of a routine, however, receives less protection
– unlike routine performances, it is not shielded against modifications. Instead, the Scrum method explicitly contains routines that impact the structure of other routines through redesign or modification in order to process input variation or cope with different external constraints. Speaking to research that has asked how fast change within routines can occur (Feldman, 2000), our data suggest that routines can become subject to modifications within very short cycles of product development cycles lasting as little as a few days in the fast paced environment of software development.

**Interdependent Routines Balance Stability and Flexibility of Processes**

One long established quality of routines is the creation of stability with regard to work processes, which supports a number of operations such as coordinative activities, workload and task distribution, or knowledge documentation (March and Simon, 1958; Cyert and March, 1963). These activities occupy a central role in product development. The routines in agile software development projects appear in the classical sense of algorithms (March and Simon, 1958) that consist of sequences of steps to be followed in order to produce the outcomes that will eventually become assembled in the final software product. For example, using Sprint iterations, the participants in the routines are able to create phases of development work toward a pre-defined product increment via a set of scheduled or event-triggered activities. The frequent interruptions that developers encounter in some teams, however, pose a threat to this stability. Increasing process stability, which is the objective of the ostensive part of the Sprint iteration routine, can become threatened when routine performances starkly deviate. To secure it, consistency in routine performances is important. Process stability is therefore increased when the performative aspects of routines receive more protection. By contrast, process flexibility is increased when there is less protection.
Our data suggest that process flexibility is enabled by the deliberate, regular, and frequent interruption of the phases of stability, for example through flexibly performing different interdependent routines that exert mutual influence. This becomes possible through providing more or less protection to different routine performances. A high degree of process flexibility results from the variable sequences to complete work tasks, as well as from the dynamic allocation of resources to tasks, which are performed as part of a network of interdependent routines. Also, considerable flexibility is enabled by frequent planning and replanning. As teams are also encouraged to alter the very routines they follow, flexibility is enabled with regard to the process of the work (software development teams call this continuous improvement). Consequently, we should expect to see increasingly flexible ostensive parts of routines in order to react to changes. Yet, our data suggest that the routines themselves do not become more flexible. Although valuing flexibility, they are maintained as a source of stability.

Hence, routines establish stability that enables flexibility. For example, this happens when process flexibility, for example reflected by the modification of task sequences, is enhanced by the routines for task size estimation and resource allocation. These routines supply teams with information upon which they base decisions regarding different possible work flows. In our data, teams are regularly and frequently provided with opportunities to reflect upon and change their course of action, and reallocate resources (process flexibility), as well as revise decisions and review long-term goals (aimed at outcome stability). These activities are so important that specific roles, such as the Product Owner who is supported in this by the Scrum Master, are charged with the task of supporting them.
II.5. A Model of Interdependent Routines

Interdependent Routines Ensure Stability and Flexibility of Outcomes

Outcome stability—that is, stability regarding the outcome of a development project, such as continued or sustained product delivery in the form of regular software updates—functions as a signaling device for customers. It has long been established by evolutionary theorists as an organizationally desirable feature that is achieved through routines (Nelson and Winter, 1982). The question of how outcome stability can be sustained even when routines need to be changed in the face of dynamic environments has originally been approached on the analytical level of populations of organizations. More recent work has emphasized the role of individuals in order to explain how change can result from deviations in the performances of routines (Feldman and Pentland, 2003; Howard-Grenville, 2005; Turner and Rindova, 2012). Our study shows that outcome stability can be regulated using several interdependent routines simultaneously. Interestingly, the Sprint routine’s output of working software, which is supported by other routines, e.g. for regular code integration and testing, can deliver remarkable flexibility regarding the specific content of the very output. As a result, several routines jointly deliver outcomes in the form of software product features, which can be revised and discussed with customers before the next Sprint iteration routine, hence providing outcome flexibility (regarding the software product) at the same time. The frequent negotiation of content, which is practiced continuously (e.g., during the daily stand-up routine with a focus on short-term rather than long-term production), is an important factor for the outcome flexibility that precedes change, and eventually, innovation.

Recognizing the need for outcome flexibility, agile software development routines are devoid of content regarding technical solutions or technologies such as programming languages, code testing and validating techniques and other development-related practices. Their empty framework delivers a minimum of stabiliz-
The balance of outcome stability and flexibility is obtained in a process that provides guidance as well as openness through varying levels of protection. Specifically, the temporal protection of content during the Sprint iteration, which is broken up between two Sprints, enables both stability and flexibility of outcomes. In this sense, agile development teams do not rely on team structures based on the separation of exploration and exploitation (Benner and Tushman, 2003), but rather follow phases of cycling, sometimes very rapidly, between the creation of stability versus flexibility. In joint and regular efforts to establish stability and flexibility simultaneously, teams are able to switch rapidly between activities emphasizing one or the other through performing routines in parallel. Interdependent routine performances serve them to enhance and balance stability and flexibility.

II.6 Discussion

We studied the employment of organizational routines in product development teams, which face the challenge of balancing competing pressures in dynamic environments. They need to balance different aspects of stability and flexibility. Whether and how organizations can simultaneously attend to both has been a long-standing puzzle that researchers have addressed using different theoretical lenses. We contribute to solving this puzzle by an empirical investigation that builds on the concept of organizational routines, and extends earlier insight into their capability to balance competing demands. While earlier work has noted how par-
participants in routines draw on internal resources, such as artifacts and connections, to balance stability and flexibility (Turner and Rindova, 2012), we show how innovation teams employ several interdependent routines that are mutually influential.

Most research on routines has been carried out in teams where the desired outcome of the work is known beforehand or can be precisely specified, such as in the service sector, where companies strive to deliver homogeneity to repeat customers who value brand recognition and the reliability of identical experiences. In many situations, such as public services or medical treatment, the stability and reliability of identical outcomes is crucial and may even be safety-relevant. Researchers were able to identify factors that supported routine participants in the delivery of stable output while maintaining flexibility regarding routine performances through the investigation of routines for a wide variety of organizational undertakings from maintaining core functions to delivering products and services in diverse industries that range from finance and manufacturing to academia and public service (Zbaracki and Bergen, 2010; Howard-Grenville, 2005; Feldman, 2000; Turner and Rindova, 2012). An unanswered question, however, is related to situations in which the outcome of an endeavor is neither clearly defined nor certain, such as faced by research and development (R&D) and new product development (NPD) teams. These teams rely on routines to reliably deliver novel outcomes, oftentimes with project goals changing dramatically over time as they reflect market dynamics. To expand insight from past routine research, we study how different routines are simultaneously employed in innovation teams to establish and balance stability and flexibility. We demonstrate how product development teams engage in balancing stability and flexibility by employing interconnected routines to achieve stability and flexibility of processes as well as products (outcomes). Our empirical study speaks to the conjecture put forward by Farjoun (2010), according to which mutual influence permits flexibility and stability to enhance each other. This becomes visible when considering that stability and
flexibility can be balanced by enhancing stability through flexible routine performances in development teams. It is also visible in the flexibility that is enhanced by stable structures in the same routines. Our empirical findings are consistent with recent theoretical conceptualizations of stability and flexibility as a duality, rather than a dualism (Farjoun, 2010). A duality incorporates two seemingly opposed concepts and acknowledges that both can be mutually beneficial and simultaneously present, whereas a dualism denies the possibility of their simultaneous existence and views them as competing and exclusive. Our analysis shows that stability and flexibility can be enhanced simultaneously. Via the impact of stability or flexibility at either the level of the process or the outcome, routines can trigger change that is directed at the routines themselves, or at other routines, or at non-routine activities.

To comprehend underlying mechanisms, we build on the notion of ostensive versus performative parts of routines (Pentland and Feldman, 2005), which provides a language for the study of how routines change and are changed, as well as how their flexible performance does not necessarily induce change (Feldman, 2000; Howard-Grenville, 2005; Turner and Rindova, 2012; Zbaracki and Bergen, 2010). By drawing on the analysis of stability and flexibility, our findings both confirm and enhance the current understanding of the functioning of routines. In particular, our contribution stresses the importance of interdependences between employed routines in serving teams to acquire and balance different desired effects. Based on our analysis of routine interdependencies, we are able to address underlying mechanisms and triggers.

II.6.1 Routine Interdependence

Our demonstration of the interdependencies of routines extends the literature on routines by answering the call to investigate
how routines interact with each other (Parmigiani and Howard-Grenville, 2011). Recent advances in this direction point out the importance of considering the collective functioning of multiple routines. For example, Kremser and Schreyögg ("The Dynamics of Interrelated Routines: Introducing the Cluster Level (forthcoming)“) find that the dynamics of bundles of routines can have important implications on the organizational potential to adapt and change. The authors also note that many questions considering the dynamics of bundles of routines still remain unanswered.

By showing how multiple routines trigger and feed into each other to provide mutual input, and how their multiplicity is orchestrated in different ways to increase stability at times and flexibility at others, we highlight the importance of viewing routines in the context of their interaction with other routines. Earlier work has noted that several routines are often combined to manage organizational activities (Pentland et al., 2012), including those directed at innovation. For example, Galunic and Weeks (2002) stress the consideration of routine ecologies within organizations to study how bundles of routines collectively compete of complement each other in a similar fashion to the behavior of individual routines. By contrast, Kremser and Schreyögg ("The Dynamics of Interrelated Routines: Introducing the Cluster Level (forthcoming)“) find that dynamics of bundles of routines are different from dynamics on the routine level.

A number of researchers has noted the importance for organizations to achieve desired outcomes by employing several routines collectively, for example, by developing higher-level capabilities based on the assemblage of routines (Zollo and Winter, 2002; Salvato and Rerup, 2011). Addressing the specifics of routine interactions, Collinson (2006) point to control routines that serve as guiding mechanisms for other routines, such as for knowledge creation, in order to link “the strategic intent of the firm with the development, integration and leveraging of knowledge.” While these control routines support the routines employed for innovation activities, they do not seem to receive input from the routines
they inform, and are located at a different hierarchical level similar to the higher-level routines described in the earlier routines literature (Winter, 2003). By contrast, our study contributes important insight from the analysis of mutually influential routines.

Routines are collective phenomena, and many have recognized the importance of interactions not only within routines, but also between them (Becker, 2004). While effects of the former have been increasingly studied, the latter have unfortunately received less attention. Our data suggest that interactions between routines are relevant when routines are performed to produce mutually important information in order to move work processes forward. In this sense, systems of routines are established that may function similarly to routines themselves. Recent research in this line has noted the importance of interactions within routines. Work by Grodal, Nelson, and Siino (2015) shows how routines entail different actions that can be viewed as intertwined subroutines (i.e., intermediate parts of a routine), which are assembled as needed by participants. There are parallels in these findings despite the difference that their work delineates one routine with different subroutines, whereas our study involves different routines that can exist independently, yet are interrelated. In particular, the task estimation, task allocation and daily stand-up routine are not subroutines of the larger Sprint iteration routine. There are agile teams that use, for example, stand-up meetings without performing Sprints. Parallel to the work by Grodal, Nelson, and Siino (2015) stands the question of how routine performances are shaped in different situational contexts. In a study of helping routines, Grodal, Nelson, and Siino (2015) demonstrate how help-seekers and help-givers assemble routines differently by skipping subroutines dependent on the context at hand. For example, help-seeking was different during meetings. This implies not only that routine performances are influenced by other organizational patterns, but also that work context shapes routines by impacting the patterns of actions within them. We believe that further investigation of routine interactions provides an important and
promising avenue for future routine research.

II.6.2 Triggering Routines

Our study shows how teams establish a balance between stability and flexibility by relying on the careful employment and modification of different, interdependent routines that are triggered either by events or by a pre-defined schedule. In a recent review, Turner (2014) has noted the much neglected aspect of temporality in routines, which is paramount, yet underresearched. Based on work by Ancona, Okhuysen, and Perlow (2001), he identifies possibilities for routine performances based on clock-time (following calendars and clocks) versus event-time (following preceding actions). Noting that routine research has paid most attention to event-based triggers and the sequence of actions (e.g., service routines performed upon a customer entering a restaurant), clock-based or scheduled triggers are oftentimes reserved for routines that occupy a fixed length gained through maturity (Turner, 2014). On the one hand, routine maturity and consistency in the temporal dimension of routines may support routine participants facing time pressures, such as those that pace innovation (Eisenhardt and Tabrizi, 1995), may “restrict opportunities for deliberation and variability in the routine performances” (Turner, 2014, p. 128). On the other hand, our data show that routine performances can be subject to modification especially when time-pressure is high, such as when deadlines require additional resources and are consequently postponed. This was the case when teams exceeded the schedule of the Sprint routine, which carries a fixed amount of accomplishable, but somewhat uncertain effort.

More importantly, we find that routines following triggers based on the clock and routines following event-based triggers can work together to produce stability and flexibility. While prescheduled triggers tend to provide process stability and flexibility regarding outcomes, event-based triggers are characteristics of
flexible processes and serve to support outcome stability. To illustrate, consider the employment of two different routines, sometimes even in parallel, which produce strikingly different effects. One is the daily stand-up meeting, which has a fixed schedule and duration, and follows a precise script, thereby enhancing process stability. It provides merely a void template as a structure for decisions and discussion of varying content (i.e., its outcome is flexible). By contrast, the task estimation routine is not evoked periodically, but rather upon demand. For example, it may be triggered during the daily stand-up routine in order to generate information necessary to update the planning board so that planning activities can continue. Its content and duration are variable, because it depends on current work tasks (it supports the increase of flexibility regarding the process), yet it produces firmly established outcomes in the form of task size estimations (outcome stability). Hence, stability and flexibility can be increased simultaneously through the performance of different routines according to a variable scheme. Routines are employed like ingredients to a recipe that each have different effects but end up to an effective composition. Routine performances are triggered in different ways and may be initiated several times for teams to achieve desired effects stemming from the collection of several routines.

II.6.3 Achieving Stability and Flexibility

On a more abstract level, the routines we found in agile software development settings operate collectively on the creation of stability and flexibility to enable organizations to simultaneously become more flexible and efficient. Table 6 (see p.151) details how different routines can produce a collective effect, such as in the case of process stability, which is increased by discouraging the deviation from routines and protecting their performances. This is supported by artifacts, such as visualizations of work-flows (which are created as part of the Sprint routine) and progress visualizations (created and updated by the daily stand-up routine).
By contrast, when process flexibility is required, routine performances are less protected to allow flexibility. In particular, this is achieved by the flexible allocation of tasks to vacant developers (or vice versa), which allows performances by different participants with varying skill levels. Process flexibility is also achieved by abstaining from the time boxing of some routine performances and thereby allowing for different development velocities.

**Figure 2:** Establishing stability or flexibility results from the stark or weak protection of routine performances and content.

<table>
<thead>
<tr>
<th>Outcome dimension</th>
<th>Flexibility</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Prohibiting deviations between routine performances (high performance protection) achieves process stability. Remaining open to content changes (low content protection) leads to outcome flexibility.</td>
<td>When neither content nor routine performances may be altered (high content and performance protection), the result is stability regarding process and outcome.</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Attending quickly to dramatic requirements changes and doing so by altering processes becomes possible when content and performances receive low protection.</td>
<td>The regular delivery of working software (high content protection, e.g. via defining a Sprint goal) produces outcome stability, while the process to achieve it may be performed flexibly (low performance protection).</td>
</tr>
</tbody>
</table>

Similarly, outcome stability stems from protection, however, with regard to content (i.e. product features). Several activities support organizations to become more efficient by achieving outcome stability, including goal setting (through planning activities in the Sprint routine), goal valuation (as an outcome of task size estimations), and the creation of task awareness, which focuses the development on previously set goal (produced by the daily stand-up routine). Outcome flexibility, on the other hand, is in-
creased by the contrary; openness to content changes in the form of product modifications are incorporated in the Sprint iteration routine (where modifications are invited via customer feedback) as well as in the daily stand-up routine (where modifications may be created).

In Figure 2, we compare how the variation of protection regarding routine performances versus content permits to focus on the increase of stability or flexibility. Using different interdependent routines, this focus supports their increase either independently or simultaneously. By varying the protection levels of the routines’ performances and content, organizations profit from increased adaptability and efficiency, which ultimately stems from the employment of interdependent routines to balance process- and outcome- stability and flexibility.

II.6.4 Implications for Organizations

Shifting the focus of routine functioning and design from individual routines to bundles of routines carries important implications for organizations and their understanding of how desired organizational outcomes are achieved. While a broader perspective introduces a level of complexity that may require increased effort to dissect specific phenomena, it moves organizational analyses forward by allowing to grasp phenomena in a wider context of mutually influential elements. As a result, it becomes possible to derive important insight to complex dynamics, for example regarding organizational change, learning, and associated antecedents.

We know from studies of intraorganizational routine ecologies that bundles of routines may change collectively in favor of improved organizational functioning (Galunic and Weeks, 2002). Through the consideration of routines interactions, one can understand more of the organizational dynamics that sometimes seem to develop a life on their own. Appreciating the complexity of routine interactions and interdependencies with each other, as
II.6. Discussion

well as with routines outside the organization, it becomes possible to systematically analyze the triggers of organizational change with regard to the creation of different types of stability and flexibility, and trace the resulting effects to routines interdependencies. Increasingly complex organizational environments demand approaches to organizational change that match their complexity (Worley and Mohrman, 2014). Incorporating routine interdependencies in models of organizational change may provide fruitful for updating approaches to change management, and could render them more effective. Especially the interaction between routines via information feeds needs to be considered when designing routines. While Pentland and Feldman (2008) argue that managers cannot establish routines, but must do so via the design artifacts, it is crucial to understand the importance of the latter with regard to their role in the information exchange between interacting routines. Thus, the modification of artifacts may affect the functioning of several routines. Further, Cacciatori (2012) points out that the joint influence of several artifacts can impact routines and may have important consequences for their modification, implying that the design of complex systems of routines cannot occur independently of considering interactions with systems of artifacts (as opposed to individual artifacts).

Further, and in line with the recognition of routine complexity, considering possible systematic interactions in product development routines has implications for the organizational knowledge creation and knowledge management (Collinson, 2006). Research has shown that routine interactions can erect knowledge barriers (Dyer and Hatch, 2006). When specific knowledge is embedded in a set of routines rather than individual routines, the transfer of knowledge becomes more difficult. Implications affect outcomes that can be either attractive or undesired, such as when trying to integrate acquired organizational units, or when protecting an organizational production technologies against competitors. While shielding companies based on the difficulty to simply copy successful processes, routine interdependencies that are insufficiently un-
derstood also hinder organizations to scale practices across teams. In fact, this transfer of knowledge within organizations is a large concern with which practitioners in agile product development are preoccupied (Ambler, 2009). While our study contributes to solving related problems, there is certainly a need for more research on the interaction of routines employed in organizations.

II.6.5 Outlook

The model that we have developed draws from empirical data collected in agile software development teams. Whereas we are confident that this setting provides a good fit for the purpose of our study, a number of questions have appeared that future research could investigate in order to develop details and further understanding.

First, we have looked at small teams in the field of new product development. Although, as we have argued, this is a good starting point for the study of stability and flexibility that is linked to routines, future research should include different industries and team sizes, as well as teams that are subject to different organizational structures. This may lead to insight regarding the role of a number of concepts, such as hierarchical differences, or team autonomy, which could have an influence on the use of routines. Second, though we noted the importance of the roles of team members in our study, we did not place specific focus on this aspect in our analysis. In addition to the role specificity that has been found to influence the performance of routines (Turner and Rindova, 2012), we discovered that several participants can share collective responsibility for the maintenance of a balance between stability and flexibility through performing different routines. They monitor the complex interplay of routines, which leads to the stability of some routines (e.g., best practices) and provides opportunities to modify others at the same time (e.g., continuous improvement). The impact of specific roles on different parts of routines is a promising aspect for future research.
Third, software development teams share important organizational environments with teams in different research and development settings that produce demand for stability and flexibility. In order to test our findings in a wider context, it could be studied whether the mechanisms we observed are similarly present in other research and development areas, and different domains of product development, as well as in different industries generally. Conforto et al. (2014) suggested that the routines employed in agile software development depend on a number of factors that are independent of industry contexts. Based on their findings, increased generalizability should result from further empirical tests of our results in project teams with different characteristics and organizational settings.

II.7 Conclusion

This study analyzes the interplay of organizational routines, as well as their role in innovation settings. We contribute to the literature by empirically investigating stability and flexibility in the context of interdependent organizational routines. Our contribution integrates the view that stability and flexibility can exist as an input or output, and are mutually influential. We advance routine theory by investigating how both are established as a result of the simultaneous employment and interaction of different routines.

Our empirical analysis suggests that stability and flexibility are not mutually exclusive but can be increased simultaneously by a set of interdependent routines. Our findings capture their implementation in the dynamic environments that characterize product development settings, where interdependent routines conjointly offer possibilities for the creation of stability and flexibility in innovation settings. Discriminating between stability and flexibility directed at either the process or the outcome of the work, our results show how teams can (but not necessarily must) enhance
flexibility and stability independently, or both simultaneously. Recognizing this distinction allows us to analyze links between the routines that support the exploration of novel solutions while focusing on efficiency and stable outcomes. Some routines can be reinforced by the stability they create, and in some cases, routines are reinforced because participants have the flexibility to do so.

We detail how organizations can benefit from increasing adaptability and efficiency through varying the protection levels of the routines’ performances and content, and employing interdependent routines to balance process- and outcome- stability and flexibility. By describing a number of routines concerned with different work practices, this paper offers insight for teams that may be valid beyond product development.

Acknowledgements

The authors declare no conflict of interest. Research for this paper was funded through internal resources. A previous version of the paper was presented at the Sixth International Symposium on Process Organization Studies in Rhodes, Greece, in June, 2014. We are thankful for the valuable feedback from the conference participants. This work has also profited immensely from the valuable comments on earlier drafts provided by Scott Turner and Katharina Dittrich. We wish to extend our special gratitude to all participants of the study for their continued collaboration, dedication, and support.

References


Gilson, Lucy L et al. (2005). „Creativity and Standardization: Complementary or Conflicting Drivers of Team Effec-


Kremser, W and G Schreyögg. „The Dynamics of Interrelated Routines: Introducing the Cluster Level (forthcoming)“. In: *Organization Science*.


Scientific Paper III

The Effects of Stability and Flexibility on Performance in Product Development: AMeta-Analysis

AUTHORS
Denniz Dönmez, denniz@ethz.ch
Matthew Kerry, mkerry@ethz.ch
Gudela Grote, ggrote@ethz.ch

AFFILIATION & ADDRESS
ETH Zurich
Department of Management, Technology and Economics
Weinbergstr. 56-58, CH-8092 Zurich, Switzerland
Keywords

Stability and Flexibility; Product Development Performance; Meta-analysis.

Abstract

The product development literature widely acknowledges the growing pressures for consistent and repeated innovation in product development. To cope and remain competitive in increasingly complex and dynamic environments, product development teams need stability to operate efficiently, and also flexibility to react quickly and adequately to environmental changes. Stability and flexibility impose organizational tensions. How to reconcile them has long puzzled researchers and practitioners. Many studies have yielded mixed findings. To address existing inconsistencies in the product development literature, we conduct a meta-analysis of stability and flexibility effects on performance that aggregates results from $k = 55$ studies ($N = 11,666$). Our findings reveal that stability and flexibility are positively related, while their effects on performance are positive and comparable in size. We further report on substantive moderators in order to contextualize our findings. Based on our systematic review, we are able to elicit the use of different conceptualizations, and shed light on the relative importance of stability and flexibility for organizations. Our results suggest that, to enhance stability and flexibility, organizational behavior is more important than structure. Further, we demonstrate that stability and flexibility are independent, which indicates that both hold complementary rather than substitutionary values. These findings have important implications for product developers, who must strive to increase both simultaneously. Our contribution points out considerations for future research, and offers guidance to practitioners who must understand and manage stability—flexibility tensions toward performance.
III.1 Introduction

Product development teams face large pressures to create innovative products while operating under conditions of uncertainty in highly dynamic environments (Eisenhardt and Tabrizi, 1995; Cooper, 2001). To cope, they are required to react quickly and adequately to unexpected events, for which flexibility is vital (Larso, Doolen, and Hacker, 2009; Leana and Barry, 2000; Manz and Stewart, 1997). At the same time, stability and efficiency play an important role to remain competitive (Brown and Eisenhardt, 1997).

Both stability and flexibility are important as organizations face enormous pressure from increasingly competitive environments forcing them to continuously innovate (McDonough, Athanassiou, and Barczak, 2006; Winby and Worley, 2014). Despite their apparent incompatibility, both concepts offer complementary benefits that organizations must balance (March, 1991; Verganti, 1999). On the one hand, stability enhances efficiency, predictability and control while reducing the need for ad hoc coordination. On the other hand, flexibility allows for learning and adaptation to unstable environments.

The dual search for stability and flexibility constitutes a fundamental tension that was identified in organization theory half a century ago (Thompson, 1967). The stability—flexibility tension remains relevant to researchers and practitioners alike to this day (Benner and Tushman, 2015). For example, a cursory literature search on ‘stability’ and ‘flexibility’ using Scopus reveals that almost two thirds of roughly $k = 51,500$ studies have been published in the last ten years. Although enormous progress has been made toward solving the challenge of managing organizational tensions, many details remain unresolved (Nosella, Cantarello, and Filippini, 2012). In particular, there remains considerable inconsistency among studies regarding magnitude and direction of findings (cf. Junni et al., 2013). There are calls for increased
focus on the microfoundations of organizations, which could help clarify and delineate the effects of stability and flexibility in more detail (Felin et al., 2012; Turner, Swart, and Maylor, 2012).

To synthesize empirical findings, we focus on the product development domain, where stability—flexibility tensions have been identified to be vital for innovation (Leonard-Barton, 1992; Adler, Goldoftas, and Levine, 1999). Our meta-analytic approach permits us to elicit understanding with regard to potential moderating effects of stability and flexibility on performance, and investigate sources of inconsistency (Rosenthal and DiMatteo, 2001).

III.2 Theoretical Framework

The high demands for stability and flexibility in product development stem from seemingly contradictory needs to achieve efficiency and control while managing high levels of uncertainty. The latter often result from unforeseeable variations, for example, in demand, organizational resources, or from challenges in the context of changing technologies. Producing quick and adequate reactions requires organizations to possess considerable flexibility. At the same time, stability is vital for efficiency and competitiveness. Stressing the importance of both concepts, Cooper and Kleinschmidt (2011) state the need to react to users’ needs and efficiency of development as two key success factors of product development.

The simultaneous demand for stability and flexibility are often modeled as tensions that permeate the work of innovation teams (Smith and Tushman, 2005; Büschgens, Bausch, and Balkin, 2013). Such tensions have drawn considerable attention from researchers. For example, in a study of innovation in the product development industry, Greve (2007) explains how tensions arise from the need to attend to different goals simultaneously. Diverse conceptualizations have been proposed to resolve organizational
tensions by attending to seemingly opposed constructs that entail the spirit of stability and flexibility. One perspective includes ambidexterity, which emphasizes different solutions. Structural ambidexterity scholars consider organizational design to achieve structural separation, for example, according to product types and product development phases, so that different tensors can be attended to in parallel (Benner and Tushman, 2015).

By contrast, contextual ambidexterity researchers point to the need for business units to simultaneously establish stability and flexibility through behavioral autonomy and decision latitude (Gibson and Birkinshaw, 2004). The emphasis of structural versus behavioral stances represents an important difference between conceptualizations (Courtright et al., 2015). In an attempt to combine both, (Adler and Borys, 1996) proposed the concept of enabling bureaucracy to establish formalization and standardization with a supportive organizational context. Integrating structural and behavioral aspects provides an empowering context, in which employees can react flexibly to their environment, as well as improve organizational processes. Individuals and teams are emphasized rather than relying on organizational level mechanisms (Parker, 2014).

Brown and Eisenhardt (1997) similarly proposed the concept of semi-structures to accommodate organizational tensions. According to this perspective, organizational leaders are charged with the task to resolve tensions by creating response structures that function efficiently, yet are able to account for variation. Examples include cognitive approaches such as decision heuristics. Eisenhardt, Furr, and Bingham (2010) introduce cognition to complement structural and behavioral aspects. In line with the departure from strategy level approaches toward the consideration of behavioral and cognitive management of organizational tensions, scholarly work increasingly aims to integrate stability and flexibility at the microfoundations of organizations (Eisenhardt, Furr, and Bingham, 2010).
III.2.1 Integrating Stability and Flexibility

Organizational scholars have adopted a variety of theoretical perspectives to address the stability—flexibility tension. Terminology abounds, in part, because of abstraction and diverse literatures. For example, stability and change (Leana and Barry, 2000), and efficiency and flexibility (Eisenhardt, Furr, and Bingham, 2010) are sourced from organizational adaptation and design, respectively. Stability is much discussed in connection to efficiency (Adler, Goldoftas, and Levine, 1999; Eisenhardt, Furr, and Bingham, 2010), however, both are often used interchangeably. Flexibility tends to be addressed in connection with change (Farjoun, 2010; Turner and Rindova, 2012), to which it is an antecedent. Stability and flexibility have more recently been integrated in their simultaneous importance across literatures (Grote, 2009). In lack of consistent operationalizations of both concepts, however, we need to further demarcate them to fruitfully continue their investigation and understand their role in resolving or managing organizational tensions.

In developing precision regarding their definitions for investigation, their mutual relationship can serve as a starting point. While researchers agree that a balance needs to be established between stability and flexibility, there is an ongoing debate regarding whether or not stability comes at the cost of flexibility and vice versa. At times both concepts seem to compete, while at others they seem complementary. Eisenhardt and Tabrizi (1995), for example, associate stability with long-term planning, and flexibility with short-term iterations. While this establishes their incompatibility, other scholars avoid temporal associations, and rather regard both as complementary. In their interpretation, stability can be detected independently of flexibility (Farjoun, 2010), which is often defined as reaction to environmental stimuli (Verganti, 1999). To further this debate, we need to inquire about the essence of stability and flexibility and ask what it is that is or is not stable or flexible. Temporal aspects serve as one element in
addressing this question. When something displays no variability and does not change, it is usually measured over time. Stability then can be defined as sameness over time. Often, the background against which stability is detected includes workflows or processes. For example, the formalization of organizational processes emphasizes predictability and control, indicating stability (Ramesh, Mohan, and Cao, 2012). Here, stability refers to organizational structures, which may concern different properties. The roles in a team may be stable without the necessity for a stable team composition regarding specific people (Valentine and Edmondson, 2015). In a different view, stability and flexibility can be (and are often implicitly) defined as behavioral properties. For example, flexibility can be detected in the capacity for reactions to environmental changes.

Subtle but important differences in the operationalizations of stability and flexibility render a comparison difficult. Given the myriad of available operationalizations, the literature can best be seen as heterogeneous and fragmented. Unsurprisingly, studies of the effects of stability and flexibility have produced mixed results. We lack clarity regarding the specifics of their relative benefits for the performance of teams and organizations, as well as meaningful advice for practitioners. While there is a large body of literature, especially with regard to stability and flexibility in product development settings, the concepts have not only been operationalized inconsistently. More importantly, their joint effects have rarely been subject to investigation. Further, studies have looked at different types of performance to establish the effects of stability and flexibility.

Our contribution provides clarity with regard to the diversity in the product development literature by systematic comparison and meta-analytic estimation of the relationship between stability and flexibility, as well as moderating effects on performance. To this end, this study collects and presents different conceptualizations of stability and flexibility by retrieving and categorizing scientific articles that simultaneously measure the effects of both
concepts on performance. In addition to aggregating their effects, we reconcile prior research in order to inform product developers and facilitate future investigations.

III.3 Methods

III.3.1 Literature Search

To locate relevant studies on stability and flexibility in product development, a systematic literature search was conducted covering published and unpublished research up to and including the year 2015. To minimize potential publication bias, we included unpublished studies such as working papers and theses in our search. Eleven scientific databases and search engines were utilized for adequate sample coverage, including Web of Science, Science Direct, Google Scholar, Business Source Premier, EconLit, PsychINFO, PsycIndex (EBSCO), ABI/Inform, PsychArticles (ProQuest), JSTOR and Scopus. Search strings were based on combinations of the terms stability and flexibility in conjunction with the concept of product development. We used asterisks (i.e., stab* and flex*) to capture synonymic utilizations (e.g., stabilizing, stabilization, stable). These search strings were combined with terms referring to our outcome variable of interest, performance, as well as common forms of its operationalization borrowed from the Iron Triangle (Atkinson, 1999); cost, time (sometimes referred to as schedule) and quality.

Three different search strategies were used to locate a maximum of relevant publications (see Figure 3 for a schematic plot of the search process.) First, we searched titles and abstracts of studies for performance in combination with the simultaneous use of both stability and flexibility. Second, we repeated the search using ‘stability OR flexibility,’ however, retaining only studies containing both concepts. This strategy captured a broader range of studies, resulting in a much larger number for inclusion in our
III.3. Methods

analysis. Third, we searched the reference sections of previously identified studies and related qualitative reviews (Grote, Kolbe, and Waller, 2012; Maynard, Kennedy, and Sommer, 2015) as well as meta-analyses on related topics (Juni et al., 2013; Mueller, Rosenbusch, and Bausch, 2013; Montoya-Weiss and Calantone, 1994). We aimed to acquire studies by contacting the authors in case we did not have access to the full text.

For each search strategy, we excluded studies that did not report statistics related to our dependent variable. For example, the majority of studies was excluded either because of domain irrelevance (i.e., the constructs were conceptualized in a way that substantially differed from our definitions and did not refer to organizational or team properties but rather to material characteristics, for example, from the field of chemistry, biology, or material sciences.) We also excluded studies when they failed to report correlations of both stability and flexibility on performance, or when they were qualitative in nature.

We applied two criteria for the inclusion of studies in this meta-analysis. First, studies had to report on the effects of stability and flexibility on performance in the context of product development. Second, we only included empirical studies that are based on quantitative analyses of primary data, without discriminating against specific industry contexts or any other characteristic. Applying these criteria, the first author screened titles and abstracts of the sample with the support of a student assistant. Based on an independently screened set of $k = 50$ studies (4% of manually screened abstracts), inter-rater agreement of this process was determined at 100% as both retained the exact same six studies. The overall sum of studies retained from the different search strategies resulted in a total of 94 studies, which were subsequently coded for the effects of stability and flexibility on performance. Of these, 55 studies contained relevant data on both stability and flexibility (see Figure 3).

To extract data from the studies in our sample, we applied
a coding scheme developed on the basis of definitions that were derived from the innovation and product development literatures. To increase reliability and ensure accuracy in our data classification, the first two authors independently conducted coding on a subset of the data for focal variables, $k = 6$ (11%) and moderator variables, $k = 13$ (24%). Cohen’s kappa score for inter-rater agreement was calculated at .84 and .79, respectively, indicating near-perfect agreement (Landis and Koch, 1977). We resolved ambiguities and disagreements through discussion.

***** INSERT FIGURE 3 ABOUT HERE *****

III.3.2 Coded Variables

*Stability.* We adopt a definition of stability as the absence of variation in terms of continuity or regularity in organizational processes (Feldman and Pentland, 2003), or in terms of team- or organizational structures. Stability is, however, not necessarily a mindless absence of change in favor of continuation, but can be a purposeful intention to exploit efficiencies.

*Flexibility.* To measure flexibility, we defined it as a property of teams or organizations that enables them to react to their environments in diverse ways. For example, flexibility can be detected in the ability to take advantage of opportunities, or to mitigate negative consequences (Qin and Nembhard, 2010). Flexibility can also be detected via actions, such as changing product designs without exhaustive planning activity.

*Performance.* We coded performance with regard to new product development, project management or company success depending on the dependent variables in the primary studies. In our analyses, we grouped studies according to different units of analyses they used.

*Stability—Flexibility Conceptualizations.* We coded different conceptualizations of stability and flexibility with regard to struct-
tural aspects (e.g., process standardization, decision latitude), behavioral aspects (e.g., prioritizing team goals, rapidly adapting to circumstances), and cognitive aspects (e.g., emotional stability, openness to novelty). We provide examples for our coding scheme in the appendix.

We assigned additional codes to capture the underlying assumptions used in the conceptualizations of stability and flexibility. One distinction that has been put forward for this purpose is between structural versus behavioral aspects (Courtright et al., 2015).

In addition to dependent and independent variables, we coded a number of study characteristics, including the methodological approaches used, sample size, time and location of data collection, and industrial context (where available). While several studies reportedly focused on the manufacturing industry, we included those that measured stability and flexibility in line with our study. For example, Carson, Wu, and Moore (2012) operationalized stability as coordinating activities for problem solving, while Martínez Sánchez et al. (2009) regarding co-development and the establishment of technology transfer. Similarly, we found flexibility operationalized as the use of design iterations (Eisenhardt and Tabrizi, 1995) or abilities to customize products and respond to requirements changes (Swink, Narasimhan, and Wang, 2007), which are clearly important in product development work.

### III.3.3 Data Analysis

All analyses were conducted using Comprehensive Meta-Analysis, a software tool (version 3.0; Borenstein et al., 2014). Random-effects estimation is assumed for all main-effect analyses as well as for moderator (or subgroup) analyses. Although a random effects approach is less powerful, it permits generalization to studies outside of the study sample (Rosenthal and DiMatteo, 2001). Effect
sizes were computed for different conceptualizations of independent variables on overall performance, and on different types of performance for overall stability and flexibility. Table 10 includes a complete list of studies.

***** INSERT TABLE 10 ABOUT HERE *****

III.4 Results

Our meta-analytic results of the main effects (presented in Table 7) suggest a positive relation between stability and flexibility. Inspection of the bivariate point estimate indicated significance ($r = .18; p < .001$). Of $k = 51$ studies reporting a relationship between stability and flexibility, 74% reported a positive bivariate correlation. Fisher’s r-z transformation was conducted in order to assess the significance of the difference between the effect of stability and flexibility on performance. Results indicated non-significance ($Z(2) = -.89; p = .38$), suggesting that the effect size for stability is comparable to that for flexibility. Effects of stability and flexibility on performance were reported in all 55 studies in our sample, and are detailed below.

***** INSERT TABLE 7 ABOUT HERE *****
### Table 7: Main effects of stability and flexibility on performance

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Effect Sizes and 95% Intervals</th>
<th>Null test</th>
<th>Fail-safe k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k$</td>
<td>$N$</td>
<td>$\bar{r}$</td>
</tr>
<tr>
<td>Stability – Flexibility</td>
<td>51</td>
<td>11,157</td>
<td>.178</td>
</tr>
<tr>
<td>Stability – Performance</td>
<td>55</td>
<td>11,666</td>
<td>.233</td>
</tr>
<tr>
<td>Flexibility – Performance</td>
<td>55</td>
<td>11,666</td>
<td>.245</td>
</tr>
</tbody>
</table>

**Notes.** ES = effect size; $k$ = number of independent samples cumulated; $N$ = cumulative sample size; $\bar{r}$ = sample-weighted effect estimate of reported (uncorrected) correlations; Conf. Int. = 95% confidence interval; Cred. Int. = 95% credibility interval; $k_{FS}$ = Orwin’s fail-save $k$. 
III.4.1 Effects of Stability

Our analyses indicate a positive and significant effect of stability on performance ($r = .23$, $p < .001$). Table 8 presents results for effects grouped by different types of performance, as well as different conceptualizations of stability. Although analysis of heterogeneity reveals no significant differences between the effects of stability according performance types, effect sizes reveal a larger point estimate of stability on project performance ($r = .32; p < .001$) than on product performance ($r = .20; p < .001$) and firm performance ($r = .19; p < .001$). Further subgroup analyses indicate that the effect on overall performance is significantly stronger ($p < .05$) for behavioral aspects ($r = .34; p < .001$) than for structural aspects ($r = .18; p < .001$). We found that the majority of studies (53%) refer to stability as a structural property, whereas 38% of studies use a behavioral operationalization. Only five studies (9%) refer to stability in terms of cognitive characteristics. Based on five studies, analysis of cognitive stability did not indicate significant effects.

***** INSERT TABLE 8 ABOUT HERE *****
Table 8: Results for stability – performance relation

<table>
<thead>
<tr>
<th>SubGroups</th>
<th>Effect Sizes and 95% Intervals</th>
<th>Null test</th>
<th>Heterogeneity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DV unit of analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$k$</td>
<td>$N$</td>
<td>$\bar{r}$</td>
</tr>
<tr>
<td>Project Performance</td>
<td>17</td>
<td>3,382</td>
<td>.315</td>
</tr>
<tr>
<td>Product Performance</td>
<td>23</td>
<td>4,421</td>
<td>.198</td>
</tr>
<tr>
<td>Firm Performance</td>
<td>15</td>
<td>3,863</td>
<td>.193</td>
</tr>
<tr>
<td>Total between</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>55</td>
<td>11,666</td>
<td>.235</td>
</tr>
<tr>
<td>IV conceptualization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Behavioral Stab.</td>
<td>21</td>
<td>3,578</td>
<td>.338</td>
</tr>
<tr>
<td>Cognitive Stab.</td>
<td>5</td>
<td>772</td>
<td>.092</td>
</tr>
<tr>
<td>Total between</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>55</td>
<td>11,666</td>
<td>.219</td>
</tr>
</tbody>
</table>

Notes. $k$ = number of independent samples cumulated; $N$ = cumulative sample size; $\bar{r}$ = sample-weighted effect estimate of reported (uncorrected) correlations; Conf. Int. = 95% confidence interval; Cred. Int. = 95% credibility interval; Significant at $^{***}p < .001$; $^{**}p < .01$; $^{*}p < .05$; $^\dagger p < .10$
III.4.2 Effects of Flexibility

The main results for flexibility effects on performance (presented in Table 7) indicate a positive and significant relationship ($r = .24; p < .001$). Although analysis of heterogeneity does not reach significance for different subgroup by types of performance, Table 9 indicates that larger effects might exist on firm performance ($r = .29; p < .001$) than on project performance ($r = .23; p < .001$) and product performance ($r = .23; p < .001$). Similar to stability conceptualizations, the majority of studies (53%) adopt a structural approach to conceptualizing flexibility, whereas 38% of studies define flexibility in terms of behavior, and 9% in terms of cognitive properties. Subgroup analyses suggest that the effect of flexibility on performance is significantly larger for behavioral ($r = .32; p < .001$) than for structural approaches ($r = .22; p < .001$). Without reaching significance, analysis of cognitive flexibility indicated the smallest effect ($r = .10; p < .001$).

***** INSERT TABLE 9 ABOUT HERE *****
Table 9: Results for flexibility – performance relation

<table>
<thead>
<tr>
<th>Subgroups</th>
<th>Effect Sizes and 95% Intervals</th>
<th>Null test</th>
<th>Heterogeneity</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV unit of analysis</td>
<td>k</td>
<td>N</td>
<td>(\bar{r})</td>
</tr>
<tr>
<td>Project Performance</td>
<td>17</td>
<td>3,382</td>
<td>.225</td>
</tr>
<tr>
<td>Product Performance</td>
<td>23</td>
<td>4,421</td>
<td>.227</td>
</tr>
<tr>
<td>Total between</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>55</td>
<td>11,666</td>
<td>.245</td>
</tr>
</tbody>
</table>

IV Conceptualization

<table>
<thead>
<tr>
<th>Subgroups</th>
<th>Effect Sizes and 95% Intervals</th>
<th>Null test</th>
<th>Heterogeneity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Stability</td>
<td>29</td>
<td>7,599</td>
<td>.216</td>
</tr>
<tr>
<td>Behavioral Stab.</td>
<td>21</td>
<td>3,364</td>
<td>.319</td>
</tr>
<tr>
<td>Cognitive Stab.</td>
<td>5</td>
<td>703</td>
<td>.096</td>
</tr>
<tr>
<td>Total between</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>55</td>
<td>11,666</td>
<td>.230</td>
</tr>
</tbody>
</table>

Notes. \(k\) = number of independent samples cumulated; \(N\) = cumulative sample size; \(\bar{r}\) = sample-weighted effect estimate of reported (uncorrected) correlations; Conf. Int. = 95% confidence interval; Cred. Int. = 95% credibility interval; Significant at ***\(p < .001\); **\(p < .01\); *\(p < .05\); †\(p < .10\)
III.5 Discussion

In this study, our primary empirical questions addressed the bi-variate relation between stability and flexibility, and their effects on performance. The simultaneous importance of stability and flexibility has been widely recognized and received growing attention in recent years especially in the context of product innovation (Greve, 2007; Cooper and Kleinschmidt, 2011). There is, however, an ongoing debate regarding how development teams and their organizations can manage the tensions between stability and flexibility, and whether both can be attended to at the same time (Leana and Barry, 2000; Smith, Binns, and Tushman, 2010). Our contribution lies in adding important clarification to this debate with regard to conceptualizations that moderate the effects between stability, flexibility and performance.

We have presented meta-analytic evidence indicating that the overall relationship between stability and flexibility is positive and of moderate magnitude (Bosco et al., 2015). Further, we find significant and positive effects of both stability and flexibility on performance. There is strong evidence that both effects are significantly different from each other, with a slightly larger relative overall importance of flexibility compared to stability. This suggests that stability and flexibility constitute complements rather than substitutes, which, in turn, indicates that both can be increased simultaneously.

Inspection of credibility intervals revealed considerable between-study variability, which prompted our identification of moderators. Moderators were inspected with regard to performance (firm, product, project), as well as to operationalizations of stability and flexibility (structural, behaviors, cognition).

While we were not able to distinguish with certainty the effects related to performance types, our analyses reveal significant differences with respect to the operationalizations employed. Studies that adopt a structural view are most prevalent, referring
to the stability or flexibility, for example with regard to product development processes, product specifications, or team structures. On a more granular level, these structural properties can be divided further into temporal structures and organizational (including team) structures. The importance of structures for the increase of stability and flexibility is challenged, however, by many studies that use behavioral conceptualizations. Examples for behaviors include performing rapid design changes, reacting to fluctuations in product demand, or negotiating development team goals. Further, a number of studies employ cognitive conceptualizations, such as emotional stability, or flexibility in terms of openness to new ideas. Acknowledging these differences is vital for product development teams who are required to establish efficient workflows, but also to adapt development processes and creatively use given resources. Distinguishing between different forms of conceptualizations regarding stability and flexibility is relevant for both researchers and practitioners.

Qualitative inspection of our data suggests that behaviors of development teams as well as team leaders play an important role in creating stability and flexibility. For example, the frequent search for new uses of existing resources (Liu, Li, and Wei, 2009) and quick reactions to environmental changes (Martínez Sánchez et al., 2009) were found to have larger effects for performance than structural components such as weekly review meetings (Lynn, Skov, and Abel, 1999). As a result, our answer to the question of the magnitude of stability and flexibility effects on performance is contingent on differences with regard to operationalizations. This insight carries important implications with regard to managing organizational tensions between stability and flexibility; while their incompatibility exists within the same property (e.g., stable versus flexible team structures), tensions are relieved by combining complementary properties, such as stable structures that permit flexible behaviors.

In failing to explicate the characteristics of stability and flexibility, studies in our sample report both beneficial, as well as
detrimental effects on performance. For example, while the majority of studies confirm our expectation of beneficial net (overall) effects resulting from stability and flexibility, almost one in five studies finds a negative impact. Closer inspection reveals that the majority of studies reporting negative impacts of flexibility did not yield statistical significance on individual study levels, which may extenuate this finding to some extent. The association of (non-)significance with the direction of findings (i.e., positive versus negative effects) was statistically significant ($\chi^2_{(df=1)} = 9.63; p < .01$), and invites further inspection. Significant differences exist also in the case of stability. The largest effects of stability were found by Akgün and Lynn (2002b), Akgün and Lynn (2002a), and Akgün, Lynn, and Byrne (2006) who study influences of stable team goals and the stabilization of processes on project performance. By contrast, detrimental effects of stability were reported rather with regard to product (Bonner, Ruekert, and Walker, 2002; Larso, 2003) or firm performance (Liu, Li, and Wei, 2009) where the effects of stability were measured in terms of commitments that may produce rigidities and impede change. While, in this interpretation, stability stifles flexibility, we repeat stressing the need to map differences in observed effects to analytical differences.

III.5.1 Toward a Team Perspective

Our results make an important contribution to the management of stability and flexibility with regard to the pertinence of organizational behavior versus structure. The tensions between stability and flexibility have been researched primarily from a macro-level perspective. There are calls for increasing focus on the individual and team level especially in product development settings, where teams play an increasingly important role (Edmondson, 2013). Our study acknowledges the microfoundations in organizational tensions research and permits insight into the importance of structural, behavioral, and cognitive aspects. The three differ-
ent aspects are usually associated with different levels of analyses. We explicitly include all three to increase our understanding of organizational tensions. This provides opportunities to clarify performance-enhancing effects with regard to team and individual aspects, such as leadership and coordination.

While the importance of coordination for the management of organizational tensions has been established (Valentine and Edmondson, 2015), our study informs teams who depend on both stability and flexibility with regard to their relative importance and implications of conceptual differences. For example, role-based coordination (i.e., decoupling work tasks from individual team members) is an efficient and effective coordination mechanism that has been shown to equip teams with the capacity to balance stability and flexibility. In a study of hospital teams, Valentine and Edmondson (2015) demonstrate how role-based coordination structures foster stability, while also permitting flexibility. Our results suggest that the provision of teams with adequate (flexibility-permitting) structures is vital, support of stability- and flexibility-enhancing behaviors are even more important.

Structural and behavioral aspects of stability—flexibility tensions have been explored in a variety of team settings, including production crews and action and emergency response teams (Bechky and Okhuysen, 2011; Klein et al., 2006). Yet, there is a dearth of research in this regard in contemporary product development teams, which enjoy higher levels of team stability, but face different requirements with regard to flexibility (Hoda, Noble, and Marshall, 2013). In product development teams, flexibility is needed to produce dynamic rather than pre-defined outcomes to create innovation. In other words, these teams need flexibility to explore novelty, but also stability to exploit existing capabilities. Our findings inform teams in their endeavor to establish and balance both stability and flexibility, not only by suggesting that both may simultaneously be increased, but also by suggesting possible differences that influence their relative importance for different types of outcomes.
### III.5.2 Implications

Our study has important implications for practitioners and academics. Based on the aggregation of a large body of literature, our findings carry suggestions for teams and team leaders seeking to cope with pressures to manage stability and flexibility. They will need to carefully define the specifics of stability and flexibility that are required to achieve different goals. Our results provide a diverse set of exemplary operationalizations of stability and flexibility, and their diverse benefits for teams and organizations. To reap the benefits of both stability and flexibility, operationalizations should be derived according to the specific requirements of product development projects at hand. Failure to employ adequate operationalizations of stability and flexibility may result in their incompatibility, which compromises the endeavor to manage tensions between both concepts.

Most importantly, our study clarifies the importance of different types of stability and flexibility to manage organizational tensions. Our results show that organizational and team structures are important to facilitate the simultaneous increase of stability and flexibility in product development. To balance both, however, team and individual behaviors are even more important. Research points to the importance of equipping teams with adequate levels of autonomy to manage product development (Parker, 2014; Hoda, Noble, and Marshall, 2013). For successful product development, stability and flexibility enhancing behaviors should be encouraged on both team and individual levels.

Managers should consider differences of stability and flexibility effects for several levels of performance. Although performance-related subgroup analyses yield low levels of significance and fail to explain heterogeneity within groups, which suggests that further complexities are likely govern performance effects, implications may be derived from our analyses regarding the relative importance of stability versus flexibility for different performance types. More studies in our sample report larger
influences of stability than of flexibility on project performance. By contrast, studies of firm performance tend to find larger effects of flexibility than of stability. Based on our data, one could infer that, although both stability and flexibility are simultaneously vital, flexibility is relatively more important for firm performance, while stability is relatively more important for project performance. For product performance, the effect sizes of stability and flexibility were similar. While our findings leave room for the further exploration of moderating effects, they enable us to address the overarching question of whether stability and flexibility compete with each other, or rather complement each other in fulfilling important organizational functions (Farjoun, 2010; Ghemawat and Costa, 1993). Our results leave no doubt in the simultaneous importance as well as the possibility and necessity to increase both simultaneously to balance both in support of managing organizational tensions.

Acknowledging the relative importance of stability and flexibility, practitioners may find the list of conceptualizations extracted from the product development literature (Table 7) helpful. Our findings can serve as a starting point in defining elements required for the design of strategies to increase stability and flexibility simultaneously. Our results may also guide managers in defining indicators of stability and flexibility in order to develop more precise performance measures.

In addition to industry practitioners, researchers aiming to further our knowledge about stability and flexibility may also find the evidence we have compiled useful as a foundation upon which further studies can be conducted. One important step in the work on organizational tensions will be the adoption of consistent definitions of stability and flexibility to base future endeavors on a universal language. It is our hope that our results will inform the development of more thorough theory with regard to stability and flexibility beyond the context of product development.
III.5.3 Limitations and Future Research

There are several shortcomings in our study that we wish to address in the hope that they might guide the interpretation of our results, as well as inspire future research.

First, results of the moderator analyses we have performed did not yield highly significant differences between different types of performance. Although different effect sizes could be computed, we were not able to establish meaningful results on the basis of further subdividing already small sample sizes in each subgroup. The high levels of heterogeneity found among the subgroups suggest that further and more complex relationships govern the effects of different types of stability and flexibility on different types of performance. The model we were able to test is likely to suffer from oversimplification, and future studies should dedicate further effort to fully clarify moderating effects. Second, in relation to heterogeneity issues that limit our power to further detail the role of moderators, we were only able to analyze the more robust subgroups in our sample. Coding practices with regard to our sample were inhibited due to a lack of uniformity in primary studies. Difficulties result from the failure of studies to follow standards in describing their samples. For example, we were not able to discern effects for potentially relevant firm characteristics, such as firm sizes, R&D expenditure, or characteristics related to the organizational culture. Differences in the latter, for example, might further explain behavioral aspects. Also, we were not able to distinguish industry context without ambiguities, as almost one in two studies reported combined results for various industries. We invite future research to specify more details regarding the collection of data, and report more detailed analyses. As better understanding of moderating effects is likely to result from such practices, we urge researchers to develop standards of reporting more details regarding the research methods and sample characteristics.

Further, and perhaps more importantly, our study uncovers
distinct and equally valid possibilities to measure stability and flexibility in the absence of a single agreed-on operationalization, which invite further efforts directed at establishing sharper definitions. Our systematic review of the product development literature uncovers a rapidly increasing number of studies that explore a variety of aspects linked to the performance-enhancing effects of stability and flexibility. On the one hand, this leads to rapid growth in knowledge. On the other hand, considerable heterogeneity is produced, which fragments the product development literature. Many studies adopt implicit assumptions regarding both concepts. In line with criticism by Evanschitzky et al. (2012), we detect a tendency in the product development literature to borrow theoretical concepts from a variety of other fields. This fuels shortcomings with regard to the development of conceptual consistency, which is required to produce higher theoretical precision. As a result, the employment of different analytical operationalizations results in heterogeneity at the cost of the inability to derive generalizable implications. Inconsistencies in the conceptualizations of stability and flexibility also greatly increase the difficulty to compare results, and should caution future researchers to explicitly define their underlying assumptions regarding stability and flexibility.

The fragmented literature on stability and flexibility calls for more precision to defining and researching these concepts. Departing from our overview of the literature, future research should consider increasing our understanding of the ways in which stability and flexibility impact teams and their organizations. In order to discern the specifics of enhancing and balancing both, investigations of antecedents, boundary conditions and potential dependencies could be conducted based on the types of conceptualizations we elicit. We stress again that any attempt to investigate stability and flexibility should be based on clear definitions of the two concepts in order to arrive at more accurate findings. With a clear understanding of the concepts as a prerequisite to detect and correctly measure them in organizational settings, we
encourage future research to extend our findings.

III.6 Conclusion

In this study, we leverage the strength of meta-analytical techniques to investigate the impact of stability and flexibility on performance. We systematically aggregate and organize findings from the field of product development. To our knowledge, this is the first study that meta-analytically investigates the effects on performance exerted by stability and flexibility, both of which are highly important, yet difficult to achieve at the same time. We shed light on the poorly understood management of organizational tensions that challenge product developers to carefully balance stability and flexibility through the pursuit of seemingly incompatible goals. By delineating different conceptualizations of stability and flexibility, we offer important insight that contributes to the literature on product development. We clarify the question of how to resolve the competing demands of stability and flexibility and add support to the reconciliation of organizational tensions. Our findings point to the independent importance of stability and flexibility on performance in product development, suggesting that both can (and need to) be increased simultaneously. Specifically, we highlight the relative importance of behavioral approaches compared to organizational and team structures. We discuss managerial implications that may guide product developers to manage organizational tensions more effectively with regard to the contradictory requirements they pose in innovation processes (Andriopoulos and Lewis, 2009; Miron-Spektor, Erez, and Naveh, 2011). It is our hope that our contribution will support scholars in producing further research on stability and flexibility with more precision to guide product development teams.
Acknowledgements

The authors declare no conflict of interest. We are thankful to Sarah Kriech for her valuable assistance with our review of the literature. We also wish to extend our gratitude to Dana Unger, Anja Schulze and Jan Schmutz for providing valuable feedback on earlier drafts of the manuscript.

References


Bonner, J M, R W Ruekert, and O C Walker (2002). „Upper management control of new product development projects and
References

Borenstein, M et al. (2014). Comprehensive Meta-Analysis Version 3 [computer software].


DaSilveira, Giovani J C and Raffaella Cagliano (2006). „The relationship between interorganizational information systems and


González Benito, Javier (2010). „Supply strategy and business performance“.


In: Industrial and corporate change 16.5, pp. 945–975.


Heim, Gregory R and David Xiaosong Peng (2010). „The impact of information technology use on plant structure, practices, and performance: An exploratory study“.

Herhausen, D and M Schögel (2014). „Co-marketing capability: scale development and performance implications“.
In: Problems and Perspectives in Management.


Huckman, Robert S and Bradley R Staats (2011). „Fluid Tasks and Fluid Teams: The Impact of Diversity in Experience and Team Familiarity on Team Performance“.


Li, Yuzhu et al. (2011). „The role of team problem solving competency in information system development projects“. In: *JPMA* 29.7, pp. 911–922.


References


**Figure 3:** Flow chart of the study selection process and criteria for exclusion of studies.

1st STRATEGY

236 Studies containing *Stability AND Flexibility* in Title or Abstract

- 231 Studies Excluded
  - (72 duplicates, 6 inaccessible (authors were contacted), 153 studies without org. / team focus (e.g. chemistry) or without relevant statistical data)

- 5 Studies Remained in Sample (=S1)

2nd STRATEGY

3321 Studies containing *Stability OR Flexibility* in Title or Abstract

- 2297 Studies Excluded
  - (no references to organizational/team characteristics)

- 839 Studies Excluded
  - (251 duplicates, 419 qualitative studies or without relevant stat. data, 169 not related to NPD in team / org. context or did not measure *both* stability and flexibility)

- 104 were inaccessible (Authors Contacted via Email)

- 5 Relevant Studies Obtained from Authors

- 86 Studies Remained in Sample (=S2)

3rd STRATEGY

3 Studies Found Manually, e.g., via References (=S3)

- 94 Studies were Subjected to the Coding Process (=S1+S2+S3)

- 39 Studies Contained Insufficient Data

55 Studies Constitute the Final Sample for This Meta-Analysis
Table 10: Complete list of studies in the sample including their operationalizations of stability and flexibility

<table>
<thead>
<tr>
<th>Author(s), Year</th>
<th>Context (Industry)</th>
<th>Method</th>
<th>Operationalization of stability</th>
<th>Operationalization of flexibility</th>
<th>Perform. Measure</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akgün and Lynn (2002a)</td>
<td>Various industries</td>
<td>Surveys and interviews</td>
<td>Stability of project goals including technology and product design (S)</td>
<td>Incorporating changes based on learnings from pre-launch to full-scale launch, and uncovering and correcting product problem areas (B)</td>
<td>Project</td>
<td>Stability and flexibility equally influence project performance, defined as expectations regarding sales, ROI, market share, customer and senior management expectations</td>
</tr>
<tr>
<td>Akgün and Lynn (2002b)</td>
<td>USA, various industries</td>
<td>Survey</td>
<td>Stability of project goals (S)</td>
<td>Daily review meetings to discuss the status of the project, which provide opportunities for change (S)</td>
<td>Project</td>
<td>Stability and flexibility are associated with faster than expected product development and launch</td>
</tr>
<tr>
<td>Akgün et al. (2007)</td>
<td>USA; no specific industry reported</td>
<td>Survey</td>
<td>Variability in product development procedures, information sharing mechanisms, project plans, and use of tools (inv) (S)</td>
<td>Flexible processes and product features, not strictly following a plan (S)</td>
<td>Project</td>
<td>Flexibility-but not stability-is associated with project performance</td>
</tr>
</tbody>
</table>

Notes. (S) = Structural; (B) = behavioral; (C) = cognitive conceptualization of stability / flexibility; (inv) = inversely coded after carefully inspecting the according measurement instrument items. In five cases, correlations were included in inversed form, as measures were employed in the exact opposite sense of our understanding of stability or flexibility.
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akgün, Lynn, and Byrne (2006)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Implementing knowledge based on uncovering problem areas with the aim to stabilize processes (B)</td>
<td>Changing product development procedures, project plans, information-sharing mechanisms and product development tools (B)</td>
<td>Project</td>
<td>Stability—but not flexibility—is associated with project success</td>
</tr>
<tr>
<td>Aronson, Reilly, and Lynn (2006)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Having a common team goal, resolving conflicts, sharing information (B)</td>
<td>Openness to new ideas and innovation, creativity (C)</td>
<td>Project</td>
<td>Behavioral and cognitive measures of stability and flexibility are positively related to project performance</td>
</tr>
<tr>
<td>Aronson, Reilly, and Lynn (2008)</td>
<td>Various US industries</td>
<td>Survey</td>
<td>Leader’s emotional stability (C)</td>
<td>Openness to new ideas for innovation, and creativity (C)</td>
<td>Project</td>
<td>Cognitive stability and flexibility are related to performance</td>
</tr>
<tr>
<td>Barnes and Liao (2012)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Delivering products reliability (B)</td>
<td>Rapidly handling complaints and demand for nonstandard features (B)</td>
<td>Product</td>
<td>Stability and flexibility support achieving low cost of production, inventory, transportation and handling</td>
</tr>
<tr>
<td>Bierly and Daly (2007)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Improving efficiency and refining existing technologies (B)</td>
<td>Experimenting with new ideas and challenging conventional ones, developing breakthrough technologies (B)</td>
<td>Firm</td>
<td>The effect of stability and flexibility on firm performance is almost equally strong.</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Context</td>
<td>Method</td>
<td>Stability</td>
<td>Flexibility</td>
<td>Perf. Type</td>
<td>Key Findings</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Bonner, Ruekert, and Walker (2002)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Degree to which processes and tasks are specified for the team (S)</td>
<td>Team’s influence on project goals and objectives, budget, deadlines and team members (S)</td>
<td>Product</td>
<td>The effect on product performance (adherence to budget, cost and schedule, meeting performance expectations) is positive for operational influence (flexibility) and negative for process control (stability)</td>
</tr>
<tr>
<td>Carson, Wu, and Moore (2012)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Prioritizing ideas through retention and focus, eliminating alternatives (B)</td>
<td>Generating variants of ideas (B)</td>
<td>Product</td>
<td>Stability and flexibility are positively related to new product performance</td>
</tr>
<tr>
<td>Chen, Reilly, and Lynn (2012)</td>
<td>N/A</td>
<td>Survey</td>
<td>Experience with similar tasks from similar previous projects (C)</td>
<td>Autonomy to make decisions without consulting management (S)</td>
<td>Project</td>
<td>Experience and autonomy are positively related to meeting or exceeding expectations regarding sales, ROI, market share, customer and senior management expectations</td>
</tr>
<tr>
<td>Cui et al. (2015)</td>
<td>Various industries</td>
<td>Survey</td>
<td>IT systems support efficient information exchange (S)</td>
<td>IT systems allow for rapid Firm changes, are scalable, can accommodate new functionality (S)</td>
<td></td>
<td>IT related stability and flexibility are positively related to financial performance (sales growth)</td>
</tr>
</tbody>
</table>
### Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>DaSilveira and Cagliano (2006)</td>
<td>Internat’l manufacturing industry</td>
<td>Survey</td>
<td>Reliability with regard to delivering products and lead-time performance (S)</td>
<td>Ability to customize products, and change the product mix (S)</td>
<td>Product</td>
<td>Stability and flexibility are positively related to product quality</td>
</tr>
<tr>
<td>Dayan and Di Benedetto (2009)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Stable composition of the development team (S)</td>
<td>Different functions in the team, which -it is argued- should precede flexibility (scale based on Brown and Eisenheart, 1995) (S)</td>
<td>Product</td>
<td>Team stability is stronger related to product success than team flexibility</td>
</tr>
<tr>
<td>Eisenhardt and Tabrizi (1995)</td>
<td>Global manufacturing industry</td>
<td>Survey</td>
<td>Effort directed at laying out a predictable series of steps (B)</td>
<td>Using design iterations (redesigning at least 10% of the product’s parts each time) (B)</td>
<td>Project</td>
<td>The effect on development time (project schedule attainment) is positive for requirements planning (stability) and negative for adaptation (using iterations)</td>
</tr>
<tr>
<td>González Benito (2010)</td>
<td>N/A</td>
<td>Survey</td>
<td>Importance attributed to stability, e.g. with regard to supplier reliability (C)</td>
<td>Importance attributed to flexibility, e.g. regarding expectations toward suppliers to adapt to company needs (C)</td>
<td>Product</td>
<td>Performance increases when the relative importance of flexibility is increased, but firms need to combine both with quality</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Context</td>
<td>Method</td>
<td>Stability</td>
<td>Flexibility</td>
<td>Perf. Type</td>
<td>Key Findings</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
<td>--------------</td>
<td>----------------------------------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Gopal et al.</td>
<td>US automotive industry</td>
<td>Survey</td>
<td>Stability in productivity (inv) (S)</td>
<td>Ability to account for variety in products (S)</td>
<td>Firm</td>
<td>There is a positive relationship of flexibility-but not stability-with output in terms of new product launches</td>
</tr>
<tr>
<td>Heim and Peng</td>
<td>Various industries in 8 countries</td>
<td>Survey</td>
<td>Reduced variance in processes (S)</td>
<td>Customized process with which to account for variances (S)</td>
<td>Product</td>
<td>Stability-but not flexibility-is strongly related to customer satisfaction with products</td>
</tr>
<tr>
<td>Herhausen and Schögel</td>
<td>Various industries</td>
<td>Survey</td>
<td>Stable bonds with firm alliance partners (S)</td>
<td>Flexibility in resources or responsibilities regarding alliance partners (S)</td>
<td>Firm</td>
<td>Stability is much stronger related to firm performance than flexibility.</td>
</tr>
<tr>
<td>Huckman and Staats</td>
<td>Indian software dev. industry</td>
<td>Document analysis</td>
<td>Adherence to project schedules (B)</td>
<td>Flexibility to respond to changing tasks (S)</td>
<td>Product</td>
<td>Stability and Flexibility are both negatively correlated with product defects (i.e., a lack of quality)</td>
</tr>
<tr>
<td>Huo, Zhao, and Lai</td>
<td>International manufacturing industry</td>
<td>Survey</td>
<td>Striving to integrate different functions to collaborate and coordinate activities for problem solving (B)</td>
<td>Flexibility to change the product mix (S)</td>
<td>Product</td>
<td>Stability and flexibility are positively related to product performance with regard to acceptance and reliability</td>
</tr>
</tbody>
</table>
Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joshi and Sharma (2004)</td>
<td>Canadian manufacturing industry</td>
<td>Survey</td>
<td>Changing customer preferences (inv) (S)</td>
<td>Developing and testing new ideas, implementing iterations based on customer feedback (B)</td>
<td>Product</td>
<td>Flexibility—but not stability—is associated with product success</td>
</tr>
<tr>
<td>Kortmann (2014)</td>
<td>Various industries in India and the US</td>
<td>Survey</td>
<td>Efficiency regarding operations and processes (B)</td>
<td>Encouraging the seeking new ideas and remaining open to new technologies (B)</td>
<td>Project</td>
<td>Flexibility is stronger related to project performance than stability. Geogr. differences augment the effects, which are larger for the US than for India</td>
</tr>
<tr>
<td>Kristal, Huang, and Roth (2010)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Stability regarding the quick delivery of quality products to customers (B)</td>
<td>Ability to rapidly modify product development methods (S)</td>
<td>Product</td>
<td>Product performance increases more with flexibility than with stability</td>
</tr>
<tr>
<td>Larso (2003)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Efficiency and productivity, reliability, process stability (S)</td>
<td>Ability to modify existing Project products (S)</td>
<td>Project</td>
<td>Both stability and flexibility are pos. related to sales growth, profit margins, and ROI</td>
</tr>
<tr>
<td>Larso, Doolen, and Hacker (2009)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Efficiency and productivity, reliability, process stability (S)</td>
<td>Ability to make modifications to existing products (S)</td>
<td>Product</td>
<td>Both stability and flexibility are positively related to product success</td>
</tr>
<tr>
<td>Lau, Yam, and Tang (2007)</td>
<td>Hong Kong manufacturing industry</td>
<td>Survey</td>
<td>Standardization; specification of interfaces (S)</td>
<td>Offering rapid design changes and wider product ranges (B)</td>
<td>Product</td>
<td>Stability and flexibility are positively related to new product performance</td>
</tr>
</tbody>
</table>
Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al. (2011)</td>
<td>IT industry</td>
<td>Survey</td>
<td>Anticipation: capitalization on experiences from past projects, proactively managing risks (B)</td>
<td>Flexible reactions: prototyping, over-allocating resources to allow experimentation (B)</td>
<td>Product</td>
<td>Stability and flexibility are associated with product quality and long-term customer satisfaction</td>
</tr>
<tr>
<td>Li et al. (2010)</td>
<td>Chinese IT industry</td>
<td>Survey</td>
<td>Efficiency in responding to change (B)</td>
<td>Rapidly adapting to changes in the business (B)</td>
<td>Product</td>
<td>Stability and flexibility are strongly related to the quality of software features</td>
</tr>
<tr>
<td>Liu, Li, and Wei (2009)</td>
<td>Various industries in China</td>
<td>Survey</td>
<td>Stability of financial and technological resources (S)</td>
<td>Frequently finding new uses for existing resources (B)</td>
<td>Firm</td>
<td>Flexibility-but not stability-is related with capabilities to market new products</td>
</tr>
<tr>
<td>Lynn, Skov, and Abel (1999)</td>
<td>N/A</td>
<td>Survey</td>
<td>Keeping the goal stable as a team (B)</td>
<td>Weekly meetings to review the product/prototype, discuss action items and customer reaction reports (S)</td>
<td>Project</td>
<td>Stability and flexibility influence project performance</td>
</tr>
<tr>
<td>Martínez Sánchez et al. (2009)</td>
<td>Spanish manufacturing industry</td>
<td>Survey</td>
<td>Cooperating with customers and suppliers regarding co-development and technology transfer (B)</td>
<td>Reacting quickly to customer requirements, overcoming changes in the environment (B)</td>
<td>Project</td>
<td>Both product and process innovation are associated with stability and flexibility</td>
</tr>
</tbody>
</table>
Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>McComb, Green, and Dale Compton (2007)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Provision of technical solutions and processes (S)</td>
<td>Adapting working styles, adjusting approaches &amp; handling a variety of tasks (B)</td>
<td>Project</td>
<td>Stability is stronger related with performance than is flexibility</td>
</tr>
<tr>
<td>Menor, Kristal, and Rosenzweig (2007)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Structural capital in the form of standardization (S)</td>
<td>Ability to adjust or modify processes to speedily accommodate changes (S)</td>
<td>Product</td>
<td>Both stability and flexibility are strongly related to product innovation</td>
</tr>
<tr>
<td>Miron-Spektor and Beenen (2015)</td>
<td>N/A</td>
<td>Experiment with students</td>
<td>Clear idea of the product (C)</td>
<td>Variety of knowledge domains used (C)</td>
<td>Product</td>
<td>The impact of stability is stronger than that of flexibility on the novelty and usefulness of the product and its likelihood of success</td>
</tr>
<tr>
<td>Patel, Terjesen, and Li (2012)</td>
<td>US manufacturing industry</td>
<td>Survey</td>
<td>Stability of alliances in R&amp;D, manufacturing and other org. functions (S)</td>
<td>Flexibility in producing different product features (S)</td>
<td>Firm</td>
<td>Flexibility has a slightly larger effect than stability on sales growth, employee growth, and operating profit growth</td>
</tr>
<tr>
<td>Prajogo and McDermott (2011)</td>
<td>Var. product develop. industries in Austral.</td>
<td>Survey</td>
<td>Stability and continuity through routinization, formalization and structure (S)</td>
<td>Flexibility based on decentralization, innovation and change (S)</td>
<td>Product</td>
<td>Flexibility is stronger related to product performance than stability</td>
</tr>
</tbody>
</table>
Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qu et al. (2012)</td>
<td>N/A</td>
<td>Survey</td>
<td>Stability of process use (inv) (S)</td>
<td>Changes to organizational Firm structure and practices, including strategy (S)</td>
<td></td>
<td>Flexibility (structural changes) but not stability (of process use) positively impact market share and customer base</td>
</tr>
<tr>
<td>Roh (2009)</td>
<td>International manufacturing industry</td>
<td>Document analysis</td>
<td>Programmes designed to increase performance (S)</td>
<td>Ability to produce promptly and accurately according to customer requests (S)</td>
<td>Firm</td>
<td>Stability and flexibility have positive effects on increases in sales or market share</td>
</tr>
<tr>
<td>Salomo, Weise, and Gemuenden (2007)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Stable project goals, resource allocation and project timeframe (S)</td>
<td>Flexibly allocating resources, scheduling tasks, disseminating or integrating information (B)</td>
<td>Project</td>
<td>Stability and flexibility increase project success (time, budget) and market success</td>
</tr>
<tr>
<td>Sethi and Iqbal (2008)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Stable gate review criteria that are formalized and strictly adhered to (S)</td>
<td>Easily changeable project parameters and room for negotiation (inv) (S)</td>
<td>Product</td>
<td>There is a strong correlation between stability and inflexibility</td>
</tr>
<tr>
<td>Song et al. (2011)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Formal processes that lay out explicit procedures and long-term objectives (S)</td>
<td>Intentional overlap of skills and resources to motivate the exploration of different perspectives and options (S)</td>
<td>Firm</td>
<td>Firm financial performance is positively related to stability and flexibility</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Context</td>
<td>Method</td>
<td>Stability</td>
<td>Flexibility</td>
<td>Perf. Type</td>
<td>Key Findings</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Swink, Narasimhan, and Wang (2007)</td>
<td>Manufacturing industry</td>
<td>Survey</td>
<td>Delivering accurately, dependably, and speedily (B)</td>
<td>Ability to respond to changes in requirements and customize products (S)</td>
<td>Firm</td>
<td>Stability and flexibility are positively correlated with managers’ perception of customer satisfaction</td>
</tr>
<tr>
<td>Tatikonda and Rosenthal (2000)</td>
<td>USA, various industries</td>
<td>Survey</td>
<td>Formalization of procedures (S)</td>
<td>Ability to reallocate financial, personnel and equipment resources (S)</td>
<td>Project</td>
<td>Stability and flexibility are related to the success of the development project execution</td>
</tr>
<tr>
<td>Veldhuizen, Hultink, and Griffin (2006)</td>
<td>High-tech industry in the Netherlands</td>
<td>Survey</td>
<td>Establishing information flows, e.g., regarding customer satisfaction assessments (B)</td>
<td>Adopting new technologies even at cost if they are superior, focusing on new products (B)</td>
<td>Product</td>
<td>Stability and flexibility are almost equally important for customer assessed performance</td>
</tr>
<tr>
<td>Vicente et al. (2015)</td>
<td>Manufacturing industry in Portugal</td>
<td>Survey</td>
<td>Stable levels of know-how and long-term capabilities (S)</td>
<td>Flexibility regarding customer needs and to coordinate among departments (S)</td>
<td>Firm</td>
<td>Flexibility to adapt to customer needs is more important for firm performance than long-term capabilities</td>
</tr>
<tr>
<td>Wang, Libaers, and Jiao (2014)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Maintaining common goals and vision to ‘unify the future trajectory’ (B)</td>
<td>Reformulating strategies to changing market conditions and adapting operating modes (B)</td>
<td>Firm</td>
<td>Capacity to effectively produce product variety and offer new products/services</td>
</tr>
<tr>
<td>Webster and White (2009)</td>
<td>US and Japanese retail ind.</td>
<td>Survey</td>
<td>Organizational values that reflect reliability and predictability (C)</td>
<td>Organizational culture that is characterized by innovation orientation (C)</td>
<td>Firm</td>
<td>Values that reflect flexibility—but not stability—are positively related to firm performance</td>
</tr>
</tbody>
</table>
Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wei, Yi, and Guo (2013)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Refining existing ideas and implementations, relying on proven methods (B)</td>
<td>Exploring new solutions, striving to extend knowledge and information (B)</td>
<td>Product</td>
<td>Stability and flexibility regarding learning are positively related to product success rate and market potential, time to market</td>
</tr>
<tr>
<td>Wong, Boon-itt, and Wong (2011)</td>
<td>Thai manufacturing industry</td>
<td>Survey</td>
<td>Emphasizing information flows and responsiveness, integrating different functions (B)</td>
<td>Capability to make rapid changes and customize product features (S)</td>
<td>Product</td>
<td>Stability and flexibility are positively related to meeting customer needs</td>
</tr>
<tr>
<td>Worren, Moore, and Cardona (2002)</td>
<td>US and UK home appliances industry</td>
<td>Survey</td>
<td>Standardization of processes (S)</td>
<td>Fostering team autonomy; encouraging innovation and flexibility (B)</td>
<td>Firm</td>
<td>Standardization and flexibility have strong effects on firm financial performance</td>
</tr>
<tr>
<td>Yang and Lai (2006)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Availability of critical knowledge through intranet and communication tools whenever needed (S)</td>
<td>Ability to quickly collect and analyze new information and using it to adopt strategies (S)</td>
<td>Firm</td>
<td>Stability and flexibility might be negatively correlated to firm performance (results were not significant)</td>
</tr>
<tr>
<td>Yli-Renko, Sapienza, and Hay (2001)</td>
<td>Various industries</td>
<td>Survey</td>
<td>Stability regarding collaboration structures with the customer (S)</td>
<td>Flexibility regarding contract details (S)</td>
<td>Product</td>
<td>The positive effect of structural stability may be linked to high switching costs</td>
</tr>
</tbody>
</table>
Table 10 (continued): Sample studies including conceptualizations

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Context</th>
<th>Method</th>
<th>Stability</th>
<th>Flexibility</th>
<th>Perf. Type</th>
<th>Key Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang, Hu,</td>
<td>Chinese manufact. industry</td>
<td>Survey</td>
<td>Controlling processes by exerting influence over distribution partners (B)</td>
<td>Modifying agreements when unexpected situations arise (B)</td>
<td>Firm</td>
<td>Flexibility is stronger associated with performance than is stability</td>
</tr>
<tr>
<td>and Gu (2008)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following pages contain supplementary tables and forest plots which we would like to offer for inclusion in an online appendix.
Table 11: Variable Codebook for Stability – Flexibility Conceptualizations

<table>
<thead>
<tr>
<th>Focal Variable</th>
<th>Operational Definition</th>
<th>Sample Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural</td>
<td>Organizational provisions made for teams / product developers within which they can work, for example, in terms of infrastructure, resources (financial, etc.), process standardization (stability), or the endowment of teams with decision latitude (flexibility). The goal here is to contrast structures to cognitive or behavioral aspects.</td>
<td>Stability: Stable composition of the development team (Dayan and Di Benedetto, 2009). Flexibility: Team’s influence on project goals and objectives, budget, deadlines and team members (Bonner, Ruekert, and Walker, 2002)</td>
</tr>
<tr>
<td>Behavioral</td>
<td>Team or individual actions, such as assigning priorities to product features; developing a set of design solutions; regularly revisiting the desired product functionality; adhering to team goals (stability); or altering collaboration techniques; rapidly adapting to environmental stimuli (flexibility). Actions may be stable over time, or, by contrast, flexible.</td>
<td>Stability: Adherence to project schedules (Huckman and Staats, 2011). Flexibility: Experimenting with new ideas and challenging conventional ones, developing innovative technologies (Bierly and Daly, 2007)</td>
</tr>
<tr>
<td>Cognitive</td>
<td>Mental processes of individuals. Examples include decision heuristics, emotional stability or firmly established ideas (stability), or creativity, openness to novelty, etc. (flexibility).</td>
<td>Stability: Leader’s emotional stability (Aronson, Reilly, and Lynn, 2006). Flexibility: Openness to new ideas and innovation, creativity (Aronson et al., 2006)</td>
</tr>
</tbody>
</table>
### Table 12: Variable Codebook for Criterion Level

<table>
<thead>
<tr>
<th>Focal Variable</th>
<th>Operational Definition</th>
<th>Sample Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firm</td>
<td>Performance measure employed at the level of the organization</td>
<td>Firm performance, defined as Increase in market share and customer base (Qu et al., 2012); Capability to introduce new products to the market (Liu, Li, and Wei, 2009)</td>
</tr>
<tr>
<td>Product</td>
<td>Performance measure employed at the level of the product</td>
<td>Product quality, conformance with specifications and reliability (Prajogo and McDermott, 2011); Number of product defects (Huckman and Staats, 2011)</td>
</tr>
<tr>
<td>Project</td>
<td>Performance measure employed at the level of the development project</td>
<td>Performance in product and process innovation (Martínez Sánchez et al., 2009); Project efficiency (McComb, Green, and Dale Compton, 2007)</td>
</tr>
</tbody>
</table>
Figure 4: Results for stability—flexibility bivariate correlations.

<table>
<thead>
<tr>
<th>Study name</th>
<th>Statistics for each study</th>
<th>Correlation and 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correlation</td>
<td>Lower limit</td>
</tr>
<tr>
<td>Akgün &amp; Lynn 2002a</td>
<td>0.380</td>
<td>0.258</td>
</tr>
<tr>
<td>Akgün &amp; Lynn 2002b</td>
<td>0.080</td>
<td>-0.024</td>
</tr>
<tr>
<td>Akgun et al. 2006</td>
<td>-0.080</td>
<td>-0.188</td>
</tr>
<tr>
<td>Akgün et al. 2007</td>
<td>-0.330</td>
<td>-0.449</td>
</tr>
<tr>
<td>Aronson et al. 2006</td>
<td>0.380</td>
<td>0.230</td>
</tr>
<tr>
<td>Barnes &amp; Liao 2012</td>
<td>0.470</td>
<td>0.315</td>
</tr>
<tr>
<td>Bierley &amp; Daly 2001</td>
<td>0.465</td>
<td>0.294</td>
</tr>
<tr>
<td>Bierley et al. 2001</td>
<td>-0.310</td>
<td>-0.481</td>
</tr>
<tr>
<td>Carson et al. 2012</td>
<td>0.460</td>
<td>0.314</td>
</tr>
<tr>
<td>Chen et al. 2012</td>
<td>0.280</td>
<td>0.151</td>
</tr>
<tr>
<td>Cui et al. 2015</td>
<td>0.450</td>
<td>0.339</td>
</tr>
<tr>
<td>Da Silveira &amp; Cagiano 2006</td>
<td>-0.300</td>
<td>-0.402</td>
</tr>
<tr>
<td>Dayan &amp; Di Benedetto 2009</td>
<td>0.110</td>
<td>-0.086</td>
</tr>
<tr>
<td>Eisenhardt &amp; Tebrizi 1995</td>
<td>0.070</td>
<td>-0.164</td>
</tr>
<tr>
<td>González Benito 2010</td>
<td>-0.275</td>
<td>-0.405</td>
</tr>
<tr>
<td>Gopal et al. 2013</td>
<td>0.130</td>
<td>-0.095</td>
</tr>
<tr>
<td>Heim &amp; Peng 2010</td>
<td>-0.250</td>
<td>-0.360</td>
</tr>
<tr>
<td>Herhausen &amp; Schögel 2014</td>
<td>0.260</td>
<td>0.126</td>
</tr>
<tr>
<td>Huckman &amp; Staats 2011</td>
<td>0.060</td>
<td>-0.037</td>
</tr>
<tr>
<td>Huo et al. 2015</td>
<td>0.285</td>
<td>0.178</td>
</tr>
<tr>
<td>Joshi &amp; Sharma 2004</td>
<td>-0.150</td>
<td>-0.296</td>
</tr>
<tr>
<td>Kristal et al. 2010</td>
<td>0.400</td>
<td>0.267</td>
</tr>
<tr>
<td>Larro 2003</td>
<td>-0.446</td>
<td>-0.536</td>
</tr>
<tr>
<td>Lau et al. 2007</td>
<td>0.195</td>
<td>0.073</td>
</tr>
<tr>
<td>Li et al. 2010</td>
<td>0.750</td>
<td>0.335</td>
</tr>
<tr>
<td>Li et al. 2011</td>
<td>0.500</td>
<td>0.471</td>
</tr>
<tr>
<td>Liu et al. 2009</td>
<td>-0.276</td>
<td>-0.370</td>
</tr>
<tr>
<td>Lynn et al. 1999</td>
<td>0.260</td>
<td>0.062</td>
</tr>
<tr>
<td>Martinez Sánchez et al. 2009</td>
<td>0.360</td>
<td>0.215</td>
</tr>
<tr>
<td>McComber et al. 2007</td>
<td>0.270</td>
<td>0.017</td>
</tr>
<tr>
<td>Menor et al. 2007</td>
<td>0.210</td>
<td>0.092</td>
</tr>
<tr>
<td>Miron-Spektor &amp; Beenen 2015</td>
<td>0.120</td>
<td>-0.023</td>
</tr>
<tr>
<td>Patel et al. 2012</td>
<td>0.060</td>
<td>-0.007</td>
</tr>
<tr>
<td>Prajogo &amp; McDermott 2011</td>
<td>0.370</td>
<td>0.242</td>
</tr>
<tr>
<td>Qu et al. 2012</td>
<td>-0.280</td>
<td>-0.381</td>
</tr>
<tr>
<td>Roh 2006</td>
<td>0.320</td>
<td>0.238</td>
</tr>
<tr>
<td>Salomo et al. 2007</td>
<td>0.184</td>
<td>0.008</td>
</tr>
<tr>
<td>Sethi &amp; Iqbal 2008</td>
<td>-0.530</td>
<td>-0.648</td>
</tr>
<tr>
<td>Song et al. 2011</td>
<td>0.330</td>
<td>0.209</td>
</tr>
<tr>
<td>Snell et al. 2007</td>
<td>0.502</td>
<td>0.387</td>
</tr>
<tr>
<td>Tatikonda &amp; Rosenthal 2000</td>
<td>0.253</td>
<td>0.077</td>
</tr>
<tr>
<td>Veldhuizen et al. 2006</td>
<td>0.230</td>
<td>0.080</td>
</tr>
<tr>
<td>Vicente et al. 2015</td>
<td>0.480</td>
<td>0.407</td>
</tr>
<tr>
<td>Wang et al. 2014</td>
<td>0.470</td>
<td>0.362</td>
</tr>
<tr>
<td>Wei et al. 2013</td>
<td>0.540</td>
<td>0.437</td>
</tr>
<tr>
<td>Wong et al. 2011</td>
<td>0.234</td>
<td>0.077</td>
</tr>
<tr>
<td>Worren et al. 2002</td>
<td>0.260</td>
<td>0.070</td>
</tr>
<tr>
<td>Yang &amp; Lai 2006</td>
<td>-0.208</td>
<td>-0.168</td>
</tr>
<tr>
<td>Yi-Renko et al. 2001</td>
<td>-0.010</td>
<td>-0.150</td>
</tr>
<tr>
<td>Zhang et al. 2008</td>
<td>0.080</td>
<td>-0.101</td>
</tr>
<tr>
<td><strong>Random Overall</strong></td>
<td>0.178</td>
<td>0.094</td>
</tr>
</tbody>
</table>
Figure 5: Results for effects of stability on performance.

<table>
<thead>
<tr>
<th>Study name</th>
<th>Statistics for each study</th>
<th>Correlation and 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akgün &amp; Lynn 2002a</td>
<td>0.400 0.280 0.508 6.110 0.000</td>
<td></td>
</tr>
<tr>
<td>Akgün &amp; Lynn 2002b</td>
<td>0.340 0.244 0.429 6.634 0.000</td>
<td></td>
</tr>
<tr>
<td>Akgün et al. 2006</td>
<td>0.600 0.525 0.666 12.322 0.000</td>
<td></td>
</tr>
<tr>
<td>Akgün et al. 2007</td>
<td>-0.018 -0.157 0.122 -0.251 0.802</td>
<td></td>
</tr>
<tr>
<td>Aronson et al. 2006</td>
<td>0.500 0.366 0.614 6.499 0.000</td>
<td></td>
</tr>
<tr>
<td>Aronson et al. 2008</td>
<td>0.220 0.039 0.387 2.377 0.017</td>
<td></td>
</tr>
<tr>
<td>Barnes &amp; Liao 2002</td>
<td>0.310 0.179 0.430 4.510 0.000</td>
<td></td>
</tr>
<tr>
<td>Bierly &amp; Daly 2007</td>
<td>0.231 0.034 0.411 2.293 0.022</td>
<td></td>
</tr>
<tr>
<td>Bonner et al. 2002</td>
<td>-0.350 -0.516 -0.160 -3.505 0.000</td>
<td></td>
</tr>
<tr>
<td>Carson et al. 2012</td>
<td>0.245 0.069 0.406 2.705 0.007</td>
<td></td>
</tr>
<tr>
<td>Chen et al. 2012</td>
<td>0.320 0.197 0.435 4.941 0.000</td>
<td></td>
</tr>
<tr>
<td>Cui et al. 2015</td>
<td>0.340 0.212 0.457 4.983 0.000</td>
<td></td>
</tr>
<tr>
<td>Da Silveira &amp; Cagianco 2006</td>
<td>0.340 0.212 0.457 4.983 0.000</td>
<td></td>
</tr>
<tr>
<td>Dayan &amp; Di Benedetto 2009</td>
<td>0.150 -0.055 0.343 1.434 0.152</td>
<td></td>
</tr>
<tr>
<td>Eisenhardt &amp; Tabrizi 1996</td>
<td>0.360 0.140 0.568 3.131 0.002</td>
<td></td>
</tr>
<tr>
<td>González Benito 2010</td>
<td>-0.130 -0.271 0.017 -1.739 0.082</td>
<td></td>
</tr>
<tr>
<td>Gopal et al. 2013</td>
<td>0.960 0.166 0.279 0.520 0.603</td>
<td></td>
</tr>
<tr>
<td>Heim &amp; Peng 2010</td>
<td>0.320 0.201 0.430 5.084 0.000</td>
<td></td>
</tr>
<tr>
<td>Herhausen &amp; Schögel 2014</td>
<td>0.170 0.022 0.311 2.245 0.025</td>
<td></td>
</tr>
<tr>
<td>Huckman &amp; Staats 2011</td>
<td>-0.100 -0.196 -0.003 -2.022 0.043</td>
<td></td>
</tr>
<tr>
<td>Hsu et al. 2015</td>
<td>0.232 0.120 0.338 4.010 0.000</td>
<td></td>
</tr>
<tr>
<td>Joshi &amp; Sharma 2004</td>
<td>-0.340 -0.468 -0.197 -4.507 0.000</td>
<td></td>
</tr>
<tr>
<td>Kortmann 2014a</td>
<td>0.220 0.005 0.416 2.000 0.045</td>
<td></td>
</tr>
<tr>
<td>Kortmann 2014b</td>
<td>0.170 -0.055 0.378 1.487 0.137</td>
<td></td>
</tr>
<tr>
<td>Kristel et al. 2010</td>
<td>-0.170 -0.220 -0.311 -2.245 0.025</td>
<td></td>
</tr>
<tr>
<td>Larso 2003</td>
<td>-0.218 -0.328 -0.102 -3.641 0.000</td>
<td></td>
</tr>
<tr>
<td>Larso et al. 2009</td>
<td>0.317 0.206 0.420 5.395 0.000</td>
<td></td>
</tr>
<tr>
<td>Lau et al. 2007</td>
<td>0.175 0.053 0.293 2.801 0.005</td>
<td></td>
</tr>
<tr>
<td>Li et al. 2010</td>
<td>0.340 0.170 0.490 3.814 0.000</td>
<td></td>
</tr>
<tr>
<td>Li et al. 2011</td>
<td>0.680 0.320 0.769 8.930 0.000</td>
<td></td>
</tr>
<tr>
<td>Liu et al. 2009</td>
<td>-0.220 -0.317 -0.118 4.172 0.000</td>
<td></td>
</tr>
<tr>
<td>Lynn at al. 1999</td>
<td>0.420 0.239 0.573 4.294 0.000</td>
<td></td>
</tr>
<tr>
<td>Martínez Sánchez et al. 2009</td>
<td>0.410 0.270 0.533 5.388 0.000</td>
<td></td>
</tr>
<tr>
<td>McComb et al. 2007</td>
<td>0.520 0.307 0.684 4.351 0.000</td>
<td></td>
</tr>
<tr>
<td>Menor et al. 2007</td>
<td>0.320 0.207 0.424 5.358 0.000</td>
<td></td>
</tr>
<tr>
<td>Miron-Spektor &amp; Beenken 2015</td>
<td>0.180 0.038 0.335 2.482 0.013</td>
<td></td>
</tr>
<tr>
<td>Patel et al. 2012</td>
<td>0.190 0.124 0.254 5.604 0.000</td>
<td></td>
</tr>
<tr>
<td>Pragjgo &amp; McDermott 2011</td>
<td>0.260 0.124 0.387 3.678 0.000</td>
<td></td>
</tr>
<tr>
<td>Qu et al. 2012</td>
<td>-0.160 -0.266 -0.046 -2.781 0.005</td>
<td></td>
</tr>
<tr>
<td>Rieh 2009</td>
<td>0.180 0.110 0.248 5.010 0.000</td>
<td></td>
</tr>
<tr>
<td>Salomo et al. 2007</td>
<td>0.520 0.363 0.634 6.546 0.000</td>
<td></td>
</tr>
<tr>
<td>Sethi &amp; Iqbal 2008</td>
<td>0.180 0.001 0.348 1.968 0.049</td>
<td></td>
</tr>
<tr>
<td>Song et al. 2011</td>
<td>0.280 0.155 0.396 4.306 0.000</td>
<td></td>
</tr>
<tr>
<td>Swink et al. 2007</td>
<td>0.514 0.411 0.604 8.446 0.000</td>
<td></td>
</tr>
<tr>
<td>Talikonda &amp; Rosenthal 2000</td>
<td>0.250 0.074 0.411 2.783 0.006</td>
<td></td>
</tr>
<tr>
<td>Veldhuizen et al. 2006</td>
<td>0.240 0.091 0.378 3.125 0.002</td>
<td></td>
</tr>
<tr>
<td>Vicente et al. 2015</td>
<td>0.260 0.174 0.342 5.757 0.000</td>
<td></td>
</tr>
<tr>
<td>Wang et al. 2014</td>
<td>0.500 0.386 0.682 8.221 0.000</td>
<td></td>
</tr>
<tr>
<td>Webster &amp; White 2009</td>
<td>-0.190 -0.400 0.039 -1.632 0.103</td>
<td></td>
</tr>
<tr>
<td>Wei et al. 2013</td>
<td>0.230 0.099 0.354 3.394 0.001</td>
<td></td>
</tr>
<tr>
<td>Wong et al. 2011</td>
<td>0.447 0.309 0.586 5.851 0.000</td>
<td></td>
</tr>
<tr>
<td>Worren et al. 2002</td>
<td>0.352 0.170 0.511 3.877 0.000</td>
<td></td>
</tr>
<tr>
<td>Yang &amp; Lai 2006</td>
<td>-0.041 -0.171 0.069 -0.611 0.541</td>
<td></td>
</tr>
<tr>
<td>Yi-Renka et al. 2001</td>
<td>0.140 -0.001 0.275 1.953 0.051</td>
<td></td>
</tr>
<tr>
<td>Zhang et al. 2008</td>
<td>0.110 -0.071 0.284 1.195 0.232</td>
<td></td>
</tr>
</tbody>
</table>

Random Overall | **0.233 0.171 0.294 7.170 0.000** | **-1.00 -0.50 0.00 0.50 1.00** |
Figure 6: Results for effects of flexibility on performance.

<table>
<thead>
<tr>
<th>Study name</th>
<th>Correlation</th>
<th>Statistics for each study</th>
<th>Correlation and 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akgün &amp; Lynn 2002a</td>
<td>0.470</td>
<td>0.358 0.569</td>
<td>7.356 0.000</td>
</tr>
<tr>
<td>Akgün &amp; Lynn 2002b</td>
<td>0.180</td>
<td>0.077 0.279</td>
<td>3.409 0.001</td>
</tr>
<tr>
<td>Akgün et al. 2006</td>
<td>-0.020</td>
<td>-0.130 0.080</td>
<td>-0.356 0.722</td>
</tr>
<tr>
<td>Akgün et al. 2007</td>
<td>0.100</td>
<td>-0.040 0.236</td>
<td>1.396 0.162</td>
</tr>
<tr>
<td>Aronson et al. 2006</td>
<td>0.170</td>
<td>0.006 0.325</td>
<td>2.031 0.042</td>
</tr>
<tr>
<td>Aronson et al. 2008</td>
<td>0.290</td>
<td>0.114 0.449</td>
<td>3.174 0.002</td>
</tr>
<tr>
<td>Barnes &amp; Liao 2012</td>
<td>0.290</td>
<td>0.168 0.412</td>
<td>4.201 0.000</td>
</tr>
<tr>
<td>Blerly &amp; Daly 2007</td>
<td>0.263</td>
<td>0.067 0.430</td>
<td>2.521 0.012</td>
</tr>
<tr>
<td>Bonner et al. 2002</td>
<td>0.410</td>
<td>0.227 0.565</td>
<td>4.178 0.000</td>
</tr>
<tr>
<td>Carson et al. 2012</td>
<td>0.074</td>
<td>-0.107 0.250</td>
<td>0.802 0.423</td>
</tr>
<tr>
<td>Chen et al. 2012</td>
<td>0.260</td>
<td>0.130 0.381</td>
<td>3.847 0.000</td>
</tr>
<tr>
<td>Cui et al. 2015</td>
<td>0.480</td>
<td>0.384 0.583</td>
<td>7.987 0.000</td>
</tr>
<tr>
<td>Da Silva e &amp; Cegilano 2006</td>
<td>0.260</td>
<td>0.126 0.386</td>
<td>3.744 0.000</td>
</tr>
<tr>
<td>Dayan &amp; Di Benedetto 2009</td>
<td>0.070</td>
<td>-0.136 0.270</td>
<td>0.665 0.506</td>
</tr>
<tr>
<td>Eisenhardt &amp; Tabrizi 1995</td>
<td>-0.370</td>
<td>-0.554 -0.151</td>
<td>-3.226 0.001</td>
</tr>
<tr>
<td>González Benito 2010</td>
<td>-0.238</td>
<td>-0.372 -0.096</td>
<td>-3.242 0.001</td>
</tr>
<tr>
<td>Gopal et al. 2013</td>
<td>-0.060</td>
<td>-0.279 0.165</td>
<td>-0.520 0.603</td>
</tr>
<tr>
<td>Heim &amp; Peng 2010</td>
<td>-0.010</td>
<td>0.137 0.117</td>
<td>-0.153 0.878</td>
</tr>
<tr>
<td>Herhausen &amp; Schögel 2014</td>
<td>-0.070</td>
<td>-0.184 0.046</td>
<td>-1.182 0.237</td>
</tr>
<tr>
<td>Huckman &amp; Staats 2011</td>
<td>-0.090</td>
<td>-0.165 0.007</td>
<td>-1.818 0.069</td>
</tr>
<tr>
<td>Huo et al. 2015</td>
<td>0.358</td>
<td>0.297 0.491</td>
<td>7.149 0.000</td>
</tr>
<tr>
<td>Joshi &amp; Sharma 2004</td>
<td>0.310</td>
<td>0.165 0.442</td>
<td>4.080 0.000</td>
</tr>
<tr>
<td>Kortmann 2014a</td>
<td>0.280</td>
<td>0.068 0.467</td>
<td>2.573 0.010</td>
</tr>
<tr>
<td>Kortmann 2014b</td>
<td>0.300</td>
<td>0.083 0.490</td>
<td>2.881 0.007</td>
</tr>
<tr>
<td>Kristal et al. 2010</td>
<td>0.240</td>
<td>0.065 0.375</td>
<td>3.201 0.001</td>
</tr>
<tr>
<td>Larson 2003</td>
<td>0.239</td>
<td>0.124 0.348</td>
<td>4.005 0.000</td>
</tr>
<tr>
<td>Larso et al. 2009</td>
<td>0.435</td>
<td>0.334 0.527</td>
<td>7.658 0.000</td>
</tr>
<tr>
<td>Lau et al. 2007</td>
<td>0.158</td>
<td>0.035 0.276</td>
<td>2.598 0.012</td>
</tr>
<tr>
<td>Li et al. 2010</td>
<td>0.430</td>
<td>0.271 0.566</td>
<td>4.953 0.000</td>
</tr>
<tr>
<td>Li et al. 2011</td>
<td>0.540</td>
<td>0.399 0.656</td>
<td>6.507 0.000</td>
</tr>
<tr>
<td>Liu et al. 2009</td>
<td>0.422</td>
<td>0.332 0.504</td>
<td>6.397 0.000</td>
</tr>
<tr>
<td>Lynn et al. 1999</td>
<td>0.280</td>
<td>0.083 0.456</td>
<td>2.759 0.006</td>
</tr>
<tr>
<td>Martínez Sánchez et al. 2009</td>
<td>0.520</td>
<td>0.395 0.626</td>
<td>7.129 0.000</td>
</tr>
<tr>
<td>McComb et al. 2007</td>
<td>0.170</td>
<td>0.268 0.406</td>
<td>1.296 0.195</td>
</tr>
<tr>
<td>Manor et al. 2007</td>
<td>0.200</td>
<td>0.175 0.397</td>
<td>4.823 0.000</td>
</tr>
<tr>
<td>Miron-Spekter &amp; BEENEN 2015</td>
<td>0.060</td>
<td>-0.083 0.241</td>
<td>0.819 0.413</td>
</tr>
<tr>
<td>Patel et al. 2012</td>
<td>0.210</td>
<td>0.145 0.273</td>
<td>6.211 0.000</td>
</tr>
<tr>
<td>Prajogo &amp; McDermott 2011</td>
<td>0.490</td>
<td>0.375 0.560</td>
<td>7.409 0.000</td>
</tr>
<tr>
<td>Qu et al. 2012</td>
<td>0.080</td>
<td>-0.034 0.182</td>
<td>1.382 0.167</td>
</tr>
<tr>
<td>Rho 2009</td>
<td>0.300</td>
<td>0.234 0.363</td>
<td>8.522 0.000</td>
</tr>
<tr>
<td>Salomo et al. 2007</td>
<td>0.360</td>
<td>0.136 0.448</td>
<td>3.515 0.000</td>
</tr>
<tr>
<td>Seth &amp; Iqbal 2008</td>
<td>-0.110</td>
<td>-0.264 0.071</td>
<td>-1.195 0.232</td>
</tr>
<tr>
<td>Song et al. 2011</td>
<td>0.450</td>
<td>0.340 0.548</td>
<td>7.254 0.000</td>
</tr>
<tr>
<td>Swink et al. 2007</td>
<td>0.213</td>
<td>0.084 0.335</td>
<td>3.216 0.001</td>
</tr>
<tr>
<td>Tatikonda &amp; Rosenthal 2000</td>
<td>0.208</td>
<td>0.030 0.373</td>
<td>2.283 0.022</td>
</tr>
<tr>
<td>Veldhuizen et al. 2006</td>
<td>0.220</td>
<td>0.070 0.360</td>
<td>2.855 0.004</td>
</tr>
<tr>
<td>Vicente et al. 2015</td>
<td>0.340</td>
<td>0.256 0.416</td>
<td>7.680 0.000</td>
</tr>
<tr>
<td>Wang et al. 2014</td>
<td>0.640</td>
<td>0.556 0.711</td>
<td>11.347 0.000</td>
</tr>
<tr>
<td>Webster &amp; White 2006</td>
<td>0.230</td>
<td>0.003 0.434</td>
<td>1.987 0.047</td>
</tr>
<tr>
<td>Wei et al. 2013</td>
<td>0.330</td>
<td>0.205 0.445</td>
<td>4.968 0.000</td>
</tr>
<tr>
<td>Wong et al. 2011</td>
<td>0.382</td>
<td>0.237 0.511</td>
<td>4.895 0.000</td>
</tr>
<tr>
<td>Worren et al. 2002</td>
<td>0.307</td>
<td>0.121 0.472</td>
<td>3.172 0.002</td>
</tr>
<tr>
<td>Yang &amp; Lai 2006</td>
<td>-0.058</td>
<td>-0.217 0.042</td>
<td>-1.330 0.184</td>
</tr>
<tr>
<td>Yli-Renko et al. 2001</td>
<td>0.110</td>
<td>-0.031 0.247</td>
<td>1.530 0.126</td>
</tr>
<tr>
<td>Zhang et al. 2008</td>
<td>0.660</td>
<td>0.545 0.750</td>
<td>8.576 0.000</td>
</tr>
</tbody>
</table>

Random Overall 0.244 0.190 0.298 8.493 0.000 -1.00 -0.50 0.00 0.50 1.00
Curriculum Vitae

Denniz Dönmez

ETH Zürich
Department of Management, Technology, and Economics
Organization, Work and Technology Group WEV K508,
Weinbergstrasse 56/58, 8092 Zürich, Switzerland

Education

since 2010 PhD candidate at ETH Zurich
Department of Management, Technology, and Economics.
Thesis advisor: Prof. Dr. G. Grote

2013 BSc Economics and Management
London School of Economics and Political Science
University of London

2010 Dipl.-Ing. Electrical Engineering and Information Technology
Technische Universität München

2010 Honours Degree in Technology Management
Center for Digital Technology and Management
Technische Universität München
& Ludwig Maximilians Universität München

Work Experience

2008 Teseon GmbH, Garching, Germany
Research assistant
2006 Sonnenenergie für Westafrika e.V., Ouagadougou, Burkina Faso
Product development engineer

2003-06 Attocube Systems AG, Munich
Working student in nanotechnology research and development

2002-03 Gerontopsychiatrische Tagesstätte Riemeisterfenn, Berlin
Civil service

Publications

Manuscripts for publication in peer-reviewed journals

1. Dönmez, D., Grote, G. Two Sides of The Same Coin – How Agile Software Development Teams Manage Uncertainty as Threats and Opportunities. Manuscript invited for revise-and-resubmit at Information and Software Technology


Conference papers


**Book chapters**


Invited talks

1. Design Thinking (workshop), May 2015, Ergon Informatik AG, Zurich

2. Risks, Roles, and Routines: Scrum from a research perspective, February 2015, Google, Zurich

3. Managing Uncertainty with Agile Software Development, October 2012, Department of Computer Science, ETH Zurich


5. Agile Development Methodologies, October 2010, Engineering Design Center (EDC), Department of Engineering, University of Cambridge