Abstract

The complexity of software projects and inherent customer demands is becoming increasingly challenging for developers and managers. Human factors in the development process are growing in importance. Consequently, understanding team dynamics is a central aspect of steady development planning and execution.

Despite the many available management systems and development tools that are being continuously improved to support teams and managers with practical process information, the equally crucial sociological aspects have typically been addressed insufficiently or not at all. In people-focused agile software processes, a first socio-technical understanding can also be promoted by sharing positive and negative development experiences during specific team meetings (e.g., sprint Retrospectives). Nevertheless, there is still a lack of systematically recorded and processed socio-technical information in software projects, making it difficult for subsequent reviews by teams and managers to characterize and understand the sometimes volatile and complex team dynamics during the process.

This thesis strives to support teams and managers in understanding and improving awareness of the team dynamics that occur in their agile software projects by introducing computer-aided sprint feedback. The concept builds on four information assets: (1) socio-technical data monitoring, (2) descriptive sprint feedback, (3) predictive sprint feedback, and (4) exploratory sprint planning. These assets unify interdisciplinary fundamentals, practical methods from software engineering, data science, organizational and social psychology. Using a design science research process for information systems, observations in several conducted studies (32 in academic project environments and three in industry) resulted in the foundations and methods for a practical feedback concept on the socio-technical aspects in sprint, prototypically realized for Jira.

A practical evaluation involved two industry projects in an action research methodology that helped improve the concept's usability and utility through practitioner reflections. The collaboration between industry and research resolved practical issues that did not arise during the design science process. Several beneficial outcomes based on the provided sprint feedback are reported and described in this study (e.g., the effect of team structures on development performance). Moreover, the reflections underscored the practical relevance of systematic feedback and the need to better understand human factors in the software development process.

Keywords: Team Dynamics, Agile Software Development, Knowledge Support

Chapter 1

Introduction

This doctoral thesis introduces a computer-aided sprint feedback concept to support an understanding of socio-technical dependencies and behavior patterns in teams during agile development processes. This concept has several interdisciplinary foundations and includes practical methods from data science, software engineering, and organizational and social psychology. It comprises an adaptive, holistic process chain for the systematic capture, processing, and characterization of team dynamics in software development projects. The work addresses three research questions concerning the practical relevance, utilization, and utility of complementary team feedback on socio-technical aspects of agile software projects.

1.1 Human Factors in Software Projects

The role of human factors in software development has been investigated over the decade and several systematic literature reviews have shown that studies of this topic have intensified in recent years [16, 111, 198]. Human nature is based on psychological factors (e.g., emotions dictate behaviors), which, in combination with social activities, can impact the success of a development team [111]. In the software engineering domain, broad and diverse research has been conducted on the human and social effects of these factors on software development teams and processes [122, 205, 207, 212]. However, much of the software engineering research conducted in the last decade has focused more on technical problems than on human aspects [111]. Thus, "Failure to include human factors may explain some of the dissatisfaction with conventional information systems development methodologies; they do not address real organizations" [11]. This underscores the need for cooperation between practitioners and researchers to address the software engineering challenges stemming from human factors and improve our understanding of the inherent effects of these factors during development processes [18, 154, 198]. In the software engineering domain, the term *team dynamic* describes interaction patterns among project members that can determine the performance of teams [62].

Human factors, social activities, and development behaviors all contribute to team dynamics, which can influence a team's progress and the achievement of project goals [47, 193]. Solid planning, previous experience, knowledge, and awareness of team dynamics can reduce the likelihood of dysfunctional development situations. A team's size and its fundamental communication structure can affect the team atmosphere and development performance in a project (estimated versus completed story points) [4]. Understanding the team dynamics in ongoing software projects means learning from past interactions and deriving knowledge that enables adjustments for future development activities.

It is particularly important to understand human factors and social interactions, not only in plan-driven processes but also, and indeed especially, in agile software development. Agile methods, such as Scrum [209], Lean [189], or Extreme Programming [23], are people-focused and based on small collaborative teams with positive motivation and continuous striving to improve work habits for better organizational and development performances over time [238, 239]. They are more adaptive than plan-driven processes and allow software teams to react faster to customer demands and change requests within short development iterations [145]. However, the increased development agility of teams also amplifies the risk of volatile human behavior, which can lead to information gaps, loner attitudes, sentiment changes, demotivation, and performance fluctuations [25, 236, 238].

In this context, it is no surprise that agile methods, allow team members to routinely share their positive and negative development experiences through cyclic sprint meetings (e.g., in sprint retrospectives or planning) [164]. Accurate sprint planning is crucial for agile teams to achieve steady development performance in short iterations [164]. However, feature re-prioritization or customer change requests are always a possibility and can derail even the most careful plans. Sprint planning is an experience-based activity that builds on retrospective knowledge gained from earlier development performances, covering both positive and negative situations [50, 59, 228]. Nevertheless, human factors are often insufficiently considered in sprint planning [145]. People have different personalities and skills, which, combined with other environmental factors, can influence performances [47]. Therefore, it is vital for teams and management to learn about how these influences have played out in the past to improve dysfunctional habits (e.g., by holistically reflecting on socio-technical strengths and weaknesses in sprints).

Early recognition and understanding of team dynamics in volatile software projects is challenging, primarily due to the often short development iterations and a lack of psychological acumen, analytical expertise, or practicable methods to enable adaptive socio-technical sprint characterization in existing processes [122, 138]. These problems limit the holistic formation of a shared knowledge base covering information on team dynamics that teams can access. Computer-aided sprint feedback can support understanding of the sometimes even complex dynamics in agile teams with the help of systematically captured and processed socio-technical data through a project. Moreover, visualizations can enhance cognitive perception and awareness as the foundation for improvement opportunities [153].

1.2 Research Motivation and Questions

The aim of this work is to enable faster team feedback on the socio-technical aspects of sprints using computer-aided data capture and processing methods. The feedback support is intended to create an opportunity to sharpen awareness and broaden understanding of team dynamics and behavioral patterns resulting from systematically observed socio-technical dependencies in agile software projects. This work investigates the practical relevance, utilization, and utility of computer-aided team feedback on socio-technical aspects during the development process.

1.2.1 Team Understanding as Motivation

The central aim of this work is to support awareness and understanding behavioral patterns in teams with the help of systematic feedback on socio-technical dependencies during the development process. Data acquisition and analysis of team dynamics in software projects can be a sophisticated and time-consuming process that requires technological support [62, 137]. However, understanding behavior patterns during projects can enable a transition from explicit to tacit knowledge in teams, as shown by the data, information, knowledge, wisdom (DIKW) pyramid in Figure 1.1.

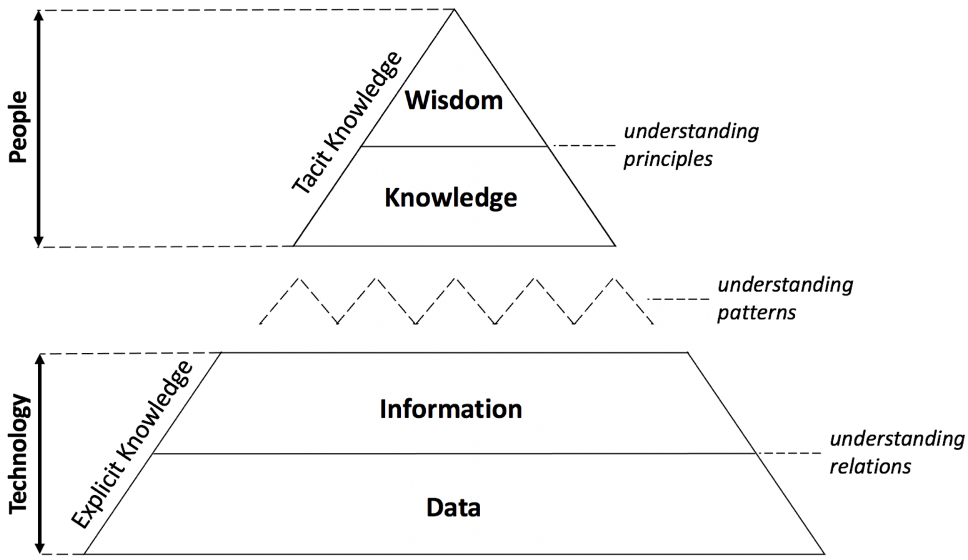

Figure 1.1: Adaption of the DIKW Pyramid, based on Ackoff [6]

Data and information can be seen as explicit knowledge that is accessible using information technologies (e.g., project management systems). By contrast, team knowledge and intellectual wisdom concerning human factors can be classed as tacit knowledge, which anchors in people's minds and includes perceptions, opinions, sentiments, personalities, and experiences [6, 200, 204].

Understanding supports various knowledge transition states (e.g., understanding the relationship between mood changes and development performance in teams enables an identification of patterns over time that must be able to be understood by the human mind to promote implicit knowledge and awareness) [6]. Intellectual wisdom is needed for decision-making and devising improvement actions based on understanding the principles (i.e., the information pattern meaning corresponding to individual experiences and knowledge).

However, the transformation of socio-technical information into knowledge requires an understanding of team behavior patterns (e.g., development performance, communication behavior, mood changes, and team structures). In this work, this understanding is supported by an information technology concept based on computer-aided sprint feedback [6, 47, 50, 139, 228]. Although new information technologies are available that facilitate knowledge acquisition and decision support (e.g., data-driven methods and models for deriving progress tendencies or bug estimations), only a few concern the socio-technical aspects of agile development teams. Moreover, in practice, capturing and interpreting team dynamics is not a trivial activity and often involves adaptions or changes to standard process routines and practices [62, 121, 122].

To be useful in practice (e.g., in sprint retrospectives), computer-aided feedback support on team behavior requires sociological (subjective) and process (objective) data to enable a holistic characterization of team dynamics [59, 134, 139]. Extended feedback mechanisms can strengthen learning in agile teams [61].

This work endeavors to support

a) the analysis of socio-technical data (explicit knowledge) from team reflections (transformed tacit knowledge) to identify behavior patterns.

b) an understanding of team dynamics in agile projects through computer-aided sprint feedback that considers both explicit and tacit knowledge.

Several practical usability and utility challenges arise in connection with these goals. Data observation and processing to obtain useful socio-technical information must be simple and must not entail significant extra efforts that could comprise existing workflows and processes [232]. There must be a clear motivation in teams to contribute of socio-technical data on experiences and perceptions. Furthermore, practicability of the sprint feedback should reflect the benefits in teams, thus enhancing the acceptance [239]. It is highly desirable to systematically support the understanding of team dynamics in agile software projects and promote team knowledge concerning socio-technical aspects.

1.2.2 Research Questions

The challenges and motivation for understanding team dynamics in agile software projects dictate this work's thematic focus. The supplementary support of intra-team knowledge and awareness of socio-technical dependencies in sprints is the central objective of this work. In volatile agile development iterations, fast feedback and knowledge sharing based on cyclic team reflections are particularly crucial information assets for understanding development performance [44, 233].

Sprint retrospectives support open discussions in teams (e.g., about achieved performances) and are useful for identifying whether improvement actions have resulted in the expected changes [59]. Therefore, functional communication structures, information flows, and a positive atmosphere in team meetings are essential for software project success [115, 205, 207]. Sprint feedback from stakeholders and project leaders supplements teams' internal perceptions and knowledge by introducing external perspectives. The latter can refer, for example, to satisfaction or disappointment with stages of the project or to software- or team-related problems, such as a lack of progress or transparency [59, 239].

The following research questions were defined to investigate the practical relevance, utilization, and utility of computer-aided team feedback for understanding team dynamics during the development process.

Research Question 1

How relevant is computer-aided sprint feedback on socio-technical aspects for agile development teams?

The first research question concerns the relevance of supplementary feedback for supporting a team's information needs and understanding of the socio-technical aspects of sprints. In addition, the question aims to identify frequent problems in agile development teams and what information can best support teams in overcoming these problems, based on researchers' and practitioners' experiences. The findings form the foundation for a computer-aided feedback concept to support a practical understanding of team dynamics during the development process.

Research Question 2

How do agile teams utilize computer-aided sprint feedback on socio-technical aspects during the development process?

The second research question focuses on the utilization of computer-aided sprint feedback on socio-technical aspects. The main focus is on determining how agile development teams acknowledge and use complementary sprint feedback on socio-technical elements along with or instead of other information resources during the development process. The question also concerns the feedback culture in teams, which must be considered for computer-aided sprint support.

Research Question 3

How is the utility of computer-aided sprint feedback on socio-technical aspects concerning the understanding for team dynamics in agile projects?

The third research question of this thesis concerns the practical utility and perceived added value of computer-aided sprint feedback in agile software projects. The central focus of this question is to disclose whether the complementary sprint feedback on the socio-technical aspects can objectively or subjectively support the practical understanding in agile teams for arisen team dynamics in the projects.

1.3 Design Science Research

In this thesis, the concept of computer-aided sprint feedback on socio-technical aspects is similar to information system research (ISR) activities. The concept was designed and prototypically realized in iterative refinements using design science research (DSR) [94, 107], which included a focus on the research question defined in Section 1.2.2, particularly for disclosing qualitative and quantitative answers. Figure 1.2 shows the applied design science research in extension of a behavioral science paradigm.

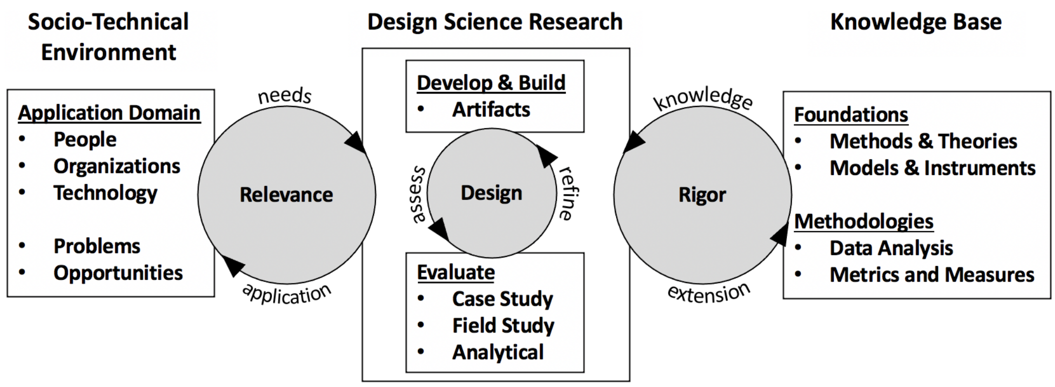

Figure 1.2: Applied Design Science Research, based on [92]

In ISR, the design science paradigm is combinable with the behavioral science paradigm, which is foundational for extending knowledge and understanding application domains concerning relevant environmental opportunities and problems (e.g., including people, organizations, and technologies) [92, 94, 161]. Both paradigms are considered in this research with the purpose of identifying and creating new artifacts (e.g., feedback assets for the communication behavior and mood courses) that support awareness and understanding of team dynamics in agile software projects through supplementary feedback support on socio-technical aspects in sprints. Moreover, DSR was used for prototyping the computer-aided sprint feedback for agile development environments, which integrates methods for capturing socio-technical data and automated feedback processing.

1.3.1 The Relevance Cycle

Modern software development depends more than ever on teams and the people behind them [47, 111, 145]. Software projects that involve agile methods benefit from highly volatile processes that enable teams to react faster to customer demands through manageable release sizes in short development iterations. However, this is not always conducive to a team's socio-psychological well-being [25, 239]. The focus is generally on steady development performance and accurate estimation, often measured using process measures such as team velocity (i.e., scheduled versus completed features) [63, 85, 111]. Problems not only relate to technologies or processes but also to socio-technical dependencies in projects, which makes teams vulnerable to sociological issues, such as dysfunctional communication structures, a hostile atmosphere, demotivation, and dissatisfaction [238]. This applies to both conventional software projects and agile projects. The difference is that in agile projects, the volatile nature of the process facilitates the identification of negative habits earlier through a holistic consideration of socio-technical dependencies in sprints before sociological team issues lead to long-term problems due to a lack of awareness.

The relevance cycle connects the contextual environment of socio-technical aspects of agile software development with the activities of DSR. As a practical example, understanding dysfunctional communication structures in agile teams requires a consideration of the different roles and interests of members, which are shaped by the relevant capabilities and characteristics of people and organizational workflows, the agile practices applied, and the information technologies used during software development. The objective is to understand and identify essential socio-technical dependencies and information-based problems in agile software projects. To this end, it is necessary to derive new and innovative feedback assets for characterizing and understanding team dynamics in sprints based on practical needs [62, 122]. Such assets can facilitate understanding of mutual influences, especially human factors, which are relevant for most projects. Observational studies, surveys, and interviews support the identification of environment-related problems and opportunities for building and evaluating computer-aided sprint feedback artifacts concerning team dynamics in agile projects [128, 133, 207].

1.3.2 The Rigor Cycle

The rigor cycle links design science activities with a knowledge base that comprises interdisciplinary theories, frameworks, instruments, constructs, models, methods, and measures (e.g., from software engineering, data science, and social and organizational psychology) relevant for this research [92]. Moreover, the rigor cycle involves the integration of previous knowledge, experience, and foundations from other disciplines into the system-aided sprint feedback concept to ensure innovation and scientific rigor. For example, previous experiences and observational studies are helpful for determining or adapting sociological measurement methods capturing (e.g., team communication and mood data) [139, 207].