



**HAL**  
open science

# Creating Business Value through Agile Project Management and Information Systems Development: The Perceived Impact of Scrum

Karlheinz Kautz, Thomas Heide Johansen, Andreas Uldahl

► **To cite this version:**

Karlheinz Kautz, Thomas Heide Johansen, Andreas Uldahl. Creating Business Value through Agile Project Management and Information Systems Development: The Perceived Impact of Scrum. Transfer and Diffusion of IT (TDIT), Jun 2014, Aalborg, Denmark. pp.150-165, 10.1007/978-3-662-43459-8\_10 . hal-01381185

**HAL Id: hal-01381185**

**<https://inria.hal.science/hal-01381185>**

Submitted on 14 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Creating Business Value Through Agile Project Management and Information Systems Development: The Perceived Impact of Scrum

Karlheinz Kautz<sup>1</sup>, Thomas Heide Johansen<sup>2</sup>, and Andreas Uldahl<sup>3</sup>,

<sup>1</sup>Faculty of Business, University of Wollongong, Wollongong NSW 2522, Australia  
kautz@uow.edu.au

<sup>2</sup>Progressive AS, DK-2730 Herlev, thj@progressive.dk

<sup>3</sup>Ernst & Young Denmark, DK-2860 Søborg, Andreas.Uldahl@dk.ey.com

**Abstract.** Value creation through information systems (IS) and information technology (IT) is a major IS research topic. However there still exists an ambiguity and fuzziness of the ‘IS business value’ concept and a lack of clarity surrounding the value creation process. This also true for organizations that develop IS/IT and for development technologies like information systems development and project management methods that are applied in the production of IS/IT. The agile method Scrum is one such technology. In the research presented here we studied productivity, quality and employee satisfaction as supported by Scrum as value creating measures. Our positive assessment is built upon subjective perceptions and goes beyond hard measures and indicators. It provides insights into individual and organisational impacts and sheds light on the value generation process. The measures we present thus deal with some of the deficiencies in current IS business value research and contribute to filling existing gaps in an IS business value research agenda.

## 1 Introduction

Value creation through information systems (IS) and information technology (IT) is one of the major research topics for IS/IT researchers. In a recent literature review of IS business value research Schryen [1] however laments the ongoing ambiguity and fuzziness of the ‘IS business value’ concept and the lack of clarity surrounding the value creation process. He proposes that to develop a consistent and comprehensive understanding of the complex phenomenon research should account for linkages between different types of performance depending on varying contexts, for different capabilities that go beyond hard indicators and measures, and for perceived impacts and benefits that are dependent on the respective stakeholders. Schryen [1] focuses on business value of IS in general, but his conclusions are also true for business organizations that develop IS and IT and for development technologies like information

systems and software development and project management methods that are applied in the production of IS and IT.

The agile development and project management method Scrum is one such technology. While numerous publications claim a positive impact of Scrum on information systems and software development, only little empirical work exists to verify these claims. To further contribute to the body of knowledge on IS value creation and the impact of Scrum we set out to answer the following research question: What kind of value is created and which impact has the introduction of the agile development and project management method Scrum on information systems and software development? We apply Schryen's [1] taxonomy of IS business value types consisting of internal value, external value, tangible and intangible values. Here we focus at the outset on internal value provided by Scrum throughout the development process. The results we present in the following are part of a larger project where we developed a framework for investigating value creation and the impact of Scrum (see [2]). In this paper however we concentrate on three of these concepts, namely productivity, quality, and employee satisfaction. In the remainder of the paper we first briefly introduce Scrum. Then we describe our theoretical background and the research setting and method. Subsequently we present and discuss our findings against the existing literature on Scrum and relate them in our conclusions to the literature on IS business value research.

## **2 SCRUM – An Agile Development and Project Management Method**

Scrum is an agile information systems and software development method with a strong focus on project management, which was formalized and tested by Schwaber and Sutherland in the mid 1990ties [3, 4]). Scrum focuses on an iterative and nimble development process, on transparency, visibility and on cooperation in and between the development team and the customers. In Scrum the development team is called the Scrum team. Unlike traditional development projects where analysts, developers and testers are typically separated, Scrum teams are built on an interdisciplinary basis and comprise all these roles in one team preferably in one physical location. This structure, as well as Scrum's focus on self-organization aims at creating team dynamics and a better understanding of the tasks to be performed jointly. In this context the role of the Product owner has the responsibility to represent the project and product externally to other stakeholders and customers and to handle and manage the tasks that appear in the product and release backlogs (see below) [3]. Internally, the role of the Scrum master will provide leadership, motivate and facilitate the team in line with the Scrum values, practices and development process.

A Scrum development process is structured through a product backlog, which is a prioritized list of required business and technical functions of the envisioned product. It might change in line with the customer's new needs. A release backlog is a prioritized subset of the total product backlog and defines the functions to be included in a release. A Scrum, performed in so-called sprints, is a set of development tasks and processes which a Scrum team carries out to achieve a given sprint goal. The length of a sprint is predefined. It typically lasts between 5 and 30 calendar days [3]. What

needs to be done during a sprint is determined by a prioritized sprint backlog, which is determined together with a sprint goal before the start of each sprint by the team and Scrum master and others, if necessary, at a planning meeting. Throughout a project a burn-down chart shows the amount of work left to do versus time over a given period [4]). In short daily Scrum meetings project members briefly present what they have done during the preceding day, which tasks they take on that day, as well as any challenges and obstacles that might have prevented them from carrying out their work without any solution being discussed. Scrums of Scrums are additional short meetings by the Scrum masters of projects, which consist of several Scrum teams. At the end of a sprint a sprint review meeting takes place where the Scrum team, the Product owner, other management, and one or more representatives from the customer [3] assess the team's development process and progress in relation to the predefined sprint goal. Finally the Scrum team, the Scrum master and possibly the Product owner hold a meeting, called a retrospective, to secure learning and further improvement in the team where both the process and the product are assessed and discussed by each individual team member.

### **3 Literature Review and Theoretical Background**

With the IS value literature as examined by Schryen [1] as a point of departure in our study we were interested in the impact of a specific method, namely Scrum on information systems and software development. Our literature review was therefore focused on that particular approach and not in general on project management methods' or agile methods' impact on information systems and software development. This limited our sources to writings which take their starting point in agile software development. We combined a concept-centric with an author-based approach [5] the value concept through the concept of impact, either by focussing on economic impact [6] or on organizational and/or individual impact [7, 8]. On this background our original search with keywords such as 'impact of Scrum', 'effect of Scrum', 'impact of Scrum implementation', and 'effect of Scrum implementation' primarily in Google, Google Scholar and IEEE sources lead to about 90 sources, of which 8 dealt more precisely with our research problem. An additional 8 sources were identified through the other mechanisms. From that literature we derived a number of concepts and for these concepts indicators for the impact of Scrum on information systems and software development processes and projects. The resulting framework consisted of the identified, interrelated concepts productivity, quality, employee satisfaction, management support, as well process clarity, and a total of 28 indicators, which defined the concepts on a more detailed level. Here we are focusing on Scrum's impact on the first three.

Productivity is a prominent concept in the IS business value literature (see f.ex. [9]; for a detailed discussion [1]); in the agile development literature it is an expression of the development team productivity [10]. There are a number of interrelated indicators that are linked to different areas that may impact on productivity. Dybå and Dingsøy [11] describe the results of a comparative case study where productivity was measured in projects driven by traditional and agile development methods based on the number of lines of code (LOC) per hour, month or employee. Guang-Yong [12] describes the measurement of productivity in the number of lines of code, and demonstrates how productivity increases gradually as a team becomes more self-organized

and manages to review its development processes to avoid the repetition of mistakes. Appelo [13] has a different view how productivity can be measured. He highlights the increased functionality to the final product as a direct indicator of improved productivity. The way he measures the functionality is the number of story points that have been completed within a given period. A story point is a number that reflects the severity of a given task. Mahnic and Vrana [14] and Mahnic and Zabkar [15] define the assessment of productivity as a ratio of the added value versus the associated financial costs as well as costs associated with bug fixes. We use these sources to investigate the indicators employee performance, the time associated with fixing bugs, and repetition of the same mistakes. Sutherland and Altman [16] use the term "perfect hours" as a label for a project participant's undisturbed and uninterrupted work. They emphasize that a project's progress and productivity should be measured by taking perfect hours combined with other indicators into account. The number of interruptions and the number of uninterrupted development hours were the two indicators we descended from these authors. Moore et al. [17] argue that increased productivity through the use of Scrum is grounded on its focus on delivering functional software in short time intervals with fixed deadlines where developers do not end up in endless development cycles in an attempt to provide perfect solutions with a product that can handle everything at one time. The avoidance of continuous development cycles and compliance with deadlines are the last two indicators we derived from these authors.

Quality is another important concept in the IS business value literature, in the form of product quality [18, 19] or system and information quality [7, 8]. In the agile development literature a strong focus is on measures that can be applied during the development process. In their comparative study Dybå and Dingsøy [11] identify various quality measures that were applied across a wide range of projects. In line with their findings Mahnic and Vrana [14] and Mahnic and Zabka [15] put forward two different indicators for measuring quality both related to error density: the number of errors as detected during the development process by the development team itself, and those reported by the customers over a fixed period of time, in both cases the measurement unit is defects or errors per 1000 lines of code (KLOC), which we adopt for our study. Dybå and Dingsøy [11] emphasize that the 'seriousness' of the errors might have an impact on the perception of quality; we follow their suggestion and also use number of bugs or minor errors as a quality indicator. As another important measure for quality to be taken into account during the development process Appelo [13] highlights that the number of errors identified by testers during the integration of software modules is an essential metric as well. Another undisputed indicator for quality is of course the overall useability of the final product [11]. Thus, although we in our study did not have access to customers directly (see below) we collect, to the extent possible, data about this indicator.

Customer and user satisfaction [7, 8, 20] as well as consumer welfare [18, 19] are significant performance measures for IS business value; in our context we identified employee satisfaction of the different team members in the development teams that perform a development task, as an important concept. These team members have the roles of project managers, analysts, developers, and testers. As such we do not focus on staff in general or those who maintain the final product. In the reviewed agile development literature there are significant differences in the perceptions of the various authors concerning employee satisfaction. Mann and Maurer [21], Mahnic and

Vrana [14] as well as Manhic and Zabka [15] argue that a reduction of overtime hours raises the overall satisfaction among team members. Mann and Maurer [21], Manhic and Vrana [14] Moore et al. [17], and Manhic and Zabka [15] also argue that improving the lines and channels of communication within the team, as well as externally, with other organizational units and with the customer increases employee satisfaction. In particular better information about project progress and more direct feedback about the work in progress are emphasised as influential indicators for employee satisfaction. Finally, a general agreement among most of the above authors, supported also by Moe and Dingsøyr [10] and Green [22], was that Scrum provides general working conditions and a working environment that have a bearing on a positive social life at the workplace as well as on work pleasure, also described as job satisfaction, that increase overall employee satisfaction. We therefore included these two indicators also into our framework.

#### **4 Research Setting and Method**

We chose a case study approach to research the impact of Scrum on information systems and software development processes and projects. The chosen case organization has approximately 40 years of experience in solving complex IT tasks. Some years ago it changed from being publically owned to private company. It has about 3,000 employees, who are involved in the development of administrative and statutory software solutions. The investigated case department falls into the latter category and has 45 employees. Its sole product is a case management system for municipal job centers, which gives administrators the opportunity to work across different platforms. For the development of the case management system, the department previously followed the traditional waterfall model. In 2011 it launched the implementation of Scrum as the preferred development model. At the time of our investigation, the department had completed three full releases with the use of Scrum. As such the department had the profile of the unit of analysis which we were looking for: an organization that had recently, within the past year, chosen to implement Scrum, and that had previously used the traditional waterfall model. With the former model still in their minds we expected the employees to make candid assessments of the impact of Scrum as compared to the past.

As we were not able to make direct measurements nor had direct access to data, such as number of interruptions, uninterrupted development hours, number of overall, integration or minor errors, over time registrations, etc., we chose to directly ask respondents about their perceptions of the given concepts. The indicators, which we had derived from the literature review, were therefore transformed into direct questions for our interviews, which we validated with 2 employees in a small pilot study before putting them to the 11 interview partners, who were available for the study. We developed 3 largely overlapping interview guides for the three stakeholder groups, with 6 developers as respondents, 4 respondents in leadership roles such as Scrum master, Product owner or unit managers and one representative from the service department, which is responsible for external liaisons. All interviews were recorded, transcribed and handed over to the respondents for approval. The results of our analysis were also presented to the participants of this study and the case organization at large.

The data collection with standardized interviews allowed both collections of qualitative and quantitative data. We first asked the respondents to numerically assess, on a scale from -5 to + 5, for each indicator its individual change, improvement or decline, as compared to the situation before the implementation of Scrum and then to evaluate its impact on the concept in question. After that quantitative judgment we asked into the reasons for these assessments, which provided rich qualitative data. This combination of data allowed for data and method triangulation to improve the validity of our findings [23]. The subsequent analysis was based on mean values for the quantitative data within each indicator; these were interpreted on the basis of the qualitative opinions. The results were then compared and discussed with regard to published Scrum guidelines and findings from the literature. It is worth pointing out that the numerical element of the collected data should be considered secondary. The interviews were intended as the primary source to collect qualitative data with a statistical element - and not vice versa. The quantitative data was exclusively used to create an indication and an overview over any specific area.

## **5 Finding and Results**

The investigation of Scrum's value creation through and impact on productivity, quality and employee satisfaction in information systems and software development was part of a larger study, which both developed and applied a comprehensive framework consisting of a further two concepts. Although a presentation of the overall result would give a more comprehensive portrait of the method's value and impact we have here focused on three concepts mostly due to length limitations. This still provides some valuable insights and where necessary we will relate to the other two concepts. As a background for our subsequent discussion in the following we summarize the results of our analysis concerning Scrum's value and impact on productivity, quality and employee satisfaction in the case unit.

### **5.1 Scrum's Impact on Productivity**

Table 1 summarizes the respondents' assessment of Scrum's impact on productivity. Despite some individual variations the respondents' mostly positive scores indicate their favourable assessment and an improvement in productivity after the implementation of Scrum.

We found that the decrease in the number of interruptions was limited, but it had led to a significant, perceived impact on productivity. In addition, in the changed process interruptions now came from an authorized person and were thus being less perceived as disturbances. The perception of the number of uninterrupted, continuous development hours had only seen a very modest increase. The respondents reasoned that the use of Scrum had led to more meetings than in the past which led to interruptions in the continuity of their work. The frequent meetings resulted, however, in a better understanding of the tasks. This was appreciated by the respondents as having a positive impact on their productivity as they thought they now both worked more efficiently and tackled unforeseen challenges much better.

Prioritizing new functionality higher than error-free deliveries had been the organization's strategy to avoid endless development cycles. Nevertheless these were expe-

rienced by the majority of the respondents. The increased focus on delivering functional software in defined, short iterations has prevented endless development and has resulted in more productivity, however not on the expense of product quality as described in the next subsection.

**Table 1.** Scrum’s impact on productivity

Productivity	Improvement	Impact on productivity	Score Range
No of interruptions	1.4	2.0	0 - 4
No of uninterrupted development hours	0.8	1.3	0 - 3
Endless development cycles	2.8	2.8	1 - 5
Compliance with deadlines	2.9	2.1	0 - 5
Repetition of mistakes	1.1	1.4	-1 - 3
Bug fixing time	0.5	1.7	-2 - 3
Employee performance	3.5	4	3 - 4

The respondents felt that Scrum’s decomposition and prioritization of tasks had positively changed the compliance to deadlines and had had a positive impact on productivity in general. With regard to the repetition of mistakes there was also positive development. Primarily the respondents' explained this progress with Scrum’s focus on self-organization and not that much with the practice of retrospectives, which are the method’s explicit mechanism for the identification of weaknesses and subsequent process and product improvements.

Bug fixing time was the area with the least perceived improvement compared to the other indicators. Despite low average ratings, respondents expressed that although the actual time spent had not decreased, bug fixing now happened at a much better and appropriate time in the process. Thus, its impact on productivity was assessed significantly higher.

The managers among the respondents assessed that the employees' performance had increased significantly. They provided two different arguments for this. First Scrum’s emphasis on process clarity resulting in visibility and transparency which we had identified as a separate concept for investigating Scrum’s impact on information systems and software development, made it compulsory for developers to publically present their work and take a position with regard to any challenges they had encountered. As a consequence they put more focus on the execution of their tasks. The other reason was related to the avoidance of project overruns. In the past overruns had always been passed through the chain of development tasks, with the results that the developers or even more so the testers became time-pressured and could not do their job properly. The shorter iterations carried out by a multidisciplinary team avoided this effect and resulted in overall better performance.

## 5.2 Scrum’s Impact on Quality

Table 2 summarizes the respondents’ assessment of Scrum’s impact on quality. There were two indicators, where the respondents had perceived improvements and a posi-



tive change. These were the number of defects per KLOC and the number of minor errors. The two parameters were interdependent, as we from the responses could conclude that the reduction of the number of defects per KLOC had mostly come from the reduction of the number of minor errors; there had been little change in the number of serious errors, especially no change in the number of integration errors, which were usually categorised as serious defects. The reason that there had been an improvement in this area, was that with the implementation of Scrum, also unit testing as part of the Scrum team's tasks jointly performed by specific testers and the developers was introduced. This rationale was spelled out explicitly in the responses to our questions concerning the number of defects per KLOC and the number of minor errors in specified development unit and modules. In view of this, we conclude that the noteworthy improvement in this area had come about through the interplay between the introduction of unit and iterative testing as featured and emphasised by Scrum.

**Table 2.** Scrum's impact on quality

Quality	Improvement	Impact on quality	Score Range
Defects per KLOC	2.7	3.2	1 - 4
No of integration errors	0	0.6	-2 - 2
No of minor errors	1.9	1.9	0 - 4
End product usability	0	0	0 - 0

The respondents had not perceived any improvement with regard to the number of integration errors. The minimal increase of the impact on quality, according to respondents, was an outcome of the participation of staff from different professional areas and different sub-projects in the sprint meetings and their resulting increased understanding of the product under development. In addition, individual respondents believed that although the level of complexity and the number of interfaces between modules had increased, the constant number of integration errors indicated that quality had not been impaired; on the contrary, they saw this as a positive effect on the resulting quality. No change was perceived concerning the overall useability of the end product. However, in the absence of the possibility to access customers and end users, questions relating to this indicator were posed to employees in leadership roles, who admitted that the organisation in the future, beyond representatives from the service department, had to include customers directly into the development process to both improve useability and to make informed judgements about it.

### 5.3 Scrum's Impact on Employee Satisfaction

Table 3 summarizes the respondents' assessment of Scrum's impact on employee satisfaction. The examination of the concept shows both areas where respondents had experienced positive changes, but also areas where the situation was largely unchanged. The latter proved to be the case in the assessment of overtime. Some respondents felt that this area had improved for two reasons. First of all, there had been a relief that Scrum had not led to the amount of overtime work that the individual respondents had feared. In addition, Scrum had made it more fun to work, which di-

verted attention from overtime. Common to most respondents concerning this indicator was, however, that the reorganization of work process had a neutralizing effect on the negative perception of overtime.

**Table 3.** Scrum’s impact on employee satisfaction

Employee satisfaction	Improvement	Impact on employee satisfaction	Score Range
Overtime	0.4	0.9	0 - 5
Project progress	2.2	2.0	0 - 4
Communication	1.7	1.3	-3 - 4
Feedback	0.8	1.3	0 - 3
Social life	1.9	2.1	0 - 4
Work pleasure	2.8	3	-2 - 5

With regard to the possibilities to track, monitor, and follow a project’s progress the respondents perceived a clear improvement. With the exception of a few, most respondents highlighted the role of product backlogs in their responses. The majority of the respondents felt that using backlogs was rewarding, but at the same time, they found them cumbersome to work with because of a, at times, paucity of clarity and transparency which were explained with a lack of experience with this tool and expected to disappear in the future.

In relation to the assessment of lines of communication, the respondents were divided into two factions. Those in leadership roles felt there had been a deterioration since the transparency of who at management level was associated with the various Scrum teams had become blurred and elusive. In contrast, respondents in developer roles had the perception that the communication paths had clearly improved. The reason for this was that as part of the Scrum implementation clearer guidelines had been put in place about the ways any communication should take in case of problems; in addition the introduction of Scrum of Scrums had improved communication across the different teams. Concerning the amount of feedback there had not been a notable change. One respondent however put strongly forward that the amount of feedback had indeed increased, which explains the positive mean value of both the perceived improvement and its impact on the indicator.

When respondents were asked to assess the social life in the Scrum teams, most of them responded that they saw an improvement in this respect. The improvement had been achieved because professional and disciplinary boundaries between the different roles had been broken down and because staff were now sitting together in an open office landscape. Finally, pleasure of work or job satisfaction was the indicator which by far received the most top scorings of all indicators – the largest number of ‘5 scores’. All but one respondent felt that Scrum had supported the elevation of job satisfaction. Breaking tasks down in smaller bites, cross disciplinary collaboration and increased process clarity and transparency of the development process and the product under development had all together contributed to this perceived improvement.

## 6 Discussion

On this background, in the following we contrast our empirical data with the literature on agile information systems and software development and in particular the identified writings about Scrum.

### 6.1 Empirical Findings on Productivity and the Agile Development Literature

There are a number of areas that impact on productivity; according to the Scrum and agile development literature, which we reviewed for this research, these are: sprints, a focus on functional software, retrospectives, self-organization, the product backlog and the daily scrum meetings.

The iterative sprint development process [3] plays a central role in the use of Scrum. In the case unit, sprints had made it easier to comply to deadlines as tasks were now decomposed in smaller manageable items with clear definitions, which allowed for their easier handling and execution. These results are confirmed in empirical work reported by among others Augustine et al. [24], Vidgen and Wang [25] as well as Wang and Vidgen [26]. Scrum's increased focus on iterative delivery of functional software should according to the literature increase productivity while avoiding falling into endless development cycles in an attempt to develop the 'perfect piece' of software [17]. This was the effect Scrum had on the case unit, which thus was an area where the method lived up to the expectation.

Retrospectives are intended to increase the productivity among others as a result of the project participants' learning from their own and others' mistakes, so that errors and faults are not repeated in the next sprint or iteration. Retrospectives should address both, the overall application of the method, its processes and practices, but also the more specific experience in the daily development work and its relation to the resulting product [4]. The latter turned out to be an area the case unit did not focus on and thus did not benefit from in their daily work. This prioritization of topics discussed during retrospectives can be explained with the case unit's early stage of utilizing Scrum and their lack of experience with regular retrospectives. In the case organization this area should therefore get further attention with an increased focus on Scrum's practices to support learning. Self-organization in a Scrum team has among others the objective to protect and relieve individual team members from certain tasks and create an environment where they are not constantly disturbed in their work. In a successfully self-organized team, everyone has insights into the other team members' tasks, while at the same time a Scrum master is clearly identified and appointed [3]. This means that when there is a need for input from a specific team member, the other team members are not unnecessarily disturbed, as the tasks have been clearly defined, broken down and distributed. If in doubt, the Scrum master is available to facilitate or solve the problem. At the case unit this had not yet been fully achieved, which meant that employees were still interrupted and disturbed in their work and further efforts will be needed to progress. However, one of the benefits of the Scrum master role had been achieved already: the respondents expressed that the interruptions now came from the right person.

In the literature the avoidance of repeating errors is ascribed to retrospectives. As discussed above in the case unit retrospectives had not yet been applied to their full

potential, yet the perception of the respondents had been that the repetition of errors had drastically decreased. This was attributed to the influence that self-organization had. As a consequence of the increased individual developer's responsibility now, team members had become more mindful not to repeat the same mistakes. Individual and collective mindfulness have been reported as characteristics of agile development independently of a particular method or agile practice [27]. This supports that the lack of exploiting retrospectives in the case organization has been compensated by self-organization and mindfulness to lead to a positive outcome with regard to avoiding the repetition of mistakes.

In the case organization the introduction of a product backlog had primarily an effect on compliance with deadlines. As the work was now broken down to single items, there were ongoing opportunities to check whether the agreed schedule was met. Additionally, there was now the possibility to prioritize and plan the order of executing the items in an appropriate manner, which according to the literature (see e.g. [3]). further increases the overall productivity. The introduction of a product backlog at the case unit had affected both of these areas positively, and thus the overall productivity. Product backlogs can also be used to plan a specific test, debugging and error correction period in form of a dedicated item for these tasks [4]. In the case unit this did not lead directly to a reduction of the time spent on bug fixing, but it had resulted in bug fixing happening at a more appropriate point in time, which ultimately had had an impact on productivity.

Finally, daily Scrum meetings, have among others the objective to create visibility in a Scrum team. This helps that everyone in a team gains insight into what the others are working on and at the same time it makes it difficult for employees to conceal modest work efforts, since they publically have to communicate and document their results [3]. The latter had a substantial impact on employee performance in the case unit. It affected productivity positively as openly explaining why a task took longer than expected had the psychological effect that it deprived the employees of the opportunity to hide behind a task longer than necessary. In addition, the meetings had both a positive and negative impact on the number of uninterrupted development hours. The increased number of meetings had reduced the amount of uninterrupted development hours. This was outweighed, however, by the fact that the meetings created better visibility, oversight and knowledge. This allowed employees to tackle unforeseen challenges better, which had a positive effect on productivity as waste time was avoided.

## **6.2 Empirical Findings on Quality and the Agile Development Literature**

The Scrum measures, which have an influence on quality, are according to the reviewed literature: the Scrum team, the Scrum team's maturity, the sprints and the sprint reviews.

A well-functioning Scrum team raises the quality of the end product through its interdisciplinary cooperation, its team dynamics and its utilization of self-organization [3]. Based on the respondents' assessments we can conclude that case unit had managed to exploit these areas as there had been a noticeable improvement in the decrease of the number of defects per KLOC and of the number of minor errors. A Scrum team's maturity also contributes to the improvement of quality; it affects quality

through increasing the mutual understanding within the team and by raising the level of self-organization [12]. Scrum was a relatively new initiative of the case unit, and our results showed that the Scrum team had so far only achieved a limited degree of maturity, especially with regard to reaching out beyond the team boundaries. This was probably one of the main reasons that there had been no significant improvements in either the number of integration errors and end product usability.

A change of the development process to sprints can affect quality in several areas [3]. The largest impact on quality happens through a change to a process where analysis, design, programming and testing activities are not performed separately, but in parallel during defined periods of time. The case unit had performed its first iterative development cycles with Scrum as classic 30-day sprints. After a period, learning from their own practice concerning a sustainable rhythm of work, the unit decided to run 14-day sprints. This led to a further refinement and decomposition of the tasks the team and its individual members were working on, and an even greater focus on the delivery of functional software. Both actions, according to respondents, helped to reduce the number of defects per KLOC and the number of minor errors, which had a positive impact on the product quality as a whole.

According to the literature [3] a sprint review meeting should be held at the end of each sprint. At this meeting customers have the opportunity to interact directly with the development team. Quality can be raised as customers at these meetings have the opportunity to provide input, feedback, but also objections and change requests. The latter may be the case if, according to the customer there has been too big a discrepancy between the agreed sprint goal and the developed software. Before the meeting ends any disagreements should be discussed and resolved [3]. The case unit had in this area not utilized customer and user involvement as intended by the method. One possible explanation for not involving customers to a larger extent could be that the case organization was concerned that individual incoming requests would be too diverse to be fully integrated in the standard information system under development. Another reason could simply be a lack of experience with customer involvement, a deficiency the organisation intends to resolve in the future. The lack of user involvement was one of the main reasons that the perception of end product usability remained unchanged despite the implementation of Scrum. The case unit might experience better results in terms of final product usability, as the Scrum team becomes more mature in the use of the method and thus learns which actions, processes and tools, likely including active and direct customer participation, are best suited for them. In any case the case unit has to consider a change of practice in the way customer will be involved in the development process to make the most of Scrum.

### **6.3 Empirical Findings on Employee Satisfaction and the Agile Development Literature**

According to the reviewed literature the measures that have an impact on employee satisfaction are: sprints, Scrum master, Scrum teams, Scrum of Scrums, product backlog, and burn down charts.

The change to a defined development process, in the form of 14-day sprints, had an influence on employee satisfaction in the case unit. Schwaber and Beedle [3] put

forward that at the end of each sprint there should be a defined piece of developed and tested functional software. They highlight that it is up to the team itself to judge and decide how much work needs to be performed in a sprint to reach the agreed sprint goal. In the case unit the short and iterative development cycles had the effect that the employees found their work more satisfying and enjoyable. They therefore did not consider overtime as something necessarily negative. We can conclude although the case unit with regard to functional software - not every sprint ended with such a result yet - did not fully follow Schwaber and Beedle's [3] advice, this had no negative consequences, on the contrary the case unit actually experienced progress with regard to employee satisfaction, specifically increased job satisfaction and pleasure of work, due to the utilisation of sprints and the reorganization of the development processes. This effect corresponded well with what the literature suggests as potential improvement.

According to Schwaber and Beedle [3], the introduction of sprints also alleviates the social life and communal atmosphere in the teams as employees encounter progress and success due to the frequent delivery of a functional product at the end of each sprint. The case unit had commenced to celebrate milestones and started to experience these effects, however thus far not to the extent predicted by Schwaber and Beedle [3].

The Scrum master also plays a significant role in employee satisfaction. Schwaber and Beedle [3] emphasize that clearer guidelines concerning the lines of communication involving the Scrum master, as well as an active protection of the Scrum team through this role from unnecessary disturbances, both are actions that contribute to increasing employee satisfaction. The case unit had followed the recommendation from the literature and established Scrum masters and communication guidelines, which resulted in a positive outcome evidenced by the respondents' answers concerning the lines of communication. It is however important to emphasize that the improvement of communication links were only perceived at the Scrum team level and not at a senior level beyond the Scrum masters. As the case organization becomes more familiar and accustomed to the use of Scrum, Schwaber and Beedle [3] argue that management will also experience these improvements. Concerning team structures the respondents repeatedly mentioned that the destruction of disciplinary boundaries and the gathering of team members in the Scrum teams with different professional background had contributed positively to their perceptions of employee satisfaction. This is an intended effect of interdisciplinary Scrum teams [3]. Self-organization as part of Scrum teams' governance contributed to the respondents' favourable assessment. On this background we can conclude that the case unit has been able to benefit from the composition of their Scrum teams. This is also manifested through the respondents' positive statements regarding the indicators social life and work pleasure.

Scrum of Scrums should be used in large, complex development projects where several Scrum teams are associated [4]. The idea behind the Scrum of Scrums is to ensure the sharing and exploitation of the potential knowledge that exists between different Scrum teams. The responses we received document that the way the case unit had chosen to use Scrum of Scrums had affected the lines of communication between the teams in a positive direction despite the fact that no clear guidelines for this particular communication activity had been developed by the case unit. Although

it remained unclear whether the case unit used Scrum of Scrums, as the literature recommended it, the use of Scrum of Scrums did not only contribute to improve communication lines in general, but also had a positive influence on the ability to monitor project progress. The interaction between the Scrum teams raised the common understanding of the individual Scrum team's status, where they positioned themselves as compared to the other teams and with regard to the development process of the overall product.

Schwaber [4] argues that the product backlog is one of the most important tools in Scrum as breaking down the development tasks to more tangible and manageable size and to increase their visibility is essential for a well functioning product backlog. The case unit has had difficulties with the utilisation and exploitation of the product backlog: the employees felt that the product backlog was unmanageable and cumbersome to work with. This indicates that they did not use the product backlog as proposed by f. ex. Schwaber and Beedle [3]. Nevertheless, they also felt that the product backlog had helped to improve their ability to monitor project progress as there had been no monitoring tool for this before the Scrum implementation. In the context of product backlogs, another tool, namely a burn down chart is intended to provide information about the development work. Depending on the status of the burn down chart it is the Scrum master's responsibility to adjust the number of hours available for the unfinished development tasks, so that the team can reach the sprint goal on time [4]. Although the case unit used burn down charts, it was surprising that no respondents mentioned that this had contributed to any improvement with regard to monitoring and following project progress. A possible explanation could be that the respondents were not yet fully aware of the difference and the different roles the product backlog and the burn down charts play in a Scrum managed development process. This did however not have negative effects on the respondents' overall positive perception of employee satisfaction.

## **7 Conclusion and Contribution**

We have applied Schryen's [1] taxonomy of IS business value as an analytical and structuring device and demonstrate its overall viability. We identified and studied different types of performance measurements in particular productivity, quality and employee satisfaction as value creating measures and provide a useful operationalization of the concepts through 17 indicators. Our positive assessment of Scrum through these measures for the value and impact of information systems and software development confirms empirically the expectations and claims, which are made in many of the conceptual and non-academic writings we had identified in our literature review. It also fills a gap in the area of empirical studies of the value and impact of agile software development [11].

Our study is built upon subjective perceptions; as with all qualitative studies of this kind we of course have to take the danger of positive bias and a respondents' tendency of reporting future expectations rather than stating actual perceptions into account. However, the fact that the respondents reported no or only minimal impact on some of the indicators gives confidence that the reported efforts were genuine rather than showing a general positive bias. In doing so, we however follow Schryen's [1] call and go beyond hard measures and indicators. While it might be argued that

our results lack the objective strength of economic value measured in monetary revenue, they provide insights about individual and organisational impact, benefits and values as perceived by the different stakeholders. We take into account internal capabilities which are usually out of scope of the value considerations, and regard the normally disregarded subjective preferences of stakeholders. We investigate an IS business value generation process within its context and environment, in our case agile development, which usually is ignored.

While not making up for a lack of a theory of IS business value our discussion sheds light on the value generation process and the measures we present deal with some of the deficiencies in current IS business value research and contribute to filling gaps in the research agenda as put forward by Schryen [1]. His distinction of internal and external value was helpful, the distinction of tangible and intangible is however not that easy to apply. While productivity is overly linked to tangible value and employee satisfaction is clearly linked to intangible value, elements of quality are both tangible and intangible. The work presented here is a first contribution to solving the identified challenges, however further research is needed to understand the IS business value and IS value in general.

## References

1. Schryen, G.: Revisiting IS business value research: what we already know, what we still need to know, and how we can get there. *EJIS* 22, 2, 139-169 (2013)
2. Johansen, T., and Uldahl, A.: Measuring the Impact of the Implementation of the Project Management Method Scrum (in Danish). MSc Thesis. Copenhagen Business School, Denmark (2012)
3. Schwaber, K., Beedle, M.: *Agile Software Development with Scrum*. Prentice Hall, Upper Saddle River (2002)
4. Schwaber, K.: *Agile Project Management with Scrum*. Microsoft Press, Redmond, Washington (2004)
5. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MISQ* 26, 2, 13-23 (2002)
6. Kohli, R., Grover, V.: Business value of IT: an essay on expanding research directions to keep up with the times. *J. AIS* 9, 1, 23-39 (2008)
7. DeLone, W. H., McLean, E. R.: Information systems success: the quest for the dependent variable. *ISR* 3, 1, 60-95 (1992)
8. DeLone, W. H., McLean, E. R.: The DeLone and McLean model of information systems success: a ten-year update. *J. of MIS* 19, 4, 9-30 (2003)
9. Brynjolfsson, E., Hitt, L.: Paradox lost? Firm-level evidence on the returns to information systems spending. *Mgt Science* 42, 4, 541-558 (1996)
10. Moe, N., Dingsøyr, T.: Scrum and Team Effectiveness: Theory and Practice. In: *Proc. Conf. on Agile Processes in Software Engineering and Extreme Programming*, pp.11-20. Springer, Berlin (2008)
11. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. *Inf. & Soft. Tech.* 50, 9-10, 833-859 (2008)
12. Guang-Yong, H.: Study and Practice of Import Scrum Agile Software Development. In: *Int. IEEE Conf. on Communication Software and Networks*. Nanjing, China. IEEE Press, New York (2011)



13. Appelo, J.: *Management 3.0 - Leading agile Developers, Developing agile Leaders*. Addison-Wesley, Crawfordsville (2010)
14. Mahnic, V., Vrana, I.: Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process. *El.-Tech. Rev.* 74, 5, 241-247, Ljubljana (2007)
15. Mahnic, V., Zabkar, N.: Using COBIT Indicators for Measuring Scrum-based Software Development. *WSEAS Trans. on Comp.* 7, 10, 1605-1617 (2008)
16. Sutherland, J., Altman, I.: *Organizational Transformation with Scrum: How a Venture Capital Group Gets Twice as Much Done with Half the Work*”, In: 43rd HICCS. Kauai (2010)
17. Moore, R., Reff, K., Graham, J., Hackerson, B.: *Scrum at a Fortune 500 Manufacturing Company*. In: Agile Conference. St. Paul (2007)
18. Thatcher, M.E., Pingry, D.E.: An economic model of product quality and IT value. *ISR* 15, 3, 268–286 (2004)
19. Thatcher, M.E., Pingry, D.E.: Understanding the business value of information technology investments: theoretical evidence from alternative market and cost structures. *J. of MIS* 21, 2, 61–85 (2004)
20. Devaraj, S. Kohli, R.: Information technology payoff in the health-care industry: a longitudinal study. *J. of MIS* 16, 4, 41–67(2000)
21. Mann, C., Maurer, F.: A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. In *Agile Development Conference*, pp.70-79. IEEE Computer Society (2005)
22. Green, P.: *Measuring the Impact of Scrum on Product Development at Adobe Systems*. In: 44th HICSS. Kauai (2011)
23. Andersen, I.: *The apparent Reality* (in Danish). Samfundslitteratur Publisher, Frederiksberg (2006)
24. Augustine, S., Payne, B., Sencindiver, F., Woodcock, S.: Agile project management: steering from the edges. *Comm. of the ACM* 48, 12, 85-89 (2005)
25. Vidgen, R., Wang, X.: *Organizing for Agility: A Complex Adaptive Systems Perspective on Agile Software Development Process*. In: Ljunberg, J., Andersson, M. (eds.) 14<sup>th</sup> ECIS, 1316-1327. Gothenburg (2006)
26. Wang, X., Vidgen, R.: *Order and Chaos in Software Development: A comparison of two software development teams in a major IT company*. In: Winter, R., et al. (eds.) 16<sup>th</sup> ECIS, pp. 807-818. St. Gallen (2007)
27. Matook, S., Kautz, K.: *Mindfulness and Agile Software Development*. In 19<sup>th</sup> ACIS, pp. 638-647. Christchurch (2008)