Towards an artifact to support agile teams in software analytics activities

Joelma Choma*, Eduardo M. Guerra*, Tiago Silva da Silva†, Luciana A. M. Zaina‡, Filipe Figueiredo Correia§¶
*National Institute for Space Research, São José dos Campos, Brazil
Email: jh.choma@hotmail.com, guerraem@gmail.com
†Federal University of São Paulo, São José dos Campos, Brazil
Email: silvadasilva@unifesp.br
‡Federal University of São Carlos, Sorocaba, Brazil
Email: lzaina@ufscar.br
§Faculty of Engineering, University of Porto, Porto, Portugal
¶INESC TEC, FEUP Campus, Porto, Portugal
Email: filipe.correia@fe.up.pt

***Abstract*—Software analytics supports data-driven decision making, which allows software practitioners to leverage valuable insights from software data to improve their processes and many quality aspects of the software. In this paper, we present an artifact designed from a set of patterns to support agile teams to plan and manage software analysis activities, named Software Analytics Canvas. Further, we report the study undertaken to evaluate the ease of use and the utility of our canvas from the practitioners' viewpoint, and a participatory design session carried out to collect information about possible artifact improvements. In general, subjects found the artifact useful, but some of them reported difficulties in learning and understanding how to use it. In the participatory design, they pointed out improvement points and a new layout for the canvas components. The results of both studies helped us refine the proposed artifact, improving both the terms used in each element and the layout of the blocks to make more sense for its users.**

***Index Terms*—software analytics, agile teams, decision making, software data**

I. Introduction

An increasing number of companies around the world have used data analytics to make decisions about their businesses. Analytics refers to the use of analysis, data, and systematic reasoning to inform the decision making process [1].

Nowadays, researchers and practitioners already use analytics applied to software data for better decision making concerning many aspects of software quality and its development process [2]. Zhang et al. [3] coined the term "Software Analytics" (SA) to label research in this area. Since then, SA has been widely adopted by large companies. However, it has not yet reached its full potential for broad industrial adoption. For small companies, for example, SA is an open question and rarely addressed [4]. Furthermore, to the best of our knowledge, there is no consolidated approach on how to introduce software analytics concepts and practices into an agile development context.

By emphasizing short feedback cycles, agile teams regard changes as an opportunity to improve the product at any time in the development process, always focusing on delivering value [5]. In this sense, software analytics can support agile teams to make more appropriate changes based on actual data, rather than only on their personal experiences or intuitions. Additionally, the adoption of data-driven continuous improvement can help agile teams to save resources and decrease the cost of building and maintaining the software [6]. Process improvement is one of the main reasons for measurement in agile software development [7]. However, the measurement tends to be immediate and straightforward, since agile principles emphasize measuring progress in terms of working software over measuring intermediate work products [5].

Currently, various tools provide structure on top which data-driven improvement processes can be implemented. However, due the perceived complexity and effort required to set up such tools and to establish a measurement program, agreed with the urgency of the product delivery, lack of time, and others reasons, many agile teams end up not systematically using metrics to track product and process performance [8]. There is a lack tools to ease the adoption of software analytics, but also a lack of approaches to effectively change the habits within agile teams towards data-driven decision making.

Seeking a practical method to introduce *software analytics* into agile teams, we have outlined an artifact to support software analytics activities in agile environments named Software Analytics Canvas (SA Canvas). It was designed based on a set of patterns that we have identified from experience reports of researchers and practitioners in the software analytics area [9], [10]. In this paper, we present the canvas and report two studies designed to (i) evaluate the usefulness and ease-of-use perceived by the users, and (ii) refine the design of the proposed artifact. The goal of these studies is to answer the following research questions (RQs):

- RQ1: What are the users' perceptions about the usefulness and ease-of-use of the SA Canvas?
- RQ2: What characteristics can be improved in the design of the SA Canvas?

DOI reference number: 10.18293/SEKE2019-146

To answer RQ1, we recruited six software practitioners for planning and managing activities from a software analytics project using the canvas. As input to the project, we provide information from a web-based system in the Space Weather area. After three iterations with the canvas, the participants were able to evaluate the usefulness and ease of use of the artifact. To answer RQ2, we call the same subjects for a participatory design session to provide ideas for possible artifact enhancements.

II. Background

A. Software Analytics

Software analytics allows software practitioners (developers, testers, designers, and managers, to name a few) to leverage insightful information (accurate and in-depth) and actionable (with real practical value) for completing various tasks around software systems, software users, and software development processes [11]. The SA process comprehends monitoring, analysis, and understanding of software data to support the decision-making process throughout the different phases of the software lifecycle [3].

Within the SA context, we refer to the software data as any data generated from different sources such as code, bug reports and test executions recorded in software repositories (e.g., version control systems and issue-tracking systems), and information about data of usage typically stored in the log files [12]. Furthermore, SA has been used to address different type of concerns, such as issues related to the source code (e.g., code quality, bug proneness, number of defects, and amount of effort to fix bugs) [13]; development process (e.g., productivity and ROI) [4]; product business (e.g., usage of features, data quality, and user satisfaction) [14]; and software runtime properties (e.g., performance, number of transactions and error log) [15].

B. Software Analytics Patterns

Considering software analytics as an essential practice for leveraging value delivery in agile contexts, we have identified in previous studies a set of eight patterns focusing on how to incorporate SA activities into agile practices on a continuous basis to inform the decision-making process of software practitioners, including project managers, analysts and software developers from small, large, or multiple teams. Below, we present the eight SA patterns and a brief description of the proposed solutions in each of them. For a more detailed description of these patterns, see [9] and [10].

1) **What You Want to Know**: refers to defining the key issues that the development team wants to focus on, in order to guide their selection of the appropriate means for measurement, assessment and monitoring these issues throughout the project.
2) **Choose the Means**: refers to defining the data sources and most appropriate means, such as tools, techniques and other approaches for selecting and collecting data that will be useful for future decisions.
3) **Software Analytics Planning**: refers to adding tasks related to the *software analytics* on the to-do list to be prioritized with the regular project tasks according to the team's demand for information.
4) **Analytics in Small Steps**: it means that the tasks related to the *software analytics* can be distributed throughout the project, adding information to the team about the system at small portions by adjusting the granularity of the analytic activities.
5) **Reachable Improvement Goals**: refers to defining reachable improvement goals from the *software analytics* findings and break the activities down into smaller tasks to fit together with the other tasks.
6) **Learning from Experiments**: refers to create an alternative solution and perform an experiment collecting data that allow the comparison with the current solution.
7) **Define Quality Standards**: refers to define quality standards and establish minimal or maximum thresholds for any software aspect that the team intends to monitor.
8) **Suspend Measurement**: refers to suspend measurement of the issues with a low possibility of recurrence, or measurement of the issues that need to be continually monitored, but the team has defined that, for some reason (e.g., effort, cost or other project constraints), the monitoring of these issues cannot be implemented immediately.

III. Research Approach

In order to develop an artifact to support agile teams throughout the software analytics activities, our research approach follows the guidelines of Design Science Research (DSR), as proposed by Hevner et al. [16]. DSR is a problem-solving paradigm, where "knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artifact".

In this paper, we describe the design process of the SA Canvas artifact, which includes the building and evaluation cycles. A first version of the canvas emerged from patterns previously identified from experiences reports in the SA field (Section II-B). Aiming at evaluating the artifact and improving its characteristics, we have carried out a formative evaluation in two rounds (Section V).

For the first round, we selected six subjects (in pairs) to plan and manage software analysis activities using the SA Canvas. After the hands-on experience of handling the artifact, we gathered the users' perceptions of usefulness and ease of use of the proposed artifact, using the Technology Acceptance Model (TAM) [17]. According to Davis [17], *perceived usefulness* (PU) refers to "the degree to which a person believes that using a particular system would enhance his or her job performance"; and *perceived ease of use* (PEU) refers to "the degree to which a person believes that using a particular system would be free of effort". We measured both variables within the TAM through a multiple-item questionnaire using a 6-point Likert scale – from "Completely Disagree" to "Completely Agree".

For the second round, we organized a redesign session using principles of participatory design (PD) [18] to collect information for artifact improvement, involving the same participants of the first round. During the PD session, the participants were encouraged to draw sketches as an idea for the canvas redesign.

IV. Software Analytics Canvas Design

This section presents the SA Canvas template we have designed as an artifact to support the software analytics activities into agile development environments. Canvas artifacts are visual maps – structured and preformatted – used to support the collaborative teamwork in their communication processes. Canvas is considered a hands-on tool that fosters understanding, discussion, creativity, and analysis on a given matter [19]. Nowadays, there is a range of applications using canvas – e.g., development of new businesses, conception, and planning of projects, strategic alignment of projects, value proposition, among others.

The SA patterns presented in Section II-B were the basis for the creation of our canvas. As shown in Figure 1, it is composed of seven blocks with names based on the patterns. The seven names are Key Issues, Data Sources, Data Gathering, Analytics Implementation, Insights, Incremental Goals, and Quality Thresholds. Note that, we added in each block a guiding question to help beginner users.

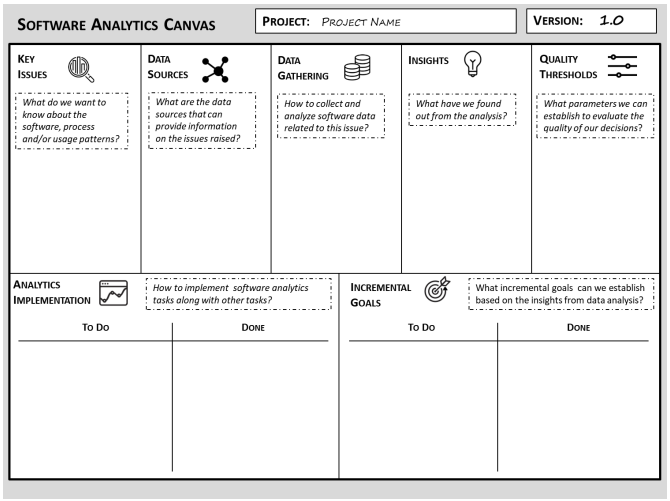

Fig. 1. Software Analytics Canvas [1st Version]

The explanation of each canvas block is presented below, including its description, the related patterns, and its corresponding guiding question.

- **Key Issues.** As a first step, the team can raise issues that need to be verified, analyzed and improved. As mentioned in Section , the issues can be, for example, related to the internal quality of the system (e.g., code quality), external quality of the system (e.g., performance, bug density, the effort required to fix defects), productivity (e.g., effort estimation), and/or usage patterns (e.g., usability, user satisfaction). This element is related to the pattern called *What You Want to Know*. The guiding question is: *What do we want to know about the software, process and/or usage patterns?*
- **Data Sources.** After defining the key issues, the team identifies what kind of data is needed to know more about the issue raised, and from which sources the data should be extracted, for example, a dump of the database system on recent transactions, source code, behaviors' user, historical data about bugs incidence, etc. For specific issues, the team may recognize the need to collect data from multiple sources for cross-referencing. The two patterns related to this element is *Choose the Means* and *Learning from Experiments*. The guiding question is: *What are the data sources that can provide information on the issues raised?*
- **Data Gathering.** After identifying the data sources, the team should decide which metrics and tools will be used to gather the data. The team can, for example, enable the collection of specific code metrics in the development environment, export from SGBD data referring to a given period, verify the need to create a specific script to extract data from the software repository, etc. Furthermore, the team needs to decide which methods and tools will be used to analyze the data collected. The team can employ from simple statistical methods (e.g., descriptive and inferential statistics) to more sophisticated methods (e.g., data mining, natural language processing, machine learning, etc.), according to the type and amount of data. Also, the team also needs to decide on the tools to support their analysis. For example, the team can opt for *software analytics platforms* which can be configured according to the team's needs. This element also is related to both *Choose the Means* and *Learning from Experiments*. The guiding question is: *How to collect and analyze software data related to this issue?*
- **Insights.** The team analyzes the results obtained from the collected data and discusses possible solutions and insights to making-decision. For example, the team find that "tests have low coverage in module X", "customers prefer this approach" and so on. Then, Insights are raised from the search for solutions. Notice that, sometimes, the data analysis did not reveal significant information about the issue raised. So, the team will decide whether to continue the investigation by collecting new data or if the issue is disregarded, once no action is necessary. This element is a trigger for *Reachable Improvement Goals* pattern. The guiding question is: *What have we found out from the analysis?*
- **Quality Thresholds.** When implementing the improvements via informed decision-making, the team can evaluate the impact of the changes by collecting feedback from stakeholders. From collecting feedback, the team will have enough information to decide whether to consider the issue resolved, or whether the issue should be monitored for longer. Concerning unresolved issues, the team will decide whether they will be re-analyzed from new data, or discarded. Ideally, the team should establish

the quality thresholds values for any issue that the team decides to evaluate or to keep in monitoring. For example, in issues related to coverage testing, the response time cannot exceed 2 seconds, or the test coverage must be at least 80%. *Define Quality Standards* is the patterns related to this element. The question is: *What parameters we can establish to evaluate the quality of our decisions?*

- **Analytics Implementation.** The team should plan how to conduct analysis activities to seek meaningful information from collected data. These activities should be included on the to-do list and prioritized along with the other development tasks. In order to avoid overloading the team, such activities – that also includes the preparation of the analytical infrastructure – can be distributed throughout the project and executed by steps resulting in something deliverable. This block is divided into two regions, the first for the works to be done, and the other to control the work done. *Software Analytics Planning* and *Analytics in Small Steps* are the patterns related to this element. The guiding question is: *How to implement software analytics tasks along with other tasks?*
- **Incremental Goals.** From their insights, the team discusses and define reachable goals to put their solutions into practice, considering that these improvements can be made incrementally. Therefore, the most important in this step is to define where the team wants to reach and what goals they want to achieve. *Reachable Improvement Goals* is the related pattern to this element. However, if the current goals have been fulfilled, the pattern is *Suspend Measurement*. The main question is: *What incremental goals can we define based on the insights from data analysis?*

V. Artifact Evaluation

This section describes the two studies conducted to evaluate the SA Canvas built to aid agile teams during the planning and managing of software analytics activities.

A. Perceived Usefulness and Ease-of-use (RQ1)

We have conducted the first study in order to investigate the users' perceptions of the usefulness and ease-of-use of the SA Canvas (RQ1). This study was conducted in the Laboratory of Computation and Applied Mathematics at the National Institute for Space Research in Brazil, where we selected six subjects from the course of Agile Projects. Participants had at least three years of experience in software development, and only one of them had no experience with agile methods. We divided the participants into three pairs, seeking to balance them based on the experience of each participant. Table I shows the participants' background and their experience time in developing software (in years).

During the study, participants were invited to plan and manage a software analytics project based on a real case using SA Canvas. For this purpose, we prepared for each team a SA Canvas using a whiteboard and a document describing the case study which should be used as input for the planning and management of the software analytics project. As for the case study, we describe a real case of EMBRACE, one of the research centers in the area of space weather at INPE. In the document, we described the context of study related to a web system to make available some of the products developed by the researchers working in the EMBRACE. In addition to the information about the products developed, we included in the document some concerns related to the web portal that the researchers of the space weather had the intention to investigate and solve. Considering the possible demands described in the document for a software analytics project, we prepared a list of possible data sources to give participants some examples. Moreover, then, we created a tutorial printed on A4 with the illustration of the canvas, the description of its components, and the guiding questions (in Portuguese).

TABLE I
Participants Characterization

Pairs	Subjects	Background	Professional Experience
G1	P1	web development	6 to 10
	P2	software architecture	16 to 20
G2	P3	systems analysis	3 to 5
	P4	systems analysis	11 to 15
G3	P5	computer programming	3 to 5
	P6	systems analysis	16 to 20

We conducted a pilot study with two researchers from the computer lab to evaluate their iteration with the SA Canvas on the whiteboard and the other materials – case study description, list of possible data sources, and canvas tutorial. From the pilot study, (i) we established that peers should have to raise at least two issues of software analytics; and that (ii) we should carry out a warm-up exercise together with all participants before they begin planning their projects. After that, we held four weekly meetings with participants, as showed in Figure 2.

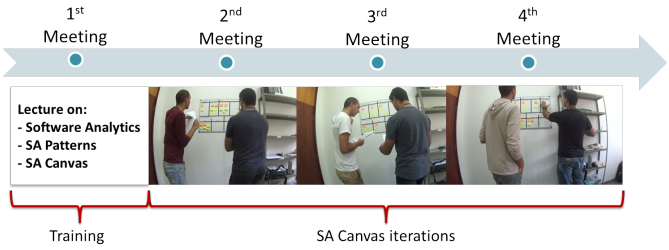

Fig. 2. Meetings over time.

1st Meeting. At the first meeting, we introduced concepts about software analytics, provided an overview of SA patterns, and presented SA Canvas using the practical example related to code coverage improvement. For more information, we asked the participants to read the two articles on software analytics patterns [9] [10] before the next meeting. After the meeting, participants read and signed an informed consent form to participate in this study.

2nd Meeting. At the second meeting, we invited the participants to a practical exercise to understand how they should use the canvas. One of the participants provided us with a real example of his own work. From this example, we helped them to plan software analytics activities on the board using sticky notes. This activity took approximately one hour.

After warming up, a pair of participants at a time should fill out four blocks of the Canvas (Key Issues, Data Sources, Data Gathering, and Analytics Implementation) based on information written in the document on the EMBRACE case study and the support material (list of possible data sources and the canvas tutorial). Additionally, we asked teams to use different color sticks for different key issues. Each pair took from 25 to 40 minutes to fill the four blocks with at least two key issues. The sessions were recorded for future analyzes, and one of the researchers observed the activities in silence, taking notes on the interaction of participants with the artifact.

After the first iteration, we prepared a report with (i) feedback about how they had used the canvas (including corrective actions), and (ii) fictional information on the evolution of the activities planned by them and executed by the developers. Also, we have introduced some insights mixed into the text to be extracted by them. Because each team raised different key issues, we then prepared three different documents for each of them.

3rd Meeting. The report's information was used as input for the second iteration when the team should adjust some fill-in mistakes pointed out by researchers, update the canvas with the tasks done, proceed in the planning their activities for the next iteration, and fill in the remaining canvas blocks from the insights. As in the first iteration, the sessions were recorded and observed by one of the researchers. The sessions took from 40 to 60 minutes. From the results of this iteration, we prepared a new report for each team with the feedback on the use of the canvas, other fictitious information about the evolution of the project, and new insights mixed into the text.

4th Meeting. As in the previous iteration, from the researchers' report, the teams should adjust some fill-in mistakes pointed out by researchers, update the canvas with the tasks done, proceed in the planning their activities for the next iteration, and fill in the remaining canvas blocks from the insights. The sessions took on average 40 minutes, and just like the previous ones, they were recorded and observed by one of the researchers. At the end of the iteration, we asked the participants individually to answer a questionnaire with questions of the TAM [17], based on their experiences during the software analytics project when they used SA Canvas to plan and manage its activities.

Findings. The questions answered by the participants can be found in the following link: https://goo.gl/ZbEcC3. Figure 3 shows the responses from the questionnaire, where to some degree all participants agree that the SA Canvas is useful. However, there was some disagreement concerning the ease of use. Some participants do not agree that it was easy to learn and understand what should be done at certain times (PEU1 and PEU2). The learning curve certainly will have an impact on the user ability to handle the artifact (PEU5). Furthermore, users tend to be more critical as to the usefulness of the artifact when its use is not effortless.

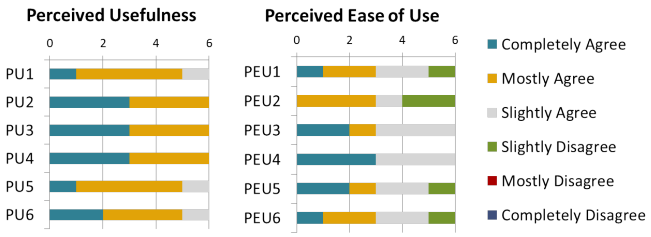

Fig. 3. Perceived usefulness and use of use.

B. Participatory Redesign (RQ2)

To identify what and how we could improve in the SA Canvas (RQ2), we conducted a participatory design session with the same participants from the previous study. The method used to conduct the redesign section was divided into three steps. In the first step, each participant received a document with the description of all the components of the canvas and a form to inform their particular suggestions to improve SA Canvas. After that, each participant should sketch a redesign proposal. In the second step, they consolidated their opinions with their peers by drawing a new sketch for canvas redesign. Lastly, in the third step, all participants discussed their redesign proposals and proposed a single sketch to the canvas. Two of the researchers participated in the session as observers making notes of relevant information. On average, each step lasted 30 minutes.

Findings. From the participants' suggestions, we can identify the need to better clarify at least three blocks:

- In "Insights", three participants suggested making it clear that insights should be described from the results of the analysis.
- The name of the "Quality Thresholds" block seems not to be suitable. Participants suggested making it clear that the values may be minimum or maximum.
- Understanding what should be considered in "Incremental Goals" was difficult for them. Some have suggested a more appropriate name since the main idea is to implement the improvements.

As a final result, the participants sketched the canvas with the following characteristics:

- At the top of the canvas, there are five blocks reserved for planning (inputs and outputs): "Key Issues", "Data Sources", "Data Gathering", "Incremental Goals", and "Quality Threshold".
- The "Quality Threshold" block has been subdivided into two parts to include a minimum acceptable value (minimum) and the goal to achieve (goal).
- At the bottom of the canvas, there is a block called "Analitycs Tasks" block instead of "Analytics Implemen-

tation". This block has been subdivided into four parts to accommodate the tasks to do, in progress, done, and the impediments.

VI. Discussion, Conclusion and Future Work

Although feedback, continuous improvement, inspection, and adaptation are always present in the agile speech, there is little evidence on measurement in practice. Often this measurements and adaptation are adopted on an ad hoc basis. Software analytics is an approach that aims at data-driven continuous improvement, but it is not very widespread in practice.

To bridge this gap, we previously identified patterns on how software analysis could be introduced into the software development process continuously. By looking for a more efficient and practical way to apply the SA patterns into agile development context, we proposed a canvas addressed to the software analytics activities taking into account how teams communicate and collaborate. SA Canvas can be considered a goal-focused approach, similar to the well-known Goal Question Metric (GQM) approach proposed by Basili et al. [20]. The goal-focused approaches are good ways to ensure that measurement goals are articulated with the metrics being collected, and also, to avoid having useless measurements.

We argue that the SA Canvas is a suitable artifact to agile teams' informative workspaces where various techniques and tools for software visualization are commonly applied as information radiators – e.g., count of velocity automated tests, continuous integration status, incident reports, and so forth [21].

In our viewpoint, our canvas has two interesting characteristics. First, it can work as a hub in terms of information flow related to software analytics projects. Information hubs can be considered spaces where information flows meet and decisions are made. Second, the proposed artifact is a situation awareness channel, which considers how people are kept informed about what is happening [22]. For Agile teams, the support of different visualization techniques and tools throughout the software development process is crucial to foster awareness and communication. Additionally, this factor can have a positive impact on the sense of purpose [8] when the professionals follow the evolution of their actions within a cycle of continuous improvement.

As a contribution of this paper, we present the methods used for formative evaluation of the artifact. As a result, we found that the SA Canvas is a useful artifact, but some participants reported difficulties in learning and understanding how to use it. To investigate what could be improved, we carried out a participatory design session, where the same subjects pointed out the components of the canvas that needed to be redefined and provided a sketch with a new layout of the canvas' blocks. These results will help us to refine the proposed artifact. In future work, we will empirically investigate the use of SA Canvas in an industrial setting and experienced agile teams in order to get more relevant research results concerning its effectiveness and impact on the software analytics process.

Acknowledgment

This work is financed by National Funds through the Portuguese funding agency FCT – Fundação para a Ciência e a Tecnologia – within project: UID/EEA/50014/2019.

References

[1] T. H. Davenport, "Make better decisions," *Harvard business review*, vol. 87, no. 11, pp. 117–123, 2009.

[2] R. P. Buse and T. Zimmermann, "Analytics for software development," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM, 2010, pp. 77–80.

[3] D. Zhang, Y. Dang, J.-G. Lou, S. Han, H. Zhang, and T. Xie, "Software analytics as a learning case in practice: Approaches and experiences," in *Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering*. ACM, 2011, pp. 55–58.

[4] R. Robbes, R. Vidal, and M. C. Bastarrica, "Are software analytics efforts worthwhile for small companies? the case of amisoft." *IEEE software*, vol. 30, no. 5, 2013.

[5] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "Manifesto for agile software development," 2001.

[6] A. E. Hassan and T. Xie, "Software intelligence: the future of mining software engineering data," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM, 2010, pp. 161–166.

[7] D. Hartmann and R. Dymond, "Appropriate agile measurement: using metrics and diagnostics to deliver business value," in *Agile Conference, 2006*. IEEE, 2006, pp. 6–pp.

[8] O. Liechti, J. Pasquier, and R. Reis, "Supporting agile teams with a test analytics platform: a case study," in *Proceedings of the 12th International Workshop on Automation of Software Testing*. IEEE Press, 2017, pp. 9–15.

[9] J. Choma, E. M. Guerra, and T. S. Silva, "Patterns for implementing software analytics in development teams," in *Proceedings of the 24th Conference on Pattern Languages of Programs*. ACM, 2017, p. 12.

[10] ——, "Learning from experiments, define quality standards, suspend measurement: Three patterns in a software analytics pattern language," in *Proceedings of the 12th Latin American Conference on Pattern Languages of Programs (SLPLoP)*. ACM, 2018, p. 10.

[11] D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang, and T. Xie, "Software analytics in practice," *IEEE software*, vol. 30, no. 5, pp. 30–37, 2013.

[12] F. Shull, "Data, data everywhere..." *IEEE software*, no. 5, pp. 4–7, 2014.

[13] J. Guo, M. Rahimi, J. Cleland-Huang, A. Rasin, J. H. Hayes, and M. Vierhauser, "Cold-start software analytics," in *Proceedings of the 13th International Workshop on Mining Software Repositories*. ACM, 2016, pp. 142–153.

[14] S. Pachidi, M. Spruit, and I. Van De Weerd, "Understanding users' behavior with software operation data mining," *Computers in Human Behavior*, vol. 30, pp. 583–594, 2014.

[15] R. Musson, J. Richards, D. Fisher, C. Bird, B. Bussone, and S. Ganguly, "Leveraging the crowd: how 48,000 users helped improve lync performance," *IEEE software*, vol. 30, no. 4, pp. 38–45, 2013.

[16] A. Hevner and S. Chatterjee, "Design science research in information systems," in *Design research in information systems*. Springer, 2010, pp. 9–22.

[17] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, pp. 319–340, 1989.

[18] S. Bφdker, K. Grφnbæk, and M. Kyng, "Cooperative design: techniques and experiences from the scandinavian scene," in *Readings in Human–Computer Interaction*. Elsevier, 1995, pp. 215–224.

[19] A. Osterwalder and Y. Pigneur, *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons, 2010.

[20] V. R. Basili, G. Caldiera, and D. H. Rombach, "The goal question metrics approach," *Encyclopedia of Software Engineering*, vol. 1, pp. 528–532, 1994.

[21] K. Beck and C. Andres, "Extreme programming explained: Embrace change," 2004.

[22] E. Berndt, D. Furniss, and A. Blandford, "Learning contextual inquiry and distributed cognition: a case study on technology use in anaesthesia," *Cognition, Technology & Work*, vol. 17, no. 3, pp. 431–449, 2015.