Objectives and Challenges of the Utilization of User-Interaction Data in Software Development

Sampo Suonsyrjä, Outi Sievi-Korte, Kari Systä, Terhi Kilamo
Tampere University of Technology
P.O.Box 527, FI-33101 Tampere, Finland
{sampo.suonsyrja, outi.sievi-korte, kari.systa, terhi.kilamo}@tut.fi

Tommi Mikkonen
University of Helsinki
P.O.Box 68, FI-00014 Helsinki, Finland
tommi.mikkonen@helsinki.fi

***Abstract*—Understanding users, their requirements and usage patterns helps building better software. To continuously improve operations, user-interaction (U-I) data can provide developers with interesting new possibilities. This study aims at gaining an understanding of how software teams can start the use of U-I data and what challenges they face with it. In this paper, we describe three cases from different organizations. This includes explaining the activities, objectives, and challenges of each team in their efforts to begin using U-I data. We conducted the study with Action Design Research (ADR) method, resulting in findings from each case by intervening in the work of these teams. As a contribution, we designed a U-I data utilization method that summarizes teams' activities. Secondly, we refined and validated the categorizations of U-I data analysis and utilization objectives, and finally we categorized the challenges that the case teams faced. Together, these contributions lay out clear steps also for other software teams in starting the utilization of U-I data.**

I. Introduction

The use of data is increasing in all fields of business and data has been regarded even as ”the main driver for innovation” [1]. Consequently, also data-driven software engineering is trending with organizations such as Google, Facebook, Microsoft, and IBM leading the way [2]. For software teams, an interesting option in this field is to collect user-interaction (U-I) data by tracking the actions of users in the user-interface level. For example, the number of times a certain button is pressed during a use session produces such U-I data. With proper analysis, these data provide concrete touch-points for software teams to what the users are actually doing with the software [3].

However, for teams who are only beginning the use of U-I data, it can be problematic to even understand what kind of objectives they could have for the collection and analysis of U-I data. For advancing the use of U-I data, a one size fits all guidance to software teams is difficult to come by, even if the emergence of general tools, such as Hotjar[1] and Google Analytics[2], have simplified the collection and analysis of data especially in the field of web development. For installable applications, the common solutions seem to focus on specific tasks such as Apache Log4j 2[3] on logging. In addition, scientific models such as the data dimension of the Stairway to Heaven [4] or the HYPEX model [5] can provide good starting point for understanding the field. However, there seems to be a lack of understanding of how organizations concretely start the utilization of U-I data and what kind of challenges are related with it. To fill the gap, we study

1) what kind of activities software teams do when starting the utilization of U-I data,
2) what kind of objectives software teams have for U-I data,
3) and what kind of challenges software teams face after starting the utilization of U-I data.

The study is a part of a large research effort, where the aim is to support data-driven software development. Previously, we have studied three case organizations (A, B & C) on how software teams choose collecting techniques for U-I data [6]. This time, we intervene with the work of the same software teams in order to start the actual use of U-I data. As contributions, (1) we design a U-I data utilization method to model and guide the workflow of how software teams start the use of U-I data. Then, (2) we examine the categorizations of U-I data analysis and utilization objectives from our previous work [7] in practice. Finally (3), we categorize the typical challenges with the utilization of U-I data.

II. Background and Related Work

Data from the Use of Software: Many terms are used to describe the data that is gathered from a software system. The data types for collected data vary by their nature in respect to time or type that is generated. One category of software data is *runtime data*, which consists of product data gathered as the system is running. Tightening this, Holmström Olsson and Bosch [8] have used the term post-deployment data for data that are generated by a product after its commercial deployment. Barik et al. [9] have approached software data somewhat similarly by events. They split this event data into two categories based on the behavior of the data, namely logs and telemetry. U-I data, as used in this paper, can be categorized as telemetry with the specification that it originates from user-interactions. Shuur et al. [10] have coined the term ”software operation knowledge” and categorized data types related to SOK into usage (e.g. U-I), performance, quality and feedback data.

Uses for Data in Software Development: In the same study, Schuur et al. [10] have defined a framework that

[1] https://www.hotjar.com
[2] www.google.com/analytics
[3] https://logging.apache.org/log4j/2.x/index.html

lists uses for software operation knowledge. Similarly, in our previous work [7], we categorized uses for post-deployment data and analyses that software teams could conduct on post-deployment data based on a questionnaire survey. In a bit more general level, Begel and Zimmerman [11] collected an extensive 145 question list of questions that software developers would like to have answers to in software development. The question of ”How do users typically use my application?” was ranked essential the most often in their survey.

Studies by Rodriguez et al. [2] and Sauvola et al. [3] investigated data collection practices and how monitoring customer usage scenarios, A/B testing and other experiment tools can be used to collect useful feedback and help with understanding of customer expectations. Also a larger literature review for customer feedback collection techniques in software was done by Fabijan et al. [12].

While those studies focus on data collection and its challenges, additional studies by Roehm et al. [13] and Guzman et al. [14] focus more on the methods that are used to analyze the data and how this can be used to enact improvements on the software product. The research by Guzman et al. has culminated to the Q-Rapids framework, where quality and functional requirements are managed together, evaluated incrementally and runtime data from the system is aggregated into indicators that can be used to support decision making in further development cycles. Somewhat similarly, the HYPEX model by Olsson et al. [5] depicts how data from the use of software systems can be not only taken into account in software development but also how to make such data a driver for the process. In an even more concrete manner, Kohavi et al. [15] have created a practical guideline on how to use the different data types gained. However, their work is limited to the web context specializing on controlled experiments such as A/B testing. Bosch [4] investigated the different levels of data use in software development. The data dimension of the Stairway to Heaven model has been extended to also include project management variables and it also includes customer satisfaction, project throughput, revenue and such. In this study, we focus specifically on U-I data.

III. Findings

A. The U-I Data Utilization Method

We conducted this part of the study with the Action Design Research (ADR) method [16] that emphasizes the joint effort of academics and practitioners. We intervened with the work of the teams to start the utilization of U-I data from February 2016 to February 2017. During this time, we collected research data that consist mainly of workshop memos written down by the first author and/or the team members. These included plans for how, why and what U-I data would be collected. We recognized that all the teams implemented very similar activities and we illustrate these with the designed U-I data utilization method. The U-I Data Utilization method is formed of three steps, each consisting of two to three activities as seen in Table I. The table also includes descriptions of how the activities were executed in the three cases. The three steps frame their activities both time-wise and in the sense of how the activities are related to each other. The steps and the activities are described as follows.

The first phase of the method is the **Proof-of-Concept** step. Two evident points in this step are to first *select the collecting technique* and to then *test the selected collecting technique* so it can be integrated with the intended target software. However, there can be other reasons for producing such a prototype as well, such as for internal marketing of the U-I data collecting or to get an approval for it from other parts of the organization. Another part in this phase is to *brainstorm the analysis and use objectives* for U-I data. This initial idea generation can be important also in the selecting of the collection technique. However, the brainstorming can take place multiple times inside this step, both before and after the technique selecting.

The second step in the U-I data utilization method is to **Aim & Deploy**. The brainstorming activity of the first step should result in an excess number of use and analysis objectives, and in most cases all of these cannot be collected in the end. This is due to multiple factors such as limitations in collecting techniques and U-I data analyzing skills of the team members. Based on these factors, the *selecting of the analysis and use objectives* is required before *concretely defining the U-I data collecting points*. The collecting technique and objective selections can affect this defining significantly. Therefore, this activity should be taken into account already in the selection of the collecting technique.

The final phase of the method is to **Collect & Analyze** the U-I data. The activities in this step – *collecting the data*, *analyzing the data with the selected objectives* and *presenting the results for use in the selected objectives* – vary greatly depending on the selections made until this step. Therefore, as our main concern of this step we point out that the capabilities of the team to do these activities and the characteristics of their software project should be taken into account much before coming to this step. As future work, we want to do research on these capabilities. However, based on the findings of this study it seems that these range from things such as how much time the team has available for the U-I data analyzing to their skills and tools of doing that analysis, and from the length of their deployment pipeline to their access to collect U-I data either freely from any actual end-user or restrictively from substitute user-groups.

B. Objectives for the Analysis and Use of U-I Data

For validating and refining the categorization of U-I data analysis and use objectives identified originally in [7], we used a directed approach to qualitative content analysis. The goal of such analyses is to validate or extend conceptually a theoretical framework or theory as described in [17].

1) Analysis Objectives: The analysis objectives found in cases are summarized in Table II. The categories are sorted according to the rank of the occurrence of the objectives. Analysis objectives targeting at evaluation of features were the most frequently selected among their categories. The analysis categories are the following.

TABLE I
STEPS AND ACTIVITIES OF THE U-I DATA UTILIZATION METHOD AND DESCRIPTIONS OF THEM IN CASES

Activity	Case A	Case B	Case C
Step1: Proof-of-Concept			
1: Brainstorming the Analysis and Use Objectives	Internal session led by a UX specialist in May 2016 resulting in a list of 46 questions	Workshop in March 2016 with the first author and two members of the software team resulting in eight objectives for U-I data	Internal session in October 2016 led by the software architect resulting with five objectives
2: Select a Collecting Technique	Monitoring the execution environment of the team's software system with a tool developed by the 1st author	Manual instrumentation in combination with an automated collecting tool	Aspect-oriented programming
3: Test the Collecting Technique	Additional demonstrative application developed and presented in a cross-team demo show	Testing with the local development environment	Testing with the local development environment
Step2: Aim & Deploy			
4: Select the Analysis and Use Objectives	23 questions selected by the first author and three team members	Three features selected for monitoring, each with two parallel implementations	No exclusions of initial questions
5: Define the Collecting Points	Tool configured by the first author and one team member. Merged to the code base.	Code snippets for monitoring the few places attached by one team member & 1st author	Data defined to be collected in selected U-I events by team member and 1st author, collecting snippets integrated by the team member
Step3: Collect & Analyze			
6: Collect Data	Company internal testing day in December 2016. Data sent over HTTP to a MySQL database.	Two-week collecting period in April-May 2016. Data sent to automated collecting tool.	Collecting period in February 2017. Data logged to users' local computers and then sent to team via email.
7: Analysis with the Objectives	Data wrangling in Rstudio and spreadsheet program. E.g. session time calculations	Usage statistics ready in automated tool	Data wrangling in Rstudio and spreadsheet program. E.g. use path visualization
8: Present the Results and Use within the Selected Objectives	Workshop session in February 2017	Presentation to all team members in May 2016	Presentation sessions to a team member in March 2017

TABLE II
ANALYSIS OBJECTIVES IN THE CASES

Analysis Objectives	*Initially*	*Selected*
Value Analysis	27	16
Pain Point Analysis	14	8
Use Paths	13	6
User Profiling	5	1
Total	59	30

TABLE III
USE OBJECTIVES IN THE CASES

Use Objectives	*Initially*	*Selected*
UX	39	19
Informed Feature Development	11	7
Resourcing and Prioritizing	5	1
Requirements Validation	4	3
Total	59	30

Value Analysis: The objective is to find out how valuable a certain feature or system is. In the most simplistic case this is done by calculating how many times the feature has been used in a certain time period and using that information in forming an estimate of the value of the feature. In addition to how often a feature is used, value evaluation can be based also on who is using, where from and at what time. Example from the internal brainstorming session's memo of case A: *"What are the features that are most commonly search for/selected?"*

Pain Point Analysis: The objective is to shed light on possible problems that users might face. These can be related to system performance, low usability etc. Data collected for these analyses likely include timestamps so that the time can be calculated and compared with others or with what was expected. Although not strictly U-I data, collecting error logs is similarly done likely to respond to this objective. Example from case A: *"How long time does it take to save a template?"*

Use Paths: The objective is to find out the order of the use of features. The commonality and abnormality of individual paths can be analyzed or the analysis can focus simply on what is the order on average. For example, a question in case C was formed as *"Use of library: What is done and what is the order?"*

User Profiling: The objective is to segment users based on the U-I data that is collected. Differentiating can be done in many ways such as expertise of user (novice and expert) or technology they use the system with (browser). For example, one analysis objective in case A was: *"What browser and browser version does the user have?"*

2) Use Objectives: The second category of objectives focuses on the different uses that the case teams had for U-I data. Table III summarizes the occurrences of objectives by each category and displays also how many of the objectives were selected for the actual collecting and analysis in the cases. User eXperience (UX) was the most popular use objective category. The use categories are the following.

Informed Feature Development: The objective is to get insights to support decision making about how or whether features should be developed (further). The insights can include pointers to which features require improvement and guidelines for new designs. These are then to be used for making informed decisions in the daily work of the software developers. For example, the team in case C targeted the "use path" analysis (mentioned as the example above) at finding out if they could develop a new feature called "recently opened" to their library functionality.

Requirements Validation: Requirements for a system might come from other stakeholders than users. The objective highlights this by aiming to validate the requirements with the users. For instance, the intention for analyzing which browsers are used (i.e. the example from the user profiling category) was to know how many browsers the software should support. For existing features these objectives can start off easily by collecting usage information, but for non-existing features approaches such as creating a minimum viable product [18] (or rather a minimum viable feature) need to be applied.

Resourcing and prioritizing: The objective is to use results for prioritizing or resourcing related decision making. Such results could be formed in a feature's value analysis, which would then affect the priority of the related development tasks in the products backlog. Similarly, additional development staff or server space could be resourced towards the high-valued features. For example, the above value analysis example from case A could work as a basis for prioritizing the development work of certain features.

UX: The objectives of this category are similar to the ones in the informed feature development category but they focus specifically on the system's User eXperience (UX). For example, a question formed as *"Does the user open the Help?"* from the case C was targeted at understanding in which situations the users miss help.

C. Challenges in the Utilization of U-I Data

We interviewed the case teams in January 2018 after a year from the initial U-I data utilizations. For analysis, we used conventional content analysis [17] to form categories of the challenges the teams faced with U-I data utilization. We found three **themes** of challenges that all of the case teams had faced after a year from starting the utilization of U-I data. These divide into *sub-themes* that characterize the main themes. In the following, we describe the themes and present translated quotes from the interviews accordingly.

Team members had **value concerns** of U-I data utilization in the sense that they thought it had either *low value* or *unclear value*. UX Specialist in Case B: "I guess, its value is not seen here so high that someone else than me would start doing it." Seeing the benefit and value under the bottom line seemed especially difficult, which then can result in other challenges such as getting organizational permissions. That again, makes the challenges in this theme significant.

The interviewees had also faced challenges related to the **difficulties in U-I data utilization**. Sub-themes of *Technical concerns and difficulties, Difficulties in the extraction and or use of insights, High effort required for U-I data utilization, Scalability concerns for U-I data utilization, Lack of experience in U-I data utilization, Using Unspecific Objectives* characterize these challenges. For example, the software architect in case C mentioned how "In this context the experience was that although we got some results out of the analysis, it was still pretty difficult to draw conclusions."

Finally, the team members brought up challenges related to the **unsuitability of U-I data utilization in the current situation**. Some interviewees felt they were faced with *lack of resources* or *lack of support from organization*. They also reported on how U-I data utilizing had *conflicts with ways of working, methods, technical environment and/or culture*. For example, the operational development manager in case A described such conflicts: "If our product was in the Internet world we would have implemented this already. But you see this is not a service but a product there behind firewalls". Additionally, the interviewees had *Privacy Concerns* as well as *Legal Concerns*.

IV. Discussion and Threats to Validity

We designed the U-I data utilization method to guide software teams that are transforming or simply trying out more data-driven software development methods. It combines the activities we identified in the cases of this study and lays out practical steps for other software teams who have similar intentions. There are some similar methods in the scientific literature. For example, a framework presented by van der Schuur et al. in [10] describes software operation knowledge and its identification, acquisition, integration, presentation and utilization in software engineering. The framework seems data centric in the sense that it is designed to reveal the potential role of software operation knowledge. At the same time, it lacks the concreteness that would help practitioners in starting the use of such knowledge. However, it takes into account also the perspectives of the company and customer and not only of the development as is in our study. Given the widening field of tasks that especially agile software teams are responsible for, the utilization objectives from the company and customer perspectives might become increasingly important for the software teams. On the other hand, the HYPEX model [5] is also highly specific with its objective on how to integrate the use of data, e.g. U-I, into software development process. Compared to our U-I data utilization method, the HYPEX model requires a radical change in the software development process whereas the method designed in this study rather focuses on supporting the existing software processes.

Considering the use and analysis objective categories, we found that the *UX* related use objectives and *Value Analysis* related analysis objectives were selected the most. Many of the objectives in the *Value Analysis* category are quite simple to approach when compared to objectives in categories such as *Use Paths*. Similarly, the objectives in *UX* category likely are more concrete than the objectives e.g. in the *Resourcing and Prioritizing (R&P)*, and they do not necessarily need other information to support decision making as is the case with *R&P*. In the three cases, only team B had experience in collecting U-I data. In this sense, it seems quite natural how all of the teams decided to start by trying out these simpler tasks even if they identified various other objectives in the first brainstorming sessions. Moreover, we think that with the extended descriptions and lively quotes from the cases, the refined analysis and use objective categories can provide software teams with helpful insights and inspiration about what to do with U-I data.

Within this research effort, we had a great opportunity to study the same cases after one year from the initial U-I data utilizations. This resulted in an in-depth look to the challenges the case teams had faced during the year. What was similar among all the cases, was that all of them had faced the challenges related to the unsuitability and difficulty of U-I data utilization. Each of them also had concerns about its value. Considering the inspirational success stories such as [18], the reports on low or unclear value of data utilization sound alarming. However, the other two challenge themes found in this study can explain this. For example, in some cases the technical environment as a whole requires radical changes to make U-I data utilization possible. The changes might be required also in areas that are not in the control of the software team. Such challenges were categorized as process and customer immaturities in [7].

Considering validity, we point out that while the number of case teams was low (3), the case organizations were highly different in their characteristics among each other. In this sense, the results of this study provide a detailed look on the activities, objectives, and challenges of software teams in the utilization of U-I data. Therefore, the results can give interesting insights to a wide audience even if the results of this study would not be generalizable for any software team. The bias from using only one researcher for most of the work in this kind of a study is significant. To lower this, the meeting notes were shown to and accepted by team members in each case. Moreover, the second, third, and fourth authors participated in the analysis of the research data. Additionally, the second author, who participated in the final interviews and in the analysis parts of this study, did not participate in the ADR part.

V. Conclusions

This study aimed at gaining an understanding of how software teams can start the use of U-I data and what challenges they face with it. In this paper, we described three cases from different organizations by explaining how the teams identified analysis and use objectives for U-I data, selected collecting techniques, collected the data and analyzed the results. We conducted the study with ADR, and as a contribution designed a U-I data utilization method that summarizes the steps of the case teams in their efforts of starting the use of U-I data. In addition, we validated and refined the categories of U-I data analysis and use objectives we had found in our previous work.

In addition, we studied the challenges software teams can face while starting the utilization of U-I data. The results contribute to the practice in the sense that the presented U-I data utilization method lays out steps for other software teams to try similar processes. At the same time, the found challenges can give valuable consideration points while the use and analysis objectives work as inspiration.

References

[1] H. H. Olsson, J. Bosch, and H. Alahyari, "Towards r&d as innovation experiment systems: A framework for moving beyond agile software development," in *Proceedings of the IASTED*, 2013, pp. 798–805.

[2] P. Rodriguez, A. Haghighatkhah, L. E. Lwakatare, S. Teppola, T. Suomalainen, J. Eskeli, T. Karvonen, P. Kuvaja, J. M. Verner, and M. Oivo, "Continuous deployment of software intensive products and services: A systematic mapping study," *Journal of Systems and Software*, vol. 123, pp. 263–291, 2017.

[3] T. Sauvola, L. E. Lwakatare, T. Karvonen, P. Kuvaja, H. H. Olsson, J. Bosch, and M. Oivo, "Towards customer-centric software development: a multiple-case study," in *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on*. IEEE, 2015, pp. 9–17.

[4] J. Bosch, *Speed, Data, and Ecosystems: Excelling in a Software-Driven World*. CRC Press, 2017.

[5] H. H. Olsson and J. Bosch, "From opinions to data-driven software r&d: a multi-case study on how to close the'open loop'problem," in *Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on*. IEEE, 2014, pp. 9–16.

[6] S. Suonsyrjä, "Eeny, meeny, miny, mo... a multiple case study on selecting a technique for user-interaction data collecting," in *International Conference on Agile Software Development*. Springer, 2017, p. To appear.

[7] S. Suonsyrjä, L. Hokkanen, H. Terho, K. Systä, and T. Mikkonen, "Post-deployment data: A recipe for satisfying knowledge needs in software development?" in *Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2016 Joint Conference of the International Workshop on*. IEEE, 2016, pp. 139–147.

[8] H. H. Olsson and J. Bosch, "Post-deployment data collection in software-intensive embedded products," in *Continuous software engineering*. Springer, 2014, pp. 143–154.

[9] T. Barik, R. DeLine, S. Drucker, and D. Fisher, "The bones of the system: a case study of logging and telemetry at microsoft," in *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM, 2016, pp. 92–101.

[10] H. van der Schuur, S. Jansen, and S. Brinkkemper, "A reference framework for utilization of software operation knowledge," in *Software Engineering and Advanced Applications (SEAA), 36th EUROMICRO Conference on*. IEEE, 2010, pp. 245–254.

[11] A. Begel and T. Zimmermann, "Analyze this! 145 questions for data scientists in software engineering," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 12–23.

[12] A. Fabijan, H. H. Olsson, and J. Bosch, "Customer feedback and data collection techniques in software r&d: a literature review," in *International Conference of Software Business*. Springer, 2015, pp. 139–153.

[13] T. Roehm, B. Bruegge, T.-M. Hesse, and B. Paech, "Towards identification of software improvements and specification updates by comparing monitored and specified end-user behavior," in *Software Maintenance (ICSM), 2013 29th IEEE International Conference on*. IEEE, 2013, pp. 464–467.

[14] L. Guzmán, M. Oriol, P. Rodríguez, X. Franch, A. Jedlitschka, and M. Oivo, "How can quality awareness support rapid software development?–a research preview," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2017, pp. 167–173.

[15] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne, "Controlled experiments on the web: survey and practical guide," *Data mining and knowledge discovery*, vol. 18, no. 1, pp. 140–181, 2009.

[16] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, "Action design research," *MIS quarterly*, pp. 37–56, 2011.

[17] H.-F. Hsieh and S. E. Shannon, "Three approaches to qualitative content analysis," *Qualitative health research*, vol. 15, no. 9, pp. 1277–1288, 2005.

[18] E. Ries, *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Books, 2011.