



The lack of sharing of customer data in large software organizations: Challenges and implications

Downloaded from: <https://research.chalmers.se>, 2024-03-20 09:17 UTC

Citation for the original published paper (version of record):

Fabijan, A., Olsson, H., Bosch, J. (2016). The lack of sharing of customer data in large software organizations: Challenges and implications. Lecture Notes in Business Information Processing, 251: 39-52.
http://dx.doi.org/10.1007/978-3-319-33515-5_4

N.B. When citing this work, cite the original published paper.

The Lack of Sharing of Customer Data in Large Software Organizations: Challenges and Implications

Aleksander Fabijan¹(✉), Helena Holmström Olsson¹, and Jan Bosch²

¹ Faculty of Technology and Society, Malmö University, Nordenskiöldsgatan 1,
211 19 Malmö, Sweden

{Aleksander.Fabijan, Helena.Holmstrom.Olsson}@mah.se

² Department of Computer Science and Engineering, Chalmers University
of Technology, Hörselgängen 11, 412 96 Göteborg, Sweden
Jan.Bosch@chalmers.se

Abstract. With agile teams becoming increasingly multi-disciplinary and including all functions, the role of customer feedback is gaining momentum. Today, companies collect feedback directly from customers, as well as indirectly from their products. As a result, companies face a situation in which the amount of data from which they can learn about their customers is larger than ever before. In previous studies, the collection of data is often identified as challenging. However, and as illustrated in our research, the challenge is not the collection of data but rather how to share this data among people in order to make effective use of it. In this paper, and based on case study research in three large software-intensive companies, we (1) provide empirical evidence that ‘lack of sharing’ is the primary reason for insufficient use of customer and product data, and (2) develop a model in which we identify what data is collected, by whom data is collected and in what development phases it is used. In particular, the model depicts critical hand-overs where certain types of data get lost, as well as the implications associated with this. We conclude that companies benefit from a very limited part of the data they collect, and that lack of sharing of data drives inaccurate assumptions of what constitutes customer value.

Keywords: Customer feedback · Product data · Qualitative and quantitative data · Data sharing practices · Data-driven development

1 Introduction

Traditional ‘waterfall-like’ methods of software development are progressively being replaced by development approaches such as e.g. agile practices that support rapid and continuous delivery of customer value [20]. Although the collection of customer feedback has always been important for R&D teams in order to better understand what customers want, it is today, when R&D teams are becoming increasingly multi-disciplinary to include all functions, that the full potential of customer data can be utilized [21]. In recent years, increasing attention has been put on the many different techniques that companies use to collect customer feedback. With connected products,

and trends such as ‘Big Data’ [1] and ‘Internet of Things’ [19], the qualitative techniques such as e.g. customer surveys, interviews and observations, are being complemented with quantitative logging and automated data collection techniques. For most companies, the increasing opportunities to collect data has resulted in rapidly growing amounts of data revealing contextual information about customer experiences and tasks, and technical information revealing system performance and operation.

However, and as recognized in recent research [8], the challenge is no longer about how to collect data. Rather, the challenge is about how to make efficient use of the large volumes of data that are continuously collected and that have the potential to reveal customer behaviors as well as product performance [1, 6, 7, 9]. Although having access to large amounts of data, most companies experience insufficient use of the data they collect, and as a result weak impact on decision-making and processes.

In this paper, we explore data collection practices in three large software-intensive companies, and we identify that ‘lack of sharing’ of data is the primary reason for insufficient use and impact of collected data. While the case companies collect large amounts of data from customers and from products in the field, they suffer from lack of practices that help them share data between people and development phases. As a result, decision-making and prioritization processes do not improve based on an accumulated data set that evolves throughout the development cycle, and organizations risk repetition of work due to lack of traceability.

The contribution of the paper is twofold. First, we identify that ‘lack of sharing’ is the primary reason for insufficient use of data and we provide empirical evidence on the challenges and implications involved in sharing of data in large software organizations. Second, and based on our empirical findings, we develop a model in which we identify what data is collected, by whom data is collected and in what development phases it is used. Our model depicts critical hand-overs where certain types of data get lost, and how this causes a situation where data does not accumulate and evolve throughout the development process. By capturing ‘current state-of-practice’, and by identifying critical hand-overs where data gets lost, the model supports companies in identifying what challenges they experience, and what implications this will result in. The awareness that the model helps create can work as valuable input when deciding what actions to take to improve sharing of data in large software-intensive organizations.

2 Background

2.1 Collection of Customer Feedback

In most companies, customer feedback is collected on a frequent basis in order to learn about how customers use products, what features they appreciate and what functionality they would like to see in new products [6, 5]. Typically, a wide range of different techniques are used to collect this feedback, spanning from qualitative techniques capturing customer experiences and behaviors [6, 7, 10], to quantitative techniques capturing product performance and operation [10–12]. While the qualitative techniques are used primarily in the early stages of development in order to understand the context in which the customer operates, the quantitative techniques are used post-deployment in order to understand the actual usage of products.

Starting with the pre-development stage, companies typically collect qualitative customer feedback using customer journeys, interviews, questionnaires and surveys [6, 7], forming the basis for the requirements generation [13]. At this stage, contextual information on the purpose of the product or a feature with functional characteristics and means of use are typically collected by customer representatives. Typically, this information is used to both define functional requirements as well as to form customer groups with similar needs and priorities, also known as personas [17].

During development, customer feedback is typically collected in prototyping sessions in which customers test the prototype, discuss it with the developers and user experience (UX) specialists, and suggest modifications of e.g. the user interface [6, 7, 14]. As a result, developers get feedback on product behaviors and initial performance data. Customer feedback is typically mixed and consists of both qualitative information on e.g. design decisions and quantitative operational data [6].

In the post-deployment stage, and when the product has been released to its customers, a number of techniques are used to collect customer and product data. First, and since the products are increasingly being connected to the Internet and equipped with data collection mechanisms, operational data, and data revealing feature usage is collected [6, 14, 15]. Typically, this data is of quantitative type and collected by the engineers that operate the product and service centers that support it. Second, if customers generate incident requests and attach the product log revealing the state of the product, error message and other details. These are important sources of information for the support engineers when troubleshooting and improving the product [10]. Also, and as recognized in previous research [15, 16], A/B testing is a commonly deployed technique to collect quantitative feedback in connected products on which version of the feature offers a better conversion or return of investment. And although increasing amounts of data are being collected, very little is actually being used. The challenges in aggregating and analyzing this data in an efficient way prevent higher levels of the organization from benefiting from it [12].

2.2 Impact and Use of Customer Data

Companies operating within transportation, telecommunications, retailing, hospitality, travel, or health care industries already today gather and store large amounts of valuable customer data [19]. These data, in combination with a holistic understanding of the resources needed in customer value-creating processes and practices, can provide the companies that fully utilize it a competitive advantage on the market [18].

However, challenges with meaningfully combining and analyzing these customer data in an efficient way are preventing companies from utilizing the full potential from it [1, 8]. Instead of a complete organization benefiting from an accumulated knowledge, it is mostly only the engineers and technicians that have an advantage in using this data for operational purposes [12]. Higher levels in the organization such as product management or customer relationship departments need to find ways of better utilizing customer data in order to unlock its potential and use it for prioritization and customer value-actualization processes [18].

3 Research Method

This research builds on an ongoing work with three case companies that use agile methods and are involved in large-scale development of software products. The study was conducted between August 2015 and December 2015. We selected the case study methodology as an empirical method because it aims at investigating contemporary phenomena in their context, and it is well suited for this kind of software engineering research [22]. This is due to the fact that objects of this study are hard to study in isolation and case studies are by definition conducted in real world settings.

Based on experience from previous projects on how to advance beyond agile practices [3, 4], we held three individual workshops with all the companies involved in this research, following up with twenty-two individual interviews. We list the participants and their roles in Table 1.

Table 1. Description of the companies and the representatives that we met with.

Company and their domain	Representatives
<p>Company A is a provider of telecommunication systems and equipment, communications networks and multimedia solutions for mobile and fixed network operators. The company has several sites and for the purpose of this study, we collaborated with representatives from one company site. The company has approximately 25.000 Employees in R&D.</p> <p>The participants marked with an asterisk (*) attended the workshop and were not available for a follow up-interview.</p>	<p>1 Product Owner 1 Product Manager 2 System Managers 2 Software Engineer 1 Release Manager 1 Area Prod. Mng.* 1 Lean Coach* 1 Section Mng.*</p>
<p>Company B is a software company specializing in navigational information, operations management and optimization solutions.</p> <p>Company B has approximately 3.000 Employees in R&D.</p> <p>All the participants attended the workshop and were interviewed.</p>	<p>1 Product Owner 1 System Architect 1 UX Designer 1 Service Manager</p>
<p>Company C is a manufacturer and supplier of transport solutions construction technology and vehicles for commercial use.</p> <p>The company has approximately 20.000 Employees in R&D.</p> <p>All the participants that attended the workshop were interviewed. In addition, one sales manager and one technology specialist wished to join the project at a later stage, and were interviewed.</p>	<p>1 Product Owner 2 Product Strategists 2 UX Managers 2 Function Owners 1 Feature Coord. 1 Sales Manager 2 Technology Spec.</p>

3.1 Data Collection

During the group workshops with the companies, we were always three researchers sharing the responsibility of asking questions and facilitating the group discussion. Notes were taken by two of the researchers and after each workshop, these notes were consolidated and shared to the third researcher and company representatives.

First, we conducted a workshop with an exercise with the post-it notes that build our inventory of the customer feedback techniques. Second, we held semi-structured group interviews with open-ended questions [2] during the workshop. These questions were asked by one of the researcher while two of the researchers were taking notes. In addition to the workshops, we conducted twenty-two individual interviews that lasted one hour in average, and were recorded using an iPhone Memo application. Individual Interviews were conducted and transcribed by one of the researchers. In total, we collected 13 pages of workshop notes, 176 post-it notes, 138 pages of interview transcriptions, and 9 graphical illustrations from the interviewees. All workshops and individual interviews were conducted in English.

3.2 Data Analysis

During analysis, the workshop notes, post-it notes, interview transcriptions and graphical illustrations were used when coding the data. The data collected were analyzed following the conventional qualitative content analysis approach where we derived the codes directly from the text data. This type of design is appropriate when striving to describe a phenomenon where existing theory or research literature is limited. Two of the researchers first independently and then jointly analyzed the collected data and derived the final codes that were consolidated with the third and independent researcher who also participated at the workshops. As soon as any questions or potential misunderstandings occurred, we verified the information with the other researcher and participating representatives from the companies.

3.3 Validity Considerations

To improve the study's construct validity, we conducted the exercise with the post-it notes and semi-structured interviews at the workshops with representatives working in several different roles and companies. This enabled us to ask clarifying questions, prevent misinterpretations, and study the phenomena from different angles. Next, we combined the workshop interviews with individual interviews. Workshop and interview notes were independently assessed by two researchers, guaranteeing inter-rater reliability. And since this study builds on ongoing work, the overall expectations between the researchers and companies were aligned and well understood.

The results of the validation cannot directly translate to other companies. However, considering external validity, and since these companies represent the current state of large-scale software development of embedded systems industry [3], we believe that the results can be generalized to other large-scale software development companies.

4 Findings

In this section, we present our empirical findings. In accordance with our research interests, we first outline the generalized data collection practices in the three case companies, i.e. what types of data that is collected in the different development phases, and by whom. Second, we identify the challenges that are associated with sharing of data in these organizations. Finally, we explore their implications.

4.1 Data Collection Practices: Current State

In the case companies, data is collected throughout the development cycle and by different roles in the organization. Typically, people working in the early phases of development collect qualitative data from customers reflecting customer environments, customer experience and customer tasks. The later in the development cycle, the more quantitative data is collected, reflecting system performance and operation when in the field. In Tables 2, 3 and 4, we illustrate the data collection practices, together with the customer feedback methods and types of data that different roles collect in each of the development stages.

Table 2. Customer data collection practices in the pre-development stage.

	Roles that collect customer feedback	Common customer feedback collection techniques	Common types of customer feedback collected
Pre-Development	Strategy specialists, Product managers, Product owners	Reading of industry press, Reading of published standards, Reading of internal reports, Reading customer visit reviews	Customer wishes, Short/Long market trends, Competitors ability of delivering the product
	Strategy specialists, Feature owners	Telephone interviews, Face-to-face interviews, Conducting group interviews	Existing product satisfaction, Future product specification, Personas and User Journeys

4.2 Data Sharing Practices: Challenges

Based on our interviews, we see that there are a number of challenges associated with sharing of data in large organizations. For example, our interviewees all report of difficulties in getting access to data that was collected by someone else and in a different development phase. Below, we identify the main challenges associated with sharing of data in the case companies:

Table 3. Customer data collection practices in the development stage.

	Roles that collect customer feedback	Common customer feedback collection techniques	Common types of customer feedback collected
Development	UX specialists, Software Engineers	System Usability Scale Form, Asking open ended questions, Demonstrating prototypes, Filming of users' product use	Acceptance of the prototype, Eye behavior and focus time, Points of pain, Bottlenecks and constraints, Interaction design sketches
	System managers, System architects, Software engineers	Consolidate feedback from other projects, Reading prototype log entries	Small improvement wishes, Configuration data, Product operational data

Table 4. Customer data collection practices in the post-deployment stage.

	Roles that collect customer feedback	Common customer feedback collection techniques	Common types of customer feedback collected
Post-Deployment	Release managers, Service managers Software engineers	Reading of customer reports, Analyzing incidents, Aggregating customer requests, Analyzing product log files	Number of incid. and req., Duration of incid. and req., Product operational data, Product performance data
	Sales managers	Reading articles in the media, Sentimental analysis Customer events participation, Reading industry press, Performing trend analysis	Opinions about the appeal of the product, Performance of the product, Business case descriptions

• *Fragmented Collection and Storage of Data*

Individuals independently collect increasing amounts of customer feedback, analyze the data they obtained, and store their findings on local repositories. Although these findings are occasionally presented at meetings, the lack of transparency and tools prevents others in the organization to use and benefit from the data. With so many different roles collecting and storing data, systematic sharing across development

phases becomes almost impossible. As a result, only those roles that work in close collaboration share data, and benefit from the analysis of this data. This situation is illustrated in the following quotes:

"... it is all in my head more or less." -Product owner, Company B

"Information exists but we don't know where it is." -UX Specialist from Company C

"I do not know everyone... So I contact only the person who is next in line." -Sales manager from Company C.

- **Filtering of Customer Data**

People collect data, and share it only within the development stage they typically work in. For example, practitioners in the development phase actively exchange product log data, interaction design sketches, quality statistics and trouble reports. Similarly, those working in the post-deployment phase exchange release notes, business case descriptions and management system issues. Attempts to communicate the significance of customer feedback and their findings across development stages are typically unsuccessful. Feedback that is shared is filtered quantitative data.

"It is like there is a wall in-between. There is a tradition that we should not talk to each other."
-Product Owner from Company C.

- **Arduous to Measure Means Hard to Share.**

The only data that is successfully shared among people and development phases, is quantitative data representing those things that can be easily measured such as e.g. system performance and operation. The case companies are successful in sharing transaction records, incident figures, feature usage data and other technical feedback that can be easily measured. However, qualitative data such as user stories, feature purpose, or the intention of a certain requirement typically stay with the people that collected that feedback. As a result, and instead of benefitting from an accumulated set of data that evolves over time, companies run the risk of using fragmented data sets that misrepresent the customer and provides an insufficient understanding of what constitutes customer value.

"Maybe 10 % of information is shared. It is very difficult. It takes so much time, to, you need to write a novel more or less and distribute it" -Product manager from Company A.

4.3 Data Sharing Practices: Implications

Due to very limited amounts of data being shared among people and across the development phases, the case companies experience a number of implications. Below, we present the implications:

- ***Non-evolving and Non-accumulating Data.***

Although quantitative data describing operational and performance requirements is typically shared, the lack of qualitative information with the context describing where, how and why a certain feature or a product is needed and how it will be used cause discrepancies in understanding the overall purpose. As a result, the data forming customer knowledge across the development stages does not accumulate and evolve. Consequently, practitioners developing the product do not fully understand the overall purpose of the product or a feature under development and develop suboptimal products that can be different from the customer wishes.

“I think now a lot of thing are developed in a sub optimized way.” -Technology Spec. from company C.

“We get feature which is broken down and then this value somehow got lost when it was broken down, then it is harder to understand what they really need it for.” -Software engineer from Company B.

- ***Repetition of Work.***

Due to the lack of access to the qualitative feedback that is collected in the early stages of development, roles in later stages that seek contextual understanding of a feature are sometimes required to collect identical feedback to the one that was already collected. Consequently, resources are spent on repeating the work that has already been done once.

“You cannot build on what is already there since you don’t know. You then repeat an activity that was already made by someone else.” -UX specialist from Company C.

- ***Inaccurate Models of Customer Value.***

Since the qualitative customer feedback is not shared across the development phases, companies risk to use only the available quantitative customer feedback to build or update the understanding of the customer. This results in inaccurate assumptions on what constitutes customer value. And as a consequence of using the feedback for prioritization on the product management level, projects that create waste risk to get prioritized.

“You think one thing is important but you don’t realize that there is another thing that was even more important.” -Technology Spec. from company C.

- ***Validation of Customer Value is a “Self-Fulfilling Prophecy”.***

Due to the fact that only quantitative customer feedback is exchanged across the development phases and development organization, companies risk to validate their products using only the effortlessly quantifiable feedback, and neglecting the rest. Instead of using the accumulated customer feedback and holistically assess their products, the validation of customer value becomes a “self-fulfilling prophecy” in that it focuses on developing and verifying things that can be quantified and provide tangible evidence.

We map the challenges with their implications for the companies and the products they develop, and summarize them in Table 5.

Table 5. The mapping of identified challenges to their implications.

Challenge	Description	Company implications	Product implications
Fragmented collection and storage of data	Sharing of data is limited across the development stages.	No evolving and accumulating of customer data and understanding.	Suboptimal products are being developed.
Filtering of customer data.	Only roles that work in close cooperation exchange feedback.	Inaccurate assumptions on customer value and repeating work.	Risk of developing wasteful features.
Arduous to measure means hard to share.	What can easily be measured and quantified is shared.	Validation of customer value is a “self-fulfilling prophecy”.	Product maximizes partial models of customer value.

5 Discussion

Multi-disciplinary teams involved in the development of a software product are using customer feedback to develop and improve the product or a feature they are responsible for. Previous research [6, 8, 9] and our empirical findings show that companies collect increasingly large amounts of customer data. Both using the qualitative techniques are used primarily in the early stages of development [6, 7, 10] to construct an understanding of the customer and the context they operate in, and quantitative techniques that are used post-deployment to monitor the actual usage of products in the field [10–12]. And although companies gather and store large amounts of valuable customer data [19], challenges with meaningfully combining and analyzing it in an efficient way [1, 8] are preventing companies from evolving the data across the development stages and accumulating the customer knowledge.

5.1 The Model: From Quantifiable Feedback to Partial Customer Value

In response to the challenges and implications presented above, we illustrate our findings and challenges in a descriptive model on Fig. 1.

In the development process, the model advocates an approach in which an internal model of customer value in companies is being created. We illustrate that companies in fragments collect a complete understating of the customer and their wishes, however, benefit only from a part of the understanding.

In our model, we distinguish between three development stages, i.e. pre-development, development and post-deployment. Although we recognize that this is a simplified view, and that most development processes are of an iterative nature, we use these stages as they typically involve similar roles, techniques, and types of feedback collected.

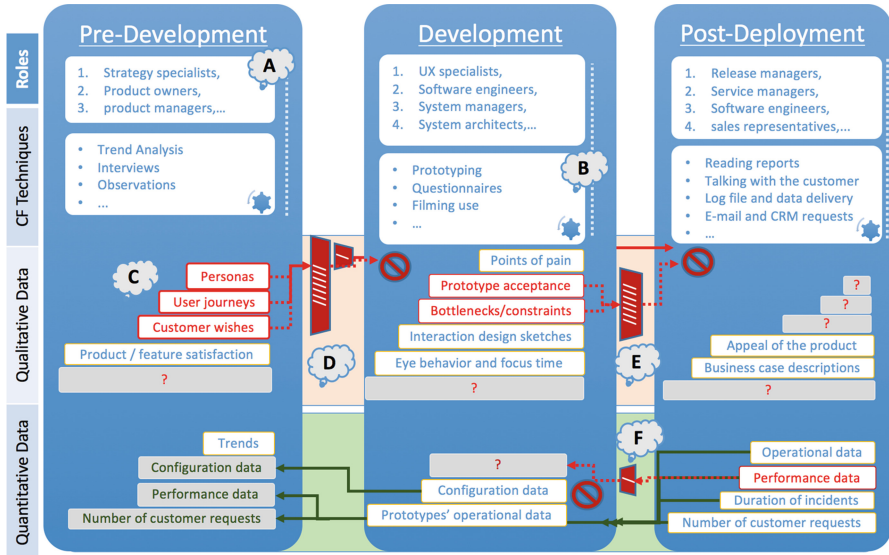


Fig. 1. Customer feedback sharing practices model.

On Fig. 1, we list a few roles that collect customer feedback (A) and different methods of how they perform the collection (B). Within each of the development stages we list the types of feedback being shared across the stages with a solid green lines and those that are not with a dashed red lines. Between development stages we identify three critical hand-over points where customer data that could and should get shared, dissipates. Instead of accumulating data being handed over, gaps across stages appear (illustrated with “?” symbols in blocks on Fig. 1).

5.1.1 The Vaporization of Customer Data.

We identify three critical hand-over blocks that cause data to disappear and prevent practitioners on project to build-on

- (1) *The PreDev Block:* While there is extensive collection of qualitative customer feedback such as user journeys and product satisfaction surveys (Illustrated with C on Fig. 1), roles working in the pre-development stage do not sufficiently supply the development part of the organization with the information they collect. Qualitative data that would inform the development stage on the context of the product under development, how it is going to be used, and who the different user groups perishes in the hand-over process between product owners and managers on one side, and software engineers and UX specialist on the other (Illustrated with D on Fig. 1). Specifically, personas, user journeys and customer wishes are the types of feedback that should be handed over to the development stage, however, they are not. Consequently, the development part of the organization is forced to repeat collection activities in order to obtain this information when in need, or continue developing the product following only the specifications /requirements that were handed to them.

- (2) *The DevOps Block*: UX specialists and software engineers collect feedback on prototypes and their acceptance, as well as where the constraints are. However, this information is only used within the development stage. As a consequence of not handing it over to the post-deployment stage service managers and software engineers (Illustrated with E on Fig. 1), operators of the product do not understand the reason behind a certain configuration when solving a problem, and at the same time, suggest alternative solutions that were already known to be unacceptable to the developers.
- (3) *The OpsDev Block*: In the post-deployment stage, release and service managers collect and exchange operational and performance data, however, do not share it with the development stage to software engineers and system managers. (Illustrated with F on Fig. 1). This prevents the roles in the development stage such as system architects from e.g. deciding on an optimal architecture for a certain type of product and customer size.

5.1.2 Unidirectional Flow of Feedback

Illustrated with red and dashed arrows on Fig. 1, the flow of feedback from the earlier stages of the development to the ones in the later stages is very limited. On the other hand, the flow of feedback from the later stages to the early ones is extensive. This both supports our finding about extensive sharing of quantitative data, which is typically available in the later stages, as well as implies that it is easier to share data about earlier releases of the software under development compared to sharing feedback about the current release. Validating the value of the current release is consequently done very late.

5.1.3 Shadow Representation of Customer Value

In the absence of the accumulated data being accessible and shared across the development stages (illustrated with missing data symbol “?” on Fig. 1), people in later stages base their prioritizations and decisions on shadow beliefs existing in the organization. Consequently, and instead of having a unique understanding of what constitutes customer value, individual development stages and roles prioritize based on their best intuition and shared quantitative data. If sharing of customer data in the direction towards the later stages is enabled, roles across the development stages will be able to conduct data-informative decisions. As seen in our findings, hazards of being purely quantitative data-driven are extensive. And with qualitative data being as accessible as quantitative, validation of customer data could be coherent, not a ‘self-fulfilling prophecy’ as it is today.

6 Conclusion

By moving away from traditional waterfall development practices and with agile teams becoming increasingly multi-disciplinary and including all functions from R&D to product management and sales [21], the role of customer feedback is increasingly gaining momentum. And although the collection of data has previously been identified as challenging, we show in our research that the challenge is not its collection, but rather how to share this data in order to make effective use of it.

In this paper, we explore the data collection practices in three large software-intensive organizations, and we identify that lack of sharing of data is the main inhibitor for effective product development and improvement. Based on our case study findings, we see that currently (1) companies benefit from a very limited part of the data they collect due to a lack of sharing of data across development phases and organizational units, (2) companies form inaccurate assumptions on what constitutes customer value and waste resources on repeating the activities that have already been performed, and (3) validation of customer in companies today is a “self-fulfilling prophecy” in that it focuses on quantifiable things that provide tangible evidence.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work’s Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work’s Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. Chen, H., Chiang, R.H., Storey, V.C.: Business intelligence and analytics: From big data to big impact. *MIS Q.* **36**, 1165–1188 (2012)
2. Dzamashvili, Fogelström, N., Gorschek, T., Svahnberg, M., et al.: The impact of agile principles on market-driven software product development. *J. Softw. Maintenance Evol. Res. Pract.* **22**, 53–80 (2010)
3. Olsson, H.H., Bosch, J.: From opinions to data-driven software R&D: a multi-case study on how to close the ‘open loop’ problem. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 9–16 (2014)
4. Olsson, H.H., Bosch, J.: Towards continuous customer validation: a conceptual model for combining qualitative customer feedback with quantitative customer observation. In: Fernandes, J.M., Machado, R.J., Wnuk, K. (eds.) *Software Business. LNBIP*, vol. 210, pp. 154–166. Springer, Heidelberg (2015)
5. Von Hippel, E.: Lead users: a source of novel product concepts. *Manage. Sci.* **32**, 791–805 (1986)
6. Fabijan, A., Olsson, H., Bosch, J.: Customer feedback and data collection techniques in software R&D: a literature review. In: Fernandes, J.M., Machado, R.J., Wnuk, K. (eds.) *Software Business. LNBIP*, vol. 210, pp. 139–153. Springer, Heidelberg (2015)
7. Cockburn, A., Williams, L.: Agile software development: It’s about feedback and change. *Computer* **36**, 39–43 (2003)
8. Bizer, C., Boncz, P., Brodie, M.L., Erling, O.: The meaningful use of big data: Four perspectives—four challenges. *ACM SIGMOD Rec.* **40**, 56–60 (2012)

9. Fabijan, A., Olsson, H.H., Bosch, J.: Early value argumentation and prediction: an iterative approach to quantifying feature value. In: Abrahamsson, P., et al. (eds.) PROFES 2015. LNCS, vol. 9459, pp. 16–23. Springer, Heidelberg (2015)
10. Bosch-Sijtsem, P., Bosch, J.: User involvement throughout the innovation process in high-tech industries. *J. Prod. Innov. Manage.* **32**(5), 793–807 (2014)
11. Olsson, H.H., Bosch, J.: Post-deployment data collection in software-intensive embedded products. In: Bosch, J. (ed.) *Continuous Software Engineering*, pp. 143–154. Springer, Heidelberg (2014)
12. Olsson, H.H., Bosch, J.: Towards data-driven product development: A multiple case study on post-deployment data usage in software-intensive embedded systems. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J. (eds.) *Lean Enterprise Software and Systems*. LNBIP, vol. 167, pp. 152–164. Springer, Heidelberg (2013)
13. Sommerville, I., Kotonya, G.: *Requirements Engineering: Processes and Techniques*. Wiley, Chichester (1998)
14. Sampson, S.E.: Ramifications of monitoring service quality through passively solicited customer feedback. *Decis. Sci.* **27**(4), 601–622 (1996)
15. Bosch, J.: Building products as innovations experiment systems. In: *Proceedings of 3rd International Conference on Software Business*, Massachusetts, 18–20 June 2012
16. Kohavi, R., Longbotham, R., Sommerfield, D., Henne, R.M.: Controlled experiments on the web: survey and practice guide. *Data Min. Knowl. Disc.* **18**(1), 140–181 (2009)
17. Aoyama, M.: Persona-and-scenario based requirements engineering for software embedded in digital consumer products, pp. 85–94 (2005)
18. Saarijärvi, H., Karjaluo, H., Kuusela, H.: Customer relationship management: The evolving role of customer data. *Mark. Intell. Plan.* **31**, 584–600 (2013)
19. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. *Comput. Netw.* **54**, 2787–2805 (2010)
20. Rodríguez, P., Haghighatkah, A., Lwakatare, L.E., Teppola, S., Suomalainen, T., Eskeli, J., Karvonen, T., Kuvaja, P., Verner, J.M., Oivo, M.: Continuous deployment of software intensive products and services: A systematic mapping study. *J. Syst. Softw.* (2015). <http://www.sciencedirect.com/science/article/pii/S0164121215002812>
21. Olsson, H.H., Alahyari, H., Bosch, J.: Climbing the “Stairway to Heaven”, in software engineering and advanced applications (SEAA). In: *2012 38th EUROMICRO Conference on Software Engineering and Advanced Applications*, Izmir, Turkey (2012)
22. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **14**(2), 131–164 (2008)