Early Value Argumentation and Prediction: An Iterative Approach to Quantifying Feature Value

Aleksander Fabijan[1(✉)], Helena Holmström Olsson[1], and Jan Bosch[2]

[1] Faculty of Technology and Society,
Malmö University, Östra Varvsgatan 11, 205 06 Malmö, Sweden
{Aleksander.Fabijan,Helena.Holmstrom.Olsson}@mah.se
[2] Department of Computer Science and Engineering,
Chalmers University of Technology,
Hörselgången 11, 412 96 Göteborg, Sweden
Jan.Bosch@chalmers.se

Abstract. Companies are continuously improving their practices and ways of working in order to fulfill always-changing market requirements. As an example of building a better understanding of their customers, organizations are collecting user feedback and trying to direct their R&D efforts by e.g. continuing to develop features that deliver value to the customer. We (1) develop an actionable technique that practitioners in organizations can use to validate feature value early in the development cycle, (2) validate if and when the expected value reflects on the customers, (3) know when to stop developing it, and (4) identity unexpected business value early during development and redirect R&D effort to capture this value. The technique has been validated in three experiments in two cases companies. Our findings show that predicting value for features under development helps product management in large organizations to correctly re-prioritize R&D investments.

Keywords: Continuous experimentation · EVAP · QCD · Data-driven development · Customer-driven development

1 Introduction

By introducing agile development practices [1], companies have taken the first step in being able to better cope with continuously changing market requirements [2, 3]. At the same time, organizations are striving to find systematic instructions that would guide them in re-prioritizing feature developments and assuring them to stop developing features when they don't deliver new value. Instead, what they wish for is a way to determine and validate feature value before it is fully developed in order to discover where to invest R&D efforts, or when to stop developing it [3, 17].

Known as customer experimentation, this is a common practice in the Business-to-Consumer (B2C) domain, while it is still to gain momentum in the Business-to-Business (B2B) domain. Data collection techniques in the B2B domain are typically designed for operational purposes and do not explicitly reveal the necessary

information for feature experimentation such as feature usage, role of the user etc. Today, and due to the fact that products are increasingly becoming connected, customer experimentation is gaining momentum also in B2B organizations [3, 11].

In this paper, and based on case study research in two B2B software-intensive companies, we investigate how feature value prediction and validation can be performed in a continuous experimentation setting in order to prevent companies from developing features that do not deliver value. We use as a foundation the QCD model [4] of customer-driven development and look into one aspect of it, e.g. how to validate hypotheses about feature value using customer data.

The contribution of this paper is the development and validation of the EVAP technique. With this technique we provide organizations with: (1) an actionable implementation of one of the aspects of the QCD model that practitioners can use to dynamically develop valuable product functionality, (2) a technique that helps teams develop just enough of a feature and stop when too little value is being created, (3) support to stop development of a when expected value is not realized, and (4) help to identify unexpected business value early during development and redirect R&D effort to capture this value.

2   Background

Von Hippel [5], coined the term 'Lead Users' in order to show that customers with strong needs often appear to be ahead of the market and that companies should involve them in order to validate concepts or emerge with new ideas. Learning and validating with users of the products is becoming increasingly important and organizations need to adapt their processes in order to take this source of feature validation into account [7, 8]. First seen in the B2C domain and recently also in B2B domain [6], products are becoming connected and by having data collection techniques in place, numerous new options are available to use this information.

2.1   The QCD Model: Qualitative and Quantitative Customer Validation

Recently, and as a model to help companies move away from early specification of requirements and towards continuous validation of hypotheses, the QCD model was introduced [4]. The model identifies a number of customer feedback techniques (CFT's) that can be used to validate feature value with customers. As seen in the model (Fig. 1), hypotheses are derived from business strategies, innovation initiatives, qualitative and quantitative customer feedback and results from on-going customer validation cycles.

The novel development approach as pictured in the QCD model captures, and expand further, on ideas that have been previously published [3, 12, 13]. In the QCD model quantitative, as well as qualitative feedback techniques are recognized.

Typically, and to initiate and experiment, a hypothesis is selected and validated on a set of customers. To support this process, numerous customer feedback collection techniques have been identified in the literature [9, 10]. In addition to the validation and
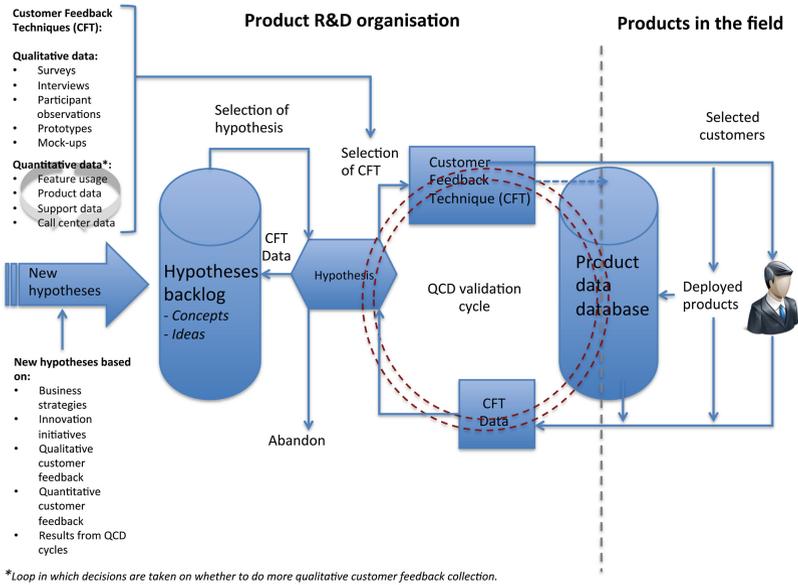
Fig. 1. The Qualitative/Quantitative Customer-driven Development (QCD) model [4].

confirmation of existing hypotheses, the data collected allows new hypothesis to be formed, making the QCD model an instrument for innovation of new features, for value discovery and for prediction of feature value.

Below, and in order to further illustrate the use of the QCD model, we specify one of its techniques, i.e. the 'Early Value Argumentation and Prediction' (EVAP).

3 Early Value Argumentation and Prediction Technique

Typically, and as recognized in previous research [17], product management (PdM) has to wait until the feature is fully developed in order to validate its expected value. To overcome this problem, and to be able to predict the value of a feature, we coin the term 'Early Value Argumentation and Prediction' (EVAP), to annotate a technique that practitioners can use in order to estimate what impact a feature will have when fully developed. We define value argumentation to be the ability to explicitly specify what the value of the feature will be when fully developed. And EVAP technique in particular works as a support for helping companies move away from early specification of requirements and towards dynamic feature prioritization (Fig. 2).

As a result of using the technique, companies can predict the value of a feature being developed, and decisions on whether to continue development or not can be taken based on real-time customer data. We illustrate this in Fig. 3, where we show the typical process of feature value validation and prediction.

In a continuous experimentation setting, a feature is first selected and its expected value is defined as a hypothesis. The product functionality is then iteratively developed
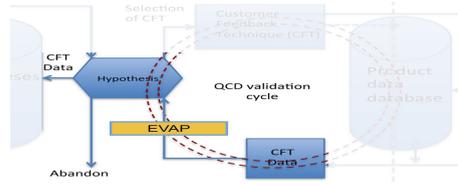
Fig. 2. 'EVAP' technique in the QCD model.

and validated on the customers. It may take several iterations of developing a minimal viable feature (MVF), referred to i*t(im) on Fig. 3, where 'i' denotes the number of iterations, before the expected value is confirmed. We annotate the time between identification of a feature expected value t(ex) and until its confirmation t(va), to be the 'value gap time' t(gap). This is the period where companies decide, using the customer data collected, whether to continue developing the functionality or stop and redirect R&D efforts.
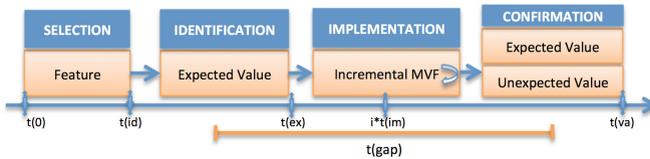
Fig. 3. 'Early Value Argumentation and Prediction' (EVAP) technique.

4   Research Method

We validated our model on a multiple-case study [14] that took place between December 2014 and June 2015. It builds on an ongoing work with two companies involved in large-scale development of software products. Case study methodology is an empirical method aimed at investigating contemporary phenomena in their context, and it is well suited for this kind of software engineering research [15]. Based on experience from previous projects on how to advance beyond agile practices [3, 9], we held two joint workshops with the companies involved in this research, and two individual workshops in-between. Since both participating companies collect quantitative as well as qualitative data, QCD model was the appropriate instrument. What companies struggled with, is an actionable implementation of one the aspects of the QCD model. We therefore completed the workshops with a concept that is now known as the EVAP technique.

4.1   Case Companies

Company A is a provider of telecommunication systems for mobile and fixed network operators. Together with company A, we performed one feature experiment focusing on a feature that reestablishes connections. Among the representatives we met were a product owner, a line and system manager and a developer.

Company B is a software company specializing in navigational information, operations management and optimization solutions. They provide nautical charts and airline avionics for several airlines around the world. In company B, we performed two feature experiments. The first one was an on-going experiment with a feature for adjusting schedule that started before December 2014, whereas the second feature experiment with the long term KPI visualization initiated in the beginning of this research. We met with a system architect, portfolio manager and UX Designer.

4.2   Data Collection and Analysis

The main data collection method in both of the joint workshops was semi-structured group interviews with open-ended questions [2]. Additionally, to the workshops, we had a number of Skype meetings, individual company visits and communicated via e-mail or Lync. In total, we met twice on joint workshops, twice at each of the companies individually, and communicated via emails at least once per month. The notes and workshop meeting minutes were analyzed following the qualitative content analysis approach [16].

5   Validation of the EVAP Technique

In company A, we validated EVAP in one feature experiment. Iteratively, and following EVAP technique, they tried different setting of the feature under development, until a significant difference in the measured counter (the number of detach requests on link failure) was visible. After the count of detaches stabilized, they were confident that the expected value of the feature is confirmed. Moreover, while validating feature value, company A monitored how the CPU load changes in time. Interestingly, they discovered that when this feature is active, the CPU load decreases.

In company B, we validated EVAP in two experiments. First, company B observed the customers using their scheduling feature. Company B wished that operators of the product use this functionality in order to improve the scenario proposed to better aid crew planner in doing the right tradeoffs between KPIs, and to compare different solutions. They discovered that the use of the feature in the first iteration was low, so they came-up with adjustments that they deployed in the next iteration. After another observation of the data collected, combined with the roles of users using this feature, they had enough evidence to argument that this feature delivers the hypothesized value. That is, the individual KPI improvements when manually adjusting a schedule.

In the third experiment, company B decided to validate if by visualizing performance indicators they could learn more about the customers and motivate them to take the improvements. Also, they did not know how well their product versions perform in reaching customers' KPIs. To overcome this, company B started to develop this functionality and show KPI trending prototypes and concepts to a test customer. First, company B visualized KPIs that they considered important. After analyzing the data, they did not see any significant difference in behavior of the customer. In the next iteration, Company B improved the feature by closely cooperating with the test

Table 1. Summary of the experiments.

	Company/Experiment	CFT and collected data	Expected and validated value	Predicted value
A	Network feature	Product logs that contain number of detaches, active users and sys. load	Reduced the number of detach requests	Reduces CPU load during a network hiccup
B	Scheduling feature	Service center logs with counts of feature usage and optimizations starts	Improving an individual KPI	Product-learning and improvements
B	Long term KPI visualization	Service center logs with counts of dashboard openings/interviews	/	Better alignment of the customers KPIs

customer and changed the visualizations so they reflect what companies consider as an important KPI. Now, customers are more interested in adopting this functionality and cooperating with Company B (Table 1).

6   Discussion and Conclusion

Organizations are struggling to know if and how much of the feature that they are developing, will actually deliver value to their customers. To overcome this problem, and to be able to predict and validate feature value, companies have to continuously be able to collect customer feedback and learn from the users [6, 10]. The QCD model, due to its nature of being very dynamic when forming hypotheses and able to handle both quantitative as well as qualitative data [4], is the current 'state of the art' instrument for continuous experimentation.

In this paper, we present EVAP technique as an actionable implementation of one aspects of the QCD model. Companies can use this technique in order to:

(1) *Dynamically develop valuable product functionality*
   We illustrate this in the three feature experiments with the two case companies, showing how they both developed valuable functionalities. For example, company A successfully developed the network feature by proving that the number of detach requests iteratively reduced by validation on a test system. Similarly, company B developed the scheduling feature, by iteratively running feature experiments with their customers and validating the value. At the same time, company B started to develop the KPI visualization feature, which is showing positive validation results, giving company B a reason to continue developing this functionality.

(2) *Develop just enough of a feature and stop when too little value is being created*
     While iterating with the network feature, company A validated the functionality
     on the test system that mimics real-life environment and found out that by
     comparing the results of two consecutive iterations, the number of detach requests
     converged and developing further would not bring more value to the product.
     They stopped developing the parts of the feature that would further reduce detach
     requests.
(3) *Stop development of a feature and remove it from the system when expected value
     is not realized*
     Although this has not been the case during this research, companies could stop
     developing a feature and remove the already implemented parts from the system,
     if they had identified that measurable value of the feature is lower as expected.
(4) *Identify unexpected business value early during development and redirect R&D
     effort immediately to capture this value.*
     We can see from the experiments that both companies predicted the value of the
     feature even before they fully developed it. In Company A, they correctly pre-
     dicted the CPU load decrease and increasingly invested in further developing this
     functionality in order to capture this value. Similarly, company B predicted the
     value of their second experiment to be positive. They are increasing their efforts
     and further developing the functionality.

To conclude, we have demonstrated how continuous feature experimentation can
be conducted in two large B2B companies, in order to use the benefits of being
customer-driven to validate and predict feature value. We do this by exploring the QCD
validation cycle and developing an iterative technique that practitioners in organiza-
tions can use to dynamically develop valuable product functionality.

In future research we plan to further detail the different techniques that are used as
part of the QCD model, and we intend to validate these in our case companies.

References

1. Cockburn, A., Williams, L.: Agile software development: it's about feedback and change.
    Computer **36**, 0039–43 (2003)
2. Dzamashvili Fogelström, N., Gorschek, T., Svahnberg, M., et al.: The impact of agile
    principles on market-driven software product development. J. Softw. Maint. 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.) ICSOB 2015. 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. Bosch, J., Eklund, U.: Eternal embedded software: towards innovation experiment systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012, Part I. LNCS, vol. 7609, pp. 19–31. Springer, Heidelberg (2012)
7. McKinley, D.: (2012). http://mcfunley.com/design-for-Continuous-Experimentation
8. Davenport, T. H.: How to design smart business experiments. In: Strategic Direction, Emerald Group Publishing Limited, Bradford (2009)
9. 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.) ICSOB 2015. LNBIP, vol. 210, pp. 139–153. Springer, Heidelberg (2015)
10. Bosch, J.: Building products as innovations experiment systems. In: Proceedings of 3rd International Conference on Software Business, Cambridge, 18–20 June 2012
11. Lindgren, E., Münch, J.: Software development as an experiment system: a qualitative survey on the state of the practice. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 117–128. Springer, Heidelberg (2015)
12. Fagerholm, F., Guinea, A.S., Mäenpää, H., et al.: Building blocks for continuous experimentation, pp. 26–35 (2014)
13. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Random House LLC, New York (2011)
14. Walsham, G.: Interpretive case studies in IS research: nature and method. Eur. J. Inf. syst. **4**, 74–81 (1995)
15. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Eng. **14**, 131–164 (2009)
16. Mayring, P.: Qualitative content analysis–research instrument or mode of interpretation. Role Researcher Qual. Psychol. **2**, 139–148 (2002)
17. Johansson, E., Bergdahl, D., Bosch, J., Olsson, H.H.: Quantitative requirements prioritization from a pre-development perspective. In: Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2015. CCIS, vol. 526, pp. 58–71. Springer, Heidelberg (2015)

Springer