Adding Value Every Sprint: A Case Study on Large-Scale Continuous Requirements Engineering*

Rashidah Kasauli[1], Eric Knauss[1], Agneta Nilsson[1], Sara Klug[2]

[1] Dept. of Computer Science and Engineering
Chalmers | University of Gothenburg, Sweden
rashida@chalmers.se, {eric.knauss,agneta.nilsson}@cse.gu.se,
[2] Ericsson AB
sara.klug@ericsson.com

Abstract. Agile development practices, such as continuous integration and continuous delivery, promise value through shorter time to market and increased flexibility. While these practices have been widely adopted in small-scale, they have shown to be challenging to adopt in large-scale, system development. This is often due to a distance between customer and developer in large scale systems, and the need to break down value from the whole system into manageable parts. The notion of value is fundamental for agile methods, especially for practices such as continuous delivery to the customer. However, how value should be handled in development practices is not clearly understood. In this paper, we investigate how the notion of adding value in every sprint has been perceived in a large-scale system development. Based on an exploratory qualitative case study, the outcome shows that it is perceived beneficial by practitioners although it comes at a price and challenges exist.

Keywords: value, continuous requirements engineering, continuous integration, continuous delivery, large-scale agile

1 Introduction

Agile software development focuses on customer collaboration and the ability to deliver customer value quickly and incrementally [8]. For this, popular agile methods such as Scrum [26] and eXtreme Programming (XP) [4] have powerful planning mechanisms in place. These methods align well with continuous integration (CI) and continuous delivery (CD) and can lead to substantially higher productivity [27] and shorter time to market. While most research on agile practices (such as continuous integration) focuses on the team scope and software only (e.g. [12]), they are increasingly applied to more and more complex development endeavors, such as large software-intensive systems [17, 23].

* In: Proc. of 3^{rd} WS on Continuous Requirements Engineering, 2017, Essen, Germany. Copyright 2017 for this paper by its authors. Copying permitted for private and academic purposes.

To our knowledge, there is a lack of empirical studies investigating large-scale systems development, its challenges with respect to requirements engineering, and suitable advice. Particulary, the notion of value and how it is used in sprints has not been investigated in literature. Yet, it is an important aspect of continuous requirements engineering, especially for organizations transitioning towards continuous delivery and deployment. In previous work, we found that particularly the distance between developers and customers can introduce challenges to the notion of value in agile developments [11]. In this paper, we present results from an explorative, qualitative case study in a large-scale agile development organization to address this research gap and to investigate the following research questions:

RQ1 What is the interpretation of value from the perspective of different roles in large-scale agile software development?

RQ2 What are the effects of the notion of adding value every sprint of individual teams? What benefits and challenges exist?

RQ3 How do you check if value has been added in each sprint?

RQ4 What improvements could support adding value every sprint in large-scale continuous software engineering?

The study shows that all interviewees see value in "adding value every sprint" as well as widely share the notion of value and how it relates to their daily work. They however reveal a diverse picture on how to check and control if value is added in the sprint. Despite being positive towards adding value every sprint, all interviewees see challenges and present constructive suggestions for improvement.

2 Background

Agile methods have drastically changed software development, relying on people skills and close customer collaboration to meet ever changing market needs and requirements rather than formalized processes and contracts as in traditional methods [5]. Agile development is characterized by short development cycles [5], face-to-face collaborations [11], continuous integration of code changes into the product baseline [12, 20], and continuous delivery of working software to meet customer demands [10]. While there is relatively rich literature on how to implement and setup agile practices (such as continuous integration) at small-scale for a project (e.g. [12, 20]), there is very limited scientific support for how to transfer this to large-scale environments (e.g. [21]).There is however research that reports on challenges with scaling of continuous integration [9, 24, 25].

With the wide adoption of agile methods, requirements engineering is no longer confined to the initial phases of software development [13]. Instead it has become a continuous process in the software development life-cycle [15]. To successfully evolve products that bring value to customers, continuous requirements engineering needs to take into account the different concerns of all the stakeholders involved in the process or project [15].

Software today has a major influence on most systems' cost, schedule, and value [7], therefore a lack of focus on the value can seriously degrade the project outcome. Agile methods promote the notion of customer value, e.g. by valuing working software over comprehensive documentation [5] and there is a rich related research that discusses the creation of customer value as key element for a company's success [3, 16, 21].

Related to this concept of customer value, value based requirements engineering (VBRE) has been proposed as an approach [16]. The VBRE approach uses selection of requirements to enhance the value of a release [3, 28]. VBRE promotes software developers to align customers' requirements, business requirements, and technological opportunities, to have a sound understanding of both technical and business implications of decisions made, and to understand the business dynamics that drive software development [2].

The notion of delivering value at the end of every sprint is the aim of agile methods [6]. However, different interpretations of the concept of customer value exist [14], frequently relating customer value to the trade-off between what the customer receives and what they invest to acquire and use a product [29]. This definition is based on the customer's perspective, and to our knowledge there exists little research on how software development teams can relate to customer value, especially in large-scale systems development.

While customer value and its role for prioritization in agile projects has been discussed critically [22], Alahyari et al. have started to investigate this gap, i.e. how value is interpreted, prioritized, assured, and measured in agile software development [1]. Based on a qualitative study with 23 participants from 14 organisations, they identified and prioritized value aspects such as *delivery process with respect to time*, *quality*, and *knowledge of feature value for customer*. In this paper, we specifically investigate the latter: What value for a customer is added during a sprint and how do developers relate to it. In accordance with Alahyari et al., we also investigate the question on how to measure or evaluate the value added in each sprint. Based on our smaller scope and stronger focus on one value aspect, we can shed more light on this aspect, e.g. how acceptance testing and sprint demos can help measuring which value has been added.

3 Research Method

In this paper, we investigate the notion of value and its use in large-scale agile system development. We employ collaborative practice research [18], which is a way to organize and conduct research in close collaboration between researchers and practitioners, drawing on the combined strengths of the practitioners' way of thinking and the reflective researchers. We designed the exploratory, qualitative case study together with a practitioner at the company and identified key areas and relevant roles for semi-structured interviews. In total we interviewed five persons, which we selected based on a convenience sampling strategy in order to cover relevant roles: two system testers, one product owner, one designer, and one function tester.

In the interview guide we covered the areas: notion of value, effects of adding value every sprint, how to check if value has been added, and suggestions of improvements. Each interview was conducted face-to-face by two researchers and lasted 45-60 minutes. Interviews were recorded and notes were taken.

Case Company: The study is conducted at one unit of a Swedish-based multinational organization offering services, software, and infrastructure in information and communication technology for telecom operators and other industries. All interviewees were involved with the development of one specific product. The unit has worked with continuous integration since 3-4 years, and employs a scaled agile approach of a feature development model, with prioritized features and 30-40 cross-functional teams working in parallel with features.

In the data analysis we focused on synthesizing the views from the different roles regarding 1) notion of value to understand and distinguish the meaning of this concept, 2) effects mentioned in terms of benefits or challenges with these practices in their daily work, 3) how they check if value has been added in the sprint, and 4) suggestions of improvements.

Discussion of Validity: As an exploratory study, the aim of this paper is to better understand the notion of value and its use in large-scale agile system development. Based on the limited size of this study, we cannot generalize our results beyond the scope of our study. Instead our aim is to identify relevant aspects for future research on a larger scale and with more companies involved. We validated our findings during a workshop, where we discussed the synthesis of results from transcripts and notes.

4 Findings

4.1 Interpretations of Value (RQ1)

We asked our interviewees what they viewed as *customer value* and *product value* to investigate if there was a perceived difference between these notions of value. While we got rather clear answers on customer value, the interpretation of product value and its relation to customer was more diverse, leading us to include *market value* as a third concept of value.

Customer Value: From the perspective of our interviewees, customer value relates to a change in the product.

> *"If we add something that the customer wants, but it shall be a change in the product."* — System Tester

More specifically, this change should relate to something a customer can sell or that makes their product or service cheaper. Customer value also relates to the relationship to the customer and become visible in development sprints as *promised features, functionality, quality, configuration, or documentation.*

> *"[. . .] also building a relationship and getting them involved. When we start we give them a demo. Then we break down things into small user stories. Then we discuss the release plan, priorities and the user stories.*

So they can influence and participate in the discussion." — Product Owner

One interviewee thought that customer value is also about providing them with the ability to influence and participate in discussions around new features.

"Different customers, they value different things." — Function Tester

Our interviewees widely agreed that customer value can differ for each customer.

Product Value: In contrast to customer value, most of our interviewees referred to product value as *"something the customer does not see"*.

"If you have certification or peer-review. Such a certification could be product value. Not sellable by itself. But now that I think about it, redesign and refactoring could be counted as product value. That, we have a lot." — Designer

Above activities add value to the product that does not directly relate to a customer need. However, as many interviewees indicated, an increase of product value will indirectly lead to customer value.

"Product value can improve development environment and indirectly improve customer value. " — System Tester

Market Value: Since customer value differs between customers, there is a risk that following each customer separately will lead to an unnecessarily complicated product.

"We have a lot of discussions on having customer specific solutions. For the product, it is not always adding value, but instead introduces complexity. So we spend a lot of time to abstract and prioritize so we do not blindly do what one customer says." — Function Tester

To mitigate this risk one needs to abstract from individual customer wishes to a combined *market value* that adds value to more than one customer.

4.2 Effects of Adding Value Every Sprint (RQ2)

We were also interested in the effects of adding value every sprint has for our interviewees. Specifically, we asked about benefits and challenges.

Benefits: All our interviewees saw benefits of focusing on value.

"If you think about sprint goals, and tie to continuous integration - yes it is beneficial. It helps with these small changes. You are not in your head thinking about things that you will do in the next year, but only about the next three weeks." — Designer

Table 1. Internal and external benefits of adding value every sprint

Type	*Benefit*	*Characteristic quote*
Internal	– Improve quality of tests and feedback	*"Today, we can always set up an integration test. We are not blocked. We can go back in the version history of one of the features."* – System Tester
	– Allows to focus on what is important now	
External	– Reduce risk to build the wrong system	*"When trying to add customer value in such small increments, you gain a lot of flexibility and can steer away from bad directions."* – *Designer*
	– Add flexibility to development	
	– Bridge distance to customer	*"Distance to customer is largely improved by feature development. Of course the customer is not sitting next to us, but they visit sometimes. And it is always clear what we are working for that is customer visible."* – System Tester
	– Get a feature out early and start learning from customer	

Generally, the benefits we saw can be divided into internal and external benefits, as shown in Table 1.

Challenges: In addition to the clear benefits, our interviewees also mentioned some challenges.

High costs of benefits: Our interviewees recognize that firstly, a high investment was necessary to get the organization to become efficient with adding value in every sprint. Secondly, it does not necessarily feel like a speedup on team level.

> *"As a team we spend a lot of time with trouble reports. In the past we squeezed the trouble reports in after the feature development. Then there was a lot to fix. Now we do the trouble reports continuously. So we are slower with the features. But the overall quality should improve, which is more important."* — Function Tester

Risk of technical debt: As shown in Table 1, a feature can be pushed out to a customer early to facilitate learning. However, the learning from the customer often comes at the expense of technical debt:

> *"This is software intended for one or two customers. It is usually not commercially supported, so you can cut corners on robustness. Send it to the customer, let them use it, get feedback on it, and make sure it is what the customer wants. So you reduce the risk of not delivering value after long development. [...] We can half the time and deliver something they can use [...]. Of course, we have some debt, but we can solve that in the coming month."* — Product Owner

Trade-off between agile and long term perspective: Our interviewees recognize the challenge to balance agility with the necessity of long term planning.

> *"How do you stay agile and maintain a 5-10 year strategy? How can you see that you go in the right direction with hundreds or thousands of user stories?"* — Designer

This is however not only due to the large scale, but also due to the difference between customer and market value mentioned earlier:

"You have to be careful about the differences between customer value and product value [we refer to this as market value]. Because you have more dialog there is also a risk that you just agree and miss out on the big picture of where the product is going." — Function Tester

Maintain high quality on main branch: It is also recognized that the quality of the main branch becomes paramount. Keeping feedback cycles short and identify which delivery to the main branch is causing problems quickly becomes critical.

"It is a new way of working. You need a culture where you are following your deliverable. The goal is to have high quality on [main branch] and most often we have. If you deliver, you need to check the portal if you broke something. But that is sometimes hard to see, e.g. when many are delivering at the same time." — System Tester

4.3 How to Check if Value is Added in Each Sprint (RQ3):

Our interviewees indicated that there is very little formal measurement:

"[Added value] is nothing that we measure. It is just captured in natural language." — Designer

"What I am using daily is 'find a good enough level' - if it costs too much to measure, than you need to scrap it." — Product Owner

However, they talked about how their agile ways of working capture the checking and control of value to a significant extent, through reviews, demonstrations, combining user stories with criteria of definition of done, and rigorous testing.

Reviews: Teams are conducting a large number of reviews, both during sprint planning and sprint review. In addition, there are special meetings across cross-functional teams that cover all features.

Sprint Demo: Sprint demos are recognized as very efficient way of demonstrating the value that has been added to the system.

"We demo the product at the end of the Sprint (not so much to the customer, but to the PO and Scrum Master). There we prove that the promised value has been added." — System Tester

Such demonstrations are conducted both for internal and external stakeholders:

"We can demo internally, and we can request external customers to come here and we can demonstrate. And there are also market and customer units that we can demo to." — Product Owner

Testing and Definition of Done: Our interviewees rely a lot on definitions of done for user stories, hence, testing becomes a powerful measurement tool for which value has been added.

"[. . .] you try to define the value when you define the user story. It might be that we are not there yet to measure the value, but we have user stories definition of done. When we define the user stories we also define acceptance criteria. So it is a binary decision on level of user story: pass or fail" — Function Tester

Table 2. Suggested Improvements with respect to Challenges

Sugg. Improvement	*Aspects to improve*	*Challenge to be addressed*
Be more pro-active	Know value for customer before they do so that we can demonstrate better.	Trade-off between agile and long term.
Component guardians	Introduce/strengthen role with responsibility to assess how a change affects the long term value of a component.	Trade-off between agile and long term; Risk of technical debt.
Increase focus on process quality	Include requirements as part of delivery to better characterize capabilities of deliveries,	Maintain high quality on main branch.
Include team earlier	Allow team to develop an understanding of customer value.	High Cost, investment for general notion of value
More detailed check	Based on partial deliveries, improved breakdown of user stories.	High Cost, investment for general notion of value.
Improve tests as measurement	Improve partial delivery of value based on explicit notion about how tests are used to measure added value.	High Cost, investment for general notion of value.
Improve demos as check	Further spread sprint demos in organization, invite more stakeholders, demonstrate main flow long before feature is implemented.	Trade-off between agile and long term.
Transparency	Better transparency about (testing) resources and dependencies to other teams.	High Cost, ability to speed up.

4.4 Suggested Improvements (RQ4)

During our interviews, we also asked the interviewees about which improvements they would suggest with respect to the scope of our investigation. Table 2 gives an overview of the improvements we collected and how they relate to challenges.

5 Discussion and Conclusion

Implications for Future Research: Value reflects the owner's or buyer's desire to retain or obtain a product [19]. Neap [19] defines *product value* as a measure expressed in units of currency equal to the cost of the product as a subjective value. Product value is influenced by the quality attributes of the software product and is related to the product price [3]. Woodruff defines the concept of *customer value*: "a customer's perceived preference for, and evaluation of, those product attributes, attribute performances, and consequences arising from use that facilitates (or blocks) achieving the customer's goals and purposes in use situations" [29]. The term customer value has many meanings but two dominate - value for the customer (customer perceived value) and value for the firm (customer lifetime value) [29]. While Neap's concept of product value differs from the interpretation of our interviewees', their interpretation of customer value relates directly to Woodruff's customer perceived value, while market and product value in our study are two aspects of customer lifetime value. In contrast to the definitions in literature, our interviewees' did not discuss the concept of cost. Evobota et al., argue that agile planning is particularly difficult to scale, because it is hard to bring together the perspectives of planning and cost [11], and we believe that this is also due to the difficulties to manage value.

Future work should identify suitable definitions of value as well as investigate how cost and value can be related to each other in a more transparent and

beneficial manner. We believe that this is crucial for understanding what value that is added in each sprint as well as defining more fine-grained measurements. In addition, we see a need for more conceptual work on requirements and testing in order to measure the value added in each sprint.

Implications for Large-Scale Agile Development Practice: Our results are particularly interesting for companies that want to embrace continuous delivery of large-scale systems, i.e. that aim at delivering new functionality to customers continuously. Based on our results we suggest that *lack of shared understanding of customer value is an impediment for continuous delivery and deployment*, highlighting the need for continuous requirements engineering practices. Specifically, we suggest that *distance to customer*, *lack of focus on sprint goal*, and *lack of quality on test infrastructure* will lead to inefficient continuous delivery. The company studied has addressed these critical areas to a large extent and our interviews suggest that this investment was crucial. Finally, we recommend the continuous requirements engineering practices of *adding value every sprint*, *establishing a definition of done for each user story*, and *linking user stories to requirements and tests* as these were deemed beneficial for continuous delivery in our interviews.

Acknowledgments: We are grateful for the support and insightful discussions with our industry partners and we thank all interviewees. This work has been partly supported by the Software Center, Proj. 1 "Implications of Continuous Deployment" and the SIDA BRIGHT project.

References

1. Alahyari, H., Svensson, R.B., Gorschek, T.: A study of value in agile software development organizations. Journal of Systems and Software (2016)
2. Aurum, A., Wohlin, C.: A value-based approach in requirements engineering: explaining some of the fundamental concepts. In: Int. Working Conf. on Reqts. Eng.: Foundation for Softw. Qual. (REFSQ). pp. 109–115. Springer (2007)
3. Barney, S., Aurum, A., Wohlin, C.: A product management challenge: Creating software product value through requirements selection. Journal of Systems Architecture 54(6), 576–593 (2008)
4. Beck, K.: Extreme programming explained: embrace change. Addison-Wesley (1999)
5. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al.: Manifesto for agile software development (2001)
6. Berczuk, S.: Back to basics: The role of agile principles in success with an distributed scrum team. In: Agile Conference (AGILE). pp. 382–388 (2007)
7. Boehm, B.: Value-based software engineering. SIGSOFT Softw. Eng. Notes 28(2), 4– (Mar 2003), http://doi.acm.org/10.1145/638750.638776
8. Cohen, D., Lindvall, M., Costa, P.: Agile software development. Tech. rep., DACS SOAR Report (2003)

9. Debbiche, A., Diener, M., Svensson, R.B.: Challenges when adopting continuous integration: A case study. In: Proc. of 15th Int. Conf. of Product Focused SW Dev. and Process Impr. (Profes). pp. 17–32. Helsinki, Finland (2014)
10. Dingsøyr, T., Moe, N.B.: Towards principles of large-scale agile development. In: International Conference on Agile Software Development. pp. 1–8. Springer (2014)
11. Evbota, F., Knauss, E., Sandberg, A.: Scaling up the planning game: Collaboration challenges in large-scale agile product development. In: Proc. of 17th Int'l Conf. on Agile Softw. Dev. (XP). Edinburgh, UK (2016)
12. Fowler, M.: Continuous integration. Tech. rep. (2006), http://martinfowler .com/articles/continuousIntegration.html last visit: 01 2017
13. Golra, F.R., Beugnard, A., Dagnat, F., Guerin, S., Guychard, C.: Continuous requirements engineering using model federation. In: Proc. of 24th Int. Reqts. Eng. Conf. (RE). pp. 347–352. IEEE (2016)
14. Kauppinen, M., Savolainen, J., Lehtola, L., Komssi, M., Tohonen, H., Davis, A.: From feature development to customer value creation. In: Proc. of 17th IEEE Int. Reqts. Eng. Conf. (RE). pp. 275–280. IEEE (2009)
15. Kirikova, M.: Continuous requirements engineering in freedom framework: A position paper. In: Proc. of 2nd WS on Cont. Reqts. Eng. (CRE). Gothenburg, Sweden (2016)
16. Komssi, M., Kauppinen, M., Töhönen, H., Lehtola, L., Davis, A.M.: Roadmapping problems in practice: value creation from the perspective of the customers. Requirements Engineering 20(1), 45–69 (2015)
17. Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. Addison-Wesley Professional (2011)
18. Mathiassen, L.: Collaborative practice research. Information Technology and People 15(4), 321–345 (2002)
19. Neap, H.S., Celik, T.: Value of a product: A definition. Int. Journal of Value-Based Management 12(2), 181–191 (1999)
20. Neely, S., Stolt, S.: Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy). In: Proc. of Agile Conference. pp. 121–128. IEEE, Nashville TN, USA (2013)
21. Olsson, H.H., Bosch, J., Alahyari, H.: Customer-specific teams for agile evolution of large-scale embedded systems. In: 2013 39th Euromicro Conference on Software Engineering and Advanced Applications. pp. 82–89. IEEE (2013)
22. Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., Wieringa, R.: Do we know enough about requirements prioritization in agile projects: Insights from a case study. In: Proc. of 18th Int. Requts Eng. Conf. (RE 10). Sydney, Australia (2010)
23. Reifer, D., Maurer, F., Erdogmus: Scaling agile methods. IEEE Software 20(4), 12–14 (2001)
24. Roberts, M.: Enterprise continuous integration using binary dependencies. In: Extreme Progr. and Agile Proc. in Softw. Eng. (XP '04). pp. 194–201. Springer (2004)
25. Rogers, R.: Scaling continuous integration. In: Extreme Programming and Agile Processes in Software Engineering. pp. 68–76. Springer (2004)
26. Schwaber, K.: Agile project management with Scrum. Microsoft Press (2004)
27. Stahl, D., Bosch, J.: Modelling continuous integration practice differences in industry software development. Systems and Software 87, 48–59 (2014)
28. Wohlin, C., Aurum, A.: Criteria for selecting software requirements to create product value: An industrial empirical study. In: Value-based software engineering, pp. 179–200. Springer (2006)
29. Woodruff, R.B.: Customer value: the next source for competitive advantage. Journal of the academy of marketing science 25(2), 139–153 (1997)