Value Creation and Delivery in Agile Software Development: Overcoming Stakeholder Conflicts

Kati Kuusinen(✉)

University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark
kaku@mmmi.sdu.dk

Abstract. Agile software development aims at early and continuous value delivery. Yet the concept of value in agile development is underdefined and the meaning can be different for different stakeholders. Successful value delivery requires continuous collaboration with relevant stakeholders which is a main challenge in agile development. In fact, most software project failures are caused by poor communication and misunderstandings between stakeholders. This position paper discusses the meaning of value for business owners, customers, users, software developers, and user experience specialists and works towards an understanding on how to align and articulate value and its delivery in a software project.

Keywords: Software design tradeoffs · Agile software development
Collaboration in software development · Value creation
Requirements engineering

1 Introduction

Value creation is a continuous process throughout the development life cycle in agile software development and it can be described as follows. User or stakeholder needs are frequently written in the user story format: "*as a <role> , I want <a goal> so that <benefit /value>*" which captures both the requirement and its value. To create user stories the development team needs first to identify the relevant stakeholder roles, dig out what those roles value and what kind of value proposition would then help the team in trying to make the role happier or solve their problem. Then the team needs to chunk down those values and needs to the size and format of a user story. Finally, as the last step before implementation, the created stories are to be ordered based on their business value which might or might not be in line with the original stakeholder value. After this the team implements the user story into working software and gets feedback from the stakeholders for improvement. The development team then grooms and reorders the stories after each implementation increment when they have learned more about the stakeholders and their needs. The process is repeated until the customer is satisfied or the project otherwise comes to an end.

The described process is not straightforward and there are no established guidelines or tools to support stakeholder value identification and prioritization. In fact, it often

T. Clemmensen et al. (Eds.): INTERACT 2017, LNCS 10774, pp. 123–129, 2018.
https://doi.org/10.1007/978-3-319-92081-8_12

remains unspoken in teams what value means in the project context [3]. Business value frequently represents only the most important customers' point of view and it can differ from the user value [19]. In addition to business value, the required developer effort (cost of implementation) has an impact on the order of the user stories. Thus, from the beginning of the project, there are at least four competing forces – the voice of the business owner, customer, user and developer - which might all base on conflicting values.

There are no established means to balance between these values although several approaches have been presented. Decisions are habitually made based on the business owners', product owners' or customers' gut feeling. On the other hand, as the process is iterative and incremental, decisions can and should be made as late as possible with the then understanding throughout the software lifecycle and improved later when further information is available. Nevertheless, the concept of value remains often vague as the project proceeds and a shared idea of value between different stakeholders is rarely formed [3].

Thus, in a software project, several people can work together towards undefined value goal which each of them might understand in their own way from their own perspective. The big picture of the project then becomes blurred from the beginning and does not improve towards the end either [22]. Moreover, working with different stakeholders means working with people from various disciplines and backgrounds, which inherently makes communication more difficult as the used concepts and foci are different.

This position paper discusses the values and needs of different stakeholder roles and the assumptions these roles habitually have on other roles. Furthermore, it discusses how to overcome value conflicts to develop highly valuable software. Section 2 discusses the concept of value in software engineering. Section 3 presents the five focal roles (business owner, customer, developer user, and UX specialist) and their needs and values. Finally, Sect. 4 presents conclusions over this emerging work.

2 Value in Agile Software Engineering

This section discusses the concept of value in agile software engineering literature.

Graeber [6] defines value from three perspectives; in *sociological, economic* and *linguistic* sense as the conception of what is *ultimately good in human life*, as a *person's willingness to pay a price* for certain product or service benefits and as a *meaningful difference*. The three perspectives are relevant to software development as well. Software engineering aims at enabling the creation of complex computer-based systems which will meet the needs of users in a timely manner with quality [24]. Thus, a software system is both "*the programs, documents, and data*" created during the development and "*the resultant information that somehow makes the user's world better*" [24]. In general, software developers traditionally have their focus on the programs, documents and data whereas user experience specialists focus on ensuring that the resultant information will make the user's world better. Thus, user experience specialists' task is to understand the sociological side of value whereas the business owner brings in the economic perspective. As the software project proceeds, each software increment should bring in a meaningful difference (growth) in value.

The approach where distinct business and user experience specialists bring social and economic value to the project works in traditional development where developers implement predefined requirements. However, developers are in a central role in agile and the development team should be able to make decisions that foster business, customer and user value as well as technical quality and rapid development. Multidisciplinarity and cross-functional teams help in rapid decision-making on issues related to different value types [14]. The developer must learn from other disciplines to think about economic and societal value and the other internal roles should understand something about the technical side to make the work effortless and improve the communication [13, 14]. Also, it is beneficial for the customer to understand about the economic, technical and user side of the software project to be able to make informed decisions about the scope of the project, where to have users involved and so forth [19].

In software engineering, value is frequently understood as usefulness, utility, and importance or as the relative worth or monetary worth of something [3]. These types of value often necessitate that external stakeholders outside the development team (customer, user etc.) assign the value. Thus, the team must learn what the external stakeholders such as customers and users value during a development project. However, estimating, calculating, and measuring business value of software delivery is abstruse [25]. Software is ubiquitous and increasing in size and complexity. For these reasons, software development decisions have a crucial impact on the value delivery and better ways to address the value proposition are needed.

3 Stakeholder Views on Value

This section presents value from different stakeholder perspectives. The views mostly reflect on our own previous research but are also built on other literature. The roles are according to business to business development where a company orders software from another company typically for its own internal users who will use the software in their work.

Business or product owner is the person in the company developing the software whose main role is to ensure the economic revenue for the developing company but also to guarantee the customer satisfaction. Business or product owner's view is on the business and monetary value of the project for the developing company; how to maximize the return on investment for the shareholders. The secondary goal is to keep the customer happy and to build the relationship with the customer. Thus, the product owner might, for example, drive the development of features that they know are not useful for the user but which the customer wants for some reason [17, 19]. Sure, some business owners might want to explain why such a feature would be a bad idea and suggest a more feasible solution for example, to improve the long-term customer relationship and trust between the partners. For business owner, it is good to keep in mind that customer and user values are distinct. Customer does not necessarily know what the user values although they might say so [16, 17, 19]. Moreover, assessing the impact of a business decision on user experience can be beneficial in cases where user value differs from customer value [17].

Customer is a person from the purchasing company who often manages the requirements engineering and scoping of the project. Thus, a business customer values a solution for their problem. Typically, it includes a more efficient, robust, safer, faster, automated or cheaper approach compared to the current one. It can also be a novel approach or field for the customer. Software projects are typically mainly negotiated between the customer and the business or product owner roles. Customer usually selects the way of working in the project on a high level. They decide whether users are involved, whether the project is agile and so forth. It is crucial that the person who represents the customer has the required power of decision to enable fast and agile decision-making throughout the project. It is also critical that the customer understands the importance of user involvement and does not think they can decide for the user only because they understand the business process behind the software being purchased [19].

Software developer designs and implements the software. They value the work itself [2, 4]. Their goal typically is to build working, technically sophisticated software. Many developers are motivated by the thought that someone will use the software and that they are helping other people whereas others are mainly driven by being able to solve challenging technical problems [21]. Feeling good about the work, being in control of the development tasks, sense of competence, and being able to work with the development environment without effort are associated with developers' motivation and good developer experience [15, 18]. A pitfall for a developer is to love too much the technical side of the software and forget about the user or vice versa [17].

User is the person who interacts with the system [11]. Hassenzahl [8] sees user as a person with multiple hierarchical goals they are to achieve by interacting with a system. Users have instrumental goals, so called "*do-goals*", such as making a phone call. These instrumental goals can be satisfied with traditional usability properties such as ease of use, efficiency and usefulness. Hedonic goals or "*be-goals*" on the other hand are supported by systems hedonic quality, the perceived ability to self-expression, competency, autonomy, stimulation, relatedness and popularity. In professional life, the system's ability to motivate and create sense of professionalism are indicators of hedonic quality [20].

User experience specialist is responsible for the social value of the software under development. Their main goal is to satisfy users' needs and design for good user experience. UX specialist is typically the one who ensures that users' voice is heard from the beginning and throughout the project. UX specialist diffuses the user value from what the user is saying or showing. UX specialist is especially responsible of the hedonic quality of the software since the users cannot express that by themselves. Moreover, understanding and designing for the hedonic value is difficult without deep understanding of UX [20]. Thus, the other stakeholder roles are usually not able to do it although they can successfully learn many other UX tasks [13, 14, 20].

4 Overcoming Value Clashes

This section discusses practices found in literature that can help the agile team to identify and create value in a software project.

Business value is characteristically ambiguous and it is difficult to define it accurately in an agile software project [25]. Supporting social interactions between stakeholders [1] and having value workshops [23] can make it easier to identify value and form a mutual understanding of it. Even a short workshop between the business owner and users before writing user stories can help to clarify the project focus and lead to better economic and user value [16]. Also, different stakeholder roles can be identified and participated into thinking of what value means for that role. These role-biased values are then discussed together for example in a value workshop to create a mutual understanding of the overall business value before starting the actual development. The mutual understanding can then be groomed later as required.

A software value map [12] can broaden the thinking of value. It presents various value perspectives such as those of customer, financial, internal business learning and innovation. Customer value consists of perceived value including usability, reliability, delivery time and cost and lifetime value including customer revenue and different sources of cost.

Value points [9] or benefit points [7] can be used to concretize and order identified sources of value. They are used similarly to agile story points. Whereas story points measure the required implementation effort of a user story, value points or benefit points measure its value. For example, numbers 1, 2, 4, 8, 16 or Fibonacci series can be used to evaluate the value. The scale and scoring is arbitrary and subjective and the idea is not to create absolute value scores but to enable comparing between the importance of different sources of business value.

As value is not independent from cost, Gillain et al. [5] suggest that value should be assessed together with cost-estimates. The customer can consider one feature more valuable than another per se, but if there is a substantial difference in cost, they might change their opinion. One practical tool for assessing both value and cost is a scale that takes both value and cost points into account. This encourages to select between features instead of giving high value points to all of them.

Agile embraces change. The overall value can be unknown when the project starts and it can be challenging to conceptualize it. Therefore, revisiting and reordering the sources of value in increment reviews can be beneficial. Also, assessing the ability of the implemented software to generate expected value can make it easier to focus the project and in making estimations of the anticipated business value of the future increments. Continuous customer and user involvement helps in reassessing value as most of the business value ought to be assigned by external stakeholders [3].

5 Conclusion

This position paper presented views on value identification and creation in agile business to business software development. Early and continuous value delivery is a core function of agile software development. However, the value itself often remains undefined in agile projects and each stakeholder role might take it as given from their own perspective. That can lead to misunderstandings and make the project goals unclear. It can also lead to arbitrary decisions on product scope which may endanger the delivery of good user experience. This paper presented common pitfalls and thinking biases different stakeholder roles might fall into if they are not aware of those. Furthermore, this paper presented practices that can help in the identification and prioritization of value in agile software projects. Future work includes observing the value creation process in organizations to generate a sounder understanding of value sources and conflicts between them.

Acknowledgments. The research behind this paper has been conducted while I was working at Tampere University of Technology, Finland and for the Agile Research Network (ARN http://agileresearchnetwork.org) which is a collaboration between researchers at the Open University and the University of Central Lancashire, UK.

References

1. Alahyari, H., Svensson, R.B., Gorschek, T.: A study of value in agile software development organizations. J. Syst. Softw. **125**, 271–288 (2017)
2. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in software engineering: a systematic literature review. Inf. Softw. Technol. **50**, 860–878 (2008)
3. Dingsøyr, T., Lassenius, C.: Emerging themes in agile software development: introduction to the special section on continuous value delivery. Inf. Softw. Technol. **77**, 56–60 (2016)
4. Franca, A.C.C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., da Silva, F.Q.B.: Motivation in software engineering: a systematic review update. In: Proceedings of Evaluation and Assessment in Software Engineering (EASE), pp. 154–163 (2011)
5. Gillain, J., Jureta, I., Faulkner, S.: Planning optimal agile releases via requirements optimization. In: IEEE International Requirements Engineering Conference Workshops (REW), pp. 10–16. IEEE, September 2016
6. Graeber, D.: Toward an Anthropological Theory of Value: the False Coin of our Own Dreams. Springer, New York (2001). https://doi.org/10.1057/9780312299064
7. Hannay, J.E., Benestad, H.C., Strand, K.: Benefit points: the best part of the story. IEEE Softw. **34**(3), 73–85 (2017)
8. Hassenzahl, M., Schöbel, M., Trautmann, T.: How motivational orientation influences the evaluation and choice of hedonic and pragmatic interactive products: the role of regulatory focus. Interact. Comput. **20**(4–5), 473–479 (2008)
9. Highsmith, J.: Agile Project Management: Creating Innovative Products. Pearson Education, Upper Saddle River (2009)
10. Hoda, R., Noble, J., Marshall, S.: The impact of inadequate customer collaboration on self-organizing Agile teams. Inf. Softw. Technol. **53**(5), 521–534 (2011)
11. ISO 9241-210:2010: Ergonomics of human-system interaction. Part 210: Human-centered design for interactive systems (2010)
12. Khurum, M., Gorschek, T., Wilson, M.: The software value map—an exhaustive collection of value aspects for the development of software intensive products. J. Softw. Evol. Process **25**(7), 711–741 (2013)

13. Kuusinen, K.: Task allocation between UX specialists and developers in agile software development projects. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.) INTERACT 2015. LNCS, vol. 9298, pp. 27–44. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22698-9_3
14. Kuusinen, K.: BoB: a framework for organizing within-iteration UX work in agile development. In: Cockton, G., Lárusdóttir, M., Gregory, P., Cajander, Å. (eds.) Integrating User-Centred Design in Agile Development. HIS, pp. 205–224. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32165-3_9
15. Kuusinen, K.: Are software developers just users of development tools? Assessing developer experience of a graphical user interface designer. In: Bogdan, C., et al. (eds.) HCSE/HESSD -2016. LNCS, vol. 9856, pp. 215–233. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44902-9_14
16. Kuusinen, K., Mikkonen, T.: Designing user experience for mobile apps: long-term product owner perspective. In: 2013 20th Asia-Pacific Software Engineering Conference (APSEC), vol. 1, pp. 535–540. IEEE, December 2013
17. Kuusinen, K., Mikkonen, T., Pakarinen, S.: Agile user experience development in a large software organization: good expertise but limited impact. In: Human-Centered Software Engineering, pp. 94–111 (2012)
18. Kuusinen, K., Petrie, H., Fagerholm, F., Mikkonen, T.: Flow, intrinsic motivation, and developer experience in software engineering. In: Sharp, H., Hall, T. (eds.) Agile Processes Software Engineering, and Extreme Programming, XP 2016. LNBIP, vol. 251, pp. 104–117. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33515-5_9
19. Kuusinen, K., Väänänen-Vainio-Mattila, K.: How to make agile UX work more efficient: management and sales perspectives. In: Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, pp. 139–148. ACM, October 2012
20. Kuusinen, K., Väätäjä, H., Mikkonen, T., Väänänen, K.: Towards understanding how agile teams predict user experience. In: Cockton, G., Lárusdóttir, M., Gregory, P., Cajander, Å. (eds.) Integrating User-Centred Design in Agile Development. HIS, pp. 163–189. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32165-3_7
21. Lakhani, K.R., Wolf, R.G.: Why hackers do what they do: Understanding motivation and effort in free/open source software projects. MIT Sloan Working Paper No. 4425-03 (2003)
22. Lárusdóttir, M., Cajander, Å., Gulliksen, J.: The big picture of UX is missing in scrum projects. In: I-UxSED, pp. 42–48, October 2012
23. Paasivaara, M., Väättänen, O., Hallikainen, M., Lassenius, C.: Supporting a large-scale lean and agile transformation by defining common values. In: Dingsøyr, T., Moe, N.B., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (eds.) Agile Methods, Large-Scale Development, Refactoring, Testing, and Estimation, XP 2014. LNBIP, vol. 199, pp. 73–82. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14358-3_7
24. Pressman, R.: Software Engineering - A Practitioner's Approach, 7th edn. McGraw-Hill (2010)
25. Racheva, Z., Daneva, M., Sikkel, K., Buglione, L.: Business value is not only dollars – results from case study research on agile software projects. In: Ali Babar, M., Vierimaa, M., Oivo, M. (eds.) PROFES 2010. LNCS, vol. 6156, pp. 131–145. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13792-1_12