On Marketable User Stories for Customer Satisfaction

Pankaj Kamthan\textsuperscript{1} and Nazlie Shahmir\textsuperscript{2}

\textsuperscript{1} Concordia University, Montreal, Canada
pankaj.kamthan@concordia.ca
\textsuperscript{2} Canadian Pacific Railway Limited, Calgary, Canada
nazlie_shahmir@cpr.ca

Abstract. The profitability, reputability, and sustainability of any organization depend intrinsically on customer satisfaction. In the context of an agile project, this paper proposes a set of conceptual models that form an input to a customer-centered, market-oriented, user story engineering process aiming to provide a continual positive customer experience. In doing so, organizational constraints pertaining to economics of the user story engineering process are considered.

Keywords: Agile methodology · Axiology · Conceptual modeling · Requirements engineering · Software consumerization · User experience

1 Introduction

The consumerization of software has radically changed how software, as a commercial \textit{product}, is viewed, perceived, managed, developed, tested, promoted, released, distributed, sold, deployed, used, reproduced, and maintained. In recent years, many organizations of all sizes have adopted agile methodologies for their software development-related initiatives [1]. The viability of any commercial organization rests on \textit{customer satisfaction} [2]. It has been shown that customer satisfaction is a multi-dimensional concept contingent on many factors [3], including continual \textit{positive customer experience}.

It is acknowledged broadly that principles and practices of in one of the earliest stages in development, namely \textit{requirements engineering}, have a major influence on the direction of software development and outcome of a software project [4]. The purpose of this paper is to provide a starting point for augmenting ‘conventional’ agile RE with knowledge derived and distilled from developmental psychology, human-computer interaction, and experiential marketing in order to provide a positive customer experience.

The rest of the paper is organized as follows. In Sect. 2, background is provided and related work is highlighted. The conceptual models and, based on these models, a customer-centered and market-oriented process, necessary for user story engineering, are introduced in Sect. 3. In Sect. 4, directions for future research are outlined. Finally, in Sect. 5, concluding remarks are given.
2 Background and Related Work

2.1 Agile Methodologies and Customer Experience

The agile methodologies are characterized by the *Agile Manifesto*. The Agile Manifesto includes principles and practices that prompt software development methodologies to satisfy the needs of small-and-medium-sized enterprises (SMEs).

In the evolution of the agile methodologies, there have been a number of paradigmatic stages. The interest in this paper is in the integration of agile methodologies and user-centered design methodologies with similar goals, such as *user experience design* (UXD) [5].

Figure 1 shows the intrinsic relationship between customer experience and related relevant concepts from UXD. It can be concluded that user experience is necessary, but not sufficient, for customer experience. For example, customer experience can involve aspects beyond the use of the product, such as those related to replacing the product or its parts, or returning the product and refunding the customer.

![Fig. 1. The customer experience in context.](image)

A *customer* is an individual who pays for software, and a *user* is an individual who for some purpose interacts with software. Therefore, in general, a user may not be a customer, and, conversely, a customer may not be a user. However, in this paper, for the sake of simplicity, user and customer are used synonymously, unless otherwise noted.

2.2 Agile Requirements

In agile methodologies, software requirements are usually expressed as either *user stories* or lightweight *use cases* [4]. The interest in this paper is in user stories [6], which are supported natively by many agile methodologies, including Extreme
Programming (XP), Scrum, and their extensions, such as BizDevOps and continuous requirements engineering; by organizational initiatives, such as the Guide to the Business Analysis Body of Knowledge (BABOK); and by the ISO/IEC/IEEE 26515 Standard.

A user story statement in controlled natural language text is usually structured as: A role can goal to value. For example, “A customer can make a payment using a credit card to purchase a shopping item.” is a user story statement for a shopping system. For practical purposes, a user story statement is usually annotated with metainformation elements, such a priority, estimate, and acceptance criteria.

2.3 Marketable User Stories

Figure 2 shows a user story as part of different spatial and temporal configurations, each of which is described subsequently.

A feature is a collection of capabilities of a software system that provides value to a user. A minimal marketable feature (MMF) is an atomic unit of software value creation [7]. An MMF creates market value (for the organization) in one or more of the following ways: competitive differentiation, revenue generation, cost saving, brand projection, and enhanced loyalty. In an agile project, an MMF is developed in a single iteration of the underlying agile process, and is composed of multiple user stories. From the perspective of release engineering, a minimum viable product (MVP), which is a not necessarily deployable release 1 and aimed primarily towards internal stakeholders, consists of at least one MMF. An MVP could, over time, evolve as a deployable release \( \geq 2 \), aimed primarily towards external stakeholders, to become a minimum marketable product (MMP) [8].
3 A Framework for Customer-Centered, Marked-Oriented, User Story Engineering

3.1 Conceptual Models for a User Story

A model is a simplification of a thing for some purpose. In software engineering-related modeling, a thing can be almost anything, including a concept; a model is usually conceptual; the purpose of a model is usually reducing the size and/or complexity of a thing for the sake of understanding that thing; and a model is usually created through abstraction of a thing.

Figure 3 shows a compendium of preliminary, interrelated, conceptual models in relation to a user story statement.

Fig. 3. An abstract user story derives knowledge from conceptual models for its description.

These conceptual models, each of which is described subsequently, are a requisite for appropriate user story engineering. This is because of a number of reasons, including that these models identify and separate relevant concerns, encapsulate thought processes, lend an understanding to and provide rationale for some of the activities during user story engineering, are an outcome of preparation necessary for user story engineering, can serve as sources during requirements traceability, and, being independent of any technique, technology, or tool, can be reused in similar contexts if necessary.
The development of these conceptual models requires a suitable combination of ethnographic techniques (such as interviews, observations, and surveys) and ideation techniques (such as brainstorming, mind mapping, and sketching).

These conceptual models could be complemented by a glossary to facilitate communication among stakeholders, create a common ground for terminological reference, and reduce the potential for ambiguity.

Finally, if deemed necessary, the conceptual models as well as other agile project artifacts could be placed on a suitable Social Web-based version control system, such as Wiki, to foster collaboration, support evolution, and expedite dissemination.

3.1.1 Role Model
This model provides an understanding of the characteristics and behavior of a typical user (playing a particular role with respect to the software system) [9]. There can be different kinds of role models. For example, a persona (an archetypical user of a software system) or a proto-persona, structured according to the ISO/IEC/IEEE 26515 Standard, is a candidate for the desired model [10]. In general, the mapping between the set of people and the set of roles is many-to-many.

3.1.2 Goal Model
This model provides an understanding of the goals of a typical user for using the software system. The construction of an appropriate goal model is especially crucial as both the role model and the value model depend on it. There can be different kinds of goal models. For example, the following organization of goals can be used for the desired model [11]: an experience goal (is at a visceral level and is about how a user wants to feel while using the system), an end goal (is at a behavioral level and is about what a user wants to do with the system, and a life goal (is at a reflective level and is about who a user wants to be by using the system).

3.1.3 Value Model
This model provides an understanding of the values that could be gained by the use of the software system. There can be different kinds of value models. For example, one possible model is a model for consumption values [12], which is aligned with the Maslow’s Hierarchy of Needs and provides the following classification of values: conditional (say, value associated with a COVID-19 tracing application), emotional (say, value associated with a computer game), epistemic (say, value associated with the knowledge of a new shopping portal), functional (say, value associated with a statistical computing software), and social (say, value associated with an electronic social network).
3.2 Elements of Customer-Centered, Market-Oriented, User Story Engineering

3.2.1 A Customer-Centered, Market-Oriented, User Story Engineering Process

Figure 4 shows a single iteration of the desired customer-centered, market-oriented, user story engineering process. The solid lines indicate the transitions between steps, and the dashed lines suggest that the feedback is optional.

![Diagram of the user story engineering process]

**Fig. 4.** The basic steps of a customer-centered, market-oriented, user story engineering process.

The steps of the aforementioned process can be explained as follows:

1. **Express** the user stories for all the MMFs aimed for a given iteration of an agile process. This would typically involve authoring, reviewing, and publishing the user stories. The knowledge from the aforementioned conceptual models is relevant to this step. The collection of user stories could be organized visually using techniques such as *user story mapping*, or, more generally, *customer journey mapping*. An economic challenge in this step is the accrual of user story debt.

2. **Experiment** with the MMFs by developing a high-fidelity UXD prototype. The goal of such a prototype should not only be to maximize user experience, but also to minimize the cost of interaction for the users. For example, such cost can manifest itself in form of inconvenience, unnecessary cognitive complexity, loss of information, unanticipated latency, or (excessive) time and effort spent in recovering from mistakes. This requires empathy of designers with the users [13], and an alignment between the mental models of designers and users [14]. It also requires adequate support in the prototype for error handling and helpful error messaging. To realize that, the prototype could rely on established marketing design knowledge, such as principles of persuasion [15], device knowledge, and information and interaction design knowledge, such as patterns [16]. An economic challenge in this step is the cost of representativeness of the prototype.
3. **Evaluate** the prototype for experience and satisfaction (by involving actual users, for example, through crowdsourcing [17] and survey, respectively). This could lead to feedback about the product (which could prompt a refinement of the user stories and/or the prototype), as well as about the process (which could prompt an improvement of the user story engineering process). An economic challenge in this step is the cost of testing, including regression testing, and the cost of user story analytics, including analysis of data extracted from the feedback.

The previous steps could be repeated to create an MVP, and, eventually, an MMP.

### 3.2.2 Economics of Customer-Centered, Market-Oriented, User Story Engineering

It is crucial that the value an MMP provides to an organization does not conflict with the value it provides to the (potential) customers. Indeed, for the agile software development to be successful, the aim of the business model should be to maximize the intersection of the two types of aforementioned values, namely the “Win-Win” value, as shown in Fig. 5. This can be done by applying design thinking [18] and lean principles [19] suitably, and, at the same time, avoiding dark patterns [20] that aim for exploitability rather than usability, and reducing the potential for negative uses [21] that emanate from the violations of, say, privacy, safety, and/or security.

![Fig. 5. The “Win-Win” value proposition.](image)

The “Win-Win” value is neither automatic, nor free-of-cost. Indeed, a successful realization of “Win-Win” value is subject to trade-offs between certain organizational constraints pertaining to economics.

Figure 6 shows an equilateral triangle, representing indispensable, minimal, and (for the sake of simplicity) equally-significant constraints of a commercial agile project. It can be concluded from an examination of this triangle that it is necessary to make compromises among ‘high value’, ‘low cost’, and ‘short time-to-market’, and that it is usually possible to achieve only one out of three.
For example, an MMP aimed at providing ‘high value’ needs to be of ‘high quality’, and such an MMP, in turn, tends to take longer to develop and costs more; conversely, it is possible to shorten the initial development time, but that comes at the expense of reducing the value provided by the delivered MMP, assuming cost is kept almost constant.

Thus, practical considerations necessitate equilibrium among the constraints (and that, evidently, requires moderation of each of the constraints). To provide some flexibility, however, this equilibrium is not represented as a point but rather as a region.

### 4 Directions for Future Research

#### 4.1 Empirical Studies on Conceptual Modeling in Agile Methodologies

Even though the paper advocates the creation and integration of a number of conceptual models, currently most agile methodologies do not have native support for extensive conceptual modeling. The agile methodologies, however, continue to evolve because of the continual feedback loop between their theory and practice. Therefore, conducting empirical studies in organizations with appropriate agile process maturity levels for eliciting the technical as well as non-technical challenges, analyzing associated risks, and using *earned value management* (EVM) as well as other metrics for calculating the *return on investment* (ROI) in the constructions of the conceptual models, is of research interest.

#### 4.2 User Stories for Affective Values

The attention in this paper is almost exclusively on *utilitarian values*. To provide a positive user experience, it is also important to attend to *hedonic values* [11], such as enjoyment and engagement. The type of enjoyment in browsing a shopping system, however, can be different from playing a computer game, and the type of engagement in an interaction with a rental car reservation system is different from that of a banking system. It has been observed that due consideration of the hedonic values in software
development may not only help the software satisfy the customers better, it may also help the organization gain competitive advantage by being able to market the product better [22]. Therefore, investigating the degree to which customers place significance on what kinds of hedonic values in their interaction with different types of software systems, and the implications of it for user story engineering, is of research interest.

4.3 Market-Oriented User Story Education

There are certain characteristics inherent to the nature of user stories (namely, abstractness, futuredness, and insipidness) that already pose a challenge in teaching user stories. Furthermore, in a real world context, non-technical concerns often constrain technical concerns. However, the students in ‘conventional’ software requirements engineering education are usually not exposed to realities of software-as-a-business, to intricacies of marketing aspects of a software product, or to the operational environment of a software product. Therefore, introducing such topics in software requirements engineering-related courses, and exploring their impact on the motivation and performance of students, is of research interest.

5 Conclusion

A market-oriented approach to software development opens new vistas for customers, but also brings new challenges for the software project team to be overcome. An agile customer-centered, market-oriented, methodology holds considerable promise for an organization committed to software development-related activities. If the industrial surveys are any sign, this trend is likely to continue for the foreseeable future.

A successful realization of prospective benefits of such an endeavor, however, requires the organization to be explicitly sensitive to the needs, values, and preferences of its customers, and be able to manage the trade-offs involved. It also requires customer satisfaction to be forethought rather than afterthought. This is possible by equipping and executing user story engineering with conceptual models that encapsulate knowledge related to customers, as this paper has attempted to show.

Acknowledgement. The authors are grateful to CCE for a Professional Development Grant.

References

1. Brunetti, G., Feld, T., Heuser, L., Schnitter, J., Webel, C.: Future Business Software: Current Trends in Business Software Development. Springer International Publishing (2014)
2. Fečková, I.: An index method for measurement of customer satisfaction. TQM Mag. 16(1), 57–66 (2004)
3. Klaus, P.: Measuring customer experience: how to develop and execute the most profitable customer experience strategies. Palgrave Macmillan, London (2015)
4. Wiegers, K., Beatty, J.: Software Requirements, Third Edition. Microsoft Press, Redmond (2013)
5. Brown, D.D.: Agile User Experience Design: A Practitioner’s Guide to Making It Work. Morgan Kaufmann, Burlington (2013)

6. Cohn, M.: User Stories Applied: For Agile Software Development. Addison-Wesley, Boston (2004)

7. Denne, M., Cleland-Huang, J.: Software by Numbers: Low-Risk, High-Return Development. Prentice-Hall, Upper Saddle River (2003)

8. Aschauer, B., Hruschka, P., Lauenroth, K., Meuten, M., Rogers, G.: Handbook of RE@Agile According to the IREB Standard. International Requirements Engineering Board (2019)

9. Wendel, S.: Designing for Behavior Change: Applying Psychology and Behavioral Economics. O’Reilly Media, Sebastopol (2014)

10. Cooper, A., Reimann, R., Cronin, D., Noessel, C.: About Face: The Essentials of Interaction Design. Wiley, Hoboken (2014)

11. Norman, D.A.: The Design of Everyday Things. Basic Books, New York (2013)

12. Sheth, J.N., Newman, B.L., Gross, B.L.: Why we buy what we buy: a theory of consumption values. J. Bus. Res. 22(2), 159–170 (1991)

13. Kamthan, P., Shahmir, N.: Effective user stories are affective. In: The Eleventh International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2017), 7–10 November 2017, Philadelphia, USA (2017)

14. Young, I.: Mental Models: Aligning Design Strategy with Human Behavior. Rosenfeld Media, New York (2008)

15. Cialdini, R.B.: Influence: The Psychology of Persuasion. HarperCollins, New York (2007)

16. Tidwell, J., Brewer, C., Valencia, A.: Designing Interfaces: Patterns for Effective Interaction Design, Third Edition. O’Reilly Media, Sebastopol (2020)

17. LaToza, T.D., van der Hoek, A.: Crowdsourcing in software engineering: models, motivations, and challenges. IEEE Softw. 33(1), 74–80 (2016)

18. Maedche, A., Botzenhardt, A., Neer, L.: Software for People: Fundamentals, Trends and Best Practices. Springer-Verlag, Heidelberg (2012)

19. Harris, M.D.S.: The Business Value of Software. CRC Press, Boca Raton (2018)

20. Gray, C.M., Kou, Y., Battles, B., Hoggatt, J., Toombs, A.L.: The dark (patterns) side of UX design. In: The 2018 CHI Conference on Human Factors in Computing Systems (CHI 2018), 21–26 April 2018, Montreal, Canada (2018)

21. Kamthan, P., Shahmir, N.: A characterization of negative user stories. In: The Twenty Eighth International Conference on Software Engineering and Knowledge Engineering (SEKE 2016), 1–3 July 2016, Redwood City, USA (2016)

22. Agarwal, A., Meyer, A.: Beyond usability: evaluating emotional response as an integral part of the user experience. In: The ACM CHI 2009 Conference on Human Factors in Computing Systems (CHI 2009), 4–9 April 2009, Boston, USA (2009)