

Introduction

Requirements Engineering play a vital role in product development since it is the most critical step in the Software Development Life Cycle. The tasks in every other stage depend on the effectiveness of requirements engineering. It thus determines the success or failure of a product in addition to avoiding unnecessary fixation costs. For an effective requirement engineering we need to understand what the customers and stakeholders expect from the system. Mere description of the expected system outcome is mostly insufficient. The user stories and use case modelling techniques help us in overcoming such knowledge gaps during requirements elicitation. This report consists of two sections, which are requirements modelling and process selection. The requirements modelling section outlines, a comparison between the user stories and use case modelling techniques, and an application of these techniques to process models based on their suitability. The process selection model outlines the selection of different software engineering processes to each of the MyTrip Website and Hospital Patient Records Systems.

Requirements Modelling (1,500)

Technique Comparison (750)

User stories and use cases both give us a user centric approach to requirements engineering. However, their usability cannot be interchanged. In this section we compare the advantages and disadvantages of using user stories and use cases for requirements modelling.

Advantages and Disadvantages of Use Cases

A use case captures the flow of information within the system and its interaction with the actors with much detail. The system is divided into actors and their related use cases. It thus provides a high-level description of system functionality from a business point of view without getting into technical details. Use cases represent the big picture of the product developed. It effectively captures dependency with a high level of detail and provides completeness.

However, use cases are generally too large and time consuming. They cannot be trusted to result in accurate estimates. Moreover, use case modelling is highly sensitive to system requirements, which means that even a slight change in requirements is going to result in having to redo the process all over again [2]. This may be the case when a project is in progress and the requirements are found to be changing over certain phases during the software lifecycle. For example, after writing use cases, during later stages it could be noticed that a use case is used by more actors than the ones specified in the use case description. In this case, the project duration could extend with change in functionality [2].

Advantages and disadvantages of user stories

In user stories, we dive into the technical system aspects early during requirements modelling which means that:

- there is a common understanding between the user and the developer concerning a requirement

- It is more beneficial to developers since it involves a more technical view of system requirements.
- requirement change due to technical feasibility is less likely to occur

User story developers are required to verify customer requirements various times, resulting in a product closely aligned to customer needs [3][4]. There is also significant improvement in costs and productivity since it reduces additional efforts and cost to fix the final product to meet customer requirements during later stages [3][4].

Since user stories are written in natural language, they can be interpreted in different ways by different people. A term used by a customer to describe something could result in the developers interpreting the same to mean something else [1]. For example, a story that reads, 'After 2 free meals, I want my workers to be entitled to a 20% discount for snacks or chips and drinks, twice a week so that they don't have to go elsewhere to purchase it', can be difficult to understand during product development [1]. Did it mean snacks or (chips and drinks) or did it mean (snacks or drinks) and chips [1]?

Prioritization

The motive behind the usage of use cases and user stories are not the same. Use cases are developed to signify an agreement for certain usability criteria between the customers and developers, while the motive behind, user stories is to aid in measuring the effort involved in development, to further aid in release and iteration planning [1][3][4]. Thus, their usability cannot be interchanged.

Process Models Applicability (750)

Plan driven

In the plan driven process model, the project is broken down into stages, in which each stage is comprised of activities that are driven by a plan and follow a certain schedule [6]. It facilitates a single release of the product towards the end. This process model involves gathering and finalising the product requirements prior to proceeding with its development [6]. Thus, the use case modelling would be a more suitable technique to gather requirements for this approach since it involves finalising all the requirements before proceeding to the technicalities of development. The user stories technique, on the other hand will involve revisiting the requirements several times when questions arise during product development. While, this is possible even during use case modelling, it's more likely to occur if the user stories technique is used since this technique lacks the big picture of the product. And this is necessary to proceed to the later stages of product development. Using user stories could thus result in a prolonged period of requirements gathering, thus slowing down the project. We can hence conclude that use case modelling is a more suitable technique for this process model.

Incremental

Incremental models facilitate an iterative release of a product. Each iteration comprises of requirements analysis, design, coding and testing stages. Since this model is designed to revisit the requirements analysis stage at the start of every new iteration, we can conclude

that it does not involve finalising all the requirements before proceeding to the more technical stages of product development. Thus, the user stories technique is more suited to this process model. Use case modelling for incremental process models will be inappropriate since it will extend the length of each iteration.

Agile

The agile development model works under the assumption that the requirements for a project never remain fixed during any stage at all. It involves the development of the most crucial features earlier in during product development while the other features remain uncertain [3][4]. The user stories technique thus fits most efficiently with this model since it enables the customers and developers to be in contact with each other throughout product development [4]. Moreover, user stories capture only the most important requirements along with their usability criteria directly from the customers which is precisely what the agile methodology is all about [3].

Lean

The lean software development is a kind of agile development model, which facilitates optimization of resources and development time to deliver a minimum viable product to the market. The usage of this is further studied to deliver the next iteration of the product. While agile focuses on delivering the project early in time boxed iterations, lean further focuses on optimization of resources and early delivery of the project together. It discards any feature or effort that doesn't add value to the product [5]. Since the lean development model is a sub framework of the agile development model, user stories are more suited to its requirement needs. Use case modelling for a lean development model would only involve repeated unnecessary documentation resulting in consumption of more time and effort, since it does not involve finalisation of requirements before proceeding to development.

Formal

The formal development model focuses on the usage of mathematical processes and methods for system development [6] [7]. Here, formal logic is used for system specification which is followed by an incremental approach for developing each subsystem separately [7]. Each system is then integrated and tested. Since all the requirements for the system are specified through formal logic, it would be inappropriate to use either use case modelling or user stories for requirements gathering in the formal development model. Both user stories and use case modelling involve the usage of the natural language. While one can argue that the requirements, once collected through natural language can later be transformed to formal logic specification, it could lead to errors in interpretation which could further result in escalation during the later stages.

Process Selection (1,500)

MyTrip Website (500)

The MyTrip Website is a big project since it involves a lot of functionalities and external dependencies like links to other social platforms. Moreover, it also comprises of a wide range of requirements that are not too definitive when compared to the Hospital Patient Record

System. While generating user stories for the MyTrip website, most requirements were interpreted in different ways by different team members. It was required to arrive at a mutual consensus among each of us before we could proceed with the development of user stories to convey requirements. It can thus get complicated to gather all its requirements upfront, before proceeding to development. This makes it hard to estimate the effort required in development. It is understood that the size of the project can greatly influence the certainty of the requirements [13]. Projects with larger sizes are more likely to experience a change in requirements as development commences [9][14]. Such projects are more likely to be aligned to customer needs if there was more customer interaction throughout its development. Following the agile methodology to develop the MyTrip Website will yield effective results. The agile methodology is designed to prioritize customer satisfaction over everything else and continuously work towards achieving it [14]. It involves releasing the project to customers in sprints. Each sprint is time boxed and focuses on a set of features that are crucial to that sprint. Based on the customer feedback provided, further releases are modified. Another reason why agile would be suited to this project is the basic principle of the model, that the project requirements never remain fixed throughout the project [14]. This is precisely something that is necessary to develop the MyTrip website. Since there are too many features in this project, if the requirements are not finalised before proceeding to development, it could result in a great deal of effort and resources being wasted on an insignificant requirement which has little or no value to the customer. Using agile, such situations can be avoided if we can prioritize the most important requirements and work on them during the early releases of the product.

Hospital Patient Records System (500)

The hospital patient records system consists of requirements that are more straight forward than the MyTrip website. Although, the nature of the requirements is more definitive, the hospital patient records system is a system of large size and complexity. Any errors in misinterpretation of the data stored could have a huge impact on the patients of the hospital and eventually its reputation.

There are various factors to consider while building this system, like:

- The size of the system which is determined by number of users and the number of patients
- The ability to withstand cases where the hospital expands and there are more incoming patients
- The ability to store large amounts of dynamic system data

Given these constraints, the agile methodology would be a good approach for this system to be developed. To be more precise, the Dynamic Systems Development Method would be a perfect approach for this project. The DSDM is an agile framework that gives high priority to individual interactions, software functioning, customer satisfaction and change response [10][11]. It is highly beneficial during the development of a complex system with unpredictable business needs. In other traditional approaches, while building the system to withstand so many possibilities, the most crucial business needs could be compromised on.

In the DSDM approach, instead of focusing on all different possibilities and scenarios, we build the system to cater to the more certain and prominent needs [10] [11].

The DSDM process model comprises of six stages which are [10] [11][17]:

1. Pre-project stage
2. Feasibility
3. Foundations
4. Evolutionary development
5. Deployment
6. Post-project stage

Some of the main reasons why DSDM is the right approach to this system are [17]:

- The foundations stage, which is exclusive to DSDM, helps in establishing a basic set of ground rules with the stakeholders, gives a head start while determining the major system goals for the HPRS system.
- The DSDM enables effective project governance for large projects like the HPRS in addition to following the principles of agile.
- The uncertainties in user requirements for the HPRS can be overcome by following DSDM.

Elicitation Technique (for the SE process) (500)

An effective elicitation technique for the MyTrip website is prototyping. Through prototyping, the developers and customers will remain engaged throughout the project development [15] [6]. And this goes hand in hand with the agile methodology as well since customer satisfaction is given a high priority in Agile. Through this technique,

- Misinterpretation of user requirements can be avoided and rectified early during system development [15] [6].
- Since there are too many requirements for the website it may be discovered that the users want to alter their requirements during the later stages of development.
- Easier approaches to implementing certain requirements may be discovered once the purpose of a requirement is clearer.

For HPRS, on the other hand, workshops would be a more effective requirements elicitation technique. The reason behind this, is that there are multiple stakeholders involved in this project and to get everyone on the same page, it is best if they are all available in a common place for collective decision making. Further usage of user stories could be followed in these workshops to finalise requirements effectively.

The advantages of using workshops for the HPRS system would be to [16]:

- Avoid disagreement or misinterpretation of information between multiple stakeholder parties which will adversely affect project development in the future.
- More straightforward and time saving to establish the ground rules for requirements during the foundations stage of the DSDM model.

The workshop for the DSDM model should include [18]:

- An agenda for the workshop
- One or more experienced facilitators
- Division of big groups of people into smaller groups of like-minded people as the session progresses
- Identification of tensions and competing needs to suppress it before it worsens.
- Timing each session to avoid wasting time

Summary (250)

Both prototyping and workshops are suitable elicitation techniques for the Agile methodology since both these techniques keep the developers and stakeholders engaged throughout the development process and develop a product that satisfies all or most of the customer needs.

Prototyping on one hand when used with any agile methodology ensures that the final product consists of all the necessary features and properties as per its intended use. Additionally, prototyping is more beneficial while developing products with various unknown factors like in scenarios where there is no existing product in the market with similar features and this is the first of its kind.

On the other hand, gathering all the stakeholders together and agreeing on various issues at a workshop can reduce the risk of inaccuracy in gathering requirements. The organizers are less likely to avoid misinterpretation in information due to in person communication rather than over email or telephonic conference. However, in cases where the stakeholders come from different geographic locations, it may be difficult to frequently conduct workshops, in which case a video conference will be more beneficial.

Selection Rationale (250)

For a complex website like MyTrip, with many requirements, it can be tough to determine the precedence of one over another. The agile methodology, when used along with prototyping as an elicitation technique will give the customers an actual feel of the system. Agile prioritizes delivery of a customer valued product within a short span of time through continuous collaboration through feedbacks. Thus, this would enable customers to try a product and enhance its usability according to their needs.

Prototyping when combined with the agile methodology:

- Reduces project cost
- Identifies defects at early stages

For a large system like the HPRS, with a dynamic nature, combining the DSDM process with workshops as a requirements elicitation technique is ideal. This is because in the DSDM we need to agree on certain factors before proceeding to project development. We also need to consider the vast number of stakeholders involved in the project.

Thus, the DSDM along with workshops as an elicitation technique help to [18]:

- Establish stakeholder consensus
- Avoid misinterpretation of requirements
- Save communication time lapse

Conclusion (250 words)

Establishing a common understanding of requirements between the developers and the customers is the most crucial part of requirements engineering. While traditional software process models work towards finalising the requirements before its development, the more recent process models like agile work with the assumption that requirements remain uncertain right until the product is developed.

However, even for process models like agile, it is necessary to understand what the customer wants before developing anything at all. This process itself can get tricky since what the customers claim they want and what they want are most often two different things. Techniques like user stories and use case modelling help in easing the communication between the customers and developers. Additionally, they help the developers understand the customer's purpose of using the system.

However, despite having a clear understanding of the requirements, many things could go wrong in a project during the later stages. It is thus important that we choose an appropriate elicitation technique for the software process model. This would enable the delivery of a product of high customer satisfaction.

References

- [1] M. Cohn, "Advantages of user stories for requirements," *InformIT Network*, available at: <http://www.informit.com/articles>, 2004.
- [2] A. Cockburn, *Writing effective use cases*. Addison-Wesley Professional, 2000.
- [3] G. Lucassen, F. Dalpiaz, J. M. E. M. v. d. Werf, and S. Brinkkemper, "Forging high-quality User Stories: Towards a discipline for Agile Requirements," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, 2015, pp. 126-135.
- [4] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [5] C. Ebert, P. Abrahamsson, and N. Oza, "Lean software development," *IEEE Software*, no. 5, pp. 22-25, 2012.
- [6] Sommerville, Ian. *Software Engineering*, 6th edition. Addison-Wesley, 2001.
- [7] Spivey, J. M. *The Z Notation: A Reference Manual*. Prentice Hall, 1992.
- [8] L. C. Alexander and A. M. Davis, "Criteria for selecting software process models," in *[1991] Proceedings The Fifteenth Annual International Computer Software & Applications Conference*, 11-13 Sept. 1991 1991, pp. 521-528, doi: 10.1109/cmpsac.1991.170231.
- [9] M. T. Hansen and H. Baggesen, "From CMMI and Isolation to Scrum, Agile, Lean and Collaboration," in *2009 Agile Conference*, 24-28 Aug. 2009 2009, pp. 283-288, doi: 10.1109/agile.2009.18.
- [10] J. Stapleton, *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press, 1997.
- [11] J. Stapleton, "DSDM: Dynamic systems development method," in *Proceedings Technology of Object-Oriented Languages and Systems. TOOLS 29 (Cat. No. PR00275)*, 1999, pp. 406-406: IEEE.

- [12] S. S. E. Kirchner, "Defining Requirements in Prototyping: The Holistic Prototype and Process Development," 2018. [Online]. Available: <https://www.designsociety.org/publication/40953/Defining+Requirements+in+Prototyping%3A+The+Holistic+Prototype+and+Process+Development>.
- [13] J. Dick, E. Hull, and K. Jackson, *Requirements engineering*. Springer, 2017.
- [14] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development*, vol. 9, no. 8, pp. 28-35, 2001.
- [15] B. Bähr, "Prototyping Requirements," in *Prototyping of User Interfaces for Mobile Applications* Cham: Springer International Publishing, 2017, pp. 37-57.
- [16] E. Gottesdiener, *Requirements by collaboration: workshops for defining needs*. Addison-Wesley Professional, 2002.
- [17] "The DSDM Agile Project Framework (2014 Onwards)", *Agile Business Consortium*, 2019. [Online]. Available: <https://www.agilebusiness.org/content/process>. [Accessed: 16- Apr- 2019].
- [18] "Requirements Workshop Technique: Exploring the Power of Collaboration," vol. 2019, ed. Business Analyst Learnings, 2013.