Pareto Analysis for Agile Requirements Prioritization

Tan Amelia
Computer Systems & Software Engineering
Universiti Malaysia Pahang
Pahang, Malaysia
tan.meilie@gmail.com

***Abstract* – Pareto Analysis or more popularly called the 80/20 rule is a theory in economics. In requirements prioritization, the pareto analysis is included in the category of scoping prioritization techniques, such as MoSCoW, Five-Whys, and Time-Boxing. The Pareto Analysis rule states that 80% of the value is obtained from 20% features. The result of the application of Pareto analysis is a Pareto chart that visualizes the important requirements that must be considered, in order to see the needs and contributions to software development. Currentlys, the use of Pareto analysis in the requirements phase, especially in the requirements prioritization process becomes interesting and needs more accurate empirical verification. The purpose of this paper is to illustrate and provide an understanding of the use of Pareto analysis in agile requirements prioritization.**

***Keywords* - requirements prioritization; prioritization techniques; pareto analysis; agile**

I. INTRODUCTION

In software development, knowing the needs of the product owner and fulfilling all their desires is a goal to be achieved, so that the resulting software as expected. However, based on the experience that occurred in the field, will be very difficult to meet all the desires of the product owner. This is due to the many limitations that exist in a software development project, such as time, cost and human resources. As a solution to these problems, it is necessary to determine the priority of needs so that it is known which needs to be prioritized.

One of the tools used for requirements prioritization is Pareto analysis. Based on Pareto principles, its application in the needs analysis is that 80% of the value is obtained from 20% features. Based on widely known fact that not all requirements will be implemented and some requirements are very important. With high product owner expectations, tight schedules and limited resources, prioritization becomes a common strategy to limit coverage and deliver critical functionality as quickly as possible[1].

Prioritization strategies are grouped into two categories: Scoping and Ordering. Examples of Scoping techniques are Objective Alignment, Five-Whys, MoSCoW, Time-Boxing, Risk-Based Ranking and Pareto Analysis. While in Ordering techniques include: Subjective Ranking, Group-Based Ranking, Limited Votes, Pain Ranking, Pair-Wise Comparison and Value-Based Ranking [2].

The Pareto principle states that 80% of the effect comes from 20% of the cause. In Agile product development, the same rule would be that 80% product value comes from 20% product backlog item. Furthermore, how to define what items should be included in the 20% is a prioritization requirements problem that must be solved in an optimal way.

The purpose of this paper is to explain empirical evidence on the use of Pareto analysis for requirements prioritization. The software development method used for the explanation of this paper is Scrum, which is one of the methods by applying the principles of the Agile approach.

This paper is structured as follows: Section 2 describes the materials and methods used, Section 3 analyzes the results, while Section 4 presents the conclusion which summarizes the contribution of this paper.

II. MATERIALS AND METHODS

A. Pareto Analysis for Software Engineering in Literature

In Software Development Life Cycle (SDLC), one of the most frequently researched processes is how to improve the software process. M. Iqbal and M. Rizwan in 2009 to do research for the existing activities in SDLC waterfall model to be efficient. The hope is to get 80% productivity easily, by reducing software development efforts so that the end result improves the performance of the software process. In this study, the idea of 80/20 rule (Pareto principle) is applied in software engineering process model [3].

Observations from the above research are to apply 80/20 rule to 144 waterfall model activity, by doing the questionnaire. To improve the better software process model, Function Point Analysis (FPA) and Cost Constructive Model (COCOMO) are applied in order to get the results in standard form as evidence.

The result is 85 activities to focus on to get 70% to 80% of total productivity through business reduction and improve model performance. There are 45 activities that can be ignored, omitted or delegated, as they only provide 20% productivity.

B. Agile Requirements Prioritization

The main purpose of identifying and defining requirements is to explain the need for the development of an information system function and as a guide in the architectural design system. Requirements are required by systems analyst because requirements identify what the system does and does

not do. In addition, the requirements can define what can change, what cannot change and what must change. Further, requirements are used by systems analyst as a starting point if the built system can no longer be controlled. Finally, it is important to get good clarity and definition of requirements that can be used as a guide in designing the system so that emerging problems can be solved efficiently[4].

The agile requirements prioritization approach differs from the traditional approach in terms of flexibility to requirements changes that are strongly supported by agile, while the traditional approach is rigid and impractical. The agile method allows users to change requirements when the system is tested, this is to minimize the risk of project failure due to increased user engagement.

Requirements prioritization on Agile software development is reflected in some product owner activity in Scrum such as sort items in product backlog to achieve goals, what Scrum team should do next, or optimize the value of the development team's work. All of the above talks about backlog prioritization.

C. Requirements Prioritization Techniques

There are many researchers who develop different requirements prioritization techniques depending on the needs of the users. Some of these techniques include Analytical Hierarchical Process (AHP), Hierarchy AHP, Ranking, Cumulative Voting or Hundred Dollar Test, Top Ten Requirements, Binary Search Tree, Numerical Assignment and many other new techniques [5]. However, there is still a lack of effective prioritization approaches, so there are many requirements prioritization techniques with their strengths and weaknesses. In addition, many techniques still fail to take into account all the factors that should be considered when performing priority requirements such as cost, value, risk, time to market, number of requirements and effect of non-functional requirements on functional requirements [6].

Grouping requirements prioritization techniques can be seen from several factors/aspects, namely:

1. Measurement Scale
2. Technique Categories
3. Prioritization Strategies

1. Measurement Scale

The use of prioritization is to put alternatives in order of importance. The sequence can be done by mapping the requirements in numbers or rankings. All these figures can best be interpreted with the relevance of Nominal, Ordinal, Interval and Ratio measurements [7].

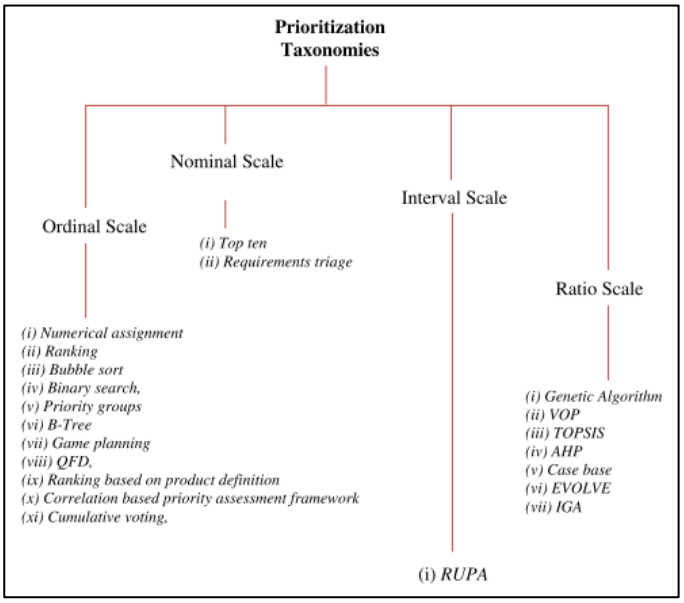

Fig 1. Prioritization taxonomies with their respective techniques [8].

2. Technique Categories

Requirements Prioritization Techniques can be split into two categories, absolute and relative priority [9]. for example giving all the requirements of a certain priority such as essential, conditional or optional. While the relative technique displays all the requirements based on their own priority values or in other words must be determined itself which needs are more important [10].

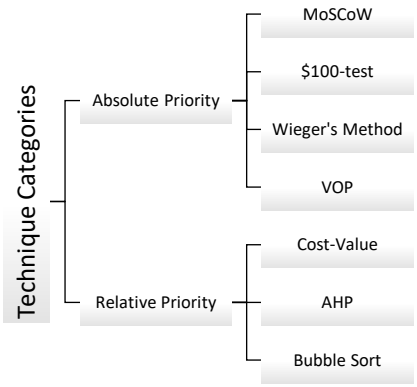

Fig 2. Requirements Prioritization Techniques Categories

3. Prioritization Strategies

Requirements prioritization strategies can be grouped into two parts: Scoping and Ordering [2]. In scoping requirements prioritization the first step is to categorize requirements such as: Must, Should, Could or Will not. Then only the category of Must and some Should be taken. While on

ordering requirements prioritization all requirements will be compared and then sorted according to the most important.

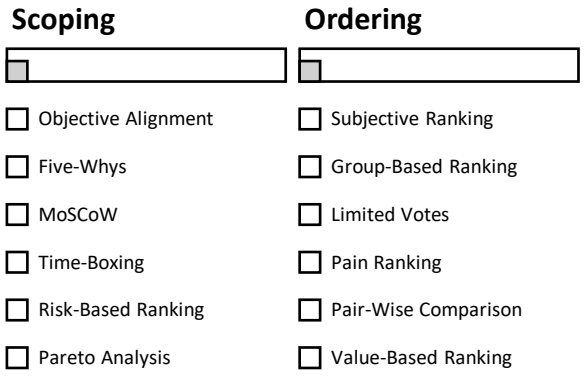

Fig 3. Prioritization Strategies

D. Pareto Analysis

Pareto analysis is a statistical technique in decision making to select a small number of tasks that produce a significant overall effect. Using the Pareto principle (also known as the 80/20 rule) in the form of the idea that by doing 20% of work can produce 80% of profits to do the whole job[11].

The purpose of Pareto analysis is to identify the most important requirements using Pareto analysis and build a Pareto chart to visualize the most important identification requirements. Pareto analysis is also part of the gap analysis to determine which requirements or needs will solve a major problem. The use of scale to improve conformity with each of the high priority features provided. While Pareto charts visualize the basic needs and how to deal with organizational problems.

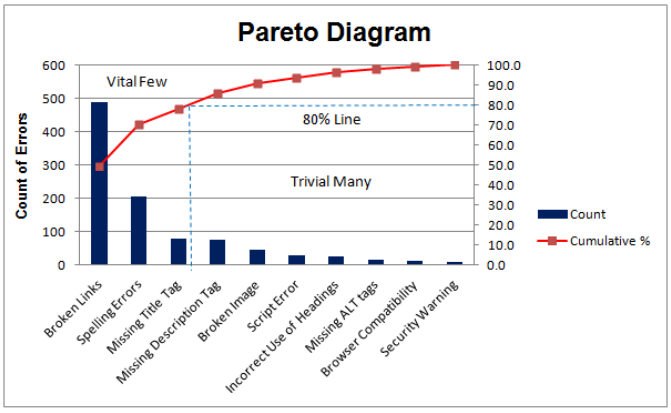

Fig 4. Pareto Analysis Diagram [12]

E. The Requirements Prioritization Process Model

Requirements prioritization is a series of steps. One of the existing models consists of 4 entities or phases, namely: initial project backlog, prioritized project backlog, sprint backlog, and implemented requirements. Initial project backlog contains all requirements at the beginning of the project. After applying the requirements prioritization technique, priority is associated with all the requirements in the prioritized project backlog. A small number of high priority requirements in the project backlog will be forwarded into the sprint backlog. The status of the requirements will change to 'implemented' after completing the iteration. Requirements in the sprint backlog that are not applied for a reason will be returned to the prioritized project backlog phase [13].

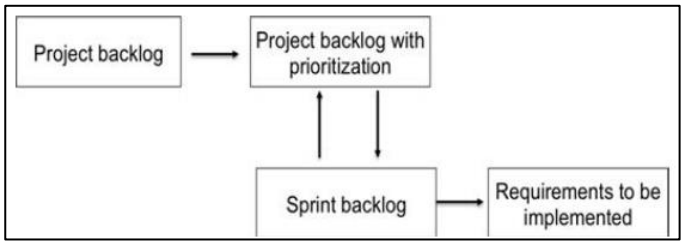

Fig 5. Prioritization process of model [13]

III. ANALYSIS OF THE RESULTS

Pareto analysis for requirements prioritization in agile development process is one promising alternative to be applied in the selection process requirements. A suitable project characteristic for applying Pareto analysis in performing priority requirements is to emphasize efficiency with minimum time duration, minority input, and multi-criteria by considering correlations of different criteria.

As for the steps to identify the main principles in using pareto analysis [12]:

1. Create a vertical bar chart with x-axis (cause) and y-axis (count the number of occurrences).
2. Sort the number of occurrences from the largest to the smallest.
3. Calculate the cumulative number of each cause, from the highest to the lowest.
4. Calculate the cumulative percentage amount for each cause in the order {Individual Cause Count} / {Total Cause Count} * 100.
5. Create a second y-axis axis with a decreasing percentage with a rise of 10 from 100% to 0%.
6. Plot the cumulative percentage of each cause on the x axis.
7. Combine points to form curves.
8. Draw a line at 80% on the y-axis parallel to the x-axis. Then form the line at the intersection with the curve on the x axis. The point on the x-axis separates the important causes on the left from less important causes on the right side.

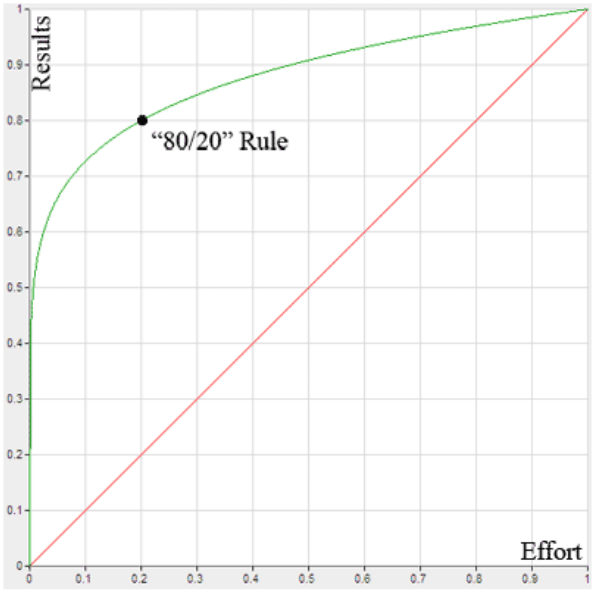

Fig 6. Pareto Chart (Source: betterexplained.com)

Under the Pareto principle does not mean only 80% of the work required. The remainder of the work is still needed to be carried out. Thus, the Pareto principle is an observation, not a natural law. The best quality must be generated from 100%, but when first focused on the critical 20% it will save time. In essence, the Pareto principle is to see the activity that produces the most results and give the right attention.

IV. CONCLUSION

Currently, many projects are showing 80/20 in results versus cost distribution. This is often associated with Pareto observation and is called the law of 80/20 [3]. Pareto analysis is a statistical technique in the field of economics which when applied to the requirement prioritization gives more definite value. Take some steps to identify the main principles in using Pareto analysis to produce a Pareto chart. In agile requirements prioritization, creating a Pareto chart to define values that help identify the most important requirements to be given more focus to get more optimal results

REFERENCES

[1] D. Port and T. Bui, "Simulating mixed agile and plan-based requirements prioritization strategies: Proof-of-concept and practical implications," *Eur. J. Inf. Syst.*, vol. 18, no. 4, pp. 317–331, 2009.

[2] M. Schedlbauer, "Requirements Prioritization," *Cathris Group*, 2013. [Online]. Available: http://is3500.weebly.com/uploads/8/6/5/9/8659576/requirements_prioritization_v1.01.pdf.

[3] M. Iqbal and M. Rizwan, "Application of 80/20 rule in software engineering Waterfall Model," *2009 Int. Conf. Inf. Commun. Technol. ICICT 2009*, no. Author 1, pp. 223–228, 2009.

[4] G. Hoff, A. Fruhling, and K. Ward, "Requirement prioritization decision factors for agile development environments," *14th Am. Conf. Inf. Syst. AMCIS 2008*, vol. 5, pp. 3298–3308, 2008.

[5] M. I. Babar, M. Ramzan, and S. a. K. Ghayyur, "Challenges and future trends in software requirements prioritization," *Int. Conf. Comput. Networks Inf. Technol.*, no. August, pp. 319–324, 2011.

[6] A. R. Asghar, S. N. Bhatti, A. Tabassum, Z. Sultan, and R. Abbas, "Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 12, pp. 300–306, 2016.

[7] P. Voola and a. V. Babu, "Comparison of requirements prioritization techniques employing different scales of measurement," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 4, p. 1, 2013.

[8] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. R. Mahrin, "A systematic literature review of software requirements prioritization research," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014.

[9] K. Logue and K. McDaid, "Handling uncertainty in agile requirement prioritization and scheduling using statistical simulation," *Proc. - Agil. 2008 Conf.*, pp. 73–82, 2008.

[10] L. Karlsson, T. Thelin, B. Regnell, P. Berander, and C. Wohlin, "Pair-wise comparisons versus planning game partitioning-experiments on requirements prioritisation techniques," *Empir. Softw. Eng.*, vol. 12, no. 1, pp. 3–33, 2007.

[11] International Software Testing Qualifications Board, "Standard glossary of terms used in software testing," pp. 1–31, 2007.

[12] B. D. Haughey, "Pareto Analysis Step by Step," *Analysis*, pp. 1–2.

[13] M. Alkandari and A. Al-Shammeri, "Enhancing the Process of Requirements Prioritization in Agile Software Development - A Proposed Model," *J. Softw.*, vol. 12, no. 6, pp. 439–453, 2017.