***Large Scale Requirement Engineering***

***Reflective Report***

Zhipeng Dong

dzp0713@gmail.com

BTH University

Sweden, Karlskrona

9208111295

**1. Article Selection**

**1.1 First Article**

**Article information:** Hatton S. Choosing the right prioritisation method[C]//Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on. IEEE, 2008: 517-526.

**Article name:** Choosing the right prioritisation method.

**Motivation for selection:**

1. Requirements priority division is a very important part of requirement engineering domain. This article is an article about the priority of requirements, which helps us understand and learn how to prioritize requirements. This is one of reason why we chose this article.

2. In this article, the author mentions several methods about requirements priority division. The author detailed describe the principle of each method and make a comparison of each method. The author also explains the advantages and disadvantages of each method. This is one of reason why we chose this article.

3. In this paper, the author teaches us what is effective requirement priority ranking. We can use the indicators defined by the author to verify that our priority ranking is valid or not. This is also a reason why we chose this article.

**1.2 Second Article**

**Article information:** och Dag, J. N., Regnell, B., Gervasi, V., & Brinkkemper, S. (2005). A linguistic-engineering approach to large-scale requirements management. IEEE software, 22(1), 32-39.

**Article name:** A linguistic-engineering approach to large-scale requirements management.

**Motivation for selection:**

1. With the rapid development of the software industry, The software industry's market competition has become more and more intense. In order to find more market opportunities, software organizations or companies must continuously elicit new requirements and reevaluate old ones as market needs evolve. But the requirements in the management process will encounter a lot of hindrances. In order to overcome these hindrances, we need to use some methods or techniques to eliminate these hindrances. Linguistic-engineering techniques is a good approach to help us eliminate these hindrances. And this article is about linguistic-engineering approach. So I select this article.

2. This article discusses market-driven requirements management. We know that market-driven is a very important part in software product development process. In order to ensure our products can be developed to be successful, we must fully consider the market requirements. In this article, the author divided market-driven requirements into two parts. They are customer wishes and product requirements. And the author provided a detailed analysis of these two parts. This is one of reason why we chose this article.

3. In this article, the author show with an open source tool how you could use linguistic approach to manage requirements in your own project. We can imitate the author’s approach to use this approach. Because of this reason, learning linguistic approach becomes simple. This is also a reason why we chose this article.

**2. Implementation Plan**

**2.1 The Plan for First Method**

In this article, the author mentions several methods about requirements priority division and the author detailed describe the principle of each method.

The proposed prioritization methods:

1. Simple Ranking

2. MoSCoW

3. AHP

4. Hundred Dollar Method

I will use MoSCoW method to implement my own project. There are 208 requirements in our project, and these requirements are belong to different types.

The first step is classify all the 208 requirements into several different types.

The second step is classify the different types of requirements into four level. They are Must have level, Should have level, Could have level and Won’t have level.

The third step is start do the release planning-focus on the Must have requirements.

The fourth step is continue do the release planning-focus on the Should have requirements.

The fifth step is continue do the release planning-focus on the Could have requirements.

The sixth step is continue do the release planning-focus on the Won’t have requirements.

The last step is implement our release planning.

**2.2 The Plan for Second Method**

In this article, the author proposed a large-scale-requirements-management method named linguistic approach. I will use linguistic approach to implement my own project. There are 208 requirements in our project. At first, I will classify these requirements into several different types. And then use linguistic approach to achieve the management of each type of requirements.

The linguistic approach will be divided into four steps:

Step 1. Flattened. Merging the label and descriptions fields of the requirements.

Step 2. Tokenized. Removal of capitals, punctuation, brackets, and so on.

Step 3. Stemmed. Remove affixes and other lexical components.

Step 4. Stop words removed. Remove common terms that are unlikely to contribute to an appropriate similarity measure.

When we completed these four steps, the amount of requirements will be reduced, and the requirements management work will become more simple.

**3. Execution**

**3.1 The Execution for First Method**

As we know that, our project include 208 requirements. At first, I download all the 208 requirements. Then I communicated with the stakeholders to deep understand all the requirements. Then I divided the 208 requirements into 10 different types. They are:

1. User interface

2. Upload and download files

3. Communicate with other users

4. Log in and log out

5. Notify

6. Manage personal information

7. Search function

8. Evaluation function

9. Course management

10. Others.

Next, according to our understanding and the roadmaps, we give labels to each requirement for four “have” levels. The four “have” levels are: Must, Should, Could and Won’t. For example, the user interface requirement and notify requirement.

No.1 User Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of RQ | Name | Must have | Should have | Could have | Won’t have |
| No.11 | Personal initial pages | **√** | --------------- | --------------- | --------------- |
| No.12 | Course start pages | **√** | --------------- | --------------- | --------------- |
| No.21 | Course information | **√** | --------------- | --------------- | --------------- |
| No.39 | Course start pages component | **√** | --------------- | --------------- | --------------- |
| No.40 | Personal initial pages component | **√** | --------------- | --------------- | --------------- |
| No.64 | Course links | **√** | --------------- | --------------- | --------------- |
| No.19 | The language for user interface | **---------------** | **√** | --------------- | --------------- |
| No.23 | Discussion forum function | **---------------** | **√** | --------------- | --------------- |
| No.144 | The layout design of course page | --------------- | **√** | --------------- | --------------- |
| No.145 | The user interface need easy to use | --------------- | **√** | -------------- | --------------- |
| No.207 | System compatibility processing | --------------- | **√** | --------------- | --------------- |
| No.66 | The calendar function | --------------- | --------------- | **√** | --------------- |
| No.68 | Show the contents of message | --------------- | --------------- | **√** | --------------- |
| No.164 | Support for custom layouts | --------------- | --------------- | **√** | --------------- |

No.5 Notify

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of RQ | Name | Must have | Should have | Could have | Won’t have |
| No.87 | Notification alerts | **√** | --------------- | --------------- | ------------- |
| No.112 | Mail generated automatically | **√** | --------------- | --------------- | ------------- |
| No.113 | Unread notifications | **√** | --------------- | --------------- | ------------- |
| No.188 | E-mail notifications | **√** | --------------- | --------------- | ------------- |
| No.196 | Receiving info | **√** | --------------- | --------------- | ------------- |
| No.203 | Grade notifications | **√** | --------------- | --------------- | ------------- |
| No.204 | Exam result notifications | **√** | --------------- | --------------- | ------------- |
| No.124 | Push notifications | ------------ | --------------- | **√** | ------------- |
| No.181 | Latest change notifications | ------------ | --------------- | **√** | ------------- |
| No.85 | User message notifications | ------------ | --------------- | --------------- | **√** |
| No.120 | Notifications | ------------ | --------------- | --------------- | **√** |
| No.190 | E-mail notifications | ------------ | --------------- | --------------- | **√** |

When we do the release planning, we will do the follow the order:

First order—> Must have : requirement number: #11, #12, #21, #39, #40, #64, #87, #112, #113, #188, #196, #203, #204.

Second order—> Should have: requirement number: #19, #23, #144, #145, #207.

Third order—> Could have: requirement number: #66, #68, #164, #124, #181.

Fourth order—> Won’t have: requirement number: #85, #120, #190.

**3.2 The Execution for Second Method**

Our project include 208 requirements. At first, I download all the 208 requirements. I not only download the requirements’ name but also download the description of these requirements. Then I made an excel table for these requirements and their description. And use linguistic approach to implement my own project.

Proof of Concept:

|  |  |  |
| --- | --- | --- |
| Number of RQ | Name | Requirements description |
| No.169 | Publish Assignment | As a teacher, I want to publish assignment so that my students can submit their results. |
| No.170 | Edit comment for the assignment published | As a teacher, I want to edit comment for the assignment I published so that my student can have more understanding about the assignment. |
| No.166 | Uploading file | As a teacher, I want to upload some files to the system so that my students can download them. |
| No.140 | Publish information function | As a manager, I need to publish some information, such like web-site maintenance information, activity notification... |
| No.184 | Download File | As a new student, I shall be able to download course materials. |
| No.183 | Publish the registrations | As a teacher, I want to publish the registrations for my students to register. Registrations can be used for team selection, time selection for presentation and so on. |
| No. 204 | Exam Notification | The system shall also notify students of exam information including date, place and time. |
| No.137 | Log in time reset | As a teacher assistant i would like the system to remember me for long time and not ask for log in information often so that i am not disturbed when i use the system for long periods of time. |
| No.195 | Login information representation | As an old engineer student, I would like to see how much time I spent on the system and how many times I logged in the system. |
| No.189 | Course Schedule | As a new student, I shall be able to get the course schedule by searching the course code. |
| ... | … | … |

Follow the four steps of linguistic approach:

Step 1. Flattened. Merging the label and descriptions fields of the requirements.

Step 2. Tokenized. Removal of capitals, punctuation, brackets, and so on.

Step 3. Stemmed. Remove affixes and other lexical components.

Step 4. Stop words removed. Remove common terms that are unlikely to contribute to an appropriate similarity measure.

The excel table become:

New Proof of Concept:

|  |  |  |
| --- | --- | --- |
| Number of RQ | Name | Requirements description |
| No.169 && No.140 &&  No.183 | publish information function | publish some information, such like web-site maintenance information, activity notification, assignment and registrations  registrations can be used for team selection, time selection for presentation and so on |
| No.170 | edit comment for the assignment published | as a teacher, I want to edit comment for the assignment I published so that my student can have more understanding about the assignment. |
| No.166 && No.184 | uploading and download file | teacher upload some files  students can download files |
| No.137 && No.195 | log in time reset  and information representation | student would like to see how much time they spent on the system and how many times they logged in the system  the system can remember the users’ information. |
| No.184 | Download File | As a new student, I shall be able to download course materials. |
| No.183 | Publish the registrations | As a teacher, I want to publish the registrations for my students to register. Registrations can be used for team selection, time selection for presentation and so on. |
| No. 204 | exam Notification | new student shall be able to get the course schedule by searching the course code. |
| ... | … | … |

**4. Lessons Learned**

**4.1 The lessons learned from the first article**

From this article, I learned a lot of method for requirement prioritization. The author introduced four main requirement prioritization methods. They are Simple ranking method, MoSCoW method, AHP method and 100$ method. The author not only describes the principle of each method but also describes the advantages and disadvantages of each method. So we can chose them according to the real project.

Because I selected MoSCoW method to implement my own project. So I put the main energy into the analysis of MoSCoW method. MoSCoW method is a numerical requirements prioritization method. It can give a hierarchical order of preference for groups of requirements. The goal of MoScoW is to deliver the best benefits for the project. It is very important for agile development project management. MoSCoW method has many benefits that suitable for requirement prioritization, such like easy to learn and perform. MoSCoW method can easy to handle the new requirements, and easy to manage the old requirements.

For large scale requirement engineering, because the number of requirements is large, so we can use MoSCoW method to divide the large number of requirements into four different levels. Then we will deal with the requirements from the first level to the last level.

**4.2 The lessons learned from the second article**

From this article, I learned a new method for large scale requirements management. It is a linguistic engineering approach. When we apply this method to our actual project, we find it can help us save time. And I think it is also a good method to help us understanding the complex requirements.

As we know that, different stakeholders will have different requirements. But these requirements may be repeated or meaningless. So If we do not label, merge, sort these requirements, rather than deal with them one by one. We will take up a lot of money and time, and we can not make sure our work is meaningful.

And from this article, I learned market-driven requirements management is a very important part in large scale requirement management. In order to ensure our products can be developed to be successful, we must fully consider the market requirements. In this article, the author divided market-driven requirements into customer wishes and product requirement. We have a deep understanding about the market-driven requirements.

**5. Reflection**

**5.1 The reflection from the first article**

From this article, I learned four requirement prioritization methods. Each method has its own advantages and disadvantages. And each of them has their own suitable situation. We can not only be limited to the use of one method when we prioritize the requirements. We have to think all the aspects and find a most suitable method.

In order to get more in-depth study of this method, I read some of the relevant literatures.

From literature[5], we known that, we need to give the four prioritization levels some flexibility. Because if the project only have “Should have” level or “Must have” level, the project will become more difficult to prioritizing. We can not clearly know which requirement need to be focus on consideration.

From other literatures, we know that consider the criterial of the MoSCoW method is important. There are five aspects we need to consider. They are:

1. Business value.

2. Risk management.

3. Challenges in the implementation process.

4. Relationship between different requirements.

5. Stakeholders.

Only make a good balance of these five criterial, the using of MoSCoW method will become more effective.

**5.2 The reflection from the second article**

From this article, I learned a new method for deal with large scale requirements. But to be honest, because of my poor experience and knowledge, I think use this method is very difficult for me.

Because I have never heard of this method before, in order to get more in-depth study of this method, I read some of the relevant literatures. From literature[3][4], I get a deep understanding about the linguistic engineering approach, but I think it is still not enough.

As we know that, our project include 208 requirements, to be honest, it is not a large project. If the number of requirements is more than 1000/10000, how to use this method? To deal with 208 requirements, we made up a excel table, to think about these requirements one by one, we use 5 days to finished that work. When the requirements becomes more and more larger, how to deal with them?

**6. Reference**

[1] Hatton S. Choosing the right prioritisation method[C]//Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on. IEEE, 2008: 517-526.

[2] och Dag J N, Regnell B, Gervasi V, et al. A linguistic-engineering approach to large-scale requirements management[J]. IEEE software, 2005, 22(1): 32-39.

[3] Bolinger D. Linguistic science and linguistic engineering[J]. Word, 1960, 16(3): 374-391.

[4] Baldwin B, Doran C, Reynar J, et al. EAGLE: An extensible architecture for general linguistic engineering[C]//Computer-Assisted Information Searching on Internet.1997: 271-283.

[5] Waters K. Prioritization using moscow[J]. Agile Planning, 2009, 12.

[6] Vestola M. A comparison of nine basic techniques for requirements prioritization[J]. Helsinki University of Technology, 2010.

[7] Prindle N H, Mendenhall F T, Boak D M, et al. The Second Iteration of the Systems Prioritization Method: A Systems Prioritization and Decision-Aiding Tool for the Waste Isolation Pilot Plant. Volume I: Synopsis of Method and Results[J]. SAND95-2017. Albuquerque, N. Mex.: Sandia National Laboratories, 1996.

[8] Chu H C, Hwang G J, Huang S X, et al. A knowledge engineering approach to developing e-libraries for mobile learning[J]. The Electronic Library, 2008, 26(3): 303-317.