***Large Scale Requirement Engineering***

***Release Planning Report***

Mingda Chen

daclaude7@gmail.com

BTH University

Sweden, Karlskrona

9408112879

***Abstract -* This article is the report of release planning assignment in Large Scale Requirement Enigineering course, this release planning work is done and this report is written by Mingda Chen individually.**

**1. Brief Product Description**

In this part, I will give a briefly product description of this online learning platform system.

**1.1 Description of Product**

This product is a Online learning platform system for Sweden University which is similar with the famous system Itslearning. This system provide a efficiency way for teacher and teacher assistant to manage the course, provide a efficiency way for students to manage his status on the course, and provide a efficiency way for student administrators to manage the student information. This system requires the stable Internet Connection and requires main server, data center and several different hardware.

**1.2 Description of the Main Features**

In this system, there will be several main module/high-level features inside and I will describe them briefly.

**Upload and Download File:** The system shall be able to allow user to upload files or download files from the system.

**Communication:** The system shall be able to allow different user to send message to each other and reply the information. Also the system shall be able to allow user to build discussion forum, everyone can send personal opinion in discussion forum.

**Log in and Log out:** User need to log in the system in order to use it and also need to log out when they want to quit the system.

**Viewing Information:** The system shall allow user to view different information(personal information, part of other's personal information, course information and so on).

**Editing Personal Information:** The system shall be able to allow user to edit his/her personal information.

**Registration:** The system shall be able to allow user to register the course they want to have, the presentation date they want to attend and the examination for the course.

**Course Management:** The system shall be able to allow teacher to manage the course.

**Notification:** The system shall be able to send notification information to the user to notify them different change of the information they care for.

**Evaluation:** The system shall be able to allow teacher to evaluate student's assignment result and examination result, evaluation means based on the good or the bad of their result and give a grade and feedback for it.

**Search:** The system shall be able to allow user to search some information based on the relative keywords.

**2. Company Size and Investment in Product**

In this part, I will introduce the information about our company and also our Investment in this product.

**2.1 Company Size**

Our company is a small size software company, our company's domain is the information management system. We have developed several systems such as student information management system for university, case information management system for law office and ship information management system for terminal management office so we have lots of experience on this aspect. And our main target is European customers. Our company's development process mainly follow the agile process especially the Scrum software development methodology.

**2.2 Team Size**

Since our company will use Scrum software development methodology to develop this online learning platform system, we will assign 4 Scrum teams to work on this product, there will be 6 people in each team. They will involved in the product at the beginning of this project, and one team will leave after the first release, because we plan to develop a working product as soon as possible which means after first release, this product can reach a usable state. So the first release is important and busy, that's the reason why we assign 4 teams to work on it. In the later release, we plan to add some detail feature into the first version so the work will be little bit easier than the first release. To put rational allocation of human resources into consideration, we will assign one team to other product. However, if something wrong happened in the later release, we will adjust it again. Since continuously validation or verification is good way to reduce the risk. And these teams will work in our company's building together.

**2.3 Resource**

**Budget:**  8,000,000 Kr

**Equipment:**  1. Enough Computers with programming software.

2. Stable Internet Connection(Fiber optic network 100Mbps).

3. Dedicated printers and scanners.

4. Dedicated cloud server for sharing different results or ideas.

5. Other common development hardware.

**Environment:** 1. Dedicated working office for each teams.

2. Dedicated meeting room for every meeting in Scrum methodology.

**Staging Environment:** 1. Stable and security staging database server to test the system.

2. Stable and security staging server to test the system.

**3. Roadmap**

In this part, I will introduce the Roadmap of this product and also the different viewpoints, and the importance of these viewpoints with the prioritization.

**3.1 Roadmap View**

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | Stage One | Stage Two | Stage Three |
| Date | Jan 2017 to Oct 2017 | Nov 2017 to Feb 2018 | Mar 2018 to June 2018 |
| Focused Theme/High-level features | 1. Upload and Download File  2. Communication  3. Course Management  4. View Information  5. Relative Quality Requirement  6. Relative User Interface | 1. Edit Personal Information  2. Register  3. Log in and out  4. Notification  5. Relative Quality Requirement  6. Relative User Interface | 1. Search  2. Evaluation  3. Others  4. Relative Quality Requirement  5. Relative User Interface |
| Satisfied Viewpoint | Old Engineer Student  Teacher  New Student  Teacher Assistant | Old Engineer Student  Teacher  New Student  Teacher Assistant  Student Administrator | Teacher  Teacher Assistant |
| Metrics | 1. The release of first version  2. Number of completed item in release backlog which is relative with the themes | Number of completed item in release backlog | Number of completed item in release backlog |

This is the feature-based product roadmap for the online learning platform system, there will be three stages in the future of this product, first stages will last for 10 months that was because in the first stages we need 6 months to develop the first version of this system and need 4 months to focus on the themes. Second themes and Third themes need 4 months to focus on specify themes. There will be lots of requirements which is relative with the themes, but you can find the brief description of these themes in the part one, and quality requirement and user interface will be implemented based on the different themes. And other requirements means that several requirements which I think is not very important and maybe out of the product scope. We will discuss more this on part six. This roadmap was designed based on the analysis of our company current status and also the importance of different viewpoints. Which means the division of theme is based on my understanding of different viewpoints also the result of prioritization of these viewpoints.

**3.2 Detailed description of different viewpoints**

Based on the information of discussion forum on Itslearning, I will summary different viewpoints and discuss their importance.

**Teacher assistant:** Teacher assistant think the most important features should be uploading file to the system for him to evaluate or for students to read, and Timely notification and easy communication. And he want the students cannot grade themselves which will be divide into the limitation of View information theme. Also he think that the system should be able to identify the type of users(teacher, student and so on).

Teacher assistant is important since every course may need one or two teacher assistant, that means there will be many teacher assistant users for this system. As a teacher assistant, they will be able to use several teacher relative features on the system. Their satisfaction degree need to put into our consideration.

**Teacher:** Teacher think that the most important features should be uploading file to the system and the system need to help teacher to collect student's assignment result. Teacher also want to evaluate the result and give the feedback. And easy communication is required in teacher's viewpoint. As a teacher, the system should allow teacher to manage the course, also the system should allow teacher to add new students into course participants list. Efficiency and convenient are the keywords.

Teacher is the most important part since there must be one or two teachers for a course, and lots of features and functions in this system are designed for teacher such as evaluate the assignment, manage the course and so on. Their viewpoints and requirements or expectation must be involved in our requirement engineering work before the development.

**Student Administrator:** Student Administrator think that he should be able to modify the student information such as telephone number, E-mail address and so on. And after teacher give grade on courses for each students, the system should help student administrator to collector the grades and calculate the total credits. Timely notification and easy communication are needed in this viewpoints too.

Student Administrator which is a jobs to manage student's information, as the large number of students, his work actually is hard and busy. Student administrators are important to our system, since several feature are design for them. We need to consider their satisfaction.

**New Student:** New student think that the system should send the log in needed information to new student by E-mail, and after the first log in, the system should identify this student as the activated user. And also the system should allow new student to register a course, after registration, new student need to view the course relative information such as course description, course schedule and so on. Timely notification is required here. And also convenient and efficiency are keywords.

New Students are very important to our system but less than the old student and the reason is simple, here we define the new students as a user that need several help when he firstly use the system. And the new students' satisfaction can be seen as the ease-to-learn and ease-to-use for the system. After the new students can easily use the system and start his university life, we will redefine them as old students.

**Old Student:** Old student think that he need be able to view the information easily and modify his personal information easily. And the system should be able to let old student reply the feedback of assignment. After the old student logging in the system, the system need to show the total time he spent on the system. Finally the old student want to cancel the registration of the outdated course. Timely notification is required in this viewpoints. Reliability, Usability, Security and Functionality are keywords here.

Old Students are the most important user for our system. That was because that the main user type are old students with the largest number. Also there will be lots of features need the involvement of old students. And their requirements will mainly influence the design and development of this system. Also after the release of this product, there will be lots of old student use this system every day. So we need to consider their requirements and expectation and satisfaction.

**Developer:** Developer think that there should be a database to store the data information for the system. And the system need to perform enough compatibility to different platform. Also the system need to manage different authority for different user type. And Security and stability are keywords in this viewpoint.

Developer is the part of viewpoint in our system, who will mainly be response for the development. Based on the professional knowledge and his personal experience, he will have some opinion on this products. And this opinion will be reflected in the final product. So we need to consider it.

**3.3 Prioritization of viewpoints**

First of all , I will show the prioritization result of these viewpoints. I divide them into 4 levels and Level 1 is the most important level.

Level 1: Old Student, Teacher

Level 2: New Student, Teacher Assistant

Level 3: Student Administrator

Level 4: Developer

And the prioritization of these viewpoints is based on following different metrics.

**Number of users:** This metrics means how many users of this type will use our system. It's hard to accurate to the detail number. But it's not hard to compare different types. And following is the prioritization of this metrics.

Old Student > New Student > Teacher / Teacher Assistant > Student Administrator >Developer

**Using Frequency:** This metrics means the frequency of using this system for a type of user. And following is my prioritization of this metrics

Old Student > Teacher / Teacher Assistant > Student Administrator >New Student > Developer

**Relative Feature Scope:** This metrics means the feature scope in our system that relate to the specific type of user. In other words, the number of features which will be used by a specific type of user. And following is my prioritization of this metrics

Teacher > Teacher Assistant > Old Student > New Student > Student Administrator > Developer

Above all, based on these metrics, I define the prioritization of these viewpoints, and the result was shown at the beginning of this part. And the building of roadmap is following the prioritization of the viewpoints.

**4. Releases**

In this part, I will describe some basic information of the releases I decide to plan, and I will discuss how I map these releases to the roadmap.

**4.1 Number of releases and its dates**

I decide to plan 4 releases for this product.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Release | One | Two | Three | Four |
| Dates | Jan 2017 to June 2017 | July 2017 to Oct 2017 | Nov 2017 to Feb 2018 | Mar 2018 to June 2018 |

If the project is fully follow the release plan, then the project will start at Jan 2017 and end at June 2018

**4.2 Relationship between releases and roadmap**

I will map the four releases to the three stages in roadmap.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stage | Stage One | | Stage Two | Stage Three |
| Release | Release One | Release Two | Release Three | Release Four |
| Date | Jan 2017 to June 2017 | July 2017 to Oct 2017 | Nov 2017 to Feb 2018 | Mar 2018 to June 2018 |
| Goal | Release the first version of this system which is already a working product.(The main requirements for each theme are completed) | Focus on the requirements which is relate to following theme to improve the system features:  1. Upload and Download File  2. Communication  3. Course Management  4. View Information | Focus on the requirements which is relate to following theme to improve the system features:  1. Edit Personal Information  2. Register  3. Log in and out  4. Notification | Focus on the requirements which is relate to following theme to improve the system features:  1. Search  2. Evaluation  3. Others |

After the building of roadmap, I plan the release based on the roadmap, as the Goal part shows, the primary goal is to develop a working product as soon as possible, and based on the roadmap, I plan the other three releases to implement other requirements based on the focus theme.

**5. Releases Strategy**

In this part, I will describe the releases strategy and my requirements prioritization methods and discuss my reason to choose them.

**5.1 Description of Strategy and Motivation**

**Theme-based Product Release Planning:** Theme-based product release planning strategy is a product management methodology that focus on to create more value for the project in the release planning aspect. The author of this strategy think that if we put several sets of features in same release or different release, it will bring more value for the product. And also several relative sets of features can be seen as a theme for the product. Theme is a meta-functionality of the release. Theme also can be defined as high-level feature. So based on the research on the theme, author propose the theme-based product release planning strategy. This strategy mainly focus on the value of one theme and the dependence or relationship between different feature. When we talk to the dependency of features. Theme-based product release planning divide them into Direct Dependency and Indirect Dependency.

Direct Dependency including weak precedence, coupling, synergy.

Weak Precedence: Feature B cannot be released before feature A.

Coupling: Feature A should be released with feature B.

Synergy: Feature A and feature B have a synergy dependency.

Indirect Dependency including strict precedence, NAND.

Strict Precedence: Feature A must be released before feature B.

NAND: Feature A should not be released with feature B.

And this theme-based product release planning method is based on the EVOLVE Ⅱand improve it by adding three new steps(Graph Transformation, Clustering, and Theme-based Plan Generation).

Graph Transformation: Based on the interdependency of different features, this step will draw a graph to show the features and their interdependency, also we can see the weight of the edge based on the thickness of the edges. This step mainly provide a visualization view.

Clustering: Use CW algorithm to cluster the different feature together based on their weight of the edge.

Theme-based Plan Generation: This step is the most important step in the theme-based product release planning method. This step requires the dynamic management of theme when we are making release planning decision. That means we need several constraints between the clusters to support the division of themes in the planning process. In other words, when we assign different themes to a release, we need to consider about the value of customer and the relationship with other themes.

There is also some disadvantages of this theme-based product release planning.

a. This methods rely on the dependency of different cluster to make cluster into theme but ignore the name of features and text description of features.

b. In the Graph Transformation part, the graph cannot distinguish the host or the parasite in a dependency relationship.

c. Too much constraints in Theme-based Plan Generation will lead lots of problem.

d. Graph Transformation and Clustering cost lots of time and resource.

**Motivation:** I will describe the reason why I choose theme-based product release planning.

a. Theme-based product release planning is suitable for some small and simple system, and this method is easy to learn and use.

b. After the review and research on all the requirements, I find out that these requirements can easily divide into several themes, which means it will reduce lots of work if I use theme-based product release planning.

c. After the review and research on all the requirements, I find out that there is not many dependencies between requirements and mainly these dependencies is between the requirements for same high-level feature(theme), that means I can reduce the dependency and constraints when I using this methods to reduce the damage from constraint explosion.

d. Theme-based product release planning is a feature-oriented method, and my roadmap is feature-oriented too, so they can map each other well in some aspect.

e. In the article which describe this methods, I agree on lots of author's opinion of release planning, for example, the definition about theme, the factors need to be considered(feature, dependency, constraints) and so on. Especially the idea that release planning is not just a feature division activity but a activity to dig out more value.

**5.2 Description of Requirements Prioritization Methods and Motivation**

**MoScoW Method:** MoScoW is a numerical requirements prioritization methods, whose goal is to deliver the best and faster business benefits for the project. And each requirement will be put into one of the four groups which are Must have, Should have, Could have and Won't have.

Must have: The requirements in Must have are very critical to the product and project plan, which will directly influence the success or failed of this project. The development of these Must have requirements will directly give business value to the project. And Must can be also defined as the "Minimum Usable Subset", which means after the development of these Must have requirements, we can get a already working product.

Should have: The requirements in Should have are important as well as the Must have, which means if Should have requirements been developed, it will be good for the project. But the difference is that Should have requirements may not be very critical in current stage, in other words, they need to be implemented but can be released in later releases.

Could have: The requirements in Could have are not very necessary in the requirements database, but the implementation of these requirements can improve the user satisfaction and improve the quality of product. So we could implement them when we have enough recourse and time.

Won't have: The requirements in Won't have are least-critical items in the requirements database, which will not be planned into release.

**Motivation:** I will describe the reason why I choose MoScoW method.

a. MoScoW method is easy to learn and easy to use.

b. In the MoScoW method, especially the Must have group is very suitable for me to decide which requirements need to put into first release since the goal of first release in my plan it to develop a working product as soon as possible.

c. MoScoW method is not suitable for the project with large requirement database. But for my project, there is only 208 requirements in the database and about 20 requirements per theme. So MoScoW is very suitable for my project.

d. MoScoW method is suitable for this situation: A team have several potential epics(high-level feature/theme) to divide in the release planning, they need to identify the MVP and MMF and other different level requirement. And this situation is similar for my project.

e. MoScoW method is a ordinal level technique, and that's what I expect in each theme.

**Kano Method:** Kano model is a useful tool for categorizing and prioritizing user requirements, which is also good to analysis the relationship between requirements and user satisfaction. Based on the different relationship between requirements and user satisfaction, this model divide them into five types, Basic Quality, Performance Quality, Excitement Quality, Neutral Quality, Reverse Quality.

Basic Quality: Basic quality means that the requirement must be developed in the product. which is the basic needs in the expectation of customers toward this product. Customer may not feel satisfy with the implementation of this needs but will definitely feel unsatisfied when this requirement is not been implemented.

Performance Quality: Performance quality also called unary demand. There is a strong relationship between requirements and user satisfaction. If the requirement in this group is met and performed well, them customer satisfaction will increase significantly. But if these requirements is not met or preformed badly, then customer satisfaction will reduce significantly.

Excitement Quality: Excitement Quality is the other hand of Basic Quality, the requirements in this group usually are out of the customer's expectation. The implementation of the requirements here will directly increase the user satisfaction, if not, that will not influence the user satisfaction.

Neutral Quality: The requirements in this group will not influence the user satisfaction.

Reverse Quality: The implementation of the requirements in this group will reduce the user satisfaction.

**Motivation:** I will describe the reason why I choose Kano model.

a. Kano model is easy to learn and easy to use.

b. Kano model analyze the relationship between user satisfaction and requirements, which can be used to fulfill the vacancy of MoScoW method.

c. Kano model is a nominal level technique, for the about 20 requirements in each requirements. I think this level is enough especially the division of Kano model is reasonable and useful for my release planning.

d. As I said in the motivation of MoScoW method, I eager to find a model to help be define the relationship between extent information and satisfaction information. Of course Kano model is such model.

**6. Execution of Release Planning**

In this part, I will describe my execution of the release planning.

**6.1 Steps to Prepare Requirements**

**Step1:** Download all the 208 requirements from GitHub with these information: Requirements Number, Requirements Name, Requirements Description and Requirements dependency. Then saved them in a Excel file, and read and understand them all to have a concept what should the system looks like. While reading, make some notes in Chinese to record the basic information of each requirement.

**Motivation:** There is no reason for this step, before dealing with the requirements, at least, I need to download them, at least read and understand them. And the information of requirements is needed. Number can help me to describe it easily, Name can help me have a quick understanding, Description is of course needed, and the dependency is the Number of requirements which is relate to this one. And the Chinese notes is because, Chinese is my native language, I can understand the meaning without thinking, so some Chinese notes will increase the efficiency of my future work.

**Step2:** Because the requirement database are built by different people, so there is lots of same or very similar requirements in the database. So the second step is to find out the duplicate requirements and meaningless requirements and put them into Duplicate list and Meaningless list(I will remain the requirements whose Number is smaller, and the dependency will transfer to the smaller number one). I will show the result of this step in 6.2.

**Motivation:** Duplicate requirements and meaningless requirements will directly influence the efficiency of my project and also duplicate requirements and meaningless requirements sometimes will make misunderstanding for the project, so I will remove them.

**Step3:** Based on the understanding of different requirements and the understanding of the system. Divide the several themes for the system and put the relative requirements into themes. I will show the result of this step in 6.2.

**Motivation:** Because my roadmap and my release planning methods will be feature-oriented. And the feature here I mean is the high-level feature, which can also be defined as themes. And the themes comes from the requirements database. Also there will be several requirements are quality requirements, User Interface requirements or others(which means no relative theme). I will divide them into different type and will consider them with the several main theme when I am making release planning.

**Step4:** Use Kano model and MoScoW method to prioritize the requirement per theme. And I will show the result of this step in 6.3.

**Motivation:** It's hard and little bit meaningless to prioritize all the requirements together, In my release planning, the most important thing is to find out the MVP of each theme, and these requirements will be put into first release for the already working product. And the other requirements will be put to later stage or release based on the focus theme in the roadmap. And the prioritization work of requirements is a important phase of requirement engineering, which will help the develop team to understand the different importance of requirements.

**6.2 Change to the requirements database**

I put the duplicate requirement into duplicate list, define the missed requirement, delete the meaningless requirement into meaningless list, make the division of these requirements based on the theme. And make some personal change.

**Duplicate: #**36(duplicate with #17), #43(duplicate with #25), **#**119&#121(duplicate with #69), #166&#185(duplicate with #118), #170(duplicate with #167), #175(duplicate with #172), #179(duplicate with #65), #187&#197(duplicate with #103), #205(duplicate with #186), #208(duplicate with #139),

**Missed:** #116

**Meaningless:** #95, #100

**Division:** I will describe the division of requirements based on theme.

Communication: #59, #67, #69, #108, #114, #117, #131, #146, #176, #186, #193, #199

Course Management: #5, #14, #15, #17, #20, #29, #45, #48, #50, #57, #60, #138, #140, #141

Information Edition: #16, #24, #25, #30, #33, #42, #44, #46, #52, #53, #54, #55, #70, #78, #80, #91, #99, #101, #126, #133, #134, #136, #147, #154, #156, #165, #167, #171, #173, #178, #180, #194, #201

Evaluation: #71, #96, #107, #125, #161, #172, #198

Log In&Out: #27, #61, #62, #63, #88, #89, #103, #104, #109, #137, #168

Notification: #85, #87, #112, #113, #120, #124, #181, #188, #190, #196, #203, #204

Registration: #65, #74, #75, #111, #135, #183, #192

Search: #77, #98, #189

Upload and Download File: #31, #72, #73, #76, #81, #82, #83, #84, #86, #118, #129, #130, #148, #149, #151, #152, #163, #169, #174, #184

View Information: #2, #4, #8, #28, #32, #34, #35, #47, #49, #51, #56, #58, #90, #97, #102, #122, #123, #132, #143, #155, #159, #160, #177, #182, #191, #195, #202

Other: #9, #18, #22, #37, #38, #92, #94, #105, #106, #110, #115, #139, #142, #150, #157, #158, #200, #206

Non Functional Requirements: #1, #3, #6, #7, #10, #13, #26, #41, #79, #93, #127, #128, #153, #162, #207, #209

User Interface: #11, #12, #19, #21, #23, #39, #40, #64, #66, #68, #144, #145, #164

**Personal Change:** Add a short notes for every requirements in Chinese to let me have quick understanding in future work.

**6.3 Description of Steps of Release Planning**

**Step1:** To follow the theme-based product release planning method, the first step should be graph transformation, but in my opinion, graph transformation is a tool to help the developer and product manager to have a visualization understanding about the different requirement and their interdependency especially the interdependency between different themes. But in my requirement database, the requirements description is clearly state, the interdependency is clearly signed and to be honest, there is less interdependency between different themes. And this work cost too much resource and time. Based on my cost-value analysis, I decided to skip this step.

**Step2:** To follow the theme-based product release planning method, the second step should be Clustering, and the work in Clustering is very similar with the step2 of my preparation work for the requirement, which is to together the requirements based on its different theme. Since I have already done the work, So I decide to skip this step.

**Step3:** To follow the theme-based product release planning method, the third step should be the generation of theme-based release plans. Based on the roadmap and the primary goal(develop a working product as soon as possible), I think about the number of releases and the date of these release. In this consideration, I also including the information of the resource(Agile team, budget). And the result of this consideration is showed in part four, I will list it here again.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Release | Release One | Release Two | Release Three | Release Four |
| Date | Jan 2017 to June 2017 | July 2017 to Oct 2017 | Nov 2017 to Feb 2018 | Mar 2018 to June 2018 |
| Goal | Release the first version of this system which is already a working product.(The main requirements for each theme are completed) | Focus on the requirements which is relate to following theme to improve the system features:  1. Upload and Download File  2. Communication  3. Course Management  4. View Information | Focus on the requirements which is relate to following theme to improve the system features:  1. Edit Personal Information  2. Register  3. Log in and out  4. Notification | Focus on the requirements which is relate to following theme to improve the system features:  1. Search  2. Evaluation  3. Others |

**Step 4:** This step is mentioned in the preparation of requirement, which is using Kano Model to prioritize the requirement per theme. The first process of Kano Model is to set a simple questionnaire to find out the relationship between requirements and user satisfaction. Following is the questionnaire I used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Question | Top Satisfaction | High Satisfaction | Middle Satisfaction | Low Satisfaction | Bottom Satisfaction |
| If this product implement this requirement, what level of satisfaction will you have? |  |  |  |  |  |
| If this product don't implement this requirement, what level of satisfaction will you have? |  |  |  |  |  |

The second process of Kano Model is to collect the result of questionnaire. The third process of Kano Model is to analyze the result based on some rules, and following is the rules I used.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Don't Implement this requirement | | | | |
| Top Satisfaction | High Satisfaction | Middle Satisfaction | Low Satisfaction | Bottom Satisfaction |
| Implement this requirement | Top Satisfaction | Suspicious Results | Suspicious Results | Excitement Quality | Performance Quality | Performance Quality |
| High Satisfaction | Suspicious Results | Suspicious Results | Excitement Quality | Performance Quality | Performance Quality |
| Middle Satisfaction | Reverse Quality | Reverse Quality | Neutral Quality | Basic Quality | Basic Quality |
| Low Satisfaction | Reverse Quality | Reverse Quality | Reverse Quality | Suspicious Results | Suspicious Results |
| Bottom Satisfaction | Reverse Quality | Reverse Quality | Reverse Quality | Suspicious Results | Suspicious Results |

And following is the result of Kano Model after I use it for my project's requirements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Basic Quality | Performance Quality | Excitement Quality | Neutral Quality | Reverse Quality |
| Communication | #69#108#117#131#176#186 | #59#146 | #67 | #114#193#199 |  |
| Course Management | #5#15#17#29#57#60#138#141 | #14#20 |  | #50#140 | #45#48 |
| Information Edition | #16#24#30#33#52#53#54#80#91#134#136#156#167#171#173#178#180#194#201 | #42#55#99#126#133#147#154#165 | #46 | #78#101 | #70 |
| Evaluation | #107#125#161#172 | #96 |  | #198 | #71 |
| Log In&Out | #27#61#62#103#109#168 | #63#88#89 | #104 | #137 |  |
| Notification | #181#188#190#203#204 | #85#112#113#196 | #87#120#124 |  |  |
| Registration | #65#74#111 | #183 | #75#135 | #192 |  |
| Search | #77#98 |  | #189 |  |  |
| Upload and Download File | #31#82#83#84#118#129#130#148#152#163#169#184 | #76#86#149#151#174 | #73#81 | #72 |  |
| View Information | #2#4#8#28#32#56#58#122#132#143#155#159#160#177#182#202 | #34#35#51 | #97#102#191 | #47#49#90#123#195 |  |
| Other | #38#142 |  |  | #9#18#22#37#92#94#105#106#110#115#139#150#157#158#200#206 |  |

**Step 5:** This step is mentioned in the preparation of requirement, which is using MoScoW Method to prioritize the requirement per theme. Following is the result of my prioritization for the requirements by MoScoW method.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Must Have | Should Have | Could Have | Won't Have |
| Communication | #67#69#117#131#176#186 | #59#108#193 | #114#146#199 |  |
| Course Management | #5#14#15#17#29#138#140 | #20#57#60#141 | #50 | #45#48 |
| Information Edition | #16#24#33#42##52#53#80#91#126#134#136#156#167#171#173#178#180#194#201 | #25#30#44#46#54#55#99#133#147#154#165 | #78#101 | #70 |
| Evaluation | #125#161#172 | #96#107 | #198 | #71 |
| Log In&Out | #27#61#62#103#109 | #63#88#89#104#168 | #137 |  |
| Notification | #87#112#181#203#204 | #113#120#188#190#196 | #85#124 |  |
| Registration | #65#74#111#183 | #135 | #192 | #75 |
| Search | #77#98 | #189 |  |  |
| Upload and Download File | #31#82#83#84#86#118#129#130#163#169#184 | #76#148#151#152#174 | #72#73#81#149 |  |
| View Information | #2#8#28#32#51#56#58#122#132#143#155#159#160#177#191#202 | #4#34#35#97#182 | #49#90#102#123#195 | #47 |
| Other | #38#142 |  | #9#18#22#37#92#94#105#106#110#115#139#150#157#158#200#206 |  |

**Step 6:** Combine the result of Kano Model and MoScoW method, I make the final decision on the requirement, some requirements must be put into first release which is important for the primary goal(develop a working product as soon as possible), some requirements can be develop in later release based on the different focus theme in roadmap, some requirements won't be implement. And the result will show in the part 7.

**7. Results of Release Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Release** | **First Release** | **Second Release** | **Third Release** | **Fourth Release** |
| **Date** | **Jan 2017 to June 2017** | **July 2017 to Oct 2017** | **Nov 2017 to Feb 2018** | **Mar 2018 to June 2018** |
| **Theme/high-level requirement** | Basic and Must have requirements in different themes | 1. Upload and Download File  2. Communication  3. Course Management  4. View Information  5. Relative Quality Requirement  6. Relative User Interface Requirement | 1. Information Edition  2. Registration  3. Log In&Out  4. Notification  5. Relative Quality Requirement  6. Relative User Interface Requirement | 1. Search  2. Evaluation  3. Others  4. Other requirement in past themes.  5. Relative Quality Requirement  6. Relative User Interface Requirement |
| **Goal** | Release a already working product | Focus on the theme to implement relative requirements | Focus on the theme to implement relative requirements | Focus on the theme to implement relative requirements |
| **Detail Requirements** | **Communication:** #69#117#131#176#186  **Course Management:** #5#15#17#29#138  **Information Edition:**#16#24#33#52#53#80#91#134#136#156#167#171#173#178#180#194#201  **Evaluation:** #125#161#172  **Log In&Out:**  #27#61#62#103  #109  **Notification:** #181#203#204  **Registration:** #65#74#111  **Search:** #77#98  **Upload and Download File:**#31#82#83#84#118#129#130#163#169#184  **View Information:** #2#8#28#32#56#58#122#132#143#155#159#160#177#202  **Other:** #38#142  **Relative QR:**  #1**#**7#10#13#26#162  #207#209  **Relative UI:**  #11#12#21#23#39#29  #145 | **Upload and Download File:** #73#76#81#86#148#149#151#152#174  **Communication:** #59#67#108#146#193  **Course Management:**  #14#20#57#60#140#141  **View Information:**  #4#34#35#47#51#97#102#182  **Relative QR:**  #41#128#153  **Relative UI:** #19#66#68 | **Information Edition:** #25#30#42#44#46#54#55#88#126#133#127#154#165  **Registration:** #135#183  **Log In&Out:**  #63#88#89#104#168  **Notification:** #85#87#112#113#120#124#188#190#196  **Relative QR:** #79  **Relative UI:** Null | **Search:** #189  **Evaluation:** #96#107  **Other:** #9#18#22#37#92#94#105#106#110#115#139#150#157#158#200#206  **Other requirement in past themes:**  #49#50#72#78#90#101##123#137#192#195#198#199  **Relative QR:**  #3#6#4#93#127  **Relative UI:**  #64#144#164 |
| **Resource（Team）** | 4 agile teams, 6 people in each team | 3 agile teams, 6 people in each team | 3 agile teams, 6 people in each team | 3 agile teams, 6 people in each team |
| **Resource（Others）** | Budget:  3,000,000kr  And all the other resource mentioned in 2.3 | Budget:  1,500,000kr  And all the other resource mentioned in 2.3 | Budget:  1,500,000kr  And all the other resource mentioned in 2.3 | Budget:  1,500,000kr  And all the other resource mentioned in 2.3 |

The Resource of team is based on the max involvement of our company, and all the equipment or needed software or system will be prepare for the agile teams(Our company already have these resource). And for the budget, I estimate that every people get 20,000kr per month. So the budget should be 2,880,000kr(first release) and 1,440,000kr(later release), the remain part is for some special case. And the total budget should be 7,500,000kr and the remain 500,000kr also is preparing for special situation.

After the building of roadmap, I plan the release based on the roadmap, as the Goal part shows, the primary goal is to develop a working product as soon as possible, and based on the roadmap, I plan the other three releases to implement other requirements based on the focus theme. So in the first release I put several basic also Must have requirements for each theme in order to develop a working product as soon as possible. And in second release I will put almost all the other requirements about the focus theme(Upload and Download File, Communication, Course Management, View Information), and same with third release( Information Edition, Registration, Log In&Out, Notification). For the last release, I put some requirements based on theme too(Search, Evaluation, Other) and also the remain requirements which is belongs to the past theme, these requirements usually are both neutral quality and Could have.