

# RELEASE PLANNING

Pakki Neetika

931110-0523

*Decide on a methodology for conducting release planning and describe it:*

Release planning (RP) helps in creating a sequence of consecutive product releases by making right decisions in selecting and assigning features so as to satisfy the important constraints such as technical, resource, budget, and risk constraints [1]. The release planning assignment of this course includes total 119 issues. These issues are needed to be prioritized based on the goal of the given milestones by using a suitable prioritization technique. All the participants of this course have gathered and discussed about all the prioritization techniques and selected MoSCoW prioritization technique. The reason for selecting this technique and the description of this technique is given below.

*Description:*

MoSCoW provides a basis for you to decide what will be done over the lifetime of a project. It considers those things that are essential and those which are nice to have but are less critical and can be omitted if necessary [2]. In MoSCoW prioritization technique, there are four kinds of priority groups which are MUST have, SHOULD have, COULD have, and WON'T have. In order to prioritise the requirements, all the requirements are grouped into the four grouping categories.

“MUST have” means that the requirements in this group must be implemented in the project. If these requirements are not included in the project, it may lead to the failure of the entire project.

“SHOULD have” means that the requirements in this group are important and adding these requirements to the project will make the product be good. And also it is not necessary that the requirements in this group needed to be delivering in the current release.

“COULD have” means that the project will be good if the requirements of this group exist. These requirements are given less importance than the requirements in “SHOULD have”

“WON'T have” the requirements in this group are given the least priority and hence they can't be implemented in the current iteration. [1][3]

The results of MoSCoW are on a nominal scale. All requirements contained in one priority group represent equal priority. No further information shows one requirement is of higher or lower priority than another requirement within one priority group [3].

MoSCoW prioritization technique includes of the following steps [2]:

1. Determine the project scope and timeframe
2. Write or draw each project element on an index card
3. Group the cards into piles of Must, Should, Could and Won't
4. Now assign values to each of the cards using different scoring.
5. Add the scores on each card, which will range from a 3 for Could/Low/Low to a 15 for Must/High/High
6. Reorder your stacks according to your weighted scores
7. Prepare your project plan based on the prioritisations you have identified
8. Monitor the project and revise the weightings as necessary as the project rolls out

*List and discuss at least three pros and three cons with the selected release planning method*

Every prioritization techniques have their own pros and cons and so those MoSCoW prioritization technique.

Advantages of MoSCoW technique:

1. It creates a common understanding with the stockholders on the importance they place on the delivery of each requirement.
2. This technique can be very useful when there is large number of requirements to be prioritised because it contains the highest degree of confidence and low difficulty rate and so it takes less time to perform and it provides high user confidence [2].
3. It is often easier for stakeholders to use than other prioritization techniques that may require stakeholders to define just how much of a higher priority one item versus another, with each item being compared to every other item.

Disadvantages of MoSCoW technique:

1. The problem with this technique is the stakeholders may classify the majority of the requirements into one group. There may raise some conflicts during the classification of requirements.
2. MoSCoW prioritisation doesn't offer any rationale for making the decision about how to rate the priority of a given requirement compared to others.
3. MoSCoW is ambiguous as to timing, particularly when it comes to the "Won't" rating. "Won't" could mean either "not in the next release" or "not ever"
4. Ensuring a common understanding of exactly what criteria should be used for assigning an item to a particular category can often be difficult.
5. MoSCoW cannot provide as meaningful a result as ordinal scale or ratio scale methods. It prioritizes requirements into different priority groups, but requirements that are in the same priority group represent equal priority; it cannot provide the relative difference between the requirements within one priority group. Some tools can help people with the scale up problem, but more resources are required and people need to spend time to learn how to use these tools [2].

*Apply it on the requirements in the LSRE-Release Planning Project repository. Describe what you do, and why.*

There are total of 119 requirements and 6 versions of milestones in our course. Each milestone has a particular goal. Based on the goals of the milestones, the requirements are taken out from the 119 requirements. This whole process is done as a team. After placing the requirements in the corresponding milestones, based on the customer value the MoSCoW technique of prioritization is applied to the requirements that are placed in each milestone.

In order to do this process, each version and their goals are considered.

## **VERSION 0.1**

Step 1: Goal of the milestone

We first need to identify and set a goal for the milestone. The goal of this milestone is that “Early version, get the bare minimums up and running so that teachers can at least keep students informed about a course.”

#### Step 2: Requirements list

In this step all the requirements are analysed and only those requirements are separated and placed in this milestone that are satisfying the goal. List of the requirements that are taken into this version based on the goal is listed below.

Requirements list: 27,62,14,12,36,21,5,40,41,53,18,56,37,54

Dependencies of these requirements: 14->56->54, 18, 27, 37, 21, 62, 40, 12, 36->53, 5->41.

#### Step 3: Prioritization of the requirements

After placing all the requirements that belong to the same milestone, these requirements are prioritised into four categories based on the customer value using the MoSCoW prioritization technique.

S. No	Must	Should	Could	Wont
1	18	56	54	
2	27	62	5	
3	40		41	
4	14			
5	37			
6	12			
7	21			
8	53			
9	36			

### Version 0.5

#### Step 1: Goal of the milestone

We have identified and set a goal for this milestone. The goal for this milestone is “The most important features are available, but perhaps not with full functionality.”

#### Step 2: Requirements list

In this step all the requirements are analysed and only those requirements are separated and placed in this milestone that are satisfying the goal. List of the requirements that are taken into this version based on the goal is listed below.

Requirements list: 33, 50, 114, 30, 16, 77, 107, 101, 45, 69, 32, 15, 26, 43, 91, 58, 22, 44, 103, 64, 6, 19, 59, 43, 94, 20, 23, 66, 15, 28, 67, 28, 109, 55

#### Step 3: Prioritization of the requirements

After placing all the requirements that belong to the same milestone, these requirements are prioritised into four categories based on the customer value using the MoSCoW prioritization technique.

S. No	Must	Should	Could	Wont
1	16	32	114	33
2	15	64	103	

3	28	55	109	
4	22	50	44	
5	20	30	91	
6	66	58	26	
7	23	43	101	
8	69	45		
9	107	67		
10	77	52		
11		59		
12		19		
13		6		
14		94		

### Version 0.8

#### Step 1: Goal of the milestone

We have identified and set a goal for this milestone. The goal for this milestone is “Full functionality for the most important features. All features at least partially implemented.”

#### Step 2: Requirements list

In this step all the requirements are analysed and only those requirements are separated and placed in this milestone that are satisfying the goal. List of the requirements that are taken into this version based on the goal is listed below.

Requirements list: 75, 33, 81, 17, 79, 35, 1, 21, 34, 78, 2, 3, 73, 63, 61, 97, 11, 92, 86, 84, 88, 31, 82, 25, 85, 51, 100, 74, 110, 101, 4, 87, 65, 83, 96, 38, 80, 102.

#### Step 3: Prioritization of the requirements

After placing all the requirements that belong to the same milestone, these requirements are prioritised into four categories based on the customer value using the MoSCoW prioritization technique.

S. No	Must	Should	Could	Wont
1	75	33	81	17
2	1	35	79	
3	21	34	78	
4	3	73	2	
5	63	61	97	
6	11	92	86	
7	31	88	84	
8	82	25	85	
9	74	51	100	
10	4	101	110	
11	87	65	83	

12		96	102	
13		38		
14		80		

### Version 0.9

Step 1: Goal of the milestone

We have identified and set a goal for this milestone. The goal for this milestone is “All functionality implemented.”

Step 2: Requirements list

In this step all the requirements are analysed and only those requirements are separated and placed in this milestone that are satisfying the goal. List of the requirements that are taken into this version based on the goal is listed below.

Requirements list: 29, 106, 112, 104, 90, 105, 111, 57, 113, 46, 47, 48, 49.

Step 3: Prioritization of the requirements

After placing all the requirements that belong to the same milestone, these requirements are prioritised into four categories based on the customer value using the MoSCoW prioritization technique.

S. No	Must	Should	Could	Wont
1	29	112	106	
2	90	104		
3	111	105		
4	57	113		
5	47	46		
6	48			
7	49			

### Version 1.0

Step 1: Goal of the milestone

We have identified and set a goal for this milestone. The goal for this milestone is “Bugs addressed most important feedback from earlier releases taken into account.”

Step 2: Requirements list

In this step all the requirements are analysed and only those requirements are separated and placed in this milestone that are satisfying the goal. List of the requirements that are taken into this version based on the goal is listed below.

Requirements list: 29, 106, 112, 104, 90, 105, 111, 57, 113, 46, 47, 48, 49.

Step 3: Prioritization of the requirements

After placing all the requirements that belong to the same milestone, these requirements are prioritised into four categories based on the customer value using the MoSCoW prioritization technique.

S. No	Must	Should	Could	Wont
1	70			
2	71			
3	72			

The prioritization of the requirements for release planning, we formed a group and used the MoSCoW technique because if the requirements are not prioritized, it would be improper to send all the requirements at one's to the release. In that case there will be high chances of missing some of the important requirements. In order to solve this issue, we have selected a prioritization technique and divided all the 19 requirements into four categories depending on the customer value so that based on the categories, the requirements are sent for release. In this way all the important features are implemented without any failure.

*Describe your experiences from using the selected release planning method, and your experiences from using GitHub's issue tracker for working with requirements and release planning*

This is not my first time using MoSCoW technique for the purpose of prioritization. I find it very simple to understand and implement this technique so it did not took me much time to get use to this technique. By using this technique, categorising the requirements into four categories is very easy and once the requirements are placed in the categories they belong to, it became easy for us to decide which requirements need to be implemented first in the release planning based on the time factor. I find this technique very useful and helpful in making the right decisions. By the implementation of this technique, our work load has reduced a lot and the remaining process was carried out in a short time. Since the prioritization of all the features/ requirements are done in a group, we had to spend lots of time in deciding what feature has to be placed among the four categories of this technique.

It is the first time for me to work with GitHub and it is a very new and different experience. I liked it because it is always an available server to store all of my git repositories either in public or private and also because it provides repositories for students and professors so that the professor can keep a track on the students and give the feedback on the work process and also allows the students and the professor to discuss. I felt that GitHub made the release planning of the requirements easier and effective when compared with normal release planning process because GitHub allows us to add issues/requirements and it allows us to attach various labels such as bugs, clarification, duplication, feature etc to those requirements. Each label has its own functionality. And this helped us to indentify where the requirement is a bug or duplication etc depending on the label that is attached. GitHub also allows us to self- assignee the requirements and also it provides us with search, sort and filter option.

References:

- [1] J. A. Khan, "Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique," no. November, pp. 53–59, 2015.
- [2] L. Hunt, ".co.uk," pp. 1–3.

- [3] M. Sciences, "The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements : A Systematic Literature Review," no. November, 2009.