Reflective Report

Author: Tan Yanwen
Personal ID: 199310079265

Article 1: Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-Secure Corporation [1]

Motivation:

The author studied a case organization and faced challenges with release planning, according to the consequently he invent a new release planning method which call the Release Iteration Planning method. The article present scientific study on Release Iteration Planning method, and using practice and improvements the method on multiple projects, finally obtained the success.

The author recorded the two project which using this method to do the release planning, which can make readers more clear understanding of the benefits of the use of the method and the way how to use the method. At the same time the author also gives a method faced challenges during first project, and detailed records in the second Project how to improve the method to solve the problems, and finally makes the process of two projects are both successful. It is also more conducive for me to know how this method to improve the process and some problems I may met and how to solve them, which is benefit for me plan ahead how to use the method applied in my release planning.

Implementation Plan:

First need to group the project team, in my own projects, I'll divide team according to the ability into four group: The design group, management group, front-end development engineers group and backend development engineers group. The group is not only the development group, it also will be in after The Release of The Iteration Planning Events plays an important role.

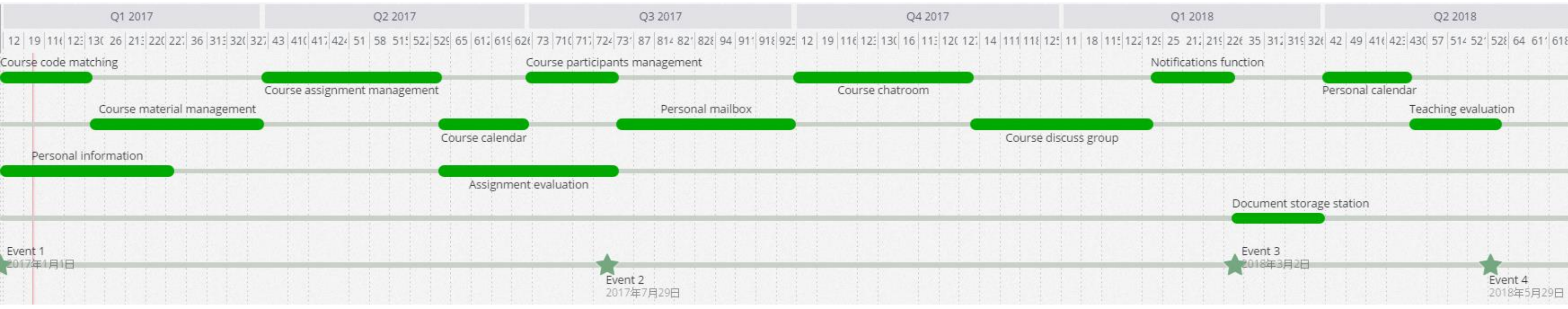
Second, we need to prepare specific requirement before the events, in the events, we will discussion and prioritization for the specific requirements of this release, if before the start of the event, management are not clear of the requirements on this release, that will cause the time wasted of this event, so settle ahead of the requirement is must.

Then according to the team divided and the requirements for the release timeline planning and every event planning, and the implementation of the specific to the project.

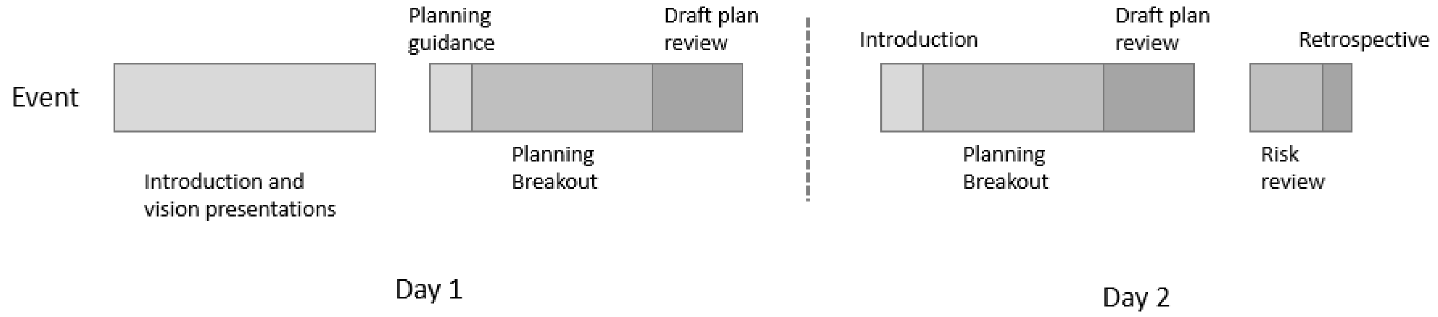
Execution:

In the project of educational administration system, I have 208 requirements at start and after remove duplicate requirements, I respectively throw them in 16 different Package, and according to different functions to throw this 16 package into four Platform, according to the Platform and the importance of the Package, custom made full release planning and event planning.

Proof of Concept:



Picture 1 Timeline of release planning



Picture 2 Example of Iteration Planning Events

Lessons Learned:

Through this article, let me know in LSRE for the requirement prepare is very important. First project in the article, it is because there is no requirement prepare before the event, that led to the first event time is long and the problem of low efficiency. At the same time, because of lack of feature prioritization and allocation also led to the release of the division of labor is not clear, no clear priority to distinguish what should do first, what to do after.

According to the article, in my own projects I pay more attention to the requirement of the early stage of the assignment, to guarantee the requirement in each release is reasonable, and use the appropriate prioritization tools to sort them, make release planning more clear and easy to develop the system.

Reflections:

Sami Jantunen has mentioned in his article [2] that most of time requirements are not known well enough, which has proved to be hard to make decisions. When the requirements are ambiguous, constrained and complex, the processes tuned for decision-making begun to providing a false sense of security based on the unjustified numerical certainty. Sometimes we fail to see this clearly because the concentration in our science-based culture on linear causal chains and on the notion of goal-seeking [3].

This means that, in initiation Package distribution of 208 requirements for project may appeared cognitive biases, when I first rough distribution of 16 package and four platform whether they are matching one by one is not sure, and after prioritization work is carried out based on this, and this may be the instability influence of the result.

Because in LSRE, one project usually has thousands or more requirements, if not classify instead of directly handle a large number of the requirement at the same time, it will be more time consuming and energy cost, so find a way when making LSRE initial requirement to proceed in a more stable the classification of the package is worth thinking about.

Article 2: RICE: Simple prioritization for product managers [4]

Motivation:

In requirement priority arrangement, we often encounter many problems, such as:

Compared to those who affect a wide range of projects, we are more willing to do what we are interested in;

Compared to those who can foresee the project directly, innovation needs tend to be seduced;

More willing to try new projects, and ignores the problems left over by history and so on.

In order to priority of these requirements do not need to be detailed by artificial selection, a reliable set of requirements prioritization rating system can help us, and RICE is which I think easy to use and understand.

Implementation Plan:

RICE was used to assess the four factors of the project requirements of the acronym: Reach, Impact, Confidence, and Effort. In my project, I will focus on different package requirement within the package sorting, and comprehensive package in each Platform internal requirement average score of RICE, using the average score to determine whether to need to remove some requirement which score is too low, the actual value is not high requirement.

Execution:

First of all, I will according to the description of each requirement for the numerical assessment of the four factors.

Reach is measured in the number of people/events per time period.

Impact use the weighted item to show: the 3 represents "major Impact", 2 represents "high Impact", 1 represents "medium effect", 0.5 represents "light", and a minimum of 0.25 represents "slight". These numbers will be graded as a weight value and multiply with other value to assess obtained evaluation results.

Confidence use percentage to show: 100% is "high Confidence" 80% is "medium", and 50% is "low".

Effort is estimated as a number of “person-months” – which means the work that one team member can do in a month.

After calculate these four factors, using the following formula to calculate the Rice score and prioritization the requirement:

$$\frac{\text{Reach * Impact * Confidence}}{\text{Effort}} = \begin{matrix}\textbf{RICE}\\ \text{SCORE}\end{matrix}$$

Proof of Concept:

Requirement	Reach	Impact	Confidence	Effort	RICE Score
Send mail	800	3	100%	4	600
Receive mail	800	3	100%	4	600
Group send email	200	2	80%	4	80
Restore deleted mail	70	0.5	80%	8	3.5
Draft box	400	2	80%	6	106

Table 1 Personal mailbox package requirement RICE score

According to the above table, we can clearly see that in the personal mailbox requirement package, send mail and receive mail is the highest priority 2 function, and draft the box and group send email is the priority of medium, restore the delete mail is far below the average score of whole function, so it can consider to give up because of low value.

Lessons Learned:

In our LSRE process, has a rating system we can clearly identify when we are doing the weighing of the pros and cons. Because in the process of dealing with large requirement, we can't analysis requirements one by one rely on human, so more often we need to have the aid of some prioritization methods for quick and easy assessment, at the same time it also can be a good reason to persuade some stakeholder who don’t trust the specific requirement

But at the same time, the author also mentioned in the article, the RICE points can't be used as a hard rules. There are many reasons you might go to do a score lower project. May be another project must base on this project has already finished, or another feature may be sold to some customers as "chip". So the score is a justified reference, rather than a template must perform step by step.

Reflections:

In addition to RICE, I also looked at a lot of other priority arrangement tool, in Javed Ali Khan's article [5], he introduced several requirement prioritization techniques to like binary search tree, AHP, spanning tree matrix, Numerical Analysis, MoSoW, simple ranking and Planning Game, learning through these method can be found that different methods have different benefits and shortcomings. Such as AHP can produce most reliable results because it is based on ratio scale, but due to the complexity of algorithm, sometimes in LSRE may need to spend too much time to build a mathematics model, and on the basis of this model to calculate, at the same time this model does not necessarily apply to all the requirement in the LSRE.

So when make the prioritization of requirements, some method you thought prefect may not really useful, choosing method should according to actual condition more over.

Reference

[1] Ville T. Heikkilä a, Maria Paasivaara, Kristian Rautiainen, Casper Lassenius, Towo Toivola,Janne Järvinen, Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-Secure Corporation, Department of Computer Science and Engineering, Aalto University, PO Box 15400, FI-00076 Aalto, Finland

[2] Sami Jantunen, Laura Lehtola, Donald C. Gause, U. Rex Dumdum, Raymond J. Barnes, The Challenge of Release Planning

[3] P. B. Checkland, "From Optimizing to Learning: A Development of systems Thyinking for the 1990," The Journal of the Operational Research Society, vol. 36, pp. 757-767, 1985.

[4] SEAN MCBRIDE, RICE: Simple prioritization for product managers, Inside intercom

[5] Javed Ali Khan; Izaz Ur Rehman; Yawar Hayat Khan; Iftikhar Javed Khan; Salman Rashid, Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique, International Journal of Modern Education and Computer Science, 11/2015, Volume 7, Issue 11