REFLECTIVE REPORT

Nitin Parasa
940803-0238
nipa15@student.bth.se

Technique 1:

23-December-2015

Article selection:

- I came across another article which stresses on the prioritisation of requirements. I find this field more interesting in MDRE context and would like to get hands on experience using techniques in this field and contribute my knowledge in the future.
- I plan to execute the Cost-Value approach for Prioritizing requirements. The article used for this is written by J. Karlsson and K. Ryan [2].

Description of Cost-Value Approach:

- The Cost-Value approach for requirements prioritisation developed by J. Karlsson ad K. Ryan helps in ranking the requirements by considering two dimensions: 1. Value to the customer 2. Cost of implementing it. In the process of using this technique we would be using the concepts of Analytical Hierarchy Process (AHP) which forms the core part of the technique. A list of steps have to be followed to implement this technique and finally arrive at a prioritised list of requirements. They are:

1. Inspection on the requirements is done for clear understanding and completeness of the requirements.
2. We use the AHP to find the relative value of the requirements.
3. Then the engineer calculates the implementation costs and relative values of each candidate requirement.
4. Then implementation costs and relative values are formulated and then the requirements are placed in cost-value graph.
5. Using the graph as an aid the stakeholders prioritise the requirements.

3-Jan-2016:

Implementation Plan:
I plan to take 3 requirements and carry out this technique. The reason I am taking only three requirements is that being this the first time implementing this it would be clear to execute all the steps involved. Moreover the technique that we are implementing here can be scaled to any number of requirements and just the calculations get a bit tough but the whole process remains the same.

Execution and Proof of Concept:

The requirements gathered are a part of the travel guide website where users can enter the system view various places and add the places they would like to visit in the future into their favourites list which will be stored for each and every user. They are:

R1: As a user, I must be able to login to the system.

R2: As a user, I want to have a responsive interface so that I have good user experience.

R3: As a user, I want to view a catalog of all places, so that I can get to know of all the places in the selected country.

R4: As a user, I want to have a filter function so that I can specifically view place that suit my interest.

R5: As a user, I want to add my places of interest to a wishlist, so that I can refer to them later.

R6: As a system, I want to authenticate users via e-mail, so that the user is verified.

R7: As a user, I want a complete description page so that I can view important information about places.

R8: As a user, I want to register and signup so that user can access profile.

R9: As a system, I want to maintain session data, so that the user data is dynamically available for all pages.

Step 1:

All requirements are clearly understood and written as per standards.

Step 2:

Carry out the AHP process:

Step 2.1: Set up 9*9 matrix.

Step 2.2: Perform Pairwise comparisons:

	R1	R2	R3	R4	R5	R6	R7	R8	R9
R1	1	3	5	3	7	0.33	3	0.2	3
R2	0.33	1	3	3	3	0.2	3	0.33	3
R3	0.2	0.33	1	5	5	0.33	3	0.2	0.33
R4	0.33	0.33	0.33	1	3	0.33	3	0.2	0.33

R5	0.14	0.33	0.33	0.33	1	0.33	3	0.14	0.33
R6	3	3	3	3	3	1	0.33	0.2	3
R7	1	1	0.33	0.33	0.33	3	1	0.2	0.2
R8	5	3	5	5	7	3	5	1	3
R9	0.33	0.33	0.33	3	3	0.33	5	0.33	1

Step 2.3: Next we perform the averaging method to determine the eigen values of the matrix.

|	|R1	|R2	|R3	|R4	|R5	|R6	|R7	|R8	|R9	|
|R1	|0.09	|0.25	|0.24	|0.12	|0.21	|0.03	|0.11	|0.07	|0.21	|
|R2	|0.03	|0.08	|0.14	|0.12	|0.09	|0.02	|0.11	|0.11	|0.21	|
|R3	|0.018	|0.02	|0.04	|0.21	|0.15	|0.03	|0.11	|0.07	|0.02	|
|R4	|0.03	|0.02	|0.009	|0.04	|0.09	|0.03	|0.11	|0.07	|0.02	|
|R5	|0.013	|0.02	|0.009	|0.01	|0.03	|0.03	|0.11	|0.05	|0.02	|
|R6	|0.28	|0.25	|0.14	|0.12	|0.09	|0.11	|0.012	|0.07	|0.21	|
|R7	|0.03	|0.02	|0.015	|0.01	|0.01	|0.33	|0.03	|0.07	|0.01	|
|R8	|0.46	|0.25	|0.24	|0.21	|0.21	|0.33	|0.18	|0.35	|0.21	|
|R9	|0.03	|0.02	|0.14	|0.12	|0.09	|0.03	|0.18	|0.11	|0.07	|

Then find the sum of each row in the matrix and divide each element by the number of requirements:

|1/9	|1.14	|
|	|0.91	|
|	|0.668	|
|	|0.42	|
|	|0.292	|
|	|1.282	|
|	|0.525	|
|	|2.38	|
|	|0.79	|

|0.126|
|0.101|
|0.074|
|0.046|
|0.032|
|0.142|
|0.058|
|0.264|
|0.087|

Step 3:

Assign the relative value to each requirement basing on the eigen value obtained above.

- R1 contains 13 percent of the requirements' total value
- R2 contains 10 percent
- R3 contains 7 percent
- R4 contains 5 percent
- R5 contains 3 percent
- R6 contains 14 percent
- R7 contains 6 percent
- R8 contains 26 percent
- R9 contains 9 percent

Then we find out the result consistency just to confirm if we were able to determine the relative value efficiently.

Next we have to find the Consistency Index: $CI = (\lambda max - n)/(n - 1)$

Multiply both matrices to find the value of λmax.

	R1	R2	R3	R4	R5	R6	R7	R8	R9
R1	1	3	5	3	7	0.33	3	0.2	3
R2	0.33	1	3	3	3	0.2	3	0.33	3
R3	0.2	0.33	1	5	5	0.33	3	0.2	0.33
R4	0.33	0.33	0.33	1	3	0.33	3	0.2	0.33
R5	0.14	0.33	0.33	0.33	1	0.33	3	0.14	0.33
R6	3	3	3	3	3	1	0.33	0.2	3
R7	1	1	0.33	0.33	0.33	3	1	0.2	0.2
R8	5	3	5	5	7	3	5	1	3
R9	0.33	0.33	0.33	3	3	0.33	5	0.33	1

And

|0.126|
|0.101|
|0.074|
|0.046|
|0.032|
|0.142|
|0.058|
|0.264|
|0.087|

Which results in

|3.323|
|1.704|
|1.142|
|0.406|
|0.188|
|2.781|
|0.352|
|9.784|
|1.432|

Then divide the first element of the resulting vector by the first element im primary vector.

|25.53 |
|16.86 |
|15.39 |
|8.72 |
|5.8 |
|19.53 |
|6.05 |
|37 |
|16.32 |

Then we find the value of λmax as :
λmax = (25.53+16.86+15.39+8.72+5.8+19.53+6.05+37+16.32)/9 = 16.7778

CI = (λmax-n)/(n-1) = (16.7778 – 9)/ (9-1) = 7.778/8 = 0.97

Finally we find the consistency ratio

CR = CI/ RI

RI indices for matrices are as given below:

|1| 2|3 |4 |5|6 |7 |8|9|10| 11|12|13|14|15|
|0.00 |0.00|0.58|0.90 | 1.12|1.24| 1.32|1.41|1.45|1.49|1.51|1.48|1.56|1.57|1.59|

CR = 0.97/1.45 = 0.66 which is a reasonable value but is occurs very frequently when people apply this method. Generally a value of 0.1 or lower is acceptable.

Step 4:

Similarly we perform the Cost implementation technique and find out the relative values as performed for value. The same sequence of steps applied for relative value determination is

applied to find out the relative costs of the requirements. It is only that the values are depicted here but not the whole process again.

The relative costs obtained are:

- R1 contains 3 percent of the requirements' total cost
- R2 contains 12 percent
- R3 contains 14 percent
- R4 contains 13 percent
- R5 contains 28 percent
- R6 contains 6 percent
- R7 contains 7 percent
- R8 contains 9 percent
- R9 contains 5 percent

Step 5:

Using the above values we have to generate a cost value diagram which helps in prioritising the requirements. For this analysis the suggested method is that three different sections are divided in the graph where each region is named as High, Medium and Low respectively. The value-to-cost ratios of each requirements are as follows:

R1:4.33
R2:0.833
R3:0.5
R4:0.33
R5:0.107
R6:2.33
R7:0.85
R8:2.88
R9:1.8

There is a limit for placing the requirements into each of these categories depending on the value-cost ratio:
High: above 2, Medium: 2 - 0.5, Low: below 0.5

The requirements in this project can be classified as:
High: R1, R6, R8

Medium: R9, R2, R7, R3

Low: R5, R4.

Lessons Learned:

While I was implementing this technique just for 9 requirements I got off the track at times and got mislead due to the simple but complex looking calculations. Looking at this I could figure out how daunting this would be for requirements of the order of 1000 where the companies implementing LSRE or MDRE face.

Without the aid of proper tools this task can prove to be a nightmare for the requirement analysts, product managers, etc. Coming to the experinece and lessons learned as a part of implementing this method is that it is not so easy as it looks. But with proper concentration and knowledge this task could be done with effeciency and is reliable in giving valid results.

Reflections:

I had a good experience working with this technique. It was interesting and challenging to use and at the same time simple to understand. It hardly took time for me to learn the concept behind this prioritisation technique. However my experiences incline towards what the author of article [2] says. There are examples of two case studies carried out by the implementation this technique which showed positive results towards the end of the requirements phase of each project.

Technique 2:

5-Jan-2016:

The second technique that I would like to impement and reflect my views upon is the Requirments Abstraction Model (RAM). I am very much fascinated with this technique because this make the process of prioritsing the requirements a lot easier. I think that placing the requirements through the RAM model gives a clear idea as of what each reuqirement actually meant and where each of them belong to. Furtherdown I would like to share my experiences working with this model.

What is RAM?

It is a model with four abstraction levels developed by Prof. Dr. Tony Gorschek and Prof. Dr. Claes Wohlin. They have developed a model where requirements can be placed on different levels and accordingly they can be broken down making it easier to compare with each other. Three constranits are mainly focussed in this model, what (requirements) a release should contain, when (time) of the release should they be implemented and at what cost. Continuous requirements engineering is supported by this model where in requirements can be placed at various levels and can be broken down into detailed ones. RAM is carried out in three steps. They are:

1. Specify: This is the elicitation phase where the requirements are gathered from the stakeholders. Here the raw requirements are taken and the extent to which the Product Manager understands these is analysed. Four attributes are specified in this step. They are: 1.

Description: The requirements here should satisfy the criteria that it should not be more than 5 sentences. Forms the central essence
2. Reason: deals with why the requirement is specified.
3. Risk/ Restrictions: it deals with the restrictions and risks of the requirements.
4. Title: it should be able to clearly depict the requirement.

2. Place: This phase deals with placing the requirements at the right abstraction level. Four abstraction levels exist. They are:
1. Product Level
2. Feature Level
3. Function Level
4. Component Level
and finally
3. Abstraction: the requirements are broken down to detailed ones in this phase.

Implementation Plan:
As a part of implementing this method I would like to make use of three requirements and then implement the RAM model on these requirements and check the level to which eah of these requirements reside. The requirements selected are a part of an online travel guide where the users can get to register and navigate through list of countries to find out various places in these countries.

7-January-2016

Execution and Proof of Concept:

Step 1:

Requirement 1:

Title: Interactive Map
Description: As a user I should be able to navigate to different countries to view the places in those countries.
Reason: easy navigability to the user.
Risks: might not be understood to the amateur users.

Requirement 2:

Title: login
Description: As a user, I should be able to log into the system.
Reason: to distinguish each user.
risk: sessions might not be correctly handled

Requirement 3:

Title: wish list

Description: As a user, I should be able to add places to my wishlist.
Reason: better and rich user experience
risk: wrong data being stored for the users.

Step 2:

Requirement 1: Feature Level

Requirement 2: Feature Level

Requirement 3: Functional Level

Step 3:

Here we perform the abstraction of the requirements specified.

Requirement 1: Interactive Map - Feature Level
This could be further broken down into:

Requirement 1.1: Countries accessible

Title: Countries Accessible
Description: what are the different countries to which the access in the map is given
reason: maintain the scope of the project within limits.
Abstaction Level: Functional Level

Requirement 2: Login - Feature Level
This requirement can be broken down into

Requirement 2.1: Email Authentication
Description: the first time the user log in it is mandatory to activate his email id.
reason: increases the security of the product.
Abstraction Level: Functional Level

Requirement 3: wishlist - Functional Level
This could not be further broken down as this is in its most abstract level.

Lessons Learned:

I was not able to completely explore this technique as some of the requirements that I have taken are straight away at the most abstract level and could not be broken down. Yet this technique is very efficient, as I was trying to break the requirements down I was able to think deep about it and make it even more clear. This way it helps the organizations in clearly understanding what each and every requirement actually mean.

References:

[1] Khurum, Mahvish, Khurum Aslam, and Tony Gorschek. "A method for early requirements triage and selection utilizing product strategies." Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific. IEEE, 2007.

[2] Karlsson, Joachim, and Kevin Ryan. "A cost-value approach for prioritizing requirements." Software, IEEE 14.5 (1997): 67-74.

[3] Gorschek, Tony, and Claes Wohlin. "Requirements abstraction model." Requirements Engineering 11.1 (2006): 79-101.