

REFLECTIVE REPORT

Pakki Neetika
931110-0523

Requirements Abstract Model:

For the implementation process of this model, I have selected 11 requirements. For the first requirement, the whole implementation process is shown below in three steps.

R1. Tracking the order product (status) : this is my first requirement.

Step 1: To elicit/ specify the requirements using the four attributes.

a. Description: this requirement helps the user to know the current status of the ordered product by tracking it.

b. Benefits: with this requirement the user will have an idea where exactly the ordered product is located.

c. Title: "Status of your order"

Step 2: In the second step, placing of the requirements in different levels is done. The placing of the requirements is done by following few questions and answers to those questions will relieve to which level each requirement belongs. If the requirements that belongs to the product level, it should be able to answer the question that says "if the requirement is comparable to the product strategies?" For the requirement to be placed in the feature level, the question is if the requirements describe "what the system should include or support?" requirement R1 fits in this level. The requirement is placed in the functional level if it answers the question that say if the requirements is functional or not, or if the requirements in the question describes what characteristics a system should provide?

Step 3: new requirements are created so as to link all the levels (product to component) for each of the requirement.

For the first requirements R1 which belongs to feature level, new requirements are created that fits in product level, functional level and component level.

Product level: system should provide online services to the customers

Feature level- Tracking the order product (status)

Functional level: the system must be able to display all the ordered products with a status option.

Component level: the system should display the date of order and the current status of the delivery product when status button is clicked.

The above process is followed to all the rest of the requirements. The table shows the requirements that were added to the remaining levels by following the RAM process.

Requirements	Product level	Feature level	Functional level	Component level
R1	System should provide online services to the customers	Tracking the order product (status)	The system must be able to display all the ordered products with a status option.	The system should display the date of order and the current status of the delivery product when status button is

				clicked
R2	The system should currently focus on the Swedish market and should be extended this to the European market in the future.	There should be a button provided for the users to change the language and country.	Language and country should be changed automatically when the users clicks the button.	A drop own box is provided through which the users can select the desired country and language before clicking the button.
R3	The product should be secured and prevent unauthorised usage.	All the users of the system should have a personal profile.	The user must be able to login into the system before the usage of product.	If the user enters an incorrect user id or password, the login page should be reloaded with information showing that incorrect login has been performed.
R4	The system should support automatic logout	The system must automatically logout if the user is not using the system for a long period of time or if the tab/browser is closed.	All the users may not want to automatically logout from the system. Hence an option is provided to enable and disable this feature.	The system must automatically logout if the user is not using it for one hour.
R5	The product should be able to support traceability.	Users should be able to find the required information through the search bar option.	As soon as the information is entered and the search button is clicked, required information should be displayed.	The search button should be enabled only if information is entered in the textbox, otherwise it should be in disabled state.
R6	The system should be flexible to add/remove the participants.	It shall be possible to add/Remove users as participants in a course.	Only the course administrator has the privilege to add/remove the users as participators in a course.	Once a particular user is added or removed as a participant of the course, the corresponding information should be automatically updated in the database.
R7	The system	There should be a	Only participators	It shall be

	should facilitate interaction among the participants of a course.	discussion form where the participants can send and receive messages to/from other participants.	in a course shall be able to access the course's discussion forum.	possible to view and add new messages in the course's discussion forum.
R8	The system should be flexible to change the privacy settings.	There should be a feature of changing the password if the user forgot the previous password.	If a user requests for a new password, a recovery mail should be sent to the user id.	The recovery email should be displayed in the inbox within 5 to 10 minutes.
R9	The system should provide access to submit and evaluate assignments.	The participants should be able to upload and download the assignments.	Only the participant has the privilege to upload assignment on their own account. The owner of the assignment and the teacher has the privilege to download the assignment.	The assignments should be downloaded to the desired location once the download button is clicked.
R10	The system must provide functionality to administer of the products	the system must provide the details of all the products	The administrator has the privileges to add the product items to the product list.	New products should be added along with the cost, picture, main ingredients and quantity of the products.
R11	The system should provide the service to the authorized users	The system should allow the user to register.	The user to be able to add his/her details into the registration form.	after successful registration, the system should display a message that the user is successful registered

Lessons learned:

After going through this article, I came to know that in market driven requirement engineering, the flow of the requirements are continuous. It is crucial to manage and handle the large number of requirements in right way at the right time because poor management of the requirements may lead to a large impact on factors such as time, success rate, cost and standards of the software product. Handling the continuous flow of the requirements along with their interdependencies is a complex task to perform because requirements arrive in different shapes and forms to development

organizations. Several interdependencies exist among different requirements and these dependencies must be handled carefully before early stages of development. To deal with such problems, Requirement Abstract Model comes into action which will allow companies to ensure comparability between requirements. In this course, when I was asked to implement any method or technique for LSRE. Before implementation of this model (RAM), I did not have any clear idea about the requirements and was confused with the interdependencies since it is the first time for me to practically deal with requirements. But while implementing this model, I was getting to know about the requirements clearly in a step by step process and adding additional requirements became an easy process as the requirements are added to distinct abstraction levels. At the initial stages I have faced problems in placing the requirements to different levels but with the help of how-to-guide at the end of the implementation of this model, I have got an idea how requirements are handled, implemented and divided into different releases in MDRE. I also understood how to deal with the dependencies practically and what kind of challenges arise while dealing with the requirements.

Reflections:

In order to relate my work and compare my experience, I have selected other article that discuss about the RAM process. The article that I have selected is “A controlled empirical evaluation of a requirements abstraction model”. The authors of this paper felt that the two main factors motivated the creation and evolution of the RAM: (1) a direct need identified in industry and (2) a suitable model was not found in the literature, i.e., a model for continuous requirements engineering catching and handling requirements on multiple levels of abstraction. For this the author of this article has presented an evaluation of RAM which is built based on needs identified in industry. The evaluation’s primary goal is to test the model’s usability and usefulness in a lab environment prior to large scale industry piloting, and is a part of a large technology transfer effort. The evaluation uses 179 subjects from three different Swedish Universities. After analysing, the results show that RAM can be tailored to satisfy the needs of different organizations and products. In the same way, it can be modified over time to reflect the current situation of a development organization, supporting the collection of requirements over time and product’s life cycle. The results from the industrial validation point to the usefulness of the model. The model will allow companies to ensure comparability between requirements, and hence it generates important input to activities such as prioritization and packaging of requirements before launching a development project. From my experience with the implementation of this model, I can tell that this model is easy to implement and if any changes in the requirement has to be done, it can be done with less complexity. I did not find it difficult to compare the requirements while placing them in different levels. I did not face any problems because I have considered only few requirements but in the case of MDRE, there will be thousands of requirements and challenging problems will arise as mentioned in the reference paper.

QUPER Model:

Since I was asked to apply the method on a larger base of requirements, I have considered 10 requirements and implemented QUPER model on these requirements.

Requirements	Identify candidate QR	Define scale and unit	Identify reference levels	Elicit quality breakpoints	Estimate cost barriers	Set candidate requirements
R1	The competitor have recently released a product with a response time of 3 sec	Scale: time Unit seconds	X: 10 sec Competitor Y: 8 sec Competitor Z: 5 sec	Utility:9 Saturation:3 Differentiation:7	Qref: 8 Q11 : 6 C11:2 Q12 : 4 C12 4	GOOD: 6 STRETCH: 4
R2	The competitor have recently released a product	Scale: time	X: 90 Competitor Y: 80	Utility:88 Saturation:6 8	Qref: 80 Q11 75 C11 2	GOOD: 80 STRETCH 76

	with mtrr of 60 minuets	Unit minuets	Competitor Z:75	Differentiati on:78	Q12 70 C12 5	
R3	The competitor have recently released a product with Load time 5 seconds	Scale: time Unit seconds	X: 15 Competitor Y: 10 Competitor Z:7	Utility:14 Saturation:6 Differentiati on:9	Qref: 11 Q11 8 C11 3 Q12 7 C12 5	GOOD:12 STRETCH 6
R4	The competitor have recently released a product with Mttf of 30 minuets	Scale: time Unit minuets	X: 60 Competitor Y: 55 Competitor Z:40	Utility:58 Saturation:3 5 Differentiati on:50	Qref: 55 Q11 40 C11 4 Q12 36 C12 8	GOOD: 56 STRETCH 45
R5	The GPM shall maintain a record for 1 year of each payment.	Scale: number of records Unit: years	X: 2 Competitor Y: 1.5 Competitor Z: 1.2	Utility:1.8 Saturation:1. 1 Differentiati on1.4	Qref: 1.5 Q11 1.3 C11 1 Q121.2 C12 3	GOOD:1.6 STRETCH1. 2
R6	The maximum number of tests that can fail when the system is shipped shall not exceed TBD 6.	Scale: number of tests Unit: TBD	X: 10 Competitor Y: 8 Competitor Z: 6	Utility:9 Saturation:5 Differentiati on7	Qref: 8 Q11 8 C11 3 Q12 6 C12 6	GOOD:8 STRETCH 6
R7	When one user updates some data, the system shall ensure that other users shall automatically see the update within 2 seconds.	Scale: time Unit seconds	X:12 Competitor Y: 8 Competitor Z:6	Utility:10 Saturation:4 Differentiati on7	Qref: 8 Q11 6 C11 3 Q12 5 C12 7	GOOD:7 STRETCH 4
R8	The typical user shall not average more than 5 minutes to install or upgrade any required GPM software on his/her personal computer.	Scale: time Unit minuets	X: 15 Competitor Y: 10 Competitor Z:7	Utility:13 Saturation:6 Differentiati on9	Qref: 10 Q11 9 C11 2 Q12 7 C12 5	GOOD: 10 STRETCH 7
R9	It shall not average more than 5 minutes to install or upgrade the software on employee's personal computers.	Scale: time Unit minuets	X:15 Competitor Y: 10 Competitor Z: 7	Utility:12 Saturation:6 Differentiati on 8	Qref: 9 Q11 8 C11 3 Q12 7 C12 7	GOOD: 9 STRETCH 7

R10	The typical seller shall be able to register an item for auction within 10 minutes.	Scale: time Unit minuets	X:20 Competitor Y: 15 Competitor Z:12	Utility:18 Saturation:1 1 Differentiati on13	Qref: 15 Q11 13 C11 4 Q12 12 C12 8	GOOD:14 STRETCH1 2
-----	---	---------------------------------	---	--	--	--------------------------

Lessons learned:

It was clearly understood to me that quality requirements is seen as a key competitive advantage in companies. The organisations operating on market-driven software product development should plan the product's releases effectively so that they can reach the market as early as possible with a competitive level of quality compared to the other products. Application of QUOER model is easy and this model has helped me to understand the positive impact of the quality requirements on the software product development as after the implementation of this method it is shown that by improving the product quality of the current state of the product, it can reach its future state. During the implementation process of this model, I got an idea on how to specify the quality requirements and how to compare the quality of our own product to the quality other products produced by the competitors. This implementation has also expended my knowledge on the importance of cost, benefit and roadmap view of the QUPER model and the importance of considering cost dependencies while selecting the requirements to different releases.

Reflection:

For the reflection of this model, another article is selected that implements the QUPER model for the purpose of relating my implementation and the experience of QUPER model. The article that I selected is "Aligning Quality Requirements and Test Results with QUPER's Roadmap View for Improved High-Level Decision-Making". In this article, a new view of the QUPER model, namely aligning QR targets and test results in QUPERs roadmap view. The alignment view of QR and test results was evaluated by 13 practitioners from one case company using semi-structured interviews for data collection. The results show that the alignment of QR and test results using QUPER's roadmap view was viewed as an important input to the decision-making process. It was also found out that the test results in the roadmap view it is possible to make the decision to accept the level of quality even if the test result shows it fails and by using the alignment in QUPER's roadmap, practitioners may have to spend less effort (and cost) to improve the level of quality. But there was one concern that was expressed by the practitioners was to have the level of quality included in the test results. From my point of view, I did not experience all the results except few that were found out in the paper because I have considered very few requirements. But I felt that this model has helped me to take right decisions y following the steps involved in this model. When discussing about the road mapping, from my experience I agree with the results in the other paper which says that this model allows you to spend less effort to improve the levels of the quality. I strongly agree to the results of that article because it is true that this model helps a lot in making a right decision at the right time.