

Reflective Report

TECHNIQUE-1

Article selection

Articles selected are [1], [2]

-- T. Gorschek and C. Wohlin, "Requirements abstraction model," *Requir. Eng.*, vol. 11, no. 1, pp. 79–101, 2006.

-- T. Gorschek, P. Garre, S. B. M. Larsson, and C. Wohlin, "Industry evaluation of the Requirements Abstraction Model," *Requir. Eng.*, vol. 12, no. 3, pp. 163–190, May 2007.

These two articles were selected in order to understand the work of Requirements Abstraction model and its application in industry. Paper 1 discusses about the working of requirements abstraction model.

About Requirement Abstraction Model (RAM)

RAM is a model supporting requirement engineering process which helps in making the requirements comparable to one other thereby helping the prioritization process. This model is developed keeping in mind the market driven development (MDD) where the requirements are more product oriented. Market driven development usually follows an incremental model whose primary goal is to select an optimal set of requirements for different releases (increments). Things like what requirements that a release should contain, what is the cost (resources) needed for implementing these requirements, decision of what features should be available depending on different customers, point at which quality should be taken into consideration determine the success of the product.

Implementation plan

In order to implement RAM, a guide given in paper 1 is followed. Coming to the requirements to work on, a set of 5 requirements are taken from hospital management system.

RQ1) Able to enter details of the patients

RQ2) Able to include other details about the patient

RQ3) Able to include billing information (insurance etc.)

RQ4) Doctor's information

Execution

Step 1: Specify the requirement

	RQ1	RQ2	RQ3	RQ4
Description	Hospital staff will enter the patient's details like their full names, photos, addresses, phone	Hospital staff will be able to upload patient's photos from MRI or similar machines to the system.	Hospital staff will be able to enter receipts of patient's care into the system.	Hospital staff will be able to enter doctor's information such as their full names, addresses, phone numbers, medical

	numbers, billing information etc.			profession, degrees and room numbers
Reason/benefit	Help user to keep track about medical record	Help users from bringing lots of papers while coming to hospital	Centralized for income tax issues	
Restriction/risks	Data should be secure	Data should only be accessed by patients		

Step 2: Place the requirements in respected level.

RQ1	Feature level
RQ2	Functional level
RQ3	Feature level
RQ4	Feature level

Step 3: Abstraction

	RQ1	RQ2	RQ3	RQ4
Product level	Ease of use	Technologically Advance	Tracking medical expenses	Promote the hospital
Feature level	-	Support for different user related medical records	-	-
Functional level	Details like name ,email id	-	Include the detailed costs related to hospitals such as pharmacy bills, consultation fee	Details including the doctors specialization, available hours
Component level	Email id is the unique identifier	File types supported are jpg,png,raw file.	Currency used is Indian rupees	Only govt recognized ,predefined specializations are allowed

Lessons learned

Applying this technique helps in understanding the requirement and level of requirements. it's also more clear to work with different abstraction levels which give most of what requirements actually is. Though it mandates issues being more specific requirement and also need expert judgement in understanding the abstraction level of requirement. Especially in large scale market driven domain

where there is continuous flow of requirements, this help in prioritization of the requirements and assist release planning process.

Usually for a company who are performing the requirement engineering process as a ad-hoc method implementing this method requires more time and improved results are realized slowly.

Reflection

Though I have understood the technique I am not able to apply it and use it to the maximum possible extent as I the no of requirements I have considered is very small and most of the requirements are from feature level of abstraction.

TECHNIQUE-2 QUPER model

Article selection

- R: Berntsson Svensson, B. Regnell (2015) "A case study evaluation of the guideline-supported QUPER model for elicitation of quality requirements", 21st International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'15), Essen, Germany, pp. 230-246, 2015.

– Berntsson Svensson & Olsson "Introducing support for release planning of quality requirements –an industrial evaluation of the QUPER model"

About QUPER

Release planning plays a major role in market driven development where the organizations compete in open market. To survive competition organization should be able to bring new, innovative, necessary features as early as possible to market which requires better more formal release planning. Along with value of the features considering quality requirements acts as added advantage. it addresses quality and cost constraints for a requirement

Implementation plan

In order to implement the QUPER model a detailed guide from article 1 is used which involves different steps.

Requirements related to an ecommerce website are taken into consideration

Feature1: loading a detailed description page

Feature2: Products page

Feature 3: Display images of the product

STEP1: Identify candidate Quality requirements

Features	Identified quality requirements
Feature 1	Response time
Feature 2	User experience
Feature 3	User-interface experience

STEP2: Define scale and unit

Identified quality requirements	Scale	Unit
Response time	Page load time	Time in millisecond(minimum better)
User experience	No of product per page	Integer number(optimal number)
User-interface experience	Resolution of the image	1024X1024 format(maximum the better)

STEP3: Identify reference levels

Identified quality requirements	Product	Reference levels
Response time	Own product	2000 milliseconds
	Competitor	1700 milliseconds
User experience	Competitor	30
	Own product	35
User-interface experience	Own product	240X240 - 1024X1024
	Competitor	512X512

STEP4: Elicit quality breakpoints

Identified quality requirements	Type of quality break points	Quality break points
Response time	Utility	3000
	Saturation	1200
	Differentiation	300
User experience	Utility	50
	Saturation	20
	Differentiation	10
User-interface experience	Utility	240X240
	Saturation	1920X1080
	Differentiation	

STEP5: Estimate cost barriers

Identified quality requirements	Qref	Present quality	RATIONALE:	cost
Response time	2000	3000	Software necessary	3 weeks
User experience	30	35	Software implementation	1 week
User-interface experience	512X512	240X240	Data base/networking support	4 weeks

STEP6: Set candidate requirements

Based on requirements value and discussions decided targets.

Identified quality requirements	Target	Reference levels
Response time	Good	2000
	Stretch	1500
User experience	Good	25-30
	Stretch	20-25
User-interface experience	Good	512X512
	Stretch	640X640

Lessons learned

Applying this technique helps in understanding the necessity of quality requirements which is necessary to improve number of satisfied customers. This technique helps in evaluating the current quality of the product, expected quality. Quality that competing products have can also be known by using this process which helps in analyzing of our current level of quality of the product. This process helps in building better products. From the articles referenced it can be seen that it does provide evaluation, but expected results are not seen.

Reflection

As the technique is only applied on the very less no of requirements cannot judge the complexity of the process. It can be said that it increases with increase in number of requirements. This process becomes more complex if the dependencies of the candidate requirements and quality requirements are considered. Impact can either be positive where improving one quality requirement also improves the quality of other quality requirement or negative where improving one quality requirements decreases the quality of other requirement.