Reflective Report

Articles selection:

The articles I choose for the assignment are,

1. Gorschek, Tony, and Claes Wohlin. "Requirements abstraction model"[1]
2. Carlshamre, Pär, et al. "An industrial survey of requirements interdependencies in software"[2]

Introduction:

Article 1:

RAM model breakdown the requirements up to four levels meeting the industrial needs. The article gives an idea of what a typical release should contain when it should be released and at what cost. The model helps in determining the scope of the project and inserting the requirements with in that scope. The model breaks down vague or abstract requirements to a detailed one, so that they can be inserted in the current scope of the project and developers can work on them.
Motivation behind selecting the article is, the model provides better visibility and traceability for the requirements. It also helps to prioritize, manage and compare the requirements which is helpful for the requirements engineers in the industry. In this article I would like to explore the pros and cons of the model being implemented in industry when a large scale project is considered.

Article 2:

The papers aim is to provide the nature of interdependencies between the requirements and categorize them to support release planning. The model helps in doing a release planning and has more version control. The model helps to differentiate which requirements to selected for release planning so that every release will have a working software. The model helps in identifying more dependencies in the same release.
Motivation for selecting the article is, the model address so many problems, if there are interdependencies between the requirements then the it causes a change in requirements while developing and finally the project needs to be reconsidered form the requirements level. The problem increases when there is a large scale of requirements. So to understand the concepts related to interdependency I opted for this model.

Implementation of article 1:

Based on the guidelines provided in the paper [1], I plan to work on four levels of abstraction:

1. Product level
2. Feature level
3. Functional level
4. Component level

I chose 17 requirements from the “course management system” of release planning assignment and planned to work on it. The requirements chosen are placed on the one of the above levels and work on the form there. The requirement when confirmed it can be in any level, that is product level is abstract level and component level is a more detailed requirement. The requirement which is in abstract level has to be broken down to be well understood. If the requirement is on an abstract level like “media support”, then the requirement is broken down till the component level i.e., functionality that should be added to the system to support the videos. Thus formed requirements are placed in the appropriate levels.

Example: “Search bar” is one of the requirements in release planning assignment. The requirement is placed in the feature level. And then it is worked on and that is broken down. The product level requirement will be “Searching system”, the functional level requirement will be “find relevant information” and the component level requirement is “Search bar is provided on the right corner of every webpage”.

Execution:

1. Product level: System Security
    a. Feature level: Authorized use of the system
    b. Functional level: 1. Signup, 2. Login, 3. Activity
        i. System should send an alert about the activity that is taking place when there is a login from other devices
        ii. When signing up a confirmation mail should be sent in order to check the authenticity.
    c. Component level:
        i. System must support traditional username and password credentials.
        ii. A system id should be presented to each user in the system.
2. Product level: System warning
    a. Feature level: alerts
    b. Functional level: 1. Incorrect login, 2. Successful login, 3. Change password
    c. Component level:
        i. When an incorrect login is attempted i.e., user enters unmatched username and password the system should show that an incorrect login is attempted.
        ii. When user is successfully logged in to the system then the system should send a mail stating a successful login.

      iii. When user changes password in the system, confirmation mail must be sent.

3. Product level: Course details
    a. Feature level: System should provide course details
    b. Functional level: 1. Every course should have a start page, 2. Start page should provide course details, aims and objectives
    c. Component level:
        i. Users with right privileges must access be able to edit content inside the course.
        ii. Users with right privileges must be able to apply for that course.
4. Product level: Course news
    a. Feature level: When the course news gets updated, alerts must be sent
    b. Functional level: 1. Participants in the course must be able to view the course news, 2. Participants in the course must get updates in the form of alerts when the news gets updated or added.
    c. Component level:
        i. The Users with right privileges should be able to edit and add or update the course news.
        ii. When a new item is added the system must send an update to the user by mail.
5. Product level: Course calendar
    a. Feature level: A calendar for the whole system
    b. Functional level: 1. Every course should have a calendar, which gets updated to the system calendar, 2. Participants of the course should be able to synchronize the calendar.
    c. Component level:
        i. Every calendar item must have the following: date, time, description.
6. Product level: System view
    a. Feature level: System should be able to support different views
    b. Functional level: 1. Users should have a personalized view of the system, 2. A predefined functionality is set and access should be provided according to that.
    c. Component level:
        i. The start page must be consisting of the following Course news, new messages, link to the profile, link to the participants, file archive, calendar.
        ii. User must be able to personalize the view on the start page.
7. Product level: Manage a course
    a. Feature level: Only course admin and course responsible should manage.
    b. Functional level: Managing is Administrating the participants in the course.
    c. Component level:
        i. Manage the participants by adding or removing them to the courses.
8. Product level: Personal profile
    a. Feature level: Every user in the system will have a profile.

   b. Functional level: A user must be able to add, edit and delete the comments in the profile.
   c. Component level:
      i. Personal profile contains the following information: first name, last name and social security number.
      ii. Additional information is: Address, e-mail id.
      iii. A link is provided in start page, which takes to the profile.
9. Product level: Interactions
   a. Feature level: System must integrate the discussion forms.
   b. Functional level: 1. Users should view and add new messages, 2. Users must be able to reply to the messages and also delete the messages.
   c. Component level:
      i. Participants of the course only should have access the course discussion form.
      ii. The format of the messages in the discussion: date, time, author, subject, content.
10. Product level: Users
    a. Feature level: The participants of the course are the users
    b. Functional level: It shall be possible to add or remove the course participants and the privileges are given to course responsible and system administrator.
    c. Component level: Appropriate buttons will be provided to the course responsible and administrator.
11. Product level: System storage
    a. Feature level: Course file archive.
    b. Functional level: 1. Files could be added to the course by the course responsible, 2. Users should be able to browse and also download.
    c. Component level: Privileges to add or remove the files will be given to course participants.
12. Product level: System users
    a. Feature level: Users should be distinguished by the roles.
    b. Functional level: Privileges are given to each participants of the course and certain functions will be limited to certain users.
    c. Component level: The roles are Course participator, Course responsible, and system administrator.
13. Product level: Course links
    a. Feature level: Attach links to the course
    b. Functional level: 1. System must support the functionality to attach the links, 2. It shall be possible to Create/Read/update/delete links.
    c. Component level: To follow up links are labeled.
14. Product level: Communication
    a. Feature level: Course chat room
    b. Functional level: In chat room it shall be made possible to chat among the course participants and also with the course responsible.
    c. Component level:

            i. When a new message is posted then mail alert shall be given to the recipient.

15. Product level: Media support
    a. Feature level: Course responsible shall be able to post videos.
    b. Functional level: Both the course participants and course responsible shall be able to view the videos.
    c. Component level: System should be made to support flash player.
16. Product level: Search system
    a. Feature level: A search bar is provided
    b. Functional level: Search can find any information available on the website.
    c. Component level: Search bar should be provided at the top of every webpage.
17. Product level: Schedule
    a. Feature level: Users are provided with a personal calendar
    b. Functional level: 1. Users should create, Read, Update, delete items on the calendar, 2. Synchronize the calendar to the systems calendar
    c. Component level: format must be date, time, description.

Implementation of article 2:

Based on the paper [2], there exists six interdependencies among the requirements they are,

1. AND – Functional relations
2. REQUIRES – Functional relation
3. TEMPORAL – can be either functional or value related
4. ICOST and CVALUE – value related
5. OR – value related

To proceed further the author instructs to prioritize the requirements without considering the interdependencies between the requirements. So the prioritization was done to all the course management system requirements. I chose to prioritize the requirements in grouping method as I had previous experience in release planning assignment. The prioritized requirements labeled as a must will be taken into consideration after a careful analysis. In order to compare the interdependencies between requirements 19 requirements of high priority are chosen. Then pairwise comparisons are performed between each of the requirements and interdependencies are identified.
Example: There are 19 requirements and by performing the pairwise comparisons the number of comparisons we get for R1 are {R1, R2}, {R1, R3}, {R1, R4}, {R1, R5}, {R1, R6}, {R1, R7}, {R1, R8},{R1, R9},{R1, R10},{R1, R11},{R1, R12},{R1, R13},{R1, R14},{R1, R15},{R1, R16},{R1, R17},{R1, R18}, {R1, R19}. For each pair the above defined interdependencies may or may not exists. I worked on excel sheet to implement the following model, though large scale might require a database. The spreadsheet I worked on is attached in the appendix.

Discussion:

Article 1:

RAM model could be easily understood and so the implementation is much simpler.

1. Placing the requirements on different levels of abstraction was easy i.e., choosing a suitable level for placing the requirements was not a difficult task to do. But talking on a large scale contrast the requirements engineers has to work heavily and taking decisions on what is the suitable level is a hectic task. At the starting of the project I felt difficult in choosing the right suitable level for the requirement. But after a careful analysis I could make a decision and overall it is easy to implement the first part. The problem was lack of experience, though in industry it could be easily implemented.
2. For the second part, the requirements had to be broken down or worked up in order to get the requirements on all levels. It was easy to break down some requirements but it was very much difficult to break down lower level requirements. It occurred to me that I was placing the requirements at wrong levels but this is due to lack of experience. Considering large scale and the industry, maintaining a traceability track record is important. I implemented on 17 requirements and result was 78 low level requirements. When the requirements are in 1000's there might be approximately 10000 low level requirements and maintaining traceability is an important task to do.

Considering MDRE process, when the requirements intake is high then the complexity is high depending on the resources that gets you requirements [3]. So maintaining a track record on traceability is important task, as I experienced it while handling only 78 requirements. By the implementation of the model it is clear that the abstraction is necessary as it does give a clear idea on implementation of the whole project.

Article 2:

Understanding the concepts on interdependencies between the requirements was easy, but I had a few hiccups while implementing the model. The pairwise comparisons are difficult to make as there are so many confusions. At times I got confused as the requirements consisted of 2 types of interdependencies. Finally, all the comparisons are made and the following were the statistics, REQUIRES – 20, AND – o6, CVALUE – 2, OR – 03. Its is clear from the above statistics that the interdependencies most of them were functionally related and by this you can tell that the it's a new product[4]. After the implementation of the model it is clear that to have a functional product for each release plan requirements should be chosen carefully. If there are any interdependencies then the whole project should be reconsidered, postponing the deadline.

Considering MDRE process, when the requirements are in 1000's it is difficult to manage the interdependencies between the requirements and so handling the release planning is also a difficult task. I used an excel sheet to demonstrate but that is confusing and I would like to use tools in industry to simplify the task.

Conclusions:

In the MDRE process, the requirements may come from users, developers and competitors. It is difficult to manage the requirements for the analysts. RAM breaks down the requirements from abstract level to a low level while providing the traceability for the acquired requirements. Through my experiences the model can also be extended with an integration with the quality requirements at different abstraction levels. RAM model can be extended with certain models like Requirements triage [5] to further simplify the process of requirement analysis.

While implementing the next model I had a few issues like using spreadsheet was difficult while making the comparisons so the model should be extended with a tool so the process becomes easy. Requirements should be analyzed carefully in order to have a functional product. The model is best when used in incremental method, as the versions could be planned easily.

References:

[1] T. Gorschek, C. Wohlin, Sektionen för teknik – avd. för programvarusystem, Blekinge Tekniska Högskola, Blekinge Institute of Technology, and School of Engineering - Dept. of Systems and Software Engineering, “Requirements Abstraction Model,” *Requir. Eng.*, vol. 11, no. 1, pp. 79 – 101, Mar. 2006.

[2] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag, “An industrial survey of requirements interdependencies in software product release planning,” in *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, 2001, pp. 84 – 91.

[3] B. Regnell and S. Brinkkemper, “Market-driven requirements engineering for software products,” in *Engineering and managing software requirements*, Springer, 2005, pp. 287–308.

[4] T. Gorschek and A. M. Davis, “Requirements engineering: In search of the dependent variables,” *Inf. Softw. Technol.*, vol. 50, no. 1, pp. 67–75, 2008.

[5] M. Khurum, K. Aslam, and T. Gorschek, “A method for early requirements triage and selection utilizing product strategies,” in *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific*, 2007, pp. 97–104.