UNIVERSITY OF LIVERPOOL

Requirements Engineering Assignment Part 2: Discussion

Department of Electrical Engineering & Electronics

November 17, 2021

Declaration

I confirm that I have read and understood the University's definitions of plagiarism and collusion from the Code of Practice on Assessment. I confirm that I have neither committed plagiarism in the completion of this work nor have I colluded with any other party in the preparation and production of this work. The work presented here is my own and in my own words except where I have clearly indicated and acknowledged that I have quoted or used figures from published or unpublished sources (including the web). I understand the consequences of engaging in plagiarism and collusion as described in the Code of Practice on Assessment.

Contents

1 Question B

Make a list of requirement elicitation techniques and thoroughly summarise each technique. Which one is arguably the most popular and why?

1.1 General Introduction

Requirement elicitation is the prerequisite for designing products that meet customers' needs and different types of elicitation are suitable for various circumstances. Commonly used elicitation techniques are (1)Analysis of existing documentation (2)Interviews (3)Questionnaires (4)Group brainstorming (5)Focus groups (6)Direct observation and (7)Prototyping [1]. The following are the specific explanations about each elicitation technique.

1.2 A list of requirement elicitation techniques

1. **Analysis of existing documents**

   This method is applied in conditions where previous products or attempts to meet customers' demand have been made and left records [2]. Engineers can have a grasp of the existing systems, the standard operation procedures, the manuscripts, errors occurred and users' responses. After recognizing what customers want and the deficiency of existing systems, engineers can better design their interviews and questionnaires with stakeholders.

2. **Interviews**

   Engineers have a face-to-face communication with stakeholders of the project, first designing questions and flexibly adjusting the conversation to dig further based on the interviewers' response. The persons to be interviewed should be carefully selected since different stakeholders have distinct concerns so the leader of the business, project manager, users and even legal consultant should all be included. Therefore, engineers can get essential information such as the client's priority to be solved, the users' preferences and the function demand of the software.

3. **Questionnaires**

   This method can quickly gather information from users or potential users of a large scale to know customers' opinions of existing programs or expectations. Questions should cover different perspectives of a program which are listed in a easy-to-answer way as well as contain open questions to obtain unexpected advice. An effective questionnaire is hard to design since questions can be ambiguous and may not have answers that truly reflected the respondents' thoughts.

4. **Group brainstorming**

   A number of stakeholders, namely the client, users, technical experts gather together in an informal environment to generate new ideas [3]. People involved just shout out ideas, which are not allowed to be criticized or discussion, only left to be determined after the brainstorming is over. It's a good way to use when creative ideas are needed but not appropriate for making a decision.

5. **Focus groups**

   This approach aims to use team dynamics to elicit a richer understanding of requirements [2]. Five to ten representatives of different aspects meet together, the users included,

to express ideas about system and spontaneous reactions are recorded [2, 3]. This is a good way to draw potential conflicts which have to be resolved later.

6. **Direct observation**

   This method refers to engineers observe the users conducting tasks with the existing systems, which can help observers get direct and accurate information in a working situation, observing how the users exploit the system and the problems encountered. Engineers keep records or videos if necessary for further research and can find out users' potential needs. One form of refinement is apprentice, where the engineer actually performs the tasks of the user and gains critical insight into the requirements [3].

7. **Prototyping**

   Engineers design a prototype for clients or users to get feedback which clarify the requirements' need and facilitate future design. This method applies for conditions with lots of uncertainty and the feedback can then used as basis for a questionnaire.

1.3 The most popular method

Interview is the most popular method for the following reasons:

1. It can directly get stakeholders' requirements without the misinterpretations of different parties. Engineers can find out reasons of customers or users' needs and make corresponding products.

2. Clients and users may having difficulty identifying and expressing their needs and engineers can design the questions to get information they need and flexibly change contents according to the interviewees' response to dig out what they truly want.

3. The interview details can be recorded and mutually checked for future references.

4. It takes comparatively less time and less difficulty to carry out interviews and collect first-hand and effective information.

2 Question C

Discuss the significance and possible advantages of using prototyping, including the different types of prototyping. Explain some problems which might arise when using prototyping.

2.1 General introduction

Prototyping enables users to get a preliminary idea of what a system looks like and its functionality before the engineers actually finish building it. It provides a direct communication between users and engineers, operation of prototypes delivers engineers designing principles and how the system will function in general to users meanwhile users give feedback to engineers for guiding future design. In this way, users can better know their needs and whether the system meet those need in early stage of projects, thus reduce the frequency of future changes and the overall costs of designing the system [1].

2.2 Different types of prototyping

Commonly used prototyping is categorized into (1)Throw-away prototyping (2)Evolutionary prototyping (3)Incremental prototyping [3].

1. **Throw-away prototyping**

   Throw-away prototyping is explicitly designed for eliciting requirements, using a preliminary program to make sure engineers really understand users' requirements and to exhibit how their requirements will be served in the completed system, but the prototype will not be the final system [1, 3]. With the use of throw-up prototypes, errors in the programming or operation will reveal and then engineers can correct or improve them in the final system. At the same time, users may find inadequacy of the system, such as unmet demands or difficulty of operating the system in certain stage. It's a communication platform where clients and developers clarify requirements and the feasibility of those requirements, which is beneficial for further design process [3].

2. **Evolutionary prototyping**

   Evolutionary prototype is an advanced version of prototype that will constitute the core of the final system and will be constantly revised and improved. Requirements gathered in the initial stage were taken consideration into and implemented in the refining process. Until the final system is delivered, evolutionary prototype can serve as an interim basis [1].

3. **Incremental prototyping**

   Incremental development combines evolutionary methods with more traditional software development disciplines. It relies on building requirements and delivery systems incrementally [3]. Parts of the systems are developed separately and then combined together to form a whole system. Any typical individual process is first gathering information, then designing and coding, the next step is to do a test, followed by integrating and maintaining. All the steps before integrating can keep a record and are under engineers' planning and management so there are standards to follow. Testing makes user feedback possible, which better fulfills requirements.

2.3 Possible problems when using prototyping

1. **Throw-away prototyping**

   Since the prototype is not a part of final product, a carefully designed one may waste time and resources. On the one hand, if something goes wrong in prototype, it's hard to define whether it is due to unreasonable requirements or limitation of the prototype; On the other, if clients are satisfied with the prototype, they may just use it and refuse to pay for the final product [3].

2. **Evolutionary prototyping**

   However, the prototyping is constantly changing, which means the design and refining process is hard to record, so the design rationale can't be kept and engineers have no standard to follow. The change occurs so rapidly that the prototype now being evaluated will soon become obsolete, then the evaluation becomes unnecessary [3]. In addition, every evaluation leads to further improvement, which can last forever. Project planning also becomes meaningless because there is no clear schedules for the process.

3. **Incremental prototyping**

   The main problem of incremental prototyping is the integration since it needs to combine various separate components. Engineers must have a clear understanding of the core requirements which can be separated into different segments, then each requirement is dealt with respectively [3]. Otherwise, components may not be compatible with each other or the components together can't meet all clients' needs. The design also has to be made very early for the following steps .

References

[1] D. McIntosh, "Requirements engineering," https://liverpool.instructure.com/courses/46000/pages/requirements-engineering?module_item_id=1258998, 2021.

[2] B.Nuseibeh and S.Easterbrook, "Requirements engineering: A roadmap," https://liverpool.instructure.com/courses/46000/pages/requirements-engineering?module_item_id=1258998, Imperial College and University of Toronto, 2021.

[3] E.Kazmierczak, "Requirementsengineering," http://ww2.cs.mu.oz.au/~dmwilm/downloads/641.pdf, The University of Melbourne, 2003.