mike6606

Nickname: mike6606
Age: 8 years 10 months
Fans: 0
Followers: 5
+ Plus follow

Search

[Look for it]

[Google search]

Frequently used links

My essays
My comment
My involvement
Latest comments
My labels

Essay archives

Read the leaderboard

Recommended leaderboards

Software Engineering Practitioners' Research Methods Chapter 8 Answers

Problem:

Why is it that many software developers don't pay enough attention to requirements engineering? Are there ever circumstances where you can skip it?

Answer:

Designing and building an elegant computer program that solves the wrong problem server no one needs. That's why it is important to understand what the customer wants before beginning to design and build a computer – based system.

But many software developers do not pay enough attention to requirements engineering, because in developers point of view –

• After all, doesn't the customer know what is required ?

• Shouldn't the end users have a good understanding of the features and functions that will provide benefit? The view is not at all correct.

Requirement engineering helps software engineers to better understand the problem they will work to solve. It encompasses the set of tasks that lead to an understanding of what the business impact of the software will be, what the customer wants and how end – users will interact with software we cannot directly skip requirement engineering but v=can be given less importance if the software to be built is very familiar.

Problem:

You have been given the responsibility to elicit requirements from a customer who tells you he is too busy to meet with you. What should you do?

Answer:

Just asking a stakeholder what their requirements are rarely works. It is said that the customer is always right. In fact the customer may be busy and normally far more urgent things to do rather than speaking to someone in a suit about requirements for a system that is not even due to be delivered.

The first thing to be sure of when eliciting requirements is that we have to get the stakeholder into a state in which they want to talk to us.

Prepare a set of questionnaire and ask him to respond in his free time.

Use the internet facility to overcome the problem of collecting requirements from a busy customer.

Be straight forward, look for the main central idea rather than looking for unnecessary defaults.

However, if the customer refuses to work with you, it's time to get both your management and the customer's management involved. If they don't have the time to help define the software, they probably won't have the inclination to use it.

Problem:

Discuss some of the problems that occur when requirements must be elicited from three or four different customers.

Answer:

The following are the problems that occur when requirements must be elicited from three or four different customers

• The requirements of the customer cannot be understood easily. The customer's requirements will change over time such that a customer with a set of requirements at one time can include another set of requirements afterward.

• It is very difficult to understand the requirements of the customers.

• The customers will have a wide range of expectations such that it may lead to disappointments at most of the time.

• The customers will change their requirements rapidly.

• Each and every customer will have some set of requirements such that their expectations can be at different level. Therefore, their expectations may not get fulfilled at most of the time.

• The customers should have the knowledge to answer the questions to be asked by the developmental team members. The customers should specify the required input and output clearly.

• The customers may get satisfied or may not get satisfied. They are the decision makers to review a product whether it matches their requirements or not.

To avoid these problems, the customer with some knowledge about the requirements should be available to the developmental team of the product at any time and answerable to the questions asked by them for the good performance and functionality of the product. Also, the customer should be very clear about the input and output data.

Problem:

Why do we say that the requirements model represents a snapshot of a system in time?

Answer:

The intent of the analysis model in to provide a description of the required informational, functional and behavioral domains for a computer based system. The model changes dynamically as software engineers learn more about the system to be built, and stakeholders understand more about what they really require. For that reason, analysis model is a snapshot of requirements at any given time. We expect it to change.

As the analysis model evolves, certain elements will become relatively stable, providing a solid foundation for design. But some elements of model may be more volatile. Indicating the customer doesn't yet fully understand requirements for the system.

Problem:

Let's assume that you've convinced the customer (you're a very good salesperson) to agree to every demand that you have as a developer. Does that make you a master negotiator? Why?

Answer:

There should be no winner and no loser in an effective negotiation. Both sides win because a "deal" that both can live with is solidified.

There is no question of master negotiator. A developer may try to convince the customer but he should remember the "Win –Win" result. The customer wins by getting the system that satisfies the majority of customer's needs and the software tam wins by working to realistic and achievable budgets and deadlines. Then is negotiation becomes master negotiation but not by convincing the customer as a salesperson.

Problem:

Develop at least three additional "context-free questions" that you might ask a stakeholder during inception.

Answer:

Context-free questions are said to be the questions that are asked during the development of a project or the project which is under construction. These questions will be used to identify the positive and negative sides of a project. These questions will give the clarity for the development of the project.

The following are the context-free questions that would be used to ask during the inception of a project.

• What are the barriers to the development of this project?

• What kind of benefits does the project will provide and who are going to get these benefits?

• Who is going to provide the payment for the work?

These questions will be asked at the inception phase of a project for the development of the project.

Problem:

Develop a requirements-gathering "kit." The kit should include a set of guidelines for conducting a requirements-gathering meeting and materials that can be used to facilitate the creation of lists and any other items that might help in defining requirements.

Answer:

Facilitated Requirements gathering "kit"

A project manager's tool kit includes different collaborative sessions. In that requirement gathering is a major session. Ii consists of

1. Team building

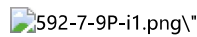2. Team communication

3. Change management

2. Prototyping and early, reiterative user testing of designs.

3. The re-use of software components.

4. A good schedule is necessary to design improvements to the next product version.

Requirements gathering session kit

This session is designed to aid in the gathering of requirements to support a project. This session is of great value when there is a need to gain alignment & consensus between stakeholders as to the requirements of a project.

Outcomes & deliverables of session kit

1. Accelerated requirement gathering

2. Consensus among user groups around project requirements.

3. Reduced rework.

4. Establishment of requirement priorities

5. Increased ownership for the project

6. Join contribution & alignment between business and IT stakeholders.

![592-7-9P-i1.png\"

Problem:

Your instructor will divide the class into groups of four or six students. Half of the group will play the role of the marketing department and half will take on the role of software engineering. Your job is to define requirements for the *SafeHome* security function described in this chapter. Conduct a requirements-gathering meeting using the guidelines presented in this chapter.

Answer:

Requirements gathering essentials:

• Fours and clarity

• Format for specifying requirements

• The author of requirements document

• The language of requirements

• Accuracy is critical

• Minimizing risk of errant interpretation

• Conclusion

Safe home security function.

• Access camera surveillance via the internet

• Security & secret cameras

• Configure sate home system parameters

• Set alarms

• Activate sensors

Problem:

Develop a complete use case for one of the following activities:

a. Making a withdrawal at an ATM.

b. Using your charge card for a meal at a restaurant.

c. Buying a stock using an online brokerage account.

e. An activity specified by your instructor.

Answer:

ATM System:–

592-7-11p-i1.png\"

Problem:

What do use case "exceptions" represent?

Answer:

The purpose of exceptions in use-case is to identify some of the situations that are not covered in the preliminary use case.

• The situations that are identified in exceptions are covered while refining the preliminary use case.

• The use case must be complete and must deliver proper meaning to the user.

In the process of redefining the preliminary use case, at each step, various questions should be asked to get a clear picture of the purpose of that step.

Some of the questions to be asked are:

• Is there any alternative action that can be taken by the actor?

• Will there be any errors conditions to be handled that the user may encounter?

Problem:

Write a user story for one of the activities listed in question 8.9.

Answer:

User stories are used to manage requirements of a system. It is a short statement of required function. Usually a user story's structure is:-

As a ,

I want to ,

So that .

A user story for also contains acceptance criteria, which verifies the correctness of the implementation of user story. An acceptance test are based on the acceptance criteria. Usually it contains Given, When and Then clauses.

The advantages of user story for stating requirements are:-

• They are easy to understand by everyone

• They lay emphasis on verbal communication between stakeholders

• They are used for iterative development and as a result details can be deferred for next iteration.

User story for making a withdrawal at an ATM can be stated as:-

As a Customer,

I want to withdraw cash from my account in an ATM,

So that I can withdraw money without going to the bank and stand in a line.

In this case, one user story acceptance criteria can be defined as:-

Given that

• Users account has that much of money,

• Customer's ATM card is valid and

• ATM machine has cash.

When the customer makes a cash transaction

Then ensure that the

• Customer's account is debited and

• customer's account does not have required balance,

• card not being valid or

• ATM machine does not have the required cash.

" " Problem:

Consider the use case you created in question 8.9, write a nonfunctional requirement for the application.

Answer:

A software might not be useable if it does not satisfy its non-functional requirements (NFR). They specify the criteria for checking the operational quality of the system. They can be Performance, Usability, Reliability, Software Quality, Security and Safety Requirements.

For making a withdrawal at an ATM system application, the NFR can be:-

• A performance requirement can be that each bank should be able to process transactions from several ATMs at the same time.

• A safety requirement is that the temperature of the ATM center should be controlled to prevent the machines overheating.

• Security requirement are

o A security guards must be present at the ATM at all times.

o A camera should be installed inside the ATM to capture and stores the videos in the ATM.

• A Quality requirement can be that the relevant communication with the ATM machine should have a readable font size.

For using a charge card for a meal at the restaurant application, the NFR can be:-

• Card number or pin, if entered, should be hidden from others.

• The system should display a transaction successful or failure message within 30 seconds of submitting information.

• The menu and ingredients at the restaurant are modified frequently, this should be done without a new system release each time.

• If the card machine is down for some reason payment by cash should be an option.

For buying a stock using an online brokerage account application, the NFR can be:-

• The online system should provide a functionality to recover transactions whenever the system is down.

• The time it takes to buy or sell stock should be almost the same as when buying from a company office.

• The system should work on desktops, laptops, tablets and mobile device platforms.

• A transaction failure or success message should be displayed to user within 30 seconds of information submission.

For searching for books (on a specific topic) using an online bookstore application, the NFR can be:-

• Result of the search should be displayed to user within 30 seconds of information submission.

• The system should work on desktops, laptops, tablets and mobile device platforms.

• The site should be able to smoothly run on all operating system, with all types of browser.

• If books are out of stock a relevant message should be displayed along with a result.

For an activity specified by an instructor, the non-functional requirements can be:-

• The instructor should pass the directions for activity to be best at 90% of the time in easily understandable language.

• The instructor can effectively instruct only 11 students at a time.

• The activity and its inherent steps should be explicitly specified by the instructor unambiguously.

Problem:

Describe what *an analysis pattern* is in your own words.

Answer:

Analysis pattern:

We can notices that certain things reoccur across all projects with in a specific application domain. These can be called as analysis patterns and represent something within the application domain that can be reused when modeling

Intent: Describes what the pattern accomplishes & represents.

Motivation: A scenario that illustrates how pattern can be used to address the problem.

Forces and context: A descriptor of external issues that can affect how the pattern is used

and also the external issues that will be resolved when pattern is

applied.

Solution: A description of how the pattern is applied to solve the problem with an

emphasis on structural and behavioral issues

Consequences: Addresses what happens when the pattern is applied.

Design: Discusses how the analysis pattern can be achieved through the use of known

design patterns.

Known uses: Examples of uses within actual system.

Related patterns: one or more patterns that are related to the named pattern.

Problem:

Using the template presented in Section 8.5.2, suggest one or more analysis pattern for the following application domains:

> a. Accounting software.
>
> b. E-mail software.
>
> c. Internet browsers.
>
> d. Word-processing software.
>
> e. Website creation software.
>
> f. An application domain specified by your instructor.

Answer: Problem:

What does *win-win* mean in the context of negotiation during the requirements engineering activity?

Answer:

The customer and the developer enter into a process of negotiation, where the customer may be asked to balance functionality performance and other product or system characteristic against cost and time to market. The intent of negotiation is to develop a project plan that meets the needs of the customer while at the same time reflecting the real – world constraints that have been placed on software team.

The best negotiations strive for a "Win – Win" result. ie.......... the customer wins by getting the system that satisfies the majority of the customer's needs and the software team wins by working to realistic and achievable budgets and deadlines.

Win – win result can' be achieved by successful completion of

1. Identification of system or subsystem's key stakeholders

2. Determination of the stokeholds "Win conditions".

3. Negotiate of stakeholders win conditions to reconcile them into a set of win – win condition for all concerned.

Answer:

It is extremely desirable to detect errors in the requirements before the design and development of the software begin. The basic objective of the requirements validation is to ensure that the SRS reflects the actual requirements accurately and correctly.

Omission, inconsistency, incorrect fact and ambiguity are the common errors in requirements. As requirements are generally textual documents that cannot be executed, inspections are eminently suitable for requirement validation. Requirement validation reviews are conducted to uncover the errors that commonly occur during requirements gathering.

Because requirement specification formally specifies something that is originally existed informally in people's minds, requirements validation must involve the clients and the users.

Problem:

What five tasks make up a comprehensive requirements monitoring program?

Answer:

Requirement monitoring aims to trail a system's runtime activities so as to detect aberrations from its requirement specification specified earlier. Requirement monitoring program are used when stakeholder choose to use incremental software development.

The five tasks that make up a comprehensive requirement monitoring program are:-

• Distributed debugging: is used to find out error and the reason for their occurrence.

• Run time verification: is used to check if software matches its earlier specified specification.

• Run time validation: is used to check if the evolving software meets earlier specified user goals.

• Business activity monitoring: is used to check if the evolving software meets earlier specified business goals.

• Evaluation and co-design: is used to provide information to various stakeholders as the system progresses.

Solution Chapter 8: UNDERSTANDING REQUIREMENTS

8.1) Understanding the requirements of a problem is among the most difficult tasks that a software engineer face since requirements change continuously, hence they tend to pay little attention to it. In some cases, an abbreviated approach may be chosen. In others, every task defined for comprehensive requirements engineering must be performed rigorously. Requirements engineering builds a bridge to design and construction and cannot be skipped.

8.2) You might try using an approach like QFD that makes use customer interviews and observation, surveys, and examination of historical data (e.g., problem reports) as raw data for the requirements gathering activity. These data are then translated into a table of requirements—called the customer voice table—that is reviewed with the customers later. A variety of diagrams, matrices, and evaluation methods are then used to extract expected requirements.

8.3) In reality, the customer and the developer enter into a process of negotiation, where the customer may be asked to balance functionality, performance, and other product or system characteristics against cost and time to market. The intent of this negotiation is to develop a project plan that meets the needs of the customer while at the same time reflecting the real-world constraints (e.g., time, people, budget) that have been placed on the software team, Unfortunately, this rarely happens, each customer has his own views. These views donot match each customer, time is another constraint that matters, each customer may not have time to meet the developer and give the requirements, this further increases the problem.

8.4) The intent of the requirements model is to provide a description of the required information, functional, and behavioral domains for a computer-based system. The model changes dynamically as software engineers learn more about the system to be built, and stakeholders understand more about what they really require. For that reason, the analysis model is a snapshot of requirements at any given time.

subsequent software engineering activities.

8.6) The first set of context-free questions focuses on the customer and other stakeholders, the overall goals, and the benefits. For example, the requirement engineer might ask:

- Who is behind the request for this work?

- Who will use the solution?

- What will be the economic benefit of a successful solution?

- Is there another source for the solution that you need?

8.7) Answers will vary

8.8) Answers will vary

8.9) Answers will vary

8.10) Use-cases describe scenarios that will be perceived differently by different actors. The exceptions portion of the use case describes error conditions that can cause interference with normal completion of the use case (and may specify the work around to prevent to restore normal system operation).

8.11) Answers will vary

8.12) Answers will vary

8.13) Anyone who has done requirements engineering on more than a few software projects begins to notice that certain things reoccur across all projects within a specific application domain. These can be called "analysis patterns" and represent something (e.g., a class, a function, a behavior) within the application domain that can be reused when modeling many applications.

8.14) Answers will vary

8.15) A "Win-Win situation is where the customer wins by getting the system or product that satisfies the majority o the customer's needs and the software team wins by working to realistic and achievable budgets and deadlines.

8.16) When the requirement validation uncovers an error, it has each requirement against a set of checklist questions. It then has a review team looking into it. The review team includes software engineers, customers, users, and other stakeholders who examine the specification looking for errors in content or interpretation, areas where clarification may be required, missing information, inconsistencies (a major problem when large products or systems are engineered), conflicting requirements, or unrealistic (unachievable) requirements.

8.17) Distributed debugging, runtime verification, runtime validation, business activity monitoring, evolution and codesign

mike6606
followers - 0 followers - 5

« Previous: Software Engineering Research Methods for Practitioners Chapter 7 Answers
» Next: Software Engineering Research Methods for Practitioners Chapter 9 Answers

Log in to view or post comments, log in now or browse the blog home page

【Alibaba Cloud】2-core 2G cloud server as low as 99 yuan/year, <> cloud products enjoy non-stop discounts

Editor's Choice:
· Modern Image Performance Optimization - Fault Tolerance and Accessibility of Image Resources
· Talk: The Evolution of Service Architecture
· I'm on timeout with Redis again
· Clever use of CSS variables to create high-end grid animations
· Record the optimization of a lock

Reading Ranking:
· 3 mistakes to avoid when using Vue 10
· ASP.NET Core Web API interface throttling
· 【Fault Bulletin】CC attack is coming again, making March worse ·
Modern Image Performance Optimization and Experience Optimization Guide - Fault tolerance and accessibility of image resources
· How to JWT authenticate and authorize WebAPI in Net6.0