CS6403 Software Engineering

TADR (Think, analyse, discuss and respond) – 4 Answers

---

1. Consider a ticket-issuing system:

An automated ticket machine sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination and the type of ticket required. Once a destination has been selected, the ticket price is displayed and customers are asked to input their credit card. Its validity is checked and the user is then asked to input their personal identifier (PIN). When the credit transaction has been validated, the ticket is issued.

Write a set of non-functional requirements for the ticket-issuing system, setting out its expected reliability and response time.

Possible non-functional requirements for the ticket issuing system include:
1. Between 0600 and 2300 in any one day, the total system down time should not exceed 5 minutes.
2. Between 0600 and 2300 in any one day, the recovery time after a system failure should not exceed 2 minutes.
3. Between 2300 and 0600 in any one day, the total system down time should not exceed 20 minutes.
All these are availability requirements – note that these vary according to the time of day. Failures when most people are traveling are less acceptable than failures when there are few customers.
4. After the customer presses a button on the machine, the display should be updated within 0.5 seconds.
5. The ticket issuing time after credit card validation has been received should not exceed 10 seconds.
6. When validating credit cards, the display should provide a status message for customers indicating that activity is taking place.
This tells the customer that the potentially time consuming activity of validation is still in progress and that the system has not simply failed.
7. The maximum acceptable failure rate for ticket issue requests is 1: 10000.

The acceptable number of incorrect tickets is not specified as this depends on whether or not the system includes trace facilities that allow customer requests to be logged. If so, a relatively high failure rate is acceptable as customers can complain and get refunds. If not, only a very low failure rate is acceptable.
Obviously, these requirements are arbitrary and there are many other possible answers.

2. Once a system has been installed and is regularly used, new requirements inevitably emerge. What could be the reasons?

   Once a system has been installed and is regularly used, new requirements inevitably emerge. This is partly a consequence of errors and omissions in the original requirements that have to be corrected. However, most changes to system requirements arise because of changes to the business environment of the system:
   1. The business and technical environment of the system always changes after installation. New hardware may be introduced and existing hardware updated. Business priorities may change, and new legislation and regulations may be introduced that require system compliance.
   2. The people who pay for a system and the users of that system are rarely the same people. System customers impose requirements because of organizational and budgetary constraints. These may conflict with end-user requirements, and, after delivery, new features may have to be added for user support if the system is to meet its goal.
   3. Large systems usually have a diverse stakeholder community, with stakeholders having different requirements. Their priorities may be conflicting or contradictory. The final requirements are inevitably a compromise, and some stakeholders have to be given priority.

3. 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?

   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.

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

   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.

5. 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?

   The best negotiations strive for a "win-win" result, hence that does make you a master negotiator. Successful completion of these initial steps achieves a win-win result, which becomes the key criterion for proceeding to subsequent software engineering activities.

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

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

7. What do you think happens when requirement validation uncovers an error? Who is involved in correcting the error?

   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. Why is it so important that the specification document should have no omissions, contradictions, or ambiguities?

Any such omissions, contradictions, and ambiguities will be carried over into the design and implementation.

9. Consider the following recipe for grilled pockwester. Ingredients:
   1 large onion
   1 can of frozen orange juice
   Freshly squeezed juice of 1 lemon
   1 cup bread crumbs
   Flour
   Milk
   3 medium-sized shallots
   2 medium-sized eggplants
   1 fresh pockwester
   1/2 cup Pouilly Fuissé

1 garlic
Parmesan cheese
4 free-range eggs

The night before, take one lemon, squeeze it, strain the juice, and freeze it. Take one large onion and three shallots, dice them, and grill them in a skillet. When clouds of black smoke start to come off, add 2 cups of fresh orange juice. Stir vigorously. Slice the lemon into paper-thin slices and add to the mixture. In the meantime, coat the mushrooms in fl our, dip them in milk, and then shake them in a paper bag with the bread crumbs. In a saucepan, heat 1/2 cup of Pouilly Fuissé. When it reaches 170°, add the sugar and continue to heat. When the sugar has caramelized, add the mushrooms. Blend the mixture for 10 minutes or until all lumps have been removed. Add the eggs. Now take the pockwester, and kill it by sprinkling it with frobs. Skin the pockwester, break it into bite-sized chunks, and add it to the mixture. Bring to a boil and simmer, uncovered. The eggs previously should have been vigorously stirred with a wire whisk for 5 minutes. When the pockwester is soft to the touch, place it on a serving platter, sprinkle with Parmesan cheese, and broil for not more than 4 minutes.

Determine the ambiguities, omissions, and contradictions in the preceding specifi cation.
(For the record, a pockwester is an imaginary sort of fi sh and frobs is slang for generic hors d'oeuvres.)

Ambiguities:

Is a clove or a head of garlic to be used?
What operations are to be performed the night before? Everything?
Which juice is to be strained, orange or lemon?
What is to be frozen, the lemon or just the juice?
Is the onion to be diced, or just the shallots?
Which mixture is to be blended for 10 minutes?
Does "blend" mean "mix by hand" or "use an electric blender?"
How are the lumps to be removed, with a strainer or a blender?
The lumps in the mixture could be the mushrooms.
Two mixtures are specified, the pockwester is added to only one of them: which?
How can you kill a "fresh" pockwester?
The terms "simmer," "bite-sized," and "soft to the touch" are imprecise.
We can broil for any time from 1 microsecond to 3 minutes, 59 seconds.
What is to be sprinkled with Parmesan — the serving platter, or the pockwester?

Omissions:

How large is a "large onion"?
What size can of orange juice?

How large a lemon?
What kind of bread for the crumbs?
How much flour?
How much milk?
How large are medium-sized shallots?
How large are medium-sized eggplants?
What size pockwester?
What size garlic?
How much Parmesan cheese?
What size eggs?
The sugar is not listed in the ingredients.
Nor are the mushrooms.
Nor are the frobs.
What size platter is to be used?

Contradictions:
The ingredients call for frozen orange juice, the recipe calls for fresh.
After the lemon has been squeezed the night before, it is then supposed to be sliced.
We are told to "stir vigorously," "slice," and "in the meantime" to coat, dip, and so on — it is impossible to do all these things at once.

The following items are certainly confusing, but they are not ambiguities, omissions, or contradictions:
The lemon juice, eggplant, and garlic are never used.
Grilling in a skillet is not usually a good idea.
Shaking up mushrooms wet with milk in a paper bag is fraught with difficulties.
The instruction to whisk the eggs should precede the instruction to add them to a mixture.
If the blending is to be done with an electric blender, 10 minutes seems to be too long.
The recipe calls for first coating the mushrooms in flour, and then dipping them in milk; this may be unorthodox, but then so is much of the rest of the recipe!
Two mixtures are specified, the pockwester is added to only one of them; nothing is mentioned regarding what is to be done with the other one.

10. What are the strengths of informal specifications?

Informal specifications are easy to use, and are easy for the client to understand.

11. What are the weaknesses of informal specifications?

Informal specifications are imprecise. They can be ambiguous, contradictory, or incomplete.

12. Why are many software organizations reluctant to use formal specifications?

Formal specifications are hard for the development team to learn; hard to use; almost impossible for most clients to understand.