

Week 3: Priorities and Use Cases

Reading: Pressman Chapter 7
Section 7.2 Establishing the Ground Work and
Section 7.4 Developing Use Cases

1. Discuss some of the problems that arise when requirements must be elicited from three or four different stakeholders.

SOLUTION:

Conflicting requirements
Different priorities for the same requirement
Different “languages” spoken by the different customers
Gatekeeping and secrets - customers don’t want others to know their needs
Scope problems of too many requirements

from Pressman: 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 do not 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.

2. Write a *user story* for one of the following activities
a. Making a withdrawal from an ATM
b. Using your charge card from meal at a restaurant
c. Searching for books on a specific topic using an online bookstore

SOLUTION: As a user searching for books by topic
(*external customer or other external entity*)
I can identify relevant books

(achieve a business goal)

So that I can buy them without spending hours searching
(value to the external customer / entity / business)

As a bank customer making a withdrawal
(external customer or other external entity)

I can gain access to my cash from many locations
(achieve a business goal)

So that I can use my cash when I need it
(value to the external customer / entity / business)

3. Develop a *UML use case* for one of the following activities:
 - a. Making a withdrawal from an ATM
 - b. Using your charge card from meal at a restaurant
 - c. Searching for books on a specific topic using an online bookstore

SOLUTION: Use case name: WithdrawCash

Flow of events:

1. The Customer inserts their card in the ATM and enters their login details
2. The Customer indicates they wish to Withdraw Cash and enters details of the account and amount
3. The ATM validates the details and checks the Customer's balance
4. If the transaction is valid then the bank updates the Customer's details and the ATM balance
5. If the transaction is valid then the ATM assembles the requested cash and receipt and delivers to the Customer
6. The Customer takes the cash, receipt and card
7. The ATM returns to entry state for the next Customer

There are several alternate paths that could be considered in this use case such as exit cases if the balance is insufficient etc.

Pressman Section 7.4 provides a more detailed template for use cases. As always, the level of detail you require depends on your application and the requirement.

4. Consider the use case you created for Q4, and write two or more non-functional requirements for the application. Then use MoSCoW to suggest possible priorities for your requirements from the point of view of one or more stakeholders. Did you find any requirements where the priorities of the stakeholders might differ?

SOLUTION: Privacy: the customer's login details shall not be revealed to any unauthorised party
Customer Priority: Must have (else I am at risk if I use an ATM)
Bank Priority: Must have (else bank am at risk if I use an ATM)

Efficiency: the withdrawal process should take no more than 1 minute
Customer Priority: Should have (else there will be long queues and inconvenience at the ATM)
Bank Priority: Could have (or should, maybe we can't guarantee this or it could compromise safety)

Availability: the ATM should be available to customers for at least 23 hours (?) of each day up to \$10,000 (?) withdrawals a day
Note: this is mostly an operational requirement outside the scope of the software system, but there are some SW issues
Customer Priority: Could have (but I can't really check it)
Bank Priority: Could have but there is a cost-benefit tradeoff here about servicing the ATMs - in reality some ATMs are probably more available than others.

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

SOLUTION:

Win-win is a solution to a conflict that both parties are happy with. Contrast this with a solution in which one party is satisfied but the other is not.

from Pressman: 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.

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

SOLUTION:

The error should be recorded in the issue tracker stating which checklist questions it fails. The review team to look into it should include SW eng, customers, users and other stakeholders. The requirement spec should say who is responsible for that requirement, and they can be consulted first. The changes made to resolve the problem should be recorded, and the requirement re-checked against the validation list. In all this: beware of setting up tick-box processes.

from Pressman: 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.