4A Software Requirements Elicitation

Oliver Au
oau@ouhk.edu.hk

Computing, The Open University of Hong Kong

http://ouhk.seprofession.com/

Unit Objective and Outline

After this unit, you should be able to elicit requirements from customers (elicit means 'draw out').

Outline

1. Challenges
   - Standish Chaos Report
   - Costs of Requirements Defects
2. Software Requirements Specification (SRS)
3. Requirements Elicitation
4. Summary

Why Communicating Software Requirements is Difficult?

1. It is difficult for customers to describe their requirements accurately and fully in words.
2. Customers don't know the requirements they stated are inadequate until they see the product built.
3. Even if the product meets their stated requirements, customers may flip-flop or get new ideas.
4. Software requirements change with the rapidly changing business environment.

Consider this sentence. Do you find it ambiguous? (A sentence or passage is ambiguous if it is open to more than one interpretation or not having an obvious meaning.)

Dogs must be carried on this escalator.

Technically, the sentence is grammatically correct. We have at least 4 ways to interpret it.

1. All dogs should have a chance to go on this escalator ride.
2. This escalator is only for dog-holders.
3. You cannot carry your pet on other escalators.
4. When riding with a dog or any other pet for that matter, carry it.

I bet everyone would choose the 4th interpretation.

Source of Ambiguity

(A) The way the sentence or passage was written.

(B) The way the reader interprets it.

Technically, the sentence is grammatically correct. We have at least 4 ways to interpret it.

1. All dogs should have a chance to go on this escalator ride.
2. This escalator is only for dog-holders.
3. You cannot carry your pet on other escalators.
4. When riding with a dog or any other pet for that matter, carry it.

I bet everyone would choose the 4th interpretation.

Source of Ambiguity

(A) The way the sentence or passage was written.
(B) The way the reader interprets it.

This example is interesting. We know that the sentence is ambiguous but we have no trouble to understand it correctly. We can generalise the case by saying that if the author writes it in the same context as the reader reads it. We are not at a risk of miscommunication.

When it comes to pet safety on escalators, people don't differ much. Even if the wording is imprecise, we end up with the same understanding. People may laugh at the sign but it serves the purpose of reminding people of the right thing to do with their pets at the escalator.

On other matters, we have more diversified understanding of the world. Users with a business background and developers with a programming background will have a difficult time to communicate on certain topics.

Project Success Factors – Standish Chaos Report

The success factors that rely on effective communication of software requirements as highlighted in purple.

Success Factors in Software Projects	
User Involvement	15.9 %
Executive Management Support	13.9 %
Clear Statement of Requirements	13 %
Proper Planning	9.6 %
Realistic Expectations	8.2 %
Smaller Project Milestones	7.7 %
Competent Staff	7.2 %
Ownership	5.3 %
Clear Vision & Objectives	2.9 %
Hard-Working, Focused Staff	2.4 %
Other	13.9 %

Project Success Factors – Standish Chaos Report

The success factors that rely on effective communication of software requirements as highlighted in purple.

Success Factors in Software Projects	
User Involvement	15.9 %
Executive Management Support	13.9 %
Clear Statement of Requirements	13 %
Proper Planning	9.6 %
Realistic Expectations	8.2 %
Smaller Project Milestones	7.7 %
Competent Staff	7.2 %
Ownership	5.3 %
Clear Vision & Objectives	2.9 %
Hard-Working, Focused Staff	2.4 %
Other	13.9 %

In the famous Chaos Report, the Standish Group (1998) summarised the experience of 365 companies on 3,682 software development projects in two tables: one for success and one for failures.

Project Challenging Factors – Standish Chaos Report

Factors that cause project to fail are highlighted in red if they rely on effective communication of software requirements.

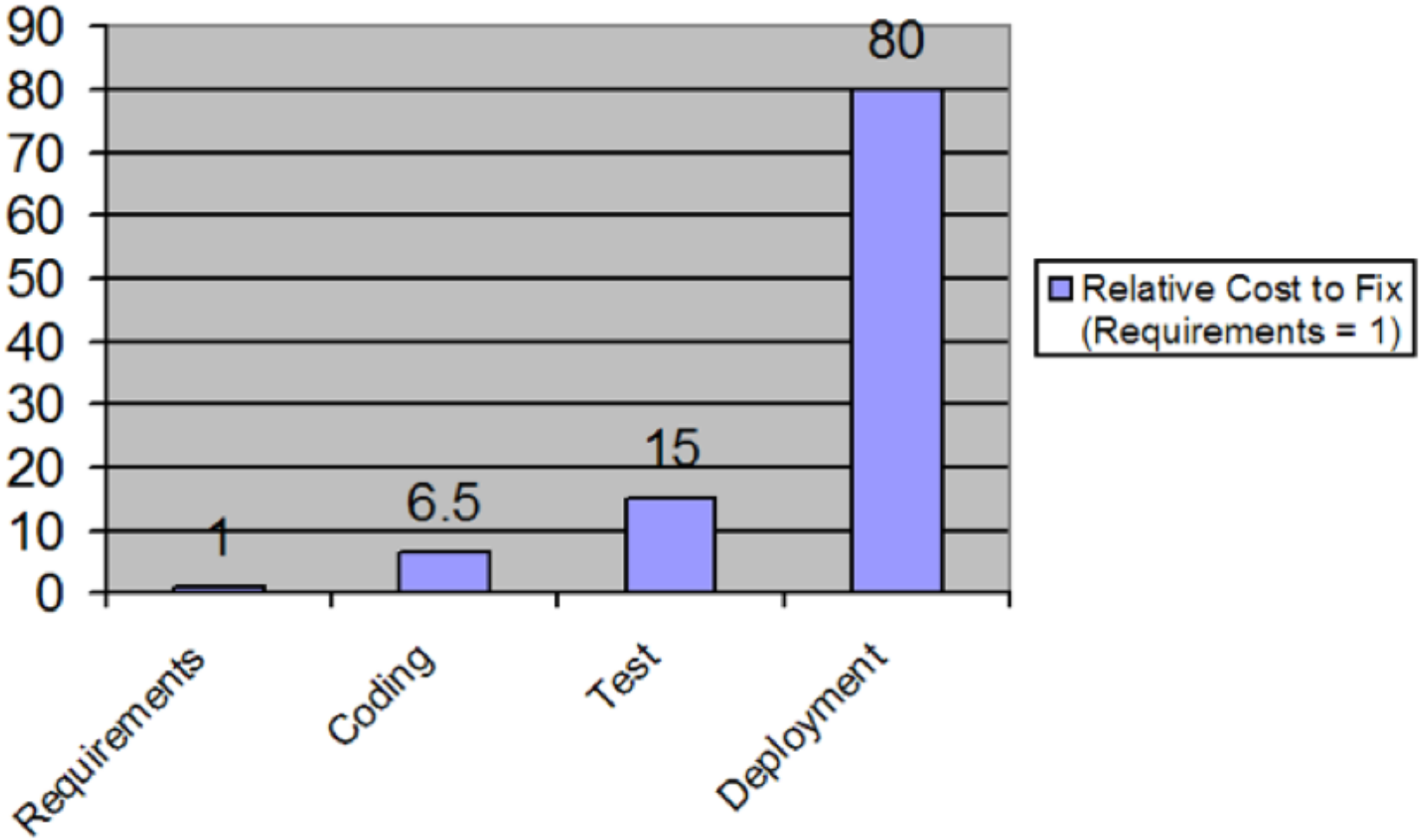
Challenging Factors in Software Projects	
Lack of User Input	12.8 %
Incomplete Requirements & Specifications	12.3 %
Changing Requirements & Specifications	11.8 %
Lack of Executive Support	7.5 %
Technology Incompetence	7 %
Lack of Resources	6.4 %
Unrealistic Expectations	5.9 %
Unclear Objectives	5.3 %
Unrealistic Time Frames	4.3 %
New Technology	3.7 %
Other	23 %

When are defects introduced?

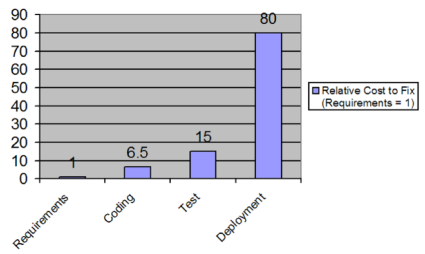
Software Development Phases	
Requirement	20%
Design	25%
Coding	35%
User Manuals	12%
Bad Fixes	8%

One-fifth of all defects were introduced during requirements elicitation and analysis. By the time, we discover a requirement error, much has been done on top already. The cost of fixing it will be high as we will see next.

Costs of Fixing Requirement Errors



Costs of Fixing Requirement Errors



It is best to fix an error shortly after you have made the error. The longer you wait, the more work will be built on top of the error, the more expensive it is to fix the error.

The costs to fix 6.5 times, 15 times and 80 times are rough estimation. Different researchers produced different numbers.

Requirement bugs are best fixed during requirement phase.

Learning Software requirements

SRS SRS stands for software requirement specification. The document was written at the project beginning to thoroughly cover functional requirements and non-functional requirements. The is the more conventional approach still used by many.

Prototypes Developers mock up the screens and their flow to get the customers to see what the eventual product would look like.

Agile At the beginning of the project, features are expressed as user stories each written in 1 to 3 sentences. Communication of detailed requirements rely on face-to-face conversation between customers and developers. No lengthy requirements documents are produced.

Learning Software requirements

SRS: SRS stands for software requirement specification. The document was written at the project beginning to thoroughly cover functional requirements and non-functional requirements. The is the more conventional approach still used by many.

Prototypes: Developers mock up the screens and their flow to get the customers to see what the eventual product would look like.

Agile: At the beginning of the project, features are expressed as user stories each written in 1 to 3 sentences. Communication of detailed requirements rely on face-to-face conversation between customers and developers. No lengthy requirements documents are produced.

The discovery of user requirements is called requirements elicitation. The person writing the functional and quality requirements needs to have good knowledge of the application domain. He or she often has the title of business analyst. Other people with the title of systems analyst, architect or project leader rely on the work of the business analyst to learn about the user requirements. The business analyst can represent users to inform systems analysts and architects of user requirements. If a business analyst does not exist, the system analyst would perform requirements elicitation. Job titles may have different meanings according to the specific organisation. You may find people doing the same job with a different title or different jobs with the same title. Followings are some requirements elicitation techniques.

Steps of Writing an SRS (1 of 2)

Who Identify stakeholders who will be affected by the project. (Stakeholders are people who are affected by the project directly or indirectly in any way.) Consider a system used by a call centre. Stakeholders include:

- The general manager (GM) of the call centre who will pay for the project
- The hotline staff who will answer the phone calls
- The customer who phone the hotline

Why Create a problem statement in a few pages or less

- state the project aims
- describe the current situation
- explain the impact of the software
- estimate costs and benefits very roughly

Steps of Writing an SRS (2 of 2)

What Specify the functional requirements listing the features required by customers.

How Specify the non-functional requirements about the quality of the product e.g. usability, reliability, performance, and supportability.

There is not a standard format of problem statements. The problem statement should be created after consulting the stakeholders. After completion, we should give the problem statement to stakeholders to review and endorse.

Requirements Elicitation Techniques

Requirements elicitation techniques help systems analysts to bring out knowledge hidden in users' minds. The techniques include:

- Questionnaires
- Interviewing
- Requirements Workshops
- Brainstorming
- Background Study

Questionnaires

Questionnaires are an effective means to obtain the views of many people at a low cost.

Closed questions Closed questions restrict the answers respondents can give. Answers can be analysed statistically and summarised in an easy-to-understand table form. Consider the question, "Which feature of the current system do you hate the most?" This is a closed question because the answer would be selected from a list of features provided by the current system.

Open questions Open questions allow respondents to write down anything in the answer. You can get interesting and unexpected information.

Mixing them After asking the user of which feature he or she dislike most (a closed question), we ask why he or she dislike it (an open question).

Interviewing

1. You meet the interviewee face-to-face.
2. You will still be asking closed questions and open questions.
3. You can tailor your questions according to the responses in earlier questions.
4. What you learn from an interview will be more qualitative than quantitative.
5. You are in a better position to detect that the interviewee is not giving you the complete and accurate answer.

Sample Questions

1. Which parts of your business operations need improvements?
2. How are the concerned tasks done today?
3. How would you improve the way the tasks are done?
4. How will the business benefit from a certain approach?

Context-free Questions for Interviews

"Context-free" means they can be asked in any context (situation).

1. Who are the users?
2. What are their experience, educational background and interests?
3. Have they used similar software applications before?
4. What kind of platforms do we need to support initially and in the future? Windows? Mac? iPad? Which versions?
5. When will the new application be needed and at what cost?
6. What features do the users need from the application?
7. What are their expectations for reliability or performance?
8. What considerations do we have regarding implementation, interfacing with existing systems, operation and legal matters?

Effective Interviewing

1. Respect interviewees whether we like their answers or not.
2. Keep them happy if we want them to be cooperative and give us the true answers.
3. When interviewees give a simple answer, we can ask them further for the reasons behind that simple answer.
4. Different stakeholders may give us conflicting information.
   (a) If the differences are about facts, we need to find out the truth.
   (b) Dealing with differences in priorities is challenging as stakeholders may insist on their different priorities. We need to strike a balance.

Requirements Workshop

1. Stakeholders meet and try to reach consensus in 1 or 2 days.
2. There should be a good facilitator in the workshop balancing the need to encourage contribution from everyone but no one dominates.
3. An agenda and a preliminary draft of requirements document would be provided to participants prior to the workshop.
4. The facilitator may predict the possibility of conflicts and talk to the opposing parties individually before the workshop to reduce their differences. For example, marketing wants the product ASAP while development team wants to spend enough time for a quality job. The workshop may garner widespread support of the project across the organisation.

Brainstorming

Brainstorming consists of two phases: idea generation and idea reduction. Brainstorming may be used in workshops or meetings.

Idea Generation

(A) Questions are asked to invite ideas. Ideas should not be evaluated at this stage. Criticism discourages people to contribute ideas.

(B) A secretary will try to write down all ideas from everyone. Each individual should keep track of the ideas he or she has proposed. He or she will compare the personal list with the secretary's master list to prevent omission.

Brainstorming (Cont.)

Idea Reduction

(A) Idea reduction should start only after idea generation has concluded.

(B) Related ideas may be grouped.

(C) Overlapped ideas may be rewritten and simplified as ene idea.

(D) We may criticise ideas and remove the bad ones now.

(E) Rank the ideas from high to low as *critical*, *important* and *useful*.

Background Study

1. Company documents help you learn the company procedures but be careful that procedures may not be observed by employees consistently or at all.
2. Even if the documents may not be accurate, they may help you better understand users? terminology. Getting acquainted with the documents will enhance your communication with users and customers.
3. Existing products from competitors should be studied too. You should build a new software that matches or surpasses existing products.
4. If software development is not your company?s core expertise, you may outsource the development project or buy off-the-shelf software.

Quotes about Journeys

1. The world is a book and those who do not travel read only one page. – St. Augustine
2. A journey of a thousand miles must begin with a single step. – Lao Tzu

Take the quiz at `http://ouhk.seprofession.com` now.

Quotes about Journeys

1. The world is a book and those who do not travel read only one page. – St. Augustine
2. A journey of a thousand miles must begin with a single step. – Lao Tzu

Take the quiz at `http://ouhk.seprofession.com` now.

Since I am preparing this set of slides while in North Dakota, U.S., it makes perfect sense for me to share with you some quotes about journey.

It is very difficult for you to become smart if you don't read. But I noticed many of our students either do not read or read poorly written materials. Do not limit your reading to the computer subject. Take time to read philosophy, novel, sociology, economic, laws and etc. Do so when you are young, it trains your brain to benefit you for the rest of your life.

When you travel, find time to mingle with the locals. Observe how they live their lives. It is fun. It is enlightening. And it can also help you in the future more than you know. For example, when you meet a new friend, you just have more to talk about. You will become a more interesting person after experiencing a different culture. People will want to be your friends even more.