UNDERSTANDING REQUIREMENTS AND REQUIREMENT ENGINEERING

Chapter 08

Acknowledgement

- Some of the contents are adapted from material by Manzil-e-Maqsood and Prof. Tony Tang

- Software Engineering – A practitioner's approach by Roger S. Pressman and Bruce. R. Maxim

- Software Engineering by Ian Sommerville

Requirements Engineering

- The process of establishing the system services and constraints is called requirements engineering.

Requirements Engineering…

- RE is the process of
  - Identifying stakeholders and their needs
  - Documenting these needs in a form that is open to analysis, communication and subsequent implementation
  - Developing a specification that will guide design

- A Requirements Specification document is the primary result of RE process

Significance of RE

- Initial step towards construction of software
- A well-defined RE process prevents potential software defects from running downstream into design and code
- Ensures that we have specified a system that properly meets the customer's needs and satisfies the customer's expectations

Significance of RE …

"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the system if done wrong. No other part is more difficult to rectify later".

Fred Brooks - No Silver Bullet: Essence and Accidents of Software Engineering, 1987.

8.1 Requirements Engineering Tasks

- Seven tasks

Inception

Elicitation

Elaboration

Negotiation

Specification

Validation

Management

Inception

- *"The seeds of major software disasters are usually sown in the first three months of commencing the software project." Caper Jones*

- Inception—ask a set of questions that establish …
  - basic understanding of the problem
  - the people who want a solution
  - the nature of the solution that is desired

Elicitation

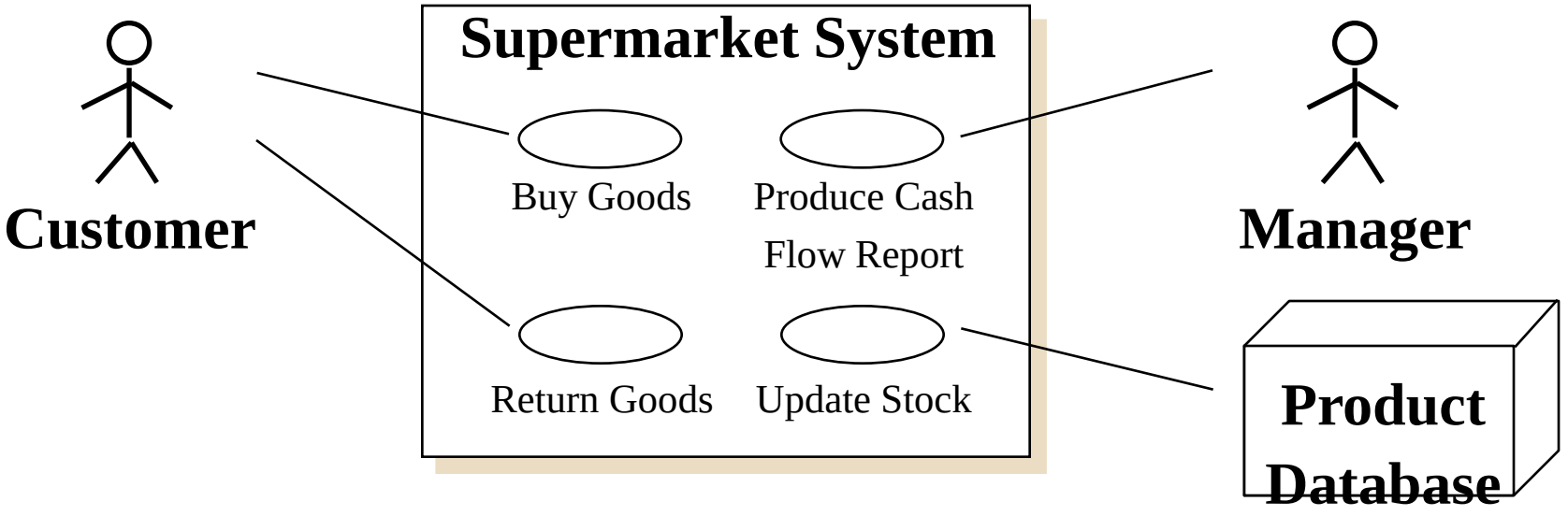
- Elicit requirements from all stakeholders
- Ask stakeholders:
  - objectives for the system
  - product fit into the business needs

- Elicitation is difficult due to:
  - Problems of scope
  - Problems of understanding
  - Problems of volatility

Elaboration

- Create an analysis model that identifies data, function and behavioral requirements
- Information obtained from the customer during the previous phases is expanded and refined.
- Analysis Modeling and Requirements Refinement
  - Requirement Analysis
  - OO Analysis
  - Class-based Modeling

Analysis Modelling: Use Case Diagram



Negotiation

- agree on a deliverable system that is realistic for developers and customers
- Understand stakeholders and their priorities
- Resolve Conflicts and Risks
- Requirements are:
  - Prioritized
  - Eliminated
  - Combined
  - Modified

Specification

- Specification—can be any one (or more) of the following:
  - A written document
  - A set of models
  - A collection of user scenarios (use-cases)
- Follow Standard Template
  - Ensures consistency and understanding
- Final work product
- Foundation for subsequent SE activities

Software Requirement Specification Example

INFO

Software Requirements Specification Template

A *software requirements specification* (SRS) is a work product that is created when a detailed description of all aspects of the software to be built must be specified before the project is to commence. It is important to note that a formal SRS is not always written. In fact, there are many instances in which effort expended on an SRS might be better spent in other software engineering activities. However, when software is to be developed by a third party, when a lack of specification would create severe business issues, or when a system is extremely complex or business critical, an SRS may be justified.

Karl Wiegers [Wie03] of Process Impact Inc. has developed a worthwhile template (available at **www.processimpact.com/process_assets/srs_template.doc**) that can serve as a guideline for those who must create a complete SRS. A topic outline follows:

Table of Contents

Revision History

1. **Introduction**
   - 1.1 Purpose
   - 1.2 Document Conventions
   - 1.3 Intended Audience and Reading Suggestions
   - 1.4 Project Scope
   - 1.5 References
2. **Overall Description**
   - 2.1 Product Perspective
   - 2.2 Product Features
   - 2.3 User Classes and Characteristics
   - 2.4 Operating Environment
   - 2.5 Design and Implementation Constraints
   - 2.6 User Documentation
   - 2.7 Assumptions and Dependencies
3. **System Features**
   - 3.1 System Feature 1
   - 3.2 System Feature 2 (and so on)
4. **External Interface Requirements**
   - 4.1 User Interfaces
   - 4.2 Hardware Interfaces
   - 4.3 Software Interfaces
   - 4.4 Communications Interfaces
5. **Other Nonfunctional Requirements**
   - 5.1 Performance Requirements
   - 5.2 Safety Requirements
   - 5.3 Security Requirements
   - 5.4 Software Quality Attributes
6. **Other Requirements**

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: Issues List

A detailed description of each SRS topic can be obtained by downloading the SRS template at the URL noted in this sidebar.

Software Engineering a Practitioner's approach by Roger S. Pressman

Validation

- Validation—a review mechanism that looks for
  - errors in content or interpretation
  - areas where clarification may be required
  - missing information
  - conflicting or unrealistic (unachievable) requirements.
- All Requirements stated are:
  - Correct
  - Unambiguous
  - Consistent
  - Complete

- Requirements Validation Method
  - Formal Technical Reviews (Inspection, Walkthroughs)
  - Review Team may include:
    - Software engineers, customers, users and other stake holders.

Example: Requirement Validation Checklist

INFO

Requirements Validation Checklist

It is often useful to examine each requirement against a set of checklist questions. Here is a small subset of those that might be asked:

- Are requirements stated clearly? Can they be misinterpreted?
- Is the source (e.g., a person, a regulation, a document) of the requirement identified? Has the final statement of the requirement been examined by or against the original source?
- Is the requirement bounded in quantitative terms?
- What other requirements relate to this requirement? Are they clearly noted via a cross-reference matrix or other mechanism?
- Does the requirement violate any system domain constraints?
- Is the requirement testable? If so, can we specify tests (sometimes called validation criteria) to exercise the requirement?
- Is the requirement traceable to any system model that has been created?
- Is the requirement traceable to overall system/product objectives?
- Is the specification structured in a way that leads to easy understanding, easy reference, and easy translation into more technical work products?
- Has an index for the specification been created?
- Have requirements associated with performance, behavior, and operational characteristics been clearly stated? What requirements appear to be implicit?

Software Engineering a Practitioner's approach by Roger S. Pressman

RE-Managment

- A set of activities that helps the team identify, control and track changes to requirements at any time.

- Once requirements are approved from the customer the requirements document forms the “baseline” for development.

8.2 Initiating RE Process

- Identify stakeholders
  - "who else do you think I should talk to?"
- Recognize multiple points of view.
  - Explore different stakeholders and you will find different views, etc

Initiating RE Process (Contd…)

- Work toward collaboration
  - Identify what the stakeholders have in common regarding requirements. And then identify where the conflict or inconsistency lies.
- The first questions
  - Context Free Questions
    - Who is behind the request for this work?
    - Who will use the solution?
    - What will be the economic benefit of a successful solution?

8.3 Eliciting Requirements

- The goal is
  - to identify the problem
  - propose elements of the solution
  - negotiate different approaches

- Collaborative Requirement Gathering
- Quality Function Deployment
- Elicitation Work Products

Collaborative Requirement Gathering

- Meetings are conducted and attended by both software engineers and customers
- Rules for preparation and participation are established
- An agenda is suggested
- A "facilitator" (can be a customer, a developer, or an outsider) controls the meeting
- A “recording mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used

Requirement Gathering Meeting

SafeHome

Conducting a Requirements-Gathering Meeting

The scene: A meeting room. The first requirements-gathering meeting is in progress.

The players: Jamie Lazar, software team member; Vinod Raman, software team member; Ed Robbins, software team member; Doug Miller, software engineering manager; three members of marketing; a product engineering representative; and a facilitator.

The conversation:

Facilitator (pointing at whiteboard): So that's the current list of objects and services for the home security function.

Marketing person: That about covers it from our point of view.

Vinod: Didn't someone mention that they wanted all *SafeHome* functionality to be accessible via the Internet? That would include the home security function, no?

Marketing person: Yes, that's right . . . we'll have to add that functionality and the appropriate objects.

Facilitator: Does that also add some constraints?

Jamie: It does, both technical and legal.

Production rep: Meaning?

Jamie: We better make sure an outsider can't hack into the system, disarm it, and rob the place or worse. Heavy liability on our part.

Doug: Very true.

Marketing: But we still need that . . . just be sure to stop an outsider from getting in.

Ed: That's easier said than done and . . .

Facilitator (interrupting): I don't want to debate this issue now. Let's note it as an action item and proceed.

(Doug, serving as the recorder for the meeting, makes an appropriate note.)

Facilitator: I have a feeling there's still more to consider here.

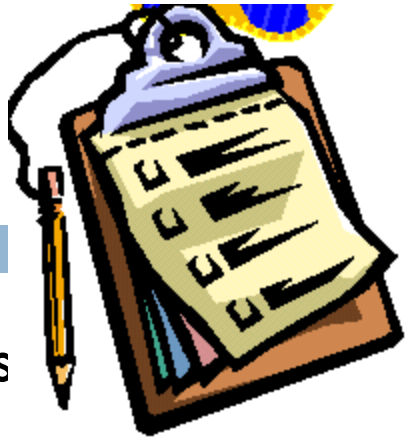
(The group spends the next 20 minutes refining and expanding the details of the home security function.)

Software Engineering a Practitioner's approach by Roger S. Pressman

Quality Function Deployment

- Particular technique, defining needs of customer into technical requirements
- QFD identifies these types of requirements:
  - Normal requirements (requested)
  - Expected requirements (ease of use; reliability; correctness; ease of installation; etc)
  - Exciting requirements (that will make this solution stand out)

Elicitation Work Products



- The deliverables at the end of the requirements usually include:
- Statement of need and feasibility
- Scope statement
- List of stakeholders
- Description of technical environment
- List of requirements
- Usage scenarios (**Use Cases**)

Customer-Developer Relationship

- Learn about the business domain
  - It is not a computer science problem – the problem lies in a different domain than computer science and you must understand it before you can solve it.
- Speak the user language
  - Use customer terminology
  - Document should be structured and written in a way that the customer finds it easy to read and understand.

Why do we need to capture requrirements?

Credit: User Story Mapping, by Jeff Patton (O'Reilly, 2014).

Mars Probe Lost Due to Simple Math Error

October 01, 1999 | ROBERT LEE HOTZ | TIMES SCIENCE WRITER

NASA lost its $125-million Mars Climate Orbiter because spacecraft engineers failed to convert from English to metric measurements when exchanging vital data before the craft was launched, space agency officials said Thursday.

A navigation team at the Jet Propulsion Laboratory used the metric system of millimeters and meters in its calculations, while Lockheed Martin Astronautics in Denver, which designed and built the spacecraft, provided crucial acceleration data in the English system of inches, feet and pounds.

As a result, JPL engineers mistook acceleration readings measured in English units of pound-seconds for a metric measure of force called newton-seconds.

In a sense, the spacecraft was lost in translation.

"That is so dumb," said John Logsdon, director of George Washington University's space policy institute. "There seems to have emerged over the past couple of years a systematic problem in the space community of insufficient attention to detail."

FROM THE ARCHIVES

Math Error Exaggerated TriZetto Loss Estimate

March 7, 2001

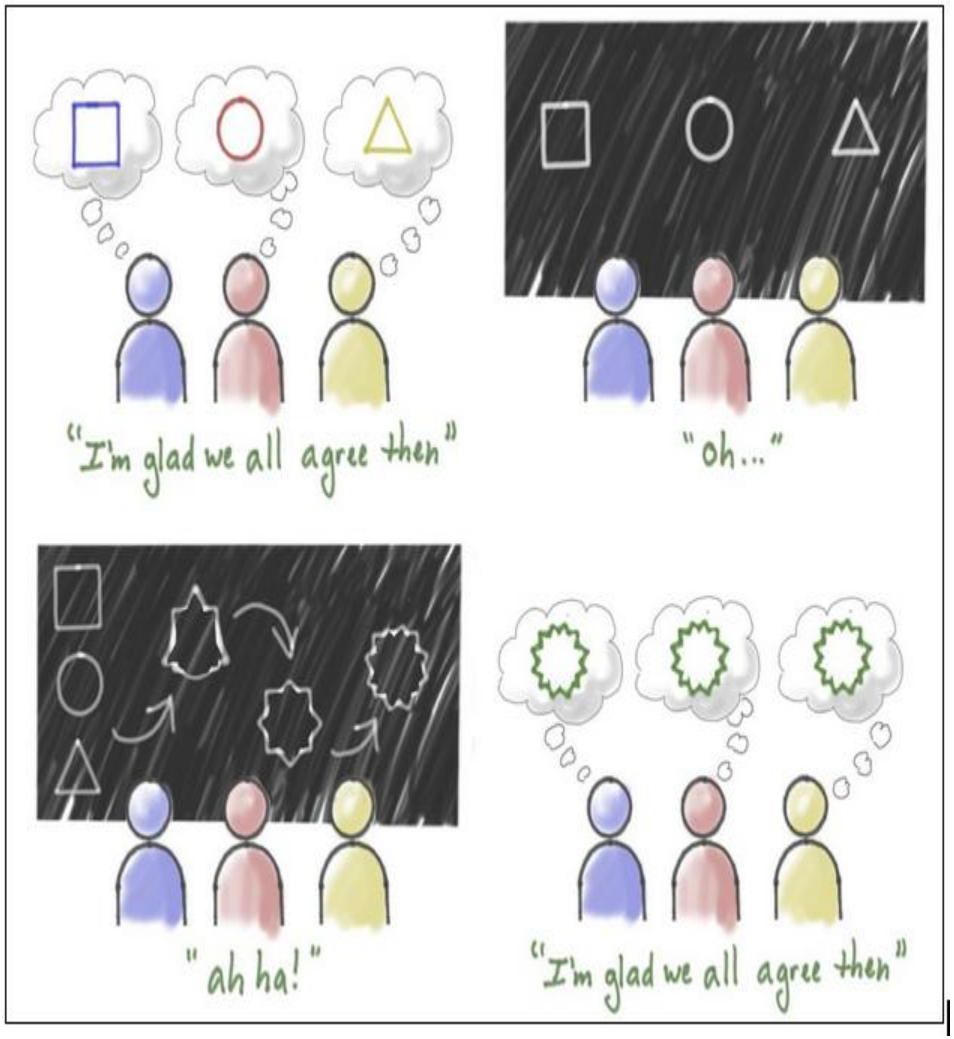
Math Error Inflated Ventura Blvd. Cost : Traffic: Mistake...

Los Angeles Times

Mars Probe Lost Due to Simple Math Error

A navigation team at the Jet Propulsion Laboratory used the metric system of millimeters and meters in its calculations, while Lockheed Martin Astronautics in Denver, which designed and built the spacecraft, provided crucial acceleration data in the English system of inches, feet and pounds.

Shared Understanding >> Documentation



Exercise on Requirement Gathering

- Make pair with your neighbor
  - One client, one developer

- What are the requirements for a coffee machine?

Next Lecture

- Requirement Modeling
  - User Stories
  - Use Cases