Identifying Needs and Establishing Requirements

Chapter 7 - PRS

By – Hao and Raj

Why do we need Requirements?

- When 38 IT professionals in the UK were asked about which project stages caused failure, respondents mentioned “requirements definition” more than any other phase.

What are requirements?

- A requirement is a statement about an intended product that specifies what it should do or how it should perform.

- Goal: To make as specific, unambigous, and clear as possible.

What kinds of specifications? (broad category)

- Functional: What the system should do
- Non-Fictional: what constraints there are on the system its development. (For example that a work processor runs on different platforms)

What requirements should be gathered?

- Functional: What the product should do.
- Data requirements: Capture the type, volatility, size/amount, persistence, accuracy and the amounts of the required data.
- Environmental requirements: a) context of use b) Social environment (eg. Collaboration and coordination) c) how good is user support likely to be d) what technologies will it run on
- User Requirements: Capture the characteristics of the intended user group.
- Usability Requirement: Usability goals associated measures for a particular product (More info on Chapter 6).

Data Gathering Techniques

- Questionnaires: Series of questions designed to elicit specific information from us. The questions may require different kinds of answers: some require a simple Yes/No, others ask us to choose from a set of pre-supplied answers.

- Interviews: Interviews involve asking someone a set of questions. Often interviews are face-to-face, but they don't have to be (more on next page).

Data Gathering Techniques (continued)

- Interviews:
  - Forum for talking to people
  - Structured, unstructured or semi-structured
  - Props, e.g. sample scenarios of use, prototypes, can be used in interviews
  - Good for exploring issues
  - But are time consuming and may be infeasible to visit everyone

Data-gathering techniques

- Focus groups and workshops: Interviews tend to be one on one, and elicit only one person's perspective. It can be very revealing to get a group of stakeholders together to discuss issues and requirements.

- **Naturalistic Observation**: It can be very difficult for humans to explain what they do or to even describe accurately how they achieve a task. (more on next page)

Data-gathering Techniques (continued)

- Naturalistic observation:
  - Spend time with stakeholders in their day-to-day tasks, observing work as it happens
  - Gain insights into stakeholders' tasks
  - Good for understanding the nature and context of the tasks
  - But, it requires time and commitment from a member of the design team, and it can result in a huge amount of data
  - **Ethnography is one form** : entire class devoted to this.

Data-gathering

- Studying documentation: Procedures and rules are often written down in a manual and these are a good source of data about the steps involved in an activity and any regulations governing a task.

Use All of the above In Combination :
Constraints of Time and Money

Task Classification

- Is the Task a set of sequential steps or is it a rapidly overlaying series of sub-tasks?
- Does the task involve high information content : High information content or low --- is it complex or simple to interpret visual displays.
- Is the task intended to be performed by a layman without much training or by a practitioner skilled in the task domain?

Data Gathering Guidelines

- Focus on identifying the stakeholders
- Involve all the stakeholder groups
- Need more than on person from stakeholder group(s)
- **Use a combination of data gathering techniques**
  For example: use observation to understand the context, interviews to target specific user groups, questionnaires to reach a wider population, and **focus groups to build a consensus view**

Data Gathering Guidelines Cont.

- Support the data-gathering sessions with suitable props, such as task descriptions and prototypes if available.
- Run a pilot session if possible to ensure that your data-gathering session is likely to go as planned

- In an ideal world, you would understand what you are looking for and what kinds of analysis you want to do, and design the data-capture exercise to collect the data you want. **However, data gathering is expensive and often a tightly constrained resource.**

Software Cost Reduction Method (SCR)

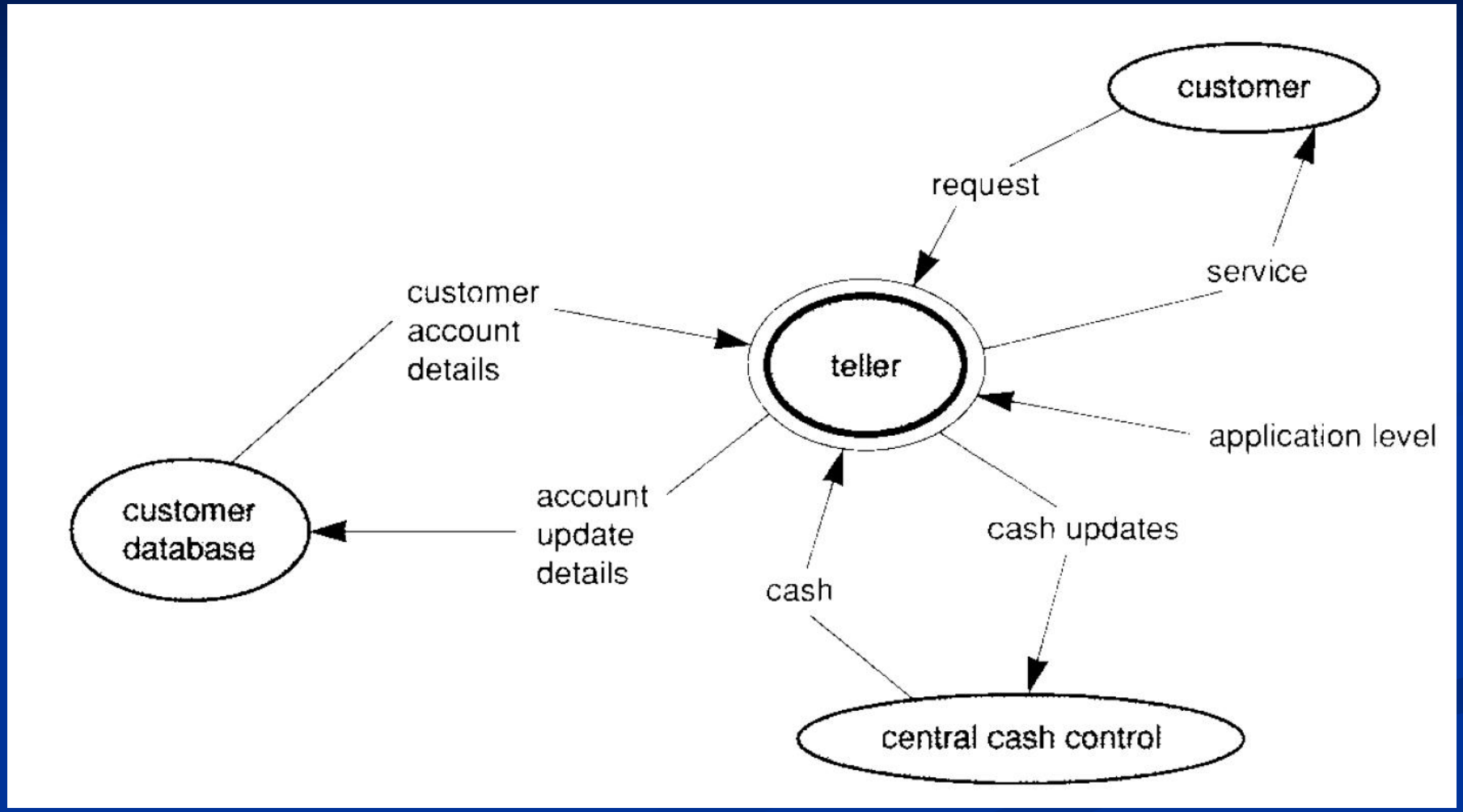
- The system behavior is described by a mathematical relation between monitored variables and controlled variables. Very mathematical…

Event	New Mode
@T(WaterPres $\geq$ Low)	Permitted
@T(WaterPres $\geq$ Permit)	High
@T(WaterPres $<$ Low)	TooLow
@T(WaterPres $<$ Permit)	Permitted

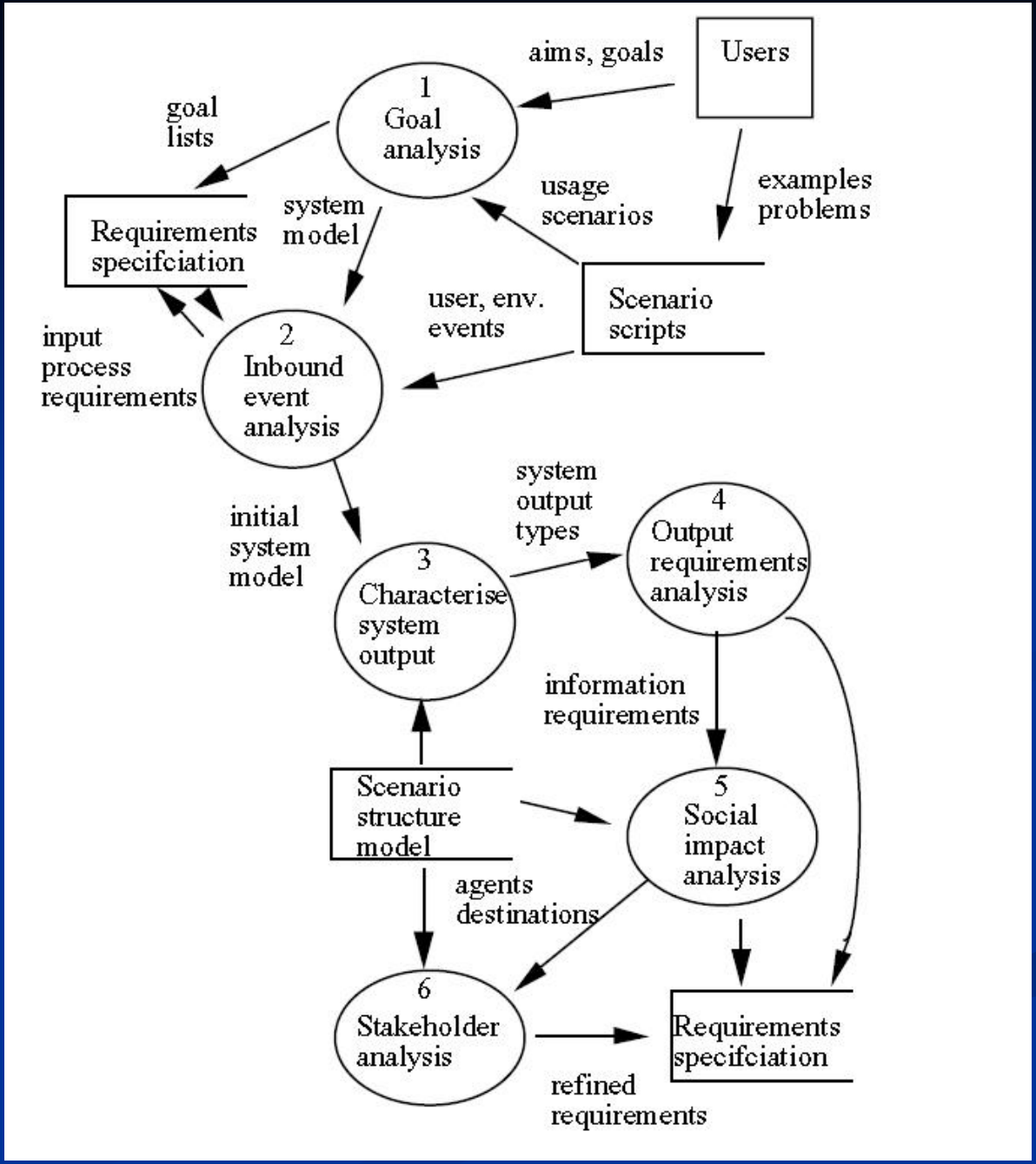
Table 7.1 Overview of data-gathering techniques used in the requirements activity

Technique	Good for	Kind of data	Advantages	Disadvantages	Detail for designing in
Questionnaires	Answering specific questions	Quantitative and qualitative data	Can reach many people with low resource	The design is crucial. Response rate may be low. Responses may not be what you want	Chapter 13
Interviews	Exploring issues	Some quantitative but mostly qualitative data	Interviewer can guide interviewee if necessary. Encourages contact between developers and users	Time consuming. Artificial environment may intimidate interviewee	Chapter 13
Focus groups and workshops	Collecting multiple viewpoints	Some quantitative but mostly qualitative data	Highlights areas of consensus and conflict. Encourages contact between developers and users	Possibility of dominant characters	Chapter 13
Naturalistic observation	Understanding context of user activity	Qualitative	Observing actual work gives insights that other techniques can't give	Very time consuming. Huge amounts of data	Chapter 12
Studying documentation	Learning about procedures, regulations and standards	Quantitative	No time commitment from users required	Day-to-day working will differ from documented procedures	N/A

Viewpoints:


customer
request
service
customer
account
details
teller
application level
customer
database
account
update
details
cash updates
cash
central cash control

- Scenarios used to create Requirements Specification (From Users to Requirements)
- We may follow this approach in our projects



Overview

- Data interpretation and analysis
- Task descriptions
- Task analysis
- Requirements management

Data interpretation and analysis

- What: structure and record description of requirement
- When: Start soon after data gathering session

Data interpretation and analysis

- Main Requirement analysis models in object-oriented systems
  - Use cases diagrams:

    consists of actors and user cases, discussed later
  - Class diagrams
  - More…
- How to develop those diagrams?

  UML tools(useful in practice)

What is UML

- Stands for **Unified Modeling Language**
- standard language

  to specify, visualize, construct, and document the artifacts of software systems, as well as other non-software systems.
- very important part of object oriented software development process.
- uses graphical notations to express the design of software projects.
- you can choose to use UML tool for establishing requirement Or just use paper and pen.
- See details in references books about UML

Task descriptions

Task descriptions

- Scenarios
  - an informal narrative story of users
  - Natural way to explain
- Use cases
  - show interaction with a system
  - show detailed understanding of the interaction
- Essential use cases
  - Improvement of use cases
- **Often used in combination**

Scenarios

- Scenarios
  - Usually the first step in establishing requirement
  - an informal narrative story of users
  - Describe
    - what users are doing or
    - How to achieve something
  - An example

Scenario for shared calendar

- In an informal interview, one potential user answered the question

" How to arrange a meeting
between several people
in the shared-calendar system ?"

- **A simple scenario for this system:**

"The user **types in all the names of the meeting participants, the length of the meeting,** together with some constraints such as roughly **when** the meeting needs to take place, and possibly **where** it needs to take place. The system then **checks** against the individuals' calendars and the central departmental calendar……"

Scenario for shared calendar

- In an informal interview, one potential user answered the question

  " How to arrange a meeting between several people in the shared-calendar system ?"

- **A simple scenario for this system:**

  User :
  - **types in all the names of the meeting participants**
  - **length of the meeting**
  - **when is the meeting**
  - **where is the meeting.**

  System:
  - **checks against the individuals' calendars and the central departmental calendar**
  - **More…**

Use cases

- Use cases
  - a special kind of scenario that breaks down system requirements into user functions
  - use case is a sequence of events performed by a user
  - **Interaction** between user and a system
  - Example

Use case for shared calendar

A sequence of events to arrange a meeting:

- 1. The user chooses the option to arrange a meeting.
- 2. The system prompts user for the names of attendees.
- 3. The user types in a list of names.
- 4. The system checks that the list is valid.
- 5. The system prompts the user for meeting constraints.
- More steps…

Note: Number indicates the steps,
shows how user and system will interact.

Use cases

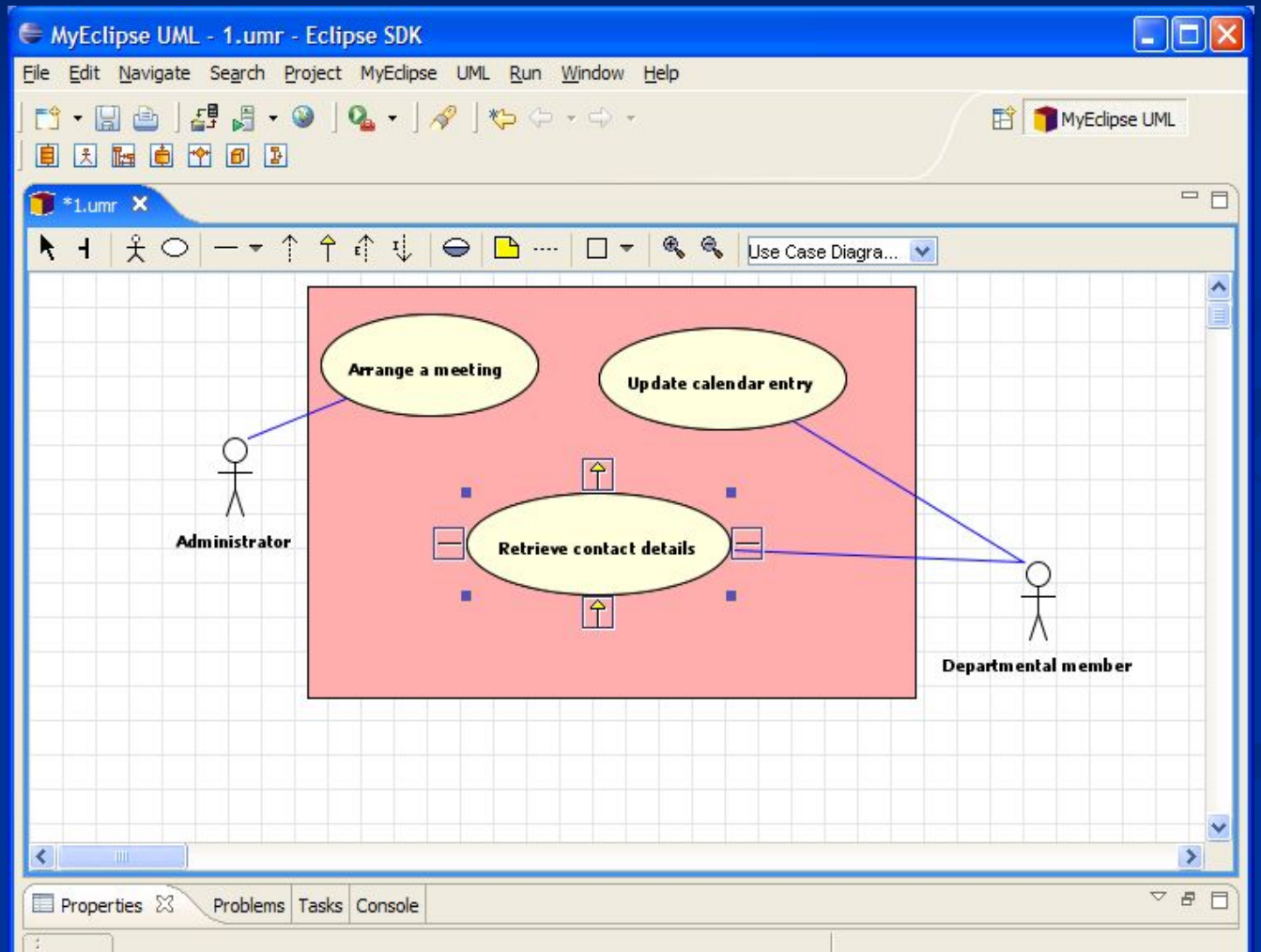
How to avoid many words in the requirement description ?

- Try graphical description!
- Actors----------use cases
  - For example

    Administrator --------- Arrange a meeting

Use case diagram example

- Use case diagram for the shared calendar system
- 3 use cases and 2 actors (Using the UML tool)



Essential use cases

- Improvement of scenarios and use cases
- structured
- Consists of description of
  - user intention
  - system responsibility

Task analysis

Task analysis

- Task descriptions versus Task analysis
  - Task descriptions are often used to envision **new** systems or devices
  - Task analysis is used mainly to investigate an **existing** situation

- Many techniques, the most popular is Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis

- Idea is simple!

  Involves breaking a task down into subtasks, then sub-sub-tasks and so on. These are grouped as plans which specify how the tasks might be performed in practice

- Start with a user goal which is examined and the main tasks for achieving it are identified

Example Hierarchical Task Analysis

0. In order to **borrow a book from the library**
    1. go to the library
    2. find the required book
        2.1 access library catalogue
        2.2 access the search screen
        2.3 enter search criteria
        2.4 identify required book
        2.5 note location
    3. go to correct shelf and retrieve book
    4. take book to checkout counter

plan 0: do 1-3-4. If book isn't on the shelf expected, do 2-3-4.
plan 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

Note: start with a user goal
plan 2 shows subtasks order
no plan 1, step1 has no subtasks

-

Requirements management

Requirements management

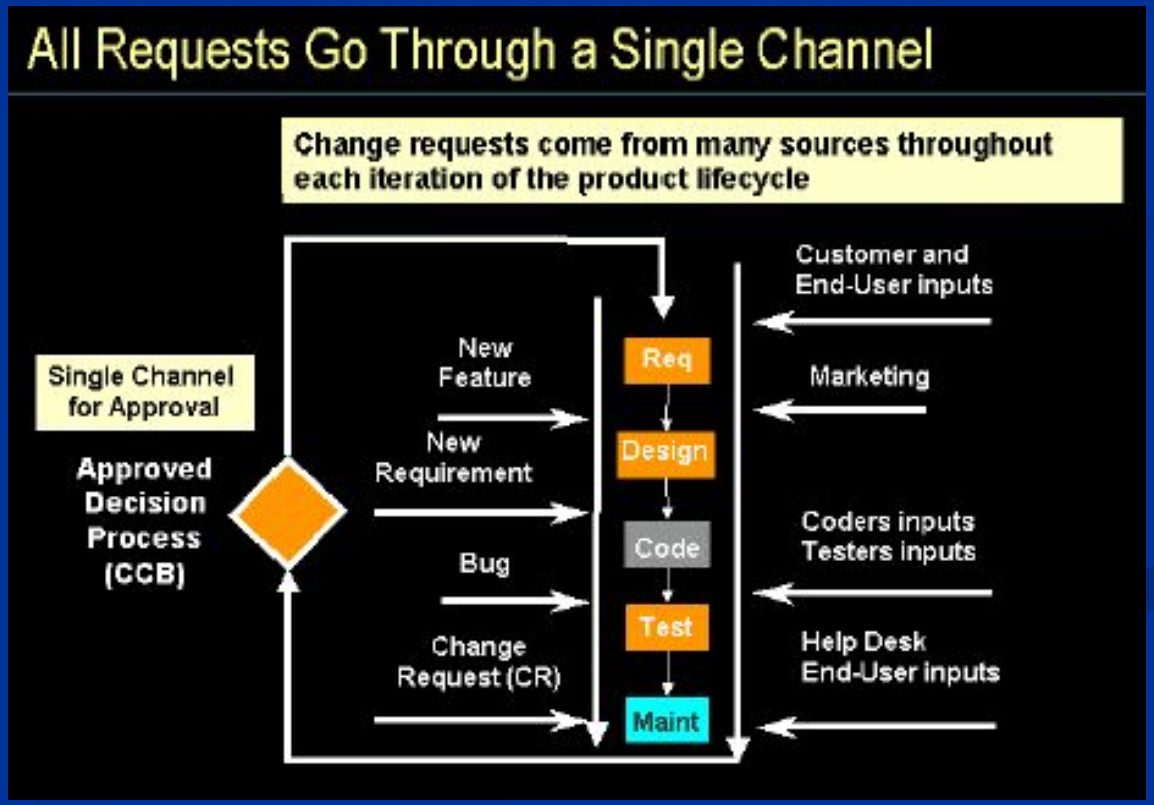
- What is it?
  - a systematic approach to eliciting, organizing, and documenting the requirements of the system,
  - a process that establishes and maintains changing requirements.
  - Important and helpful for real projects
- **Common problems**
  - No.1: Can't track change
  - No.2: Difficult to write
  - More…

How to Manage changing requirements

- Single channel of change control

  Change Control Board (CCB).

- Keep track history of requirements
- Maintain version control
- More…



How to document requirement?

- Requirement of description documents
  - Natural language and graphical
  - Widely accepted, consistent format

- Use tools to help
  - Software<IBM Rational RequisitePro>
    automating effective tool, template

Further Info:

See more details at

- http://www-306.ibm.com/software/awdtools/reqpro/

Thank you for listening!

Works Cited

- Chapter 7 from J. Preece, Y. Rogers, H. Sharp: Interaction Design: Beyond Human-Computer Interaction.
- "Applying Requirements Management with Use Cases," Roger Oberg (IBM Rational Software)
- Constance L. Heitmeyer. "*Software cost reduction*". In John J. Marciniak, editor, Encyclopedia of Software Engineering. John Waley & Sons, 2nd edition, 2002
- Alistai Sutcliffe, "Scenario-based requirements analysis." Requirements Engineering Journal, London, UK, 1998.