

SOFTWARE ENGINEERING

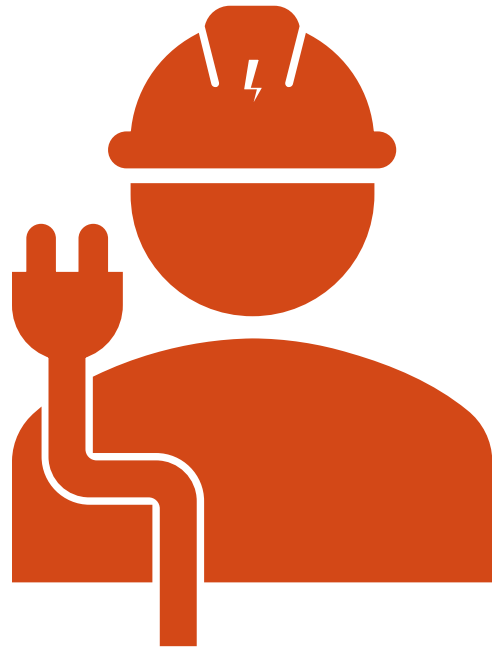
Lecture 3: Requirements Engineering: Process, Elicitation Techniques

Prof. Mohamad Kassab
m.kassab@nyu.edu



LESSON OUTLINE

- What is Requirements Engineering?
- What is a requirement?
- Requirements Classifications
- Who is a Requirements Engineer?
- Requirements Elicitation Techniques
- Context Diagram



WHAT IS REQUIREMENTS ENGINEERING?



1

How the customer explained it



2

How the project leader understood it



3

How the analyst designed it



4

How the programmer wrote it



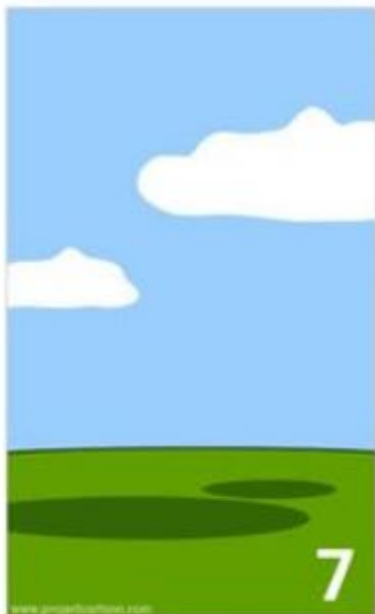
5

What the beta testers received



6

How the business consultant described it



7

How the project was documented



8

What operations installed



9

How the customer was billed



10

How it was supported



11

iSwing

What marketing advertised



12

What the customer really needed

WHAT IS REQUIREMENTS ENGINEERING?

“Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.” (Zave, 1997)

REQUIREMENTS ENGINEERING ACTIVITIES

**Requirements
elicitation/discovery**

**Requirements analysis
and reconciliation**

**Requirements
representation/modeling**

**Requirements
verification and
validation**

**Requirements
management**

**There are others, but
these are the “major”
ones**

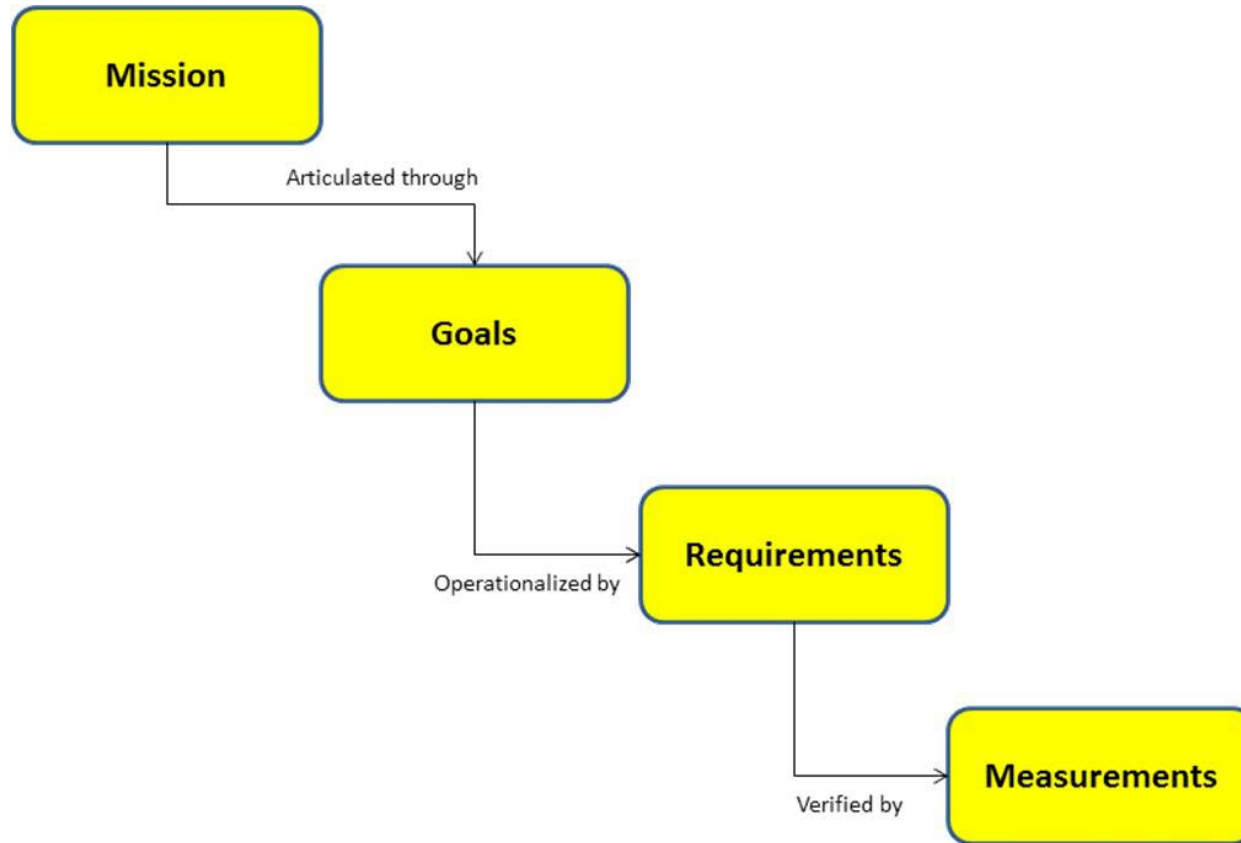
WHAT ARE REQUIREMENTS?

- A requirement is a documented description of a feature, functionality, or characteristic that a system or product must possess to fulfill stakeholders' needs and expectations.
- Requirements serve as the foundation for the design, development, and testing of a system, ensuring that the final product meets its intended purpose.



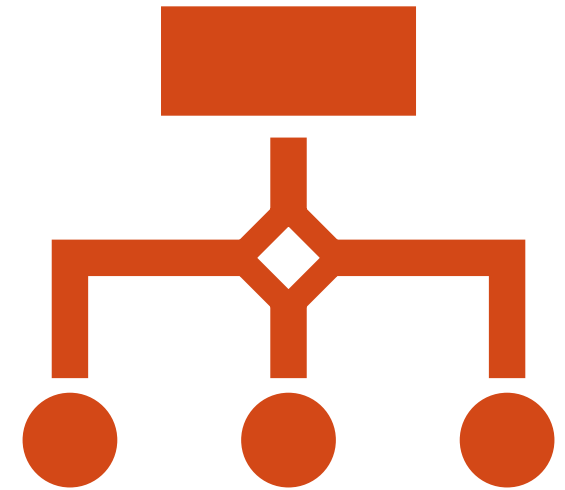
BUSINESS GOALS VS. REQUIREMENTS

- We discussed business goals during the last session. These Provide overarching purpose “project metaphor”
- **Business Goal for “Baggage Handling System”:** To automate all aspects of baggage handling from passenger origin to destination.
- On the other hand, requirements are more specific and are refined from Business Goals



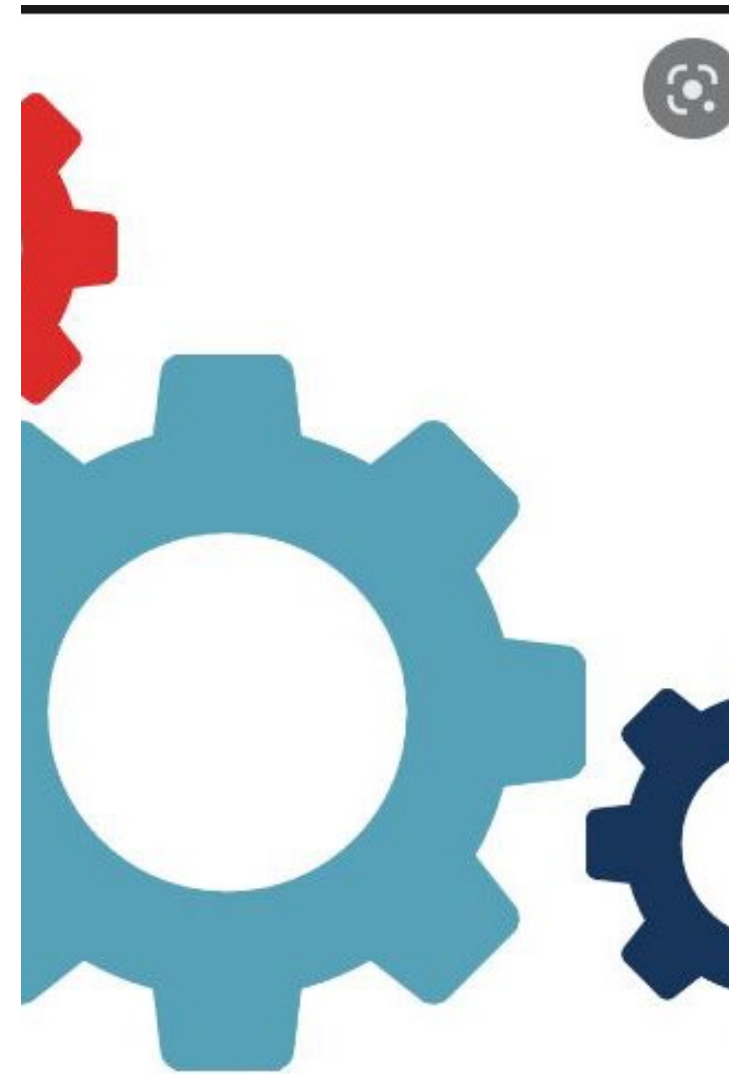
GOALS VERSUS REQUIREMENTS

- Possible classification for requirements:
- Functional requirements
- Non-functional requirements (NFRs)



FUNCTIONAL REQUIREMENTS

- The services the system should provide.
- How it will react to inputs.
- Need to explicitly state what the system should not do.
- Can be high-level and general (user requirements) or detailed, expressing inputs, outputs, exceptions, etc. (system requirements).
- Many forms of representation from NL, visual models, to formal methods.



NON-FUNCTIONAL REQUIREMENTS (NFRS)

- Requirement imposed by the environment in which the system is to exist.
- This includes timing constraints, quality properties, standard adherence, programming languages to be used, etc.
- Some of these non-functional requirements are counter intuitive (e.g., timing).
- More on NFRs later in the course.



FUNCTIONAL REQUIREMENTS FOR BAGGAGE HANDLING SYSTEM (1)

- **Baggage Check-in:**
 - The system must allow passengers to check in their baggage through self-service kiosks.
 - Baggage information (weight, size, destination) must be captured during the check-in process.
- **Baggage Sorting:**
 - The system must automatically sort baggage based on destination and flight.
 - Baggage sorting should prioritize special handling requirements (fragile, priority, etc.).
- **Baggage Tracking:**
 - The system must provide real-time tracking information to passengers regarding the location of their baggage.
 - Baggage status updates, including transfers and arrivals, must be accessible through a mobile application and airport displays.

FUNCTIONAL REQUIREMENTS FOR BAGGAGE HANDLING SYSTEM (2)

- **Security Checks:**
 - Baggage must undergo security screening in compliance with aviation regulations.
 - The system must flag and notify security personnel of any suspicious items detected during screening.
- **Transfer Handling:**
 - The system must facilitate smooth transfers of baggage between connecting flights.
 - Transfer times and processes should be optimized to minimize the risk of missed connections.
- **Baggage Claim:**
 - Baggage must be automatically routed to the correct baggage claim area.
 - Passengers must receive timely notifications on the arrival and location of their baggage at the designated claim carousel.

NFRS FOR BAGGAGE HANDLING SYSTEM (1)

- **Reliability:**

- The baggage handling system must have a reliability rate of 99.9%, minimizing the risk of lost or mishandled baggage.

- **Performance:**

- Baggage processing times, from check-in to claim, must not exceed 30 minutes on average.
- The system must handle a peak load of 1,000 baggage transactions per hour during high-traffic periods.

- **Scalability:**

- The system must scale to accommodate the growth in passenger numbers, supporting at least a 20% increase in capacity within the next five years.

NFRS FOR BAGGAGE HANDLING SYSTEM (2)

- **Security:**

- Access to the baggage handling system must be restricted to authorized personnel only.
- Data related to passenger baggage must be encrypted and secured to protect sensitive information.

- **Compliance:**

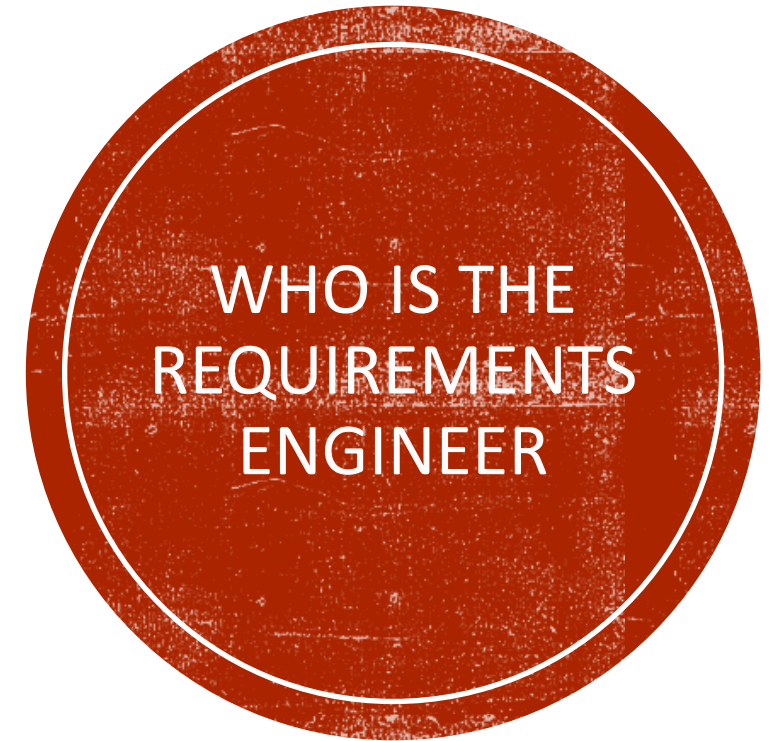
- The system must adhere to international aviation and security regulations, including IATA guidelines for baggage handling.

In companies those hold the positions of:

- Requirements Analyst
- Product Owner
- Project Manager
- Software or System Analyst

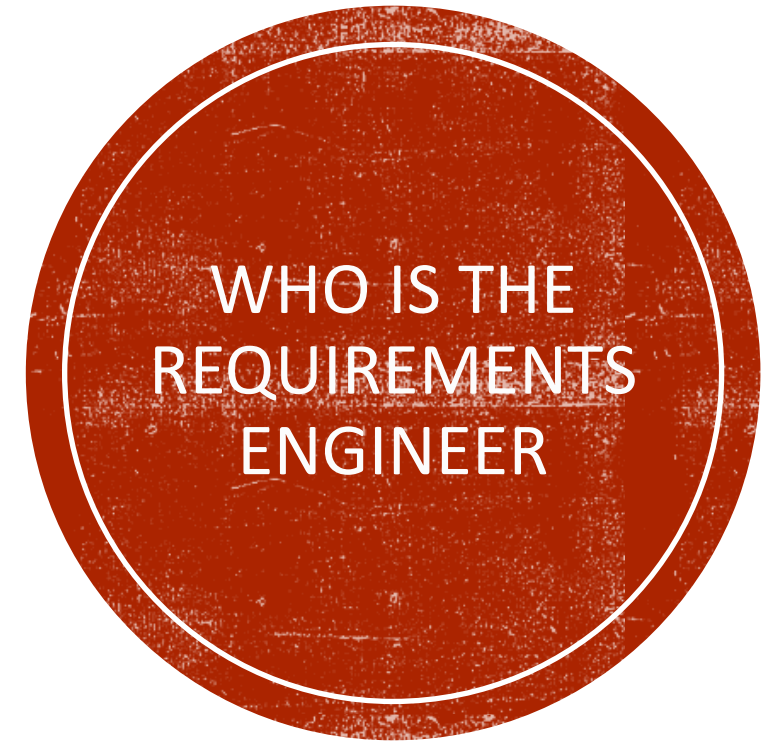
There are other positions that may have requirements engineering responsibly; for example:

- Quality Assurance Analyst
- User Experience Designer
- Process Improvement Specialist
- IT Consultant



Requirements Engineer should be:

- Be a good manager (to manage the process)
- Good listener
- Fair
- Good negotiator
- Understand the problem domain
- Multidisciplinary (e.g., traditional hard sciences and engineering augmented with communications and management skills)

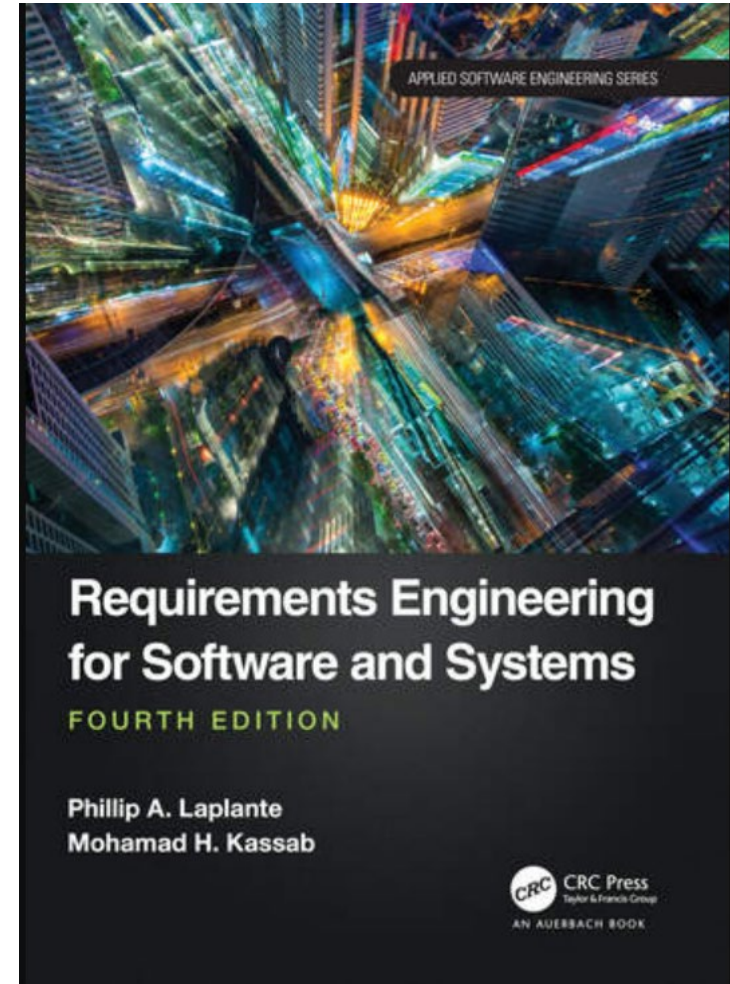


REQUIREMENTS ELICITATION TECHNIQUES

CHAPTER 3 OF REQUIREMENTS ENGINEERING FOR SOFTWARE AND SYSTEMS

MANDATORY READING

- Brainstorming
- Card Sorting
- Crowd Sourcing
- Designer as Apprentice
- Domain Analysis
- Ethnographic Observation
- Goal Based Approaches
- Group Work
- Interviews
- Introspection
- JAD
- Laddering
- Protocol Analysis
- Prototyping
- QFD (encompasses card sorting and laddering and domain analysis)
- Questionnaires
- Repertory Grids
- Reverse Engineering
- Task Analysis
- Scenarios & Use Cases
- User Stories
- Viewpoints
- Workshops



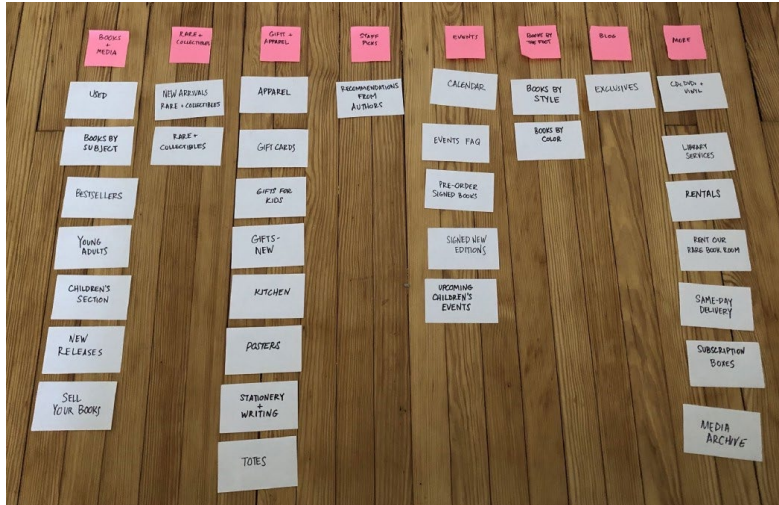


1. BRAINSTORMING

- Informal gatherings with customers and other stakeholders to generate overarching goals for the systems.
- Some preliminary requirements may be generated.
- Useful mostly for generating the business goals (mission statement), and high level features of the system.

2. CARD SORTING

- Involves having stakeholders complete a set of cards that include key information about functionality (may include ranking and rationale)
- RE then organizes these cards in some manner
- The sorted cards can be used as an input to the process to develop CRC (capability, responsibility, class) cards to determine program classes in the eventual code



2. CARD SORTING (POINT OF SALE SYSTEM)



2. CARD SORTING (POINT OF SALE SYSTEM)

Customer management

Identify customer if
returning

Priority – high

Manage customer
loyalty feature

Priority – Medium

Tax functions

Prepare sales tax
reports

Priority – High

Apply sales tax to
non-food items

Priority – Medium

Inventory features

Update inventory
records

Priority – High



3. DOMAIN ANALYSIS

- Involves assessing the “landscape” of related and competing applications
- Can be useful in identifying essential functionality and later, missing functionality
- Can be used downstream for identifying reusable components (e.g., open-source elements)

4. DESIGNER AS APPRENTICE



“A new computer system changes how its customers work. Designing such a system requires intimate knowledge of customers' work and motives to ensure that the system supports them well. The creation of a new system implicitly means designing the new work practice it will support...The fundamental problem in the relationship between customers and designers is to enable learning: how do designers learn enough about customers' work to design well?

What kind of relationship allows customers to impart deep knowledge about their work?”

4. APPRENTICESHIP



The relationship between master craftsman and apprentice stands out as a useful model.

The apprentice learns a skill from the master just as we want the designer to learn about the customers' work from the customer.

4. DESIGNER AS APPRENTICE



The critical aspects of the relationship are:

- Teaching ability is not needed
- Seeing the work reveals what matters
- Seeing the work reveals details
- Seeing the work reveals structure

4. DESIGNER AS APPRENTICE



- Customers cannot talk about their work effectively but can talk about it as it unfolds.
- Don't have to work out the best way to present it, or the motives; they just explain what they're doing.

"I'm scanning each product as it comes down the conveyor belt. If there are multiples of any one product I enter the number of same items, press the # key, then enter the price and press 'total'..."

4. DESIGNER AS APPRENTICE



- People are not aware of everything they do and sometimes why they do it.
- Some actions are result of years of experience and have subtle reasons; others are habit and no longer have a valid justification.
- The presence of an apprentice provides the opportunity for the master (customer) to think about the activities and how they came about.

4. DESIGNER AS APPRENTICE



- Patterns of working are not always obvious to the worker.
- An apprentice learns the strategies and techniques of work by observing multiple instances of a task and forming an understanding of how to do it themselves, incorporating the variations.

5. ETHNOGRAPHIC OBSERVATION

- Based on detailed (at the level of a social scientist) observations of human activity.
- Involves long periods of observation (hence, an objection)
- Direct and indirect evidence is gathered
 - The work or activity itself
 - Evidence derived from the surroundings that may not be communicated directly
- Designer as Apprentice is a special case of ethnographic approach



5. ETHNOGRAPHIC OBSERVATION



- Consider gathering requirements for a smart system.
- You spend long periods of time interviewing the customer about what she wants
- You spend time interacting with the customer as she goes about her day and ask questions (*“why are you running the dishwasher at night, why not in the morning?”*)
- You spend long periods of time passively observing the customer *“in action”* in their current home to get non-verbal clues about her wants and desires
- You gain other information from the home itself – the books on the book shelf, paintings on the wall, furniture styles, evidence of hobbies, signs of wear and tear on various appliances, etc.

6. GOAL BASED APPROACHES

- Start from mission statement and provides lower level goals brought.
- Lower-level goals are then branched out into specific high-level requirements.
- High-level requirements then generate lower-level ones.



6. GOAL BASED APPROACHES: BAGGAGE HANDLING SYSTEM

- Mission Statement: “To automate all aspects of baggage handling from passenger origin to destination.”
- Goal 1: To completely automate the tracking of baggage from check in to pick up
- Goal 2: To completely automate the routing of baggage from check in counter to plane
- Goal 3: To reduce the amount of lost luggage to .1%...



7. GROUP WORK

- General term for any kind of group meetings
- Difficult to organize and focus the many stakeholders involved
- Problems of openness and candor can occur
- Certain individuals can dominate
- Can lead to feelings of being “left out”
- RE must be very skillful in leading these sessions to avoid such problems



8. INTERVIEWS

- Obvious and easy to use technique
- Three kinds of interviews exist
 - **Unstructured** – conversational, can be hit-or-miss based on skill of interviewer
 - **Structured** – uses pre-defined questions that have been rigorously planned
 - **Semi-structured** – uses combination of the above
- Care must be taken to ensure all of the right questions are asked
- Templates are very helpful when employed with interviewing



8. INTERVIEWS: SAMPLE QUESTIONS

- Name an essential feature of the system? Why is this feature important?
- How important is this feature with respect to other features?
- What other features are dependent of this feature?
- What other features must be independent of this feature?
- What other observations can you make about this feature?

9. INTROSPECTION

- RE develops requirements based on what he “thinks” the customer wants
- Think of : Building a “Chess Game”
- Useful when the RE’s domain knowledge far exceeds the customers’
- Introspection is probably to be avoided in cases where the customer has experience in the domain



10. LADDERING

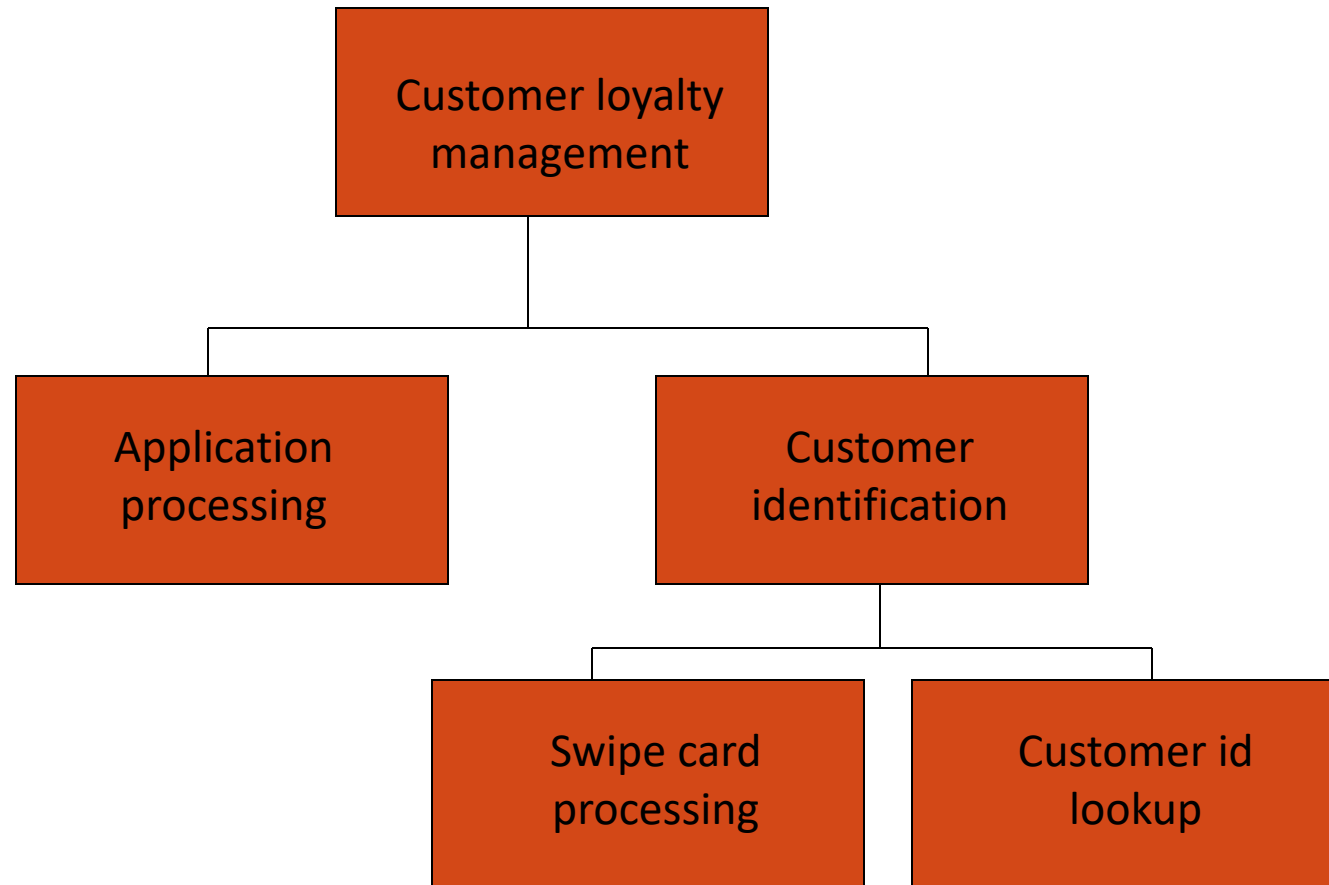


- Uses short prompting questions (“probes”) to elicit requirements
- Follow up questions dig deeper below the surface
- Assumes that information can be arranged in a hierarchical fashion
- Resultant information is then organized in some kind of tree structure

10. LADDERING: POINT OF SALE SYSTEM

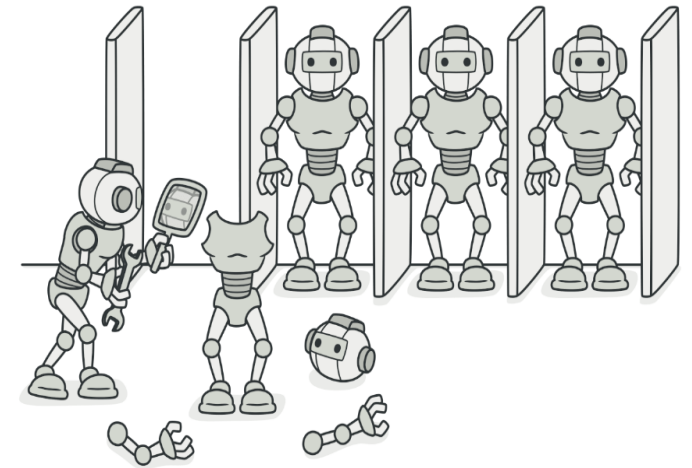
- Name a key feature of the system?
 - Customer identification
- How do you identify a customer?
 - They can swipe their loyalty card
- What if a customer forgets their card?
 - They can be looked up by phone number
- When do you get the customer's phone number?
 - When they complete the application for the loyalty card
- How do they complete the applications....

10. LADDERING



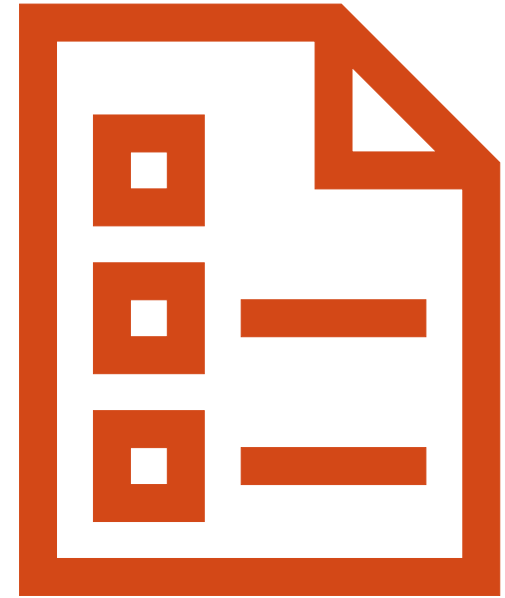
11. PROTOTYPING

- Involves construction of models of the code in order to discover new features
- Can involve working models (code) as well as non-working (storyboards, GUIs)
- Code can be throwaway and non-throwaway
- Architects use prototyping in the manner described previously
- Agile development consists of an ever-evolving non-throwaway prototype



12. QUESTIONNAIRES

- Straightforward technique consisting of survey instruments
- Used at early stages to quickly define the scope boundaries
- Survey questions can be closed (e.g., multiple choice, true false) or open-ended
- Danger in over-scoping and under-scoping if questions are not adequately framed
- Therefore, most useful when the domain is very well understood by both stakeholders and RE





12. QUESTIONNAIRES: SOME QUESTIONS FOR POINT OF SALE SYSTEM

- How many unique products (SKUs) do you carry in your inventory
a) 0-1000 b) 1001-10,000 c) 10,001-100,000 d) > 100,000
- How many different warehouse sites do you have? _____
- How many different store locations do you have? _____
- How many unique customers do you currently have? _____



13. VIEWPOINTS



- A way to organize information from the (point of view) of different constituencies.
- Various formats and applications for viewpoints in software and systems engineering
- In RE used for prioritization, agreement, and ordering of requirements
- Viewpoints incorporate a variety of information from business domain, process models, functional requirements specs, organizational models, etc.
- Viewpoints are generated for each view, and then reconciled using various approaches

14. WORKSHOPS

- Formal and informal gatherings of stakeholders to hammer out requirements issues
- Formal workshops are well planned but can be boring and tiring
- Informal workshops can be more lively, but overlook important elements



Consider the following systems

- I. A new autopilot system for a self-driving car
- II. An autonomous carpet cleaning robot
- III. A food processing plant control system
- IV. An expert system that automates cake decoration based on customer preferences

for which would it be appropriate to use Designer as Apprentice for requirements elicitation?

- ☐ II and III only
- ☐ I and II only
- ☐ IV only
- ☐ II and IV only

KNOWLEDGE CHECK

You are conducting elicitations for various user groups for a new high lift materials handling machine (basically, a long-reaching fork-lift). For the following pairs of elicitation techniques, which two are most likely to derive very similar information from a group of fork -lift drivers?

- ☐ questionnaires and interviews
- ☐ introspection and goal-based analysis
- ☐ brainstorming and viewpoints
- ☐ goal-based analysis and interviews

KNOWLEDGE CHECK

You are conducting requirements elicitation activities for a new smart washing machine/dryer combination. Which of the following problems would be least likely solved using prototyping?

- ☐ finding missing requirements
- ☐ identifying timing constraints
- ☐ identifying unwanted functionality
- ☐ identifying non-functional requirements

KNOWLEDGE CHECK

Some requirements elicitation techniques can be conducted using the Internet and mobile technologies. Of the following list of elicitation techniques, which is least likely to be effective using remote means:

- ☐ surveys
- ☐ questionnaires
- ☐ brainstorming
- ☐ Designer as Apprentice

KNOWLEDGE CHECK

In order to elicit requirements for a generic smart home (to be marketed to the public) a partitioning of the user class is needed. Which of the following categories of users best partitions the users into classes that minimizes overlap?

- ☐ occupants, visitors, intruders
- ☐ friends, family, occupants, visitors
- ☐ friends, family, children, adults
- ☐ children, adults, senior citizens, homeowners

KNOWLEDGE CHECK

ELICITATION SUMMARY

All these techniques have advantages and disadvantages (partially discussed)

Some are too general, some too specific, some rely too much on stakeholder knowledge, some not enough, etc.

Combination of techniques really the best way to go

We can group techniques in the following categories (interviews, domain-oriented, group-work, ethnography, prototyping, goals, scenarios, viewpoints)

Following tables show relationships between technique groups





IMPORTANT... KEEP IN MIND...
WHAT THE SYSTEM BOUNDARY IS...



CONTEXT DIAGRAMS

Show the system boundary

- The environment with which the system interacts
- External interfaces
- Others systems on which it relies

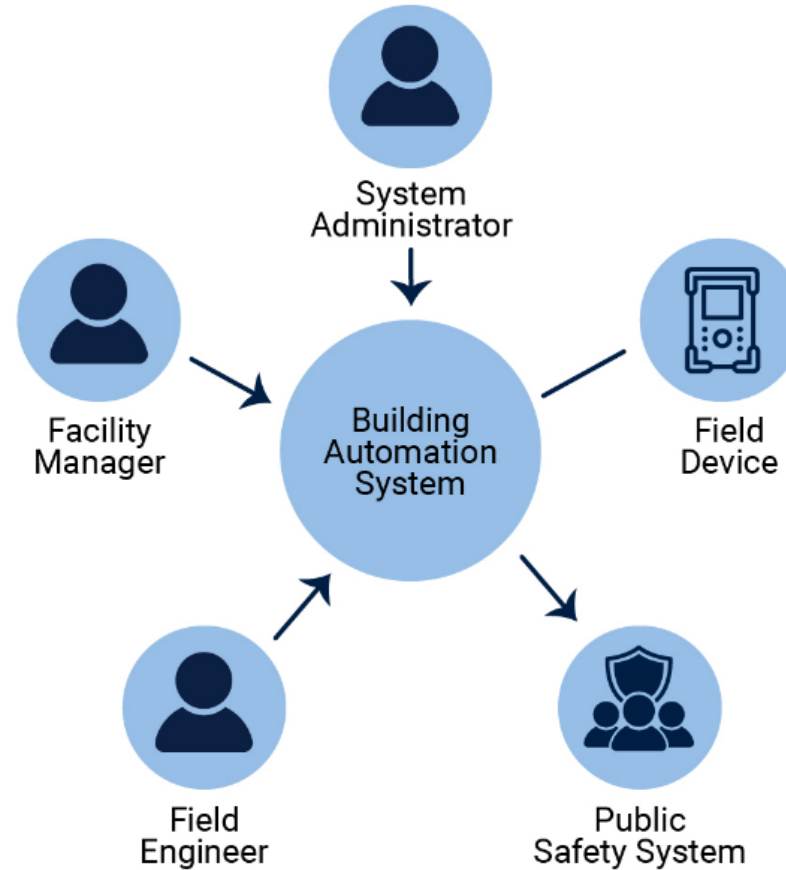
Indicate what the system's environment is

- May include users with different roles
- May include other computing systems
- Environment may be very simple, or very complex

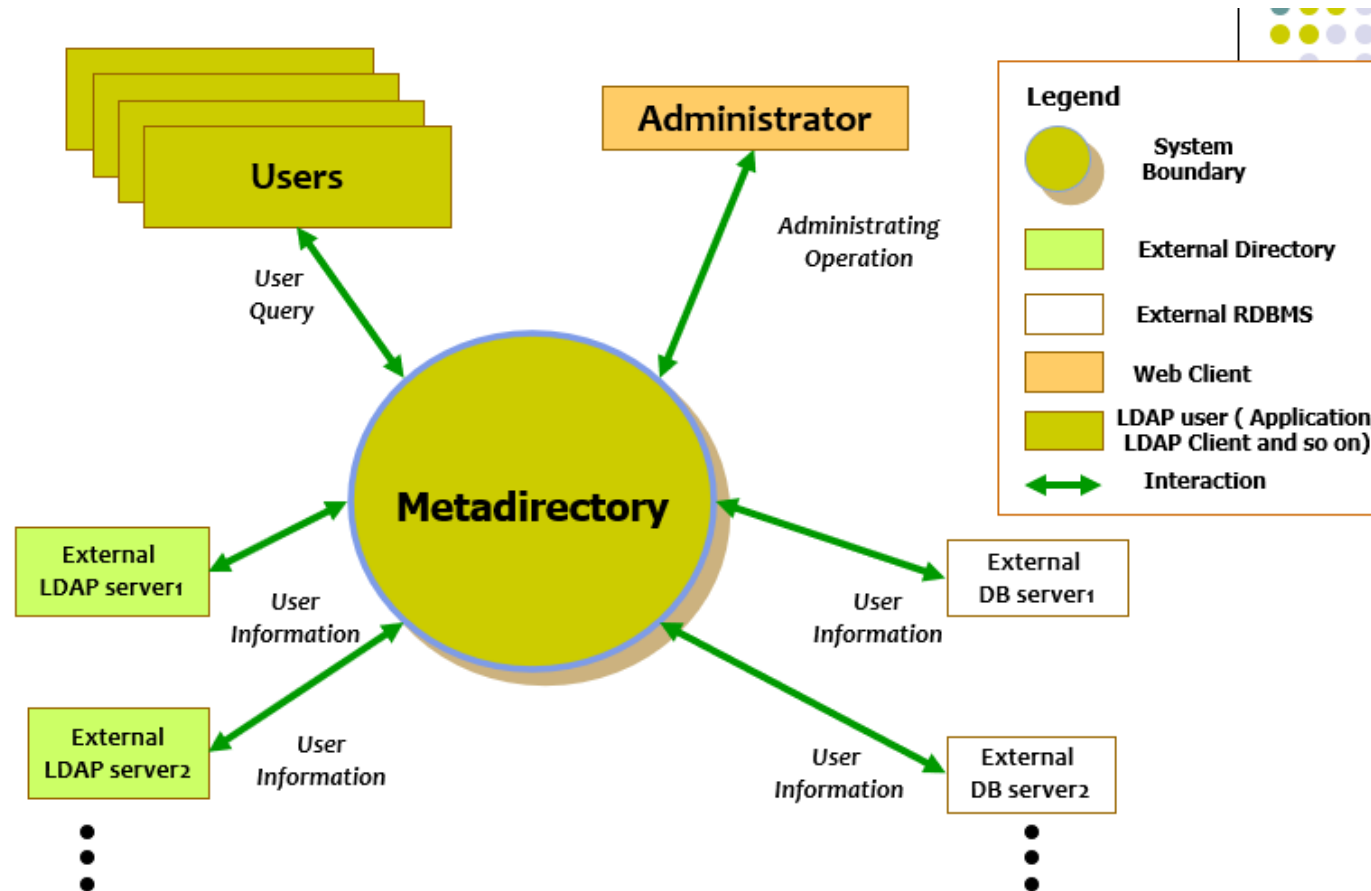
May have multiple context diagrams

- Different context diagrams for different views
- Usually given for C&C view

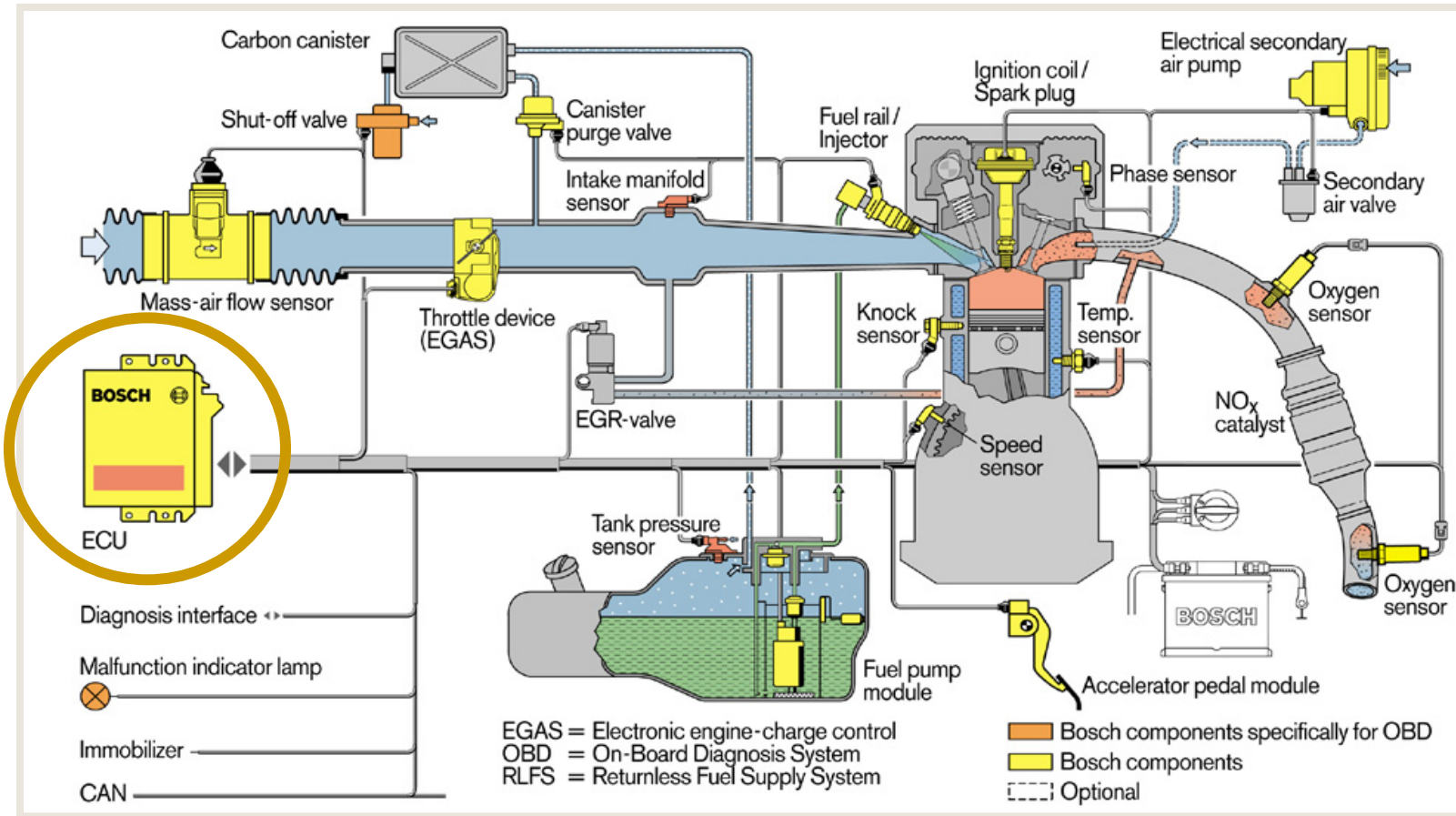
CONTEXT DIAGRAM – EXAMPLE 1



CONTEXT DIAGRAM – EXAMPLE 2



CONTEXT MAY BE COMPLEX



TOOLS TO ASSIST IN REQUIREMENTS ELICITATIONS

- Mind Mapping tools (e.g., <https://www.mindmup.com/>)
- GPTs (e.g., AlphaReq)

READINGS , ASSIGNMENTS, PROJECT TASKS



Read:

- Requirements Elicitation Techniques (Mandatory Reading available in Supplemental Readings folder)
- Chapters 7, 8 of the textbook.

Assignments & Project Tasks:



- The project is open. Continue forming the teams and working on Project Deliverable 1 (Deadline: 31/1/2025).
- Assignment 1 is open. This is an individual Assignment. (Deadline: 07/2/2025)



QUESTIONS?