SENG321: Requirements Engineering

more on

WHAT IS REQUIREMENTS ENGINEERING?

QUIZ

7 min long

Only Bright can be open; close every other window on your laptop

Place phones or other devices in your bag.

Respect the others: Stay seated and quiet for the 7 min; once submitted, you can open other windows

THIS and Next Week-> OUTLINE of LECTURE

1. Get to know your project team – **TEAMWORK!**

2. (re-) Familiarize yourself with **Github** (see GH repo) – **RFP template** available in Project space

   - when is the RFP due?

3. Identify **Real-world Organization** and Schedule/conduct Interviews – **TO DEVELOP** the **Request for Proposals**

- How does it fit within the **Overall Process**?
- **Interviewing Techniques**, **Types of Requirements**
- structure of RFP

THE STRENGTH

Your team and teamwork!

Teamwork!
The hardest and yet most rewarding part

- Open minded
- Open to taking risks
- Reliable
- Hard working / good work ethic
- Good communicator
- Accountable
- Dependable
- Responsible
- ***Respectful***

Course expectations

Act as Professionals

This is an intense and fast pace course, so learn to manage your time – another hardest skill!

DO NOT BE LATE TO CLASS! It is Disrespectful to your instructor and work group!

Coming and Participating! to lectures and Project work

Teams

Meet regularly

Self-organize: Decide on team lead roles and reflect on how that matches personalities/strengths

Run as a real project. Show initiative, commitment, autonomy.

Communication lines (in this order):

1. check course/project info in Bright/GH
2. DM your team mates
3. DM your TA (email or Teams)
4. email your instructor

Expected weekly **workload is heavy**: 3 hrs lectures plus 8-10 hours outside class reading/project work (**reconsider taking this course if you have a busy term**)

Late comers (missed the first class) should familiarize themselves with all this information on Project Deliverables and Expectations. Get in touch with your project Team ASAP.

SENG 321 Requirements Engineering Contributions to project Teamwork

To be completed in a survey (link will be provided prior to deadline)

These reports are not optional.

Each student is required to create several reports documenting the assessment of each team member of your group, including yourself. There will be an assessment **after each project iteration.**

Each report should document who did what during the iteration for each person, and your assessment of that person, again, including yourself.

Suggested criteria for assessing contribution to the group (client and description) work include:

- communication style
- completion of assigned tasks
- participation in meetings
- [illegible]
- [illegible]
- [illegible]

Gui[illegible]

- [illegible]
- [illegible]
- You will be marked on how perceptive, insightful, and reasonable your comment are.
- Provide a **letter grade** per person, alongside your paragraph-long justification.
- Do not give all A+s (or any other grade) for every team member-- this is not appropriate, and is obvious to the teaching team that a lack of effort went into your assessment.
- **Do NOT share your report with anyone. It is confidential**

  Grading: the quality of writing in these reports is assessed as 5% of the overall course grade. Furthermore, the content of these reports provide information about each team member's individual contribution to the project and will be used to adjust the overall course grade for an underperforming or overperforming team member, to reflect the marking of individual perofrmance as defined in the course outline.

AT END OF EACH ITERATION (5 TIMES)

START WITH ITERATION 0 (FIRST TWO WEEKS!)

Univers [illegible] ring

YOU ARE IN CHARGE

Listen, Engage, Reflect, Help out

Class format and participation

Most classes include both Theory (lectures) and In-class project work!

Attendance in project-related activities is mandatory, participation in class and projects part of the course mark (**10% participation mark**, mere attendance to classes does not guarantee any portion of this grade).

Communication is key to success.

LETS START

With a fun exercise

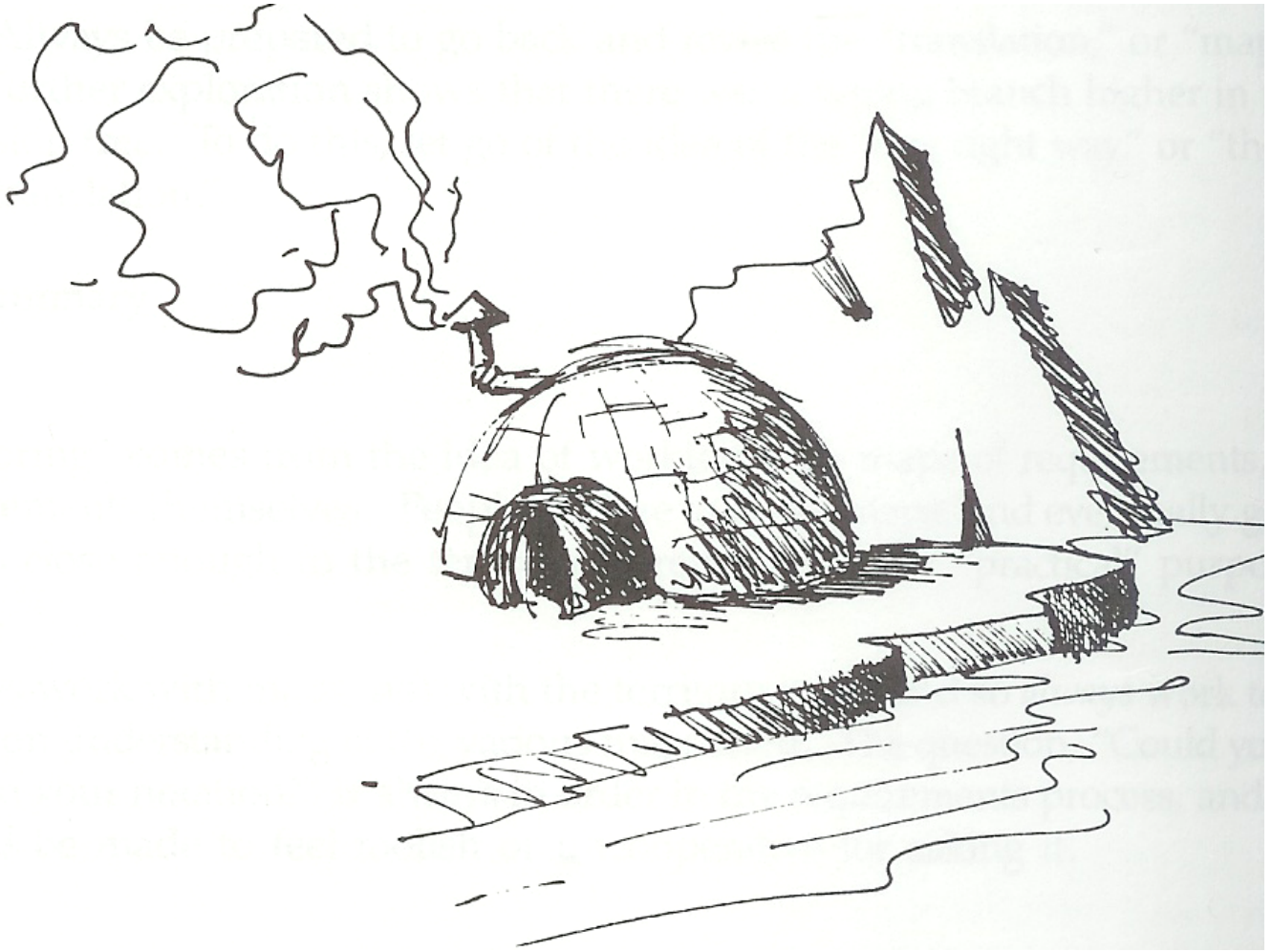
What's the best Solution?

Requirement:

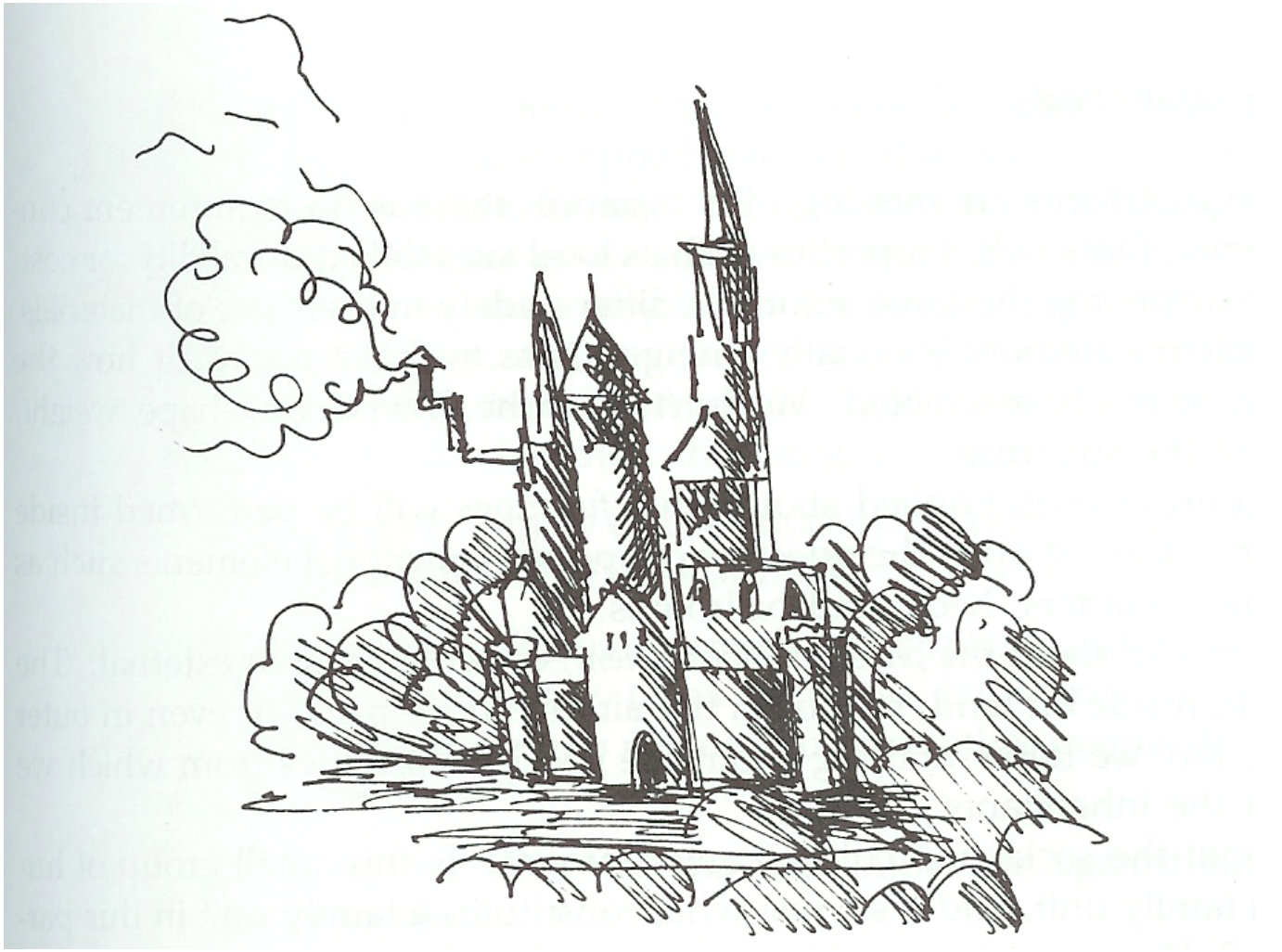
"create a means for protecting a small group of human beings from the hostile elements of their environment"

Possible solutions…

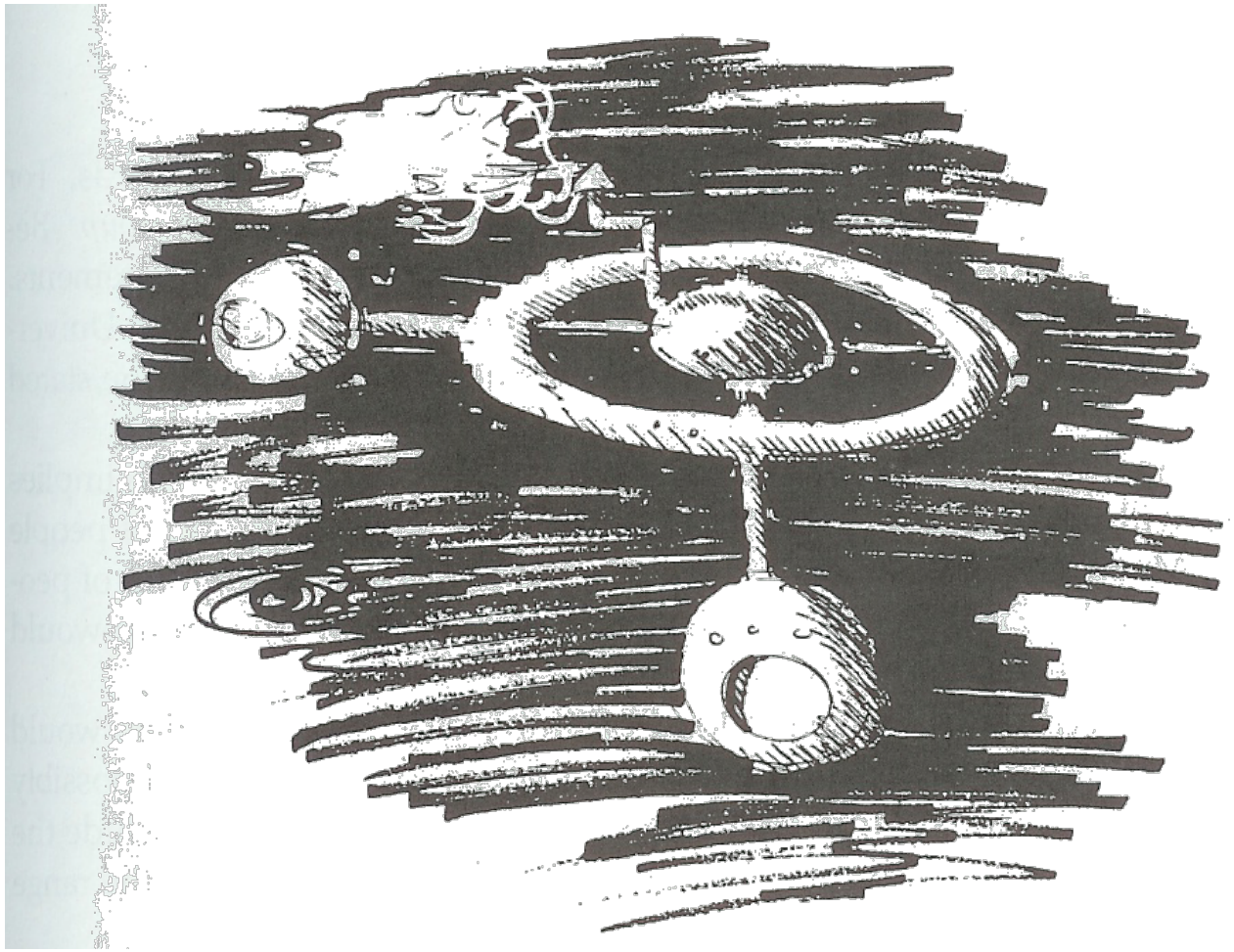
[Gause and Weinberg, 1989: Exploring requirements: quality before design]



Igloo – an indigenous home constructed of local building materials



Bavarian castle – a home constructed to impress the neighbors



Space station – a mobile home with a view

Software quality

what do you think about it?

Requirements Engineering

Software quality as **fitness for purpose**

The case of software-intensive systems

Hardware, software and human activities

… Requirements Engineering: set of activities to identify purpose in the context of human activities

BUT…

Why is this so hard?

Requirements Engineering

Complexity of purpose

human-computer interaction: intricate and complex

Design of software intensive systems and the case of **wicked problems**:

No single formulation of the problem

Continuous need for exploration

No right or wrong solution

Unique characteristics

Hard vs. soft systems methodologies

Soft systems methods suited for wicked problems

Soft Systems methods adopt a Human-centered design view where:

RE = continuing negotiation process between multiple perspectives

The design of software is inseparable from the task of defining the human activities supported by that software

Requirements Engineering

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is so 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 resulting system if done wrong. No other part is more difficult to rectify later.

[Brooks S., 1995: No silver bullet]

Requirements Engineering (def)

Requirements Engineering (RE) is a set of activities concerned with **identifying** and **communicating the purpose** of a software intensive system, and the **contexts** in which it will be used. Hence, RE acts as the bridge between the **real-world needs of users, customers**, and other **constituencies** affected by a software system, and the **capabilities and opportunities** afforded by software-intensive technologies

[Easterbrook, Chapter 1]

A spectrum of software-intensive systems

No human interaction	Limited human interaction	Tightly coupled interaction with human interaction
Networking, Middleware	Aircraft control systems	Office automation, Groupware, Information systems, Web services

The type of system may determine how requirements are elicited, analyzed and negotiated

e.g. Interviews, Ethnographic techniques
Participatory design

Requirements describe problems

Requirements

describe the problem statement (*or*, a desired effect to improve the problem situation)

and

activities to be supported by the system

Ideal: separate the problem from solutions

Practice: very difficult to accomplish

Requirements Engineering: the tension between describing the problem vs. solution

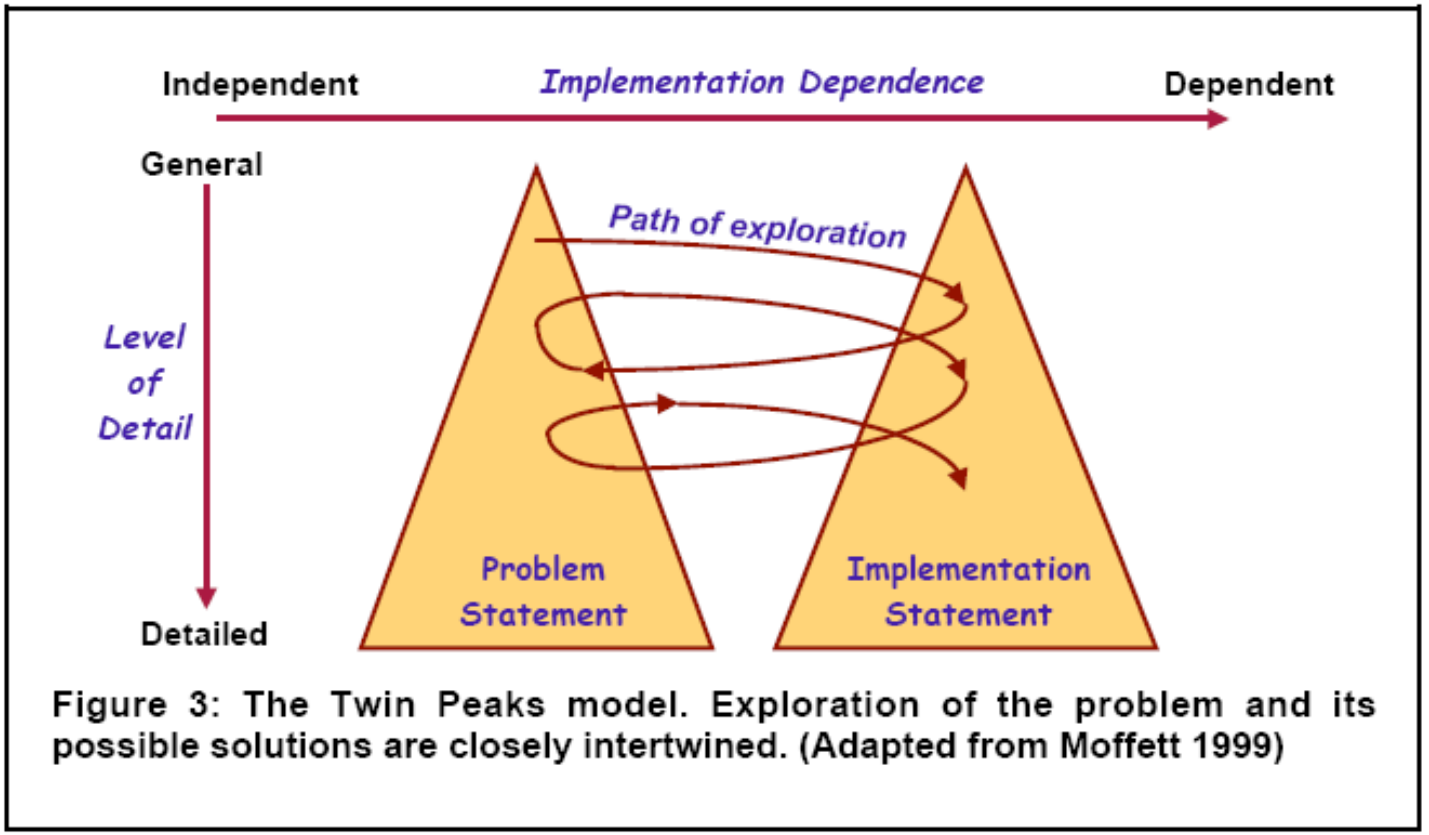
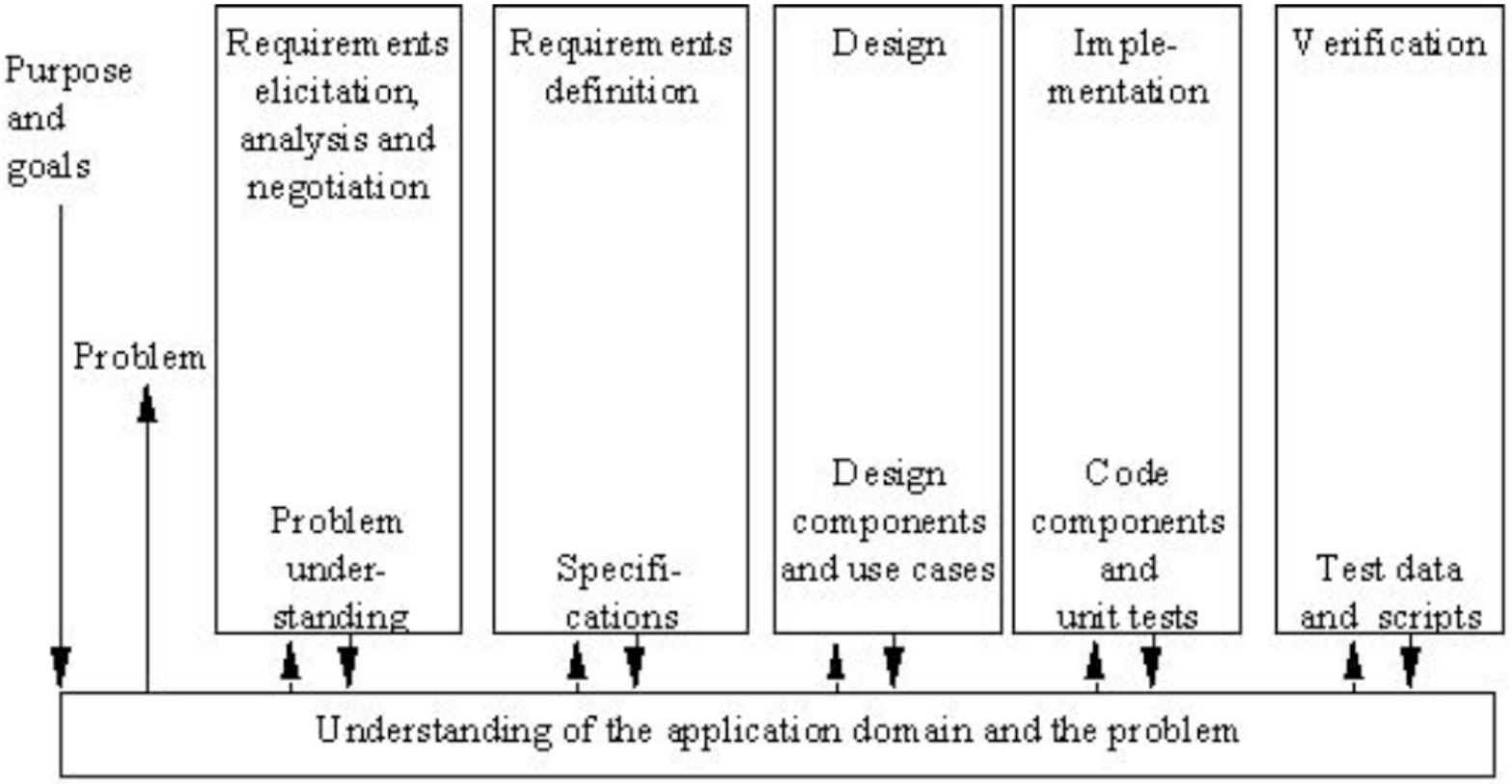


Figure 3: The Twin Peaks model. Exploration of the problem and its possible solutions are closely intertwined. (Adapted from Moffett 1999)

➔ Several approaches, waterfall vs. agile development at two extremes

Requirements Engineering in Software Development



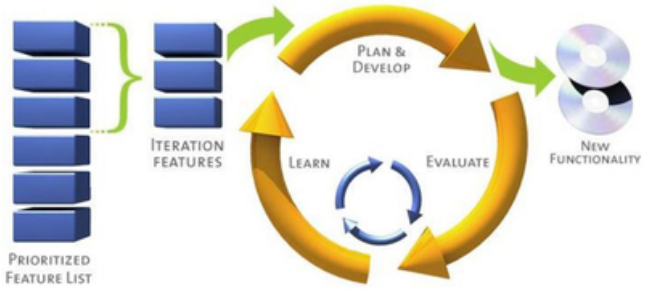
Activities of the software development process

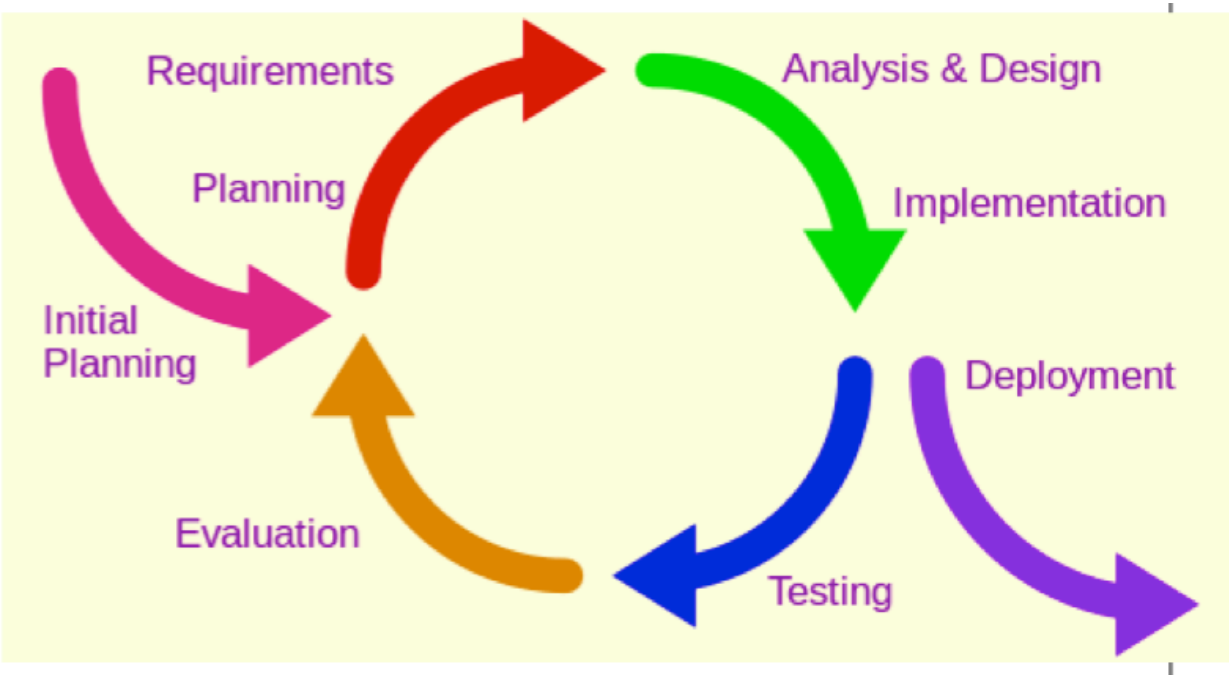
Towards processes that mitigate project risks

Continuous access to stakeholders

Incremental and iterative development approaches

E.g. SCRUM





RE effort in various development processes

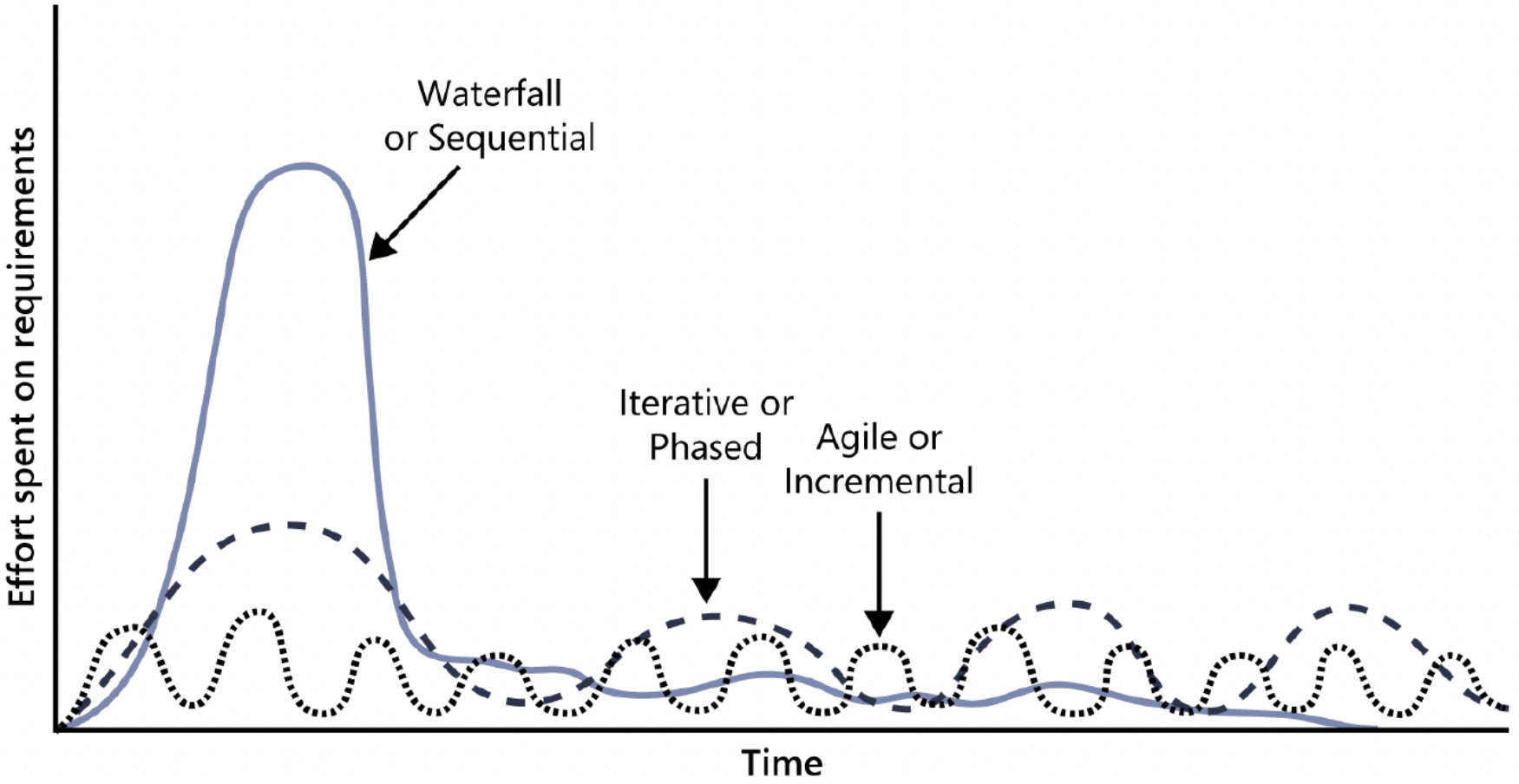
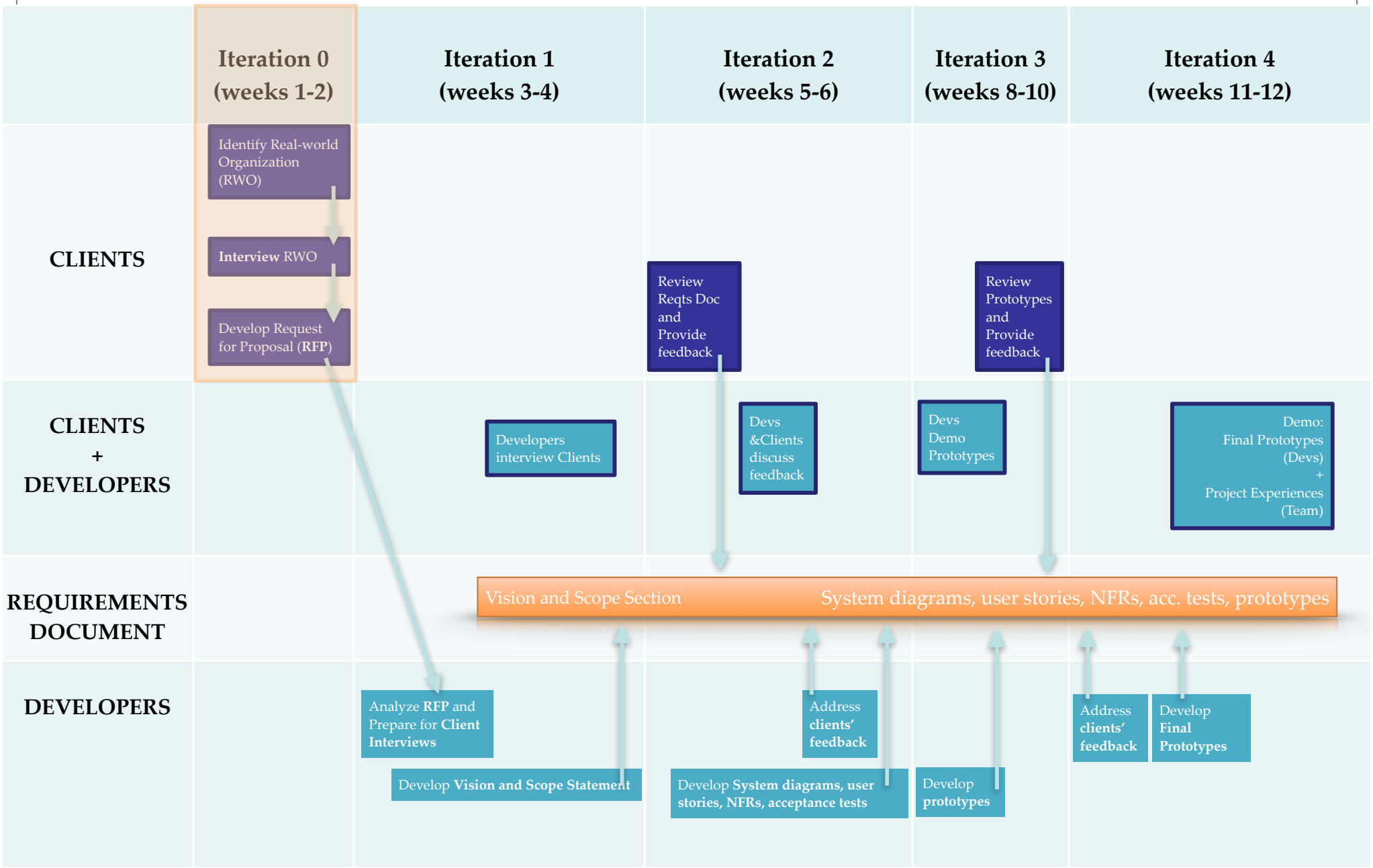


FIGURE 3-3 The distribution of requirements development effort over time varies for projects that follow different development life cycles.


Iteration 0 (weeks 1-2)
Iteration 1 (weeks 3-4)
Iteration 2 (weeks 5-6)
Iteration 3 (weeks 8-10)
Iteration 4 (weeks 11-12)
CLIENTS
Identify Real-world Organization (RWO)
Interview RWO
Develop Request for Proposal (RFP)
Review Reqts Doc and Provide feedback
Review Prototypes and Provide feedback
CLIENTS + DEVELOPERS
Developers interview Clients
Devs &Clients discuss feedback
Devs Demo Prototypes
Demo: Final Prototypes (Devs) + Project Experiences (Team)
REQUIREMENTS DOCUMENT
Vision and Scope Section
System diagrams, user stories, NFRs, acc. tests, prototypes
DEVELOPERS
Analyze RFP and Prepare for Client Interviews
Develop Vision and Scope Statement
Address clients' feedback
Develop System diagrams, user stories, NFRs, acceptance tests
Develop prototypes
Address clients' feedback
Develop Final Prototypes

Requirements Specification: Bridging two different worlds

application domain vs. machine domain
[Jackson, The meaning of requirements, 1997]

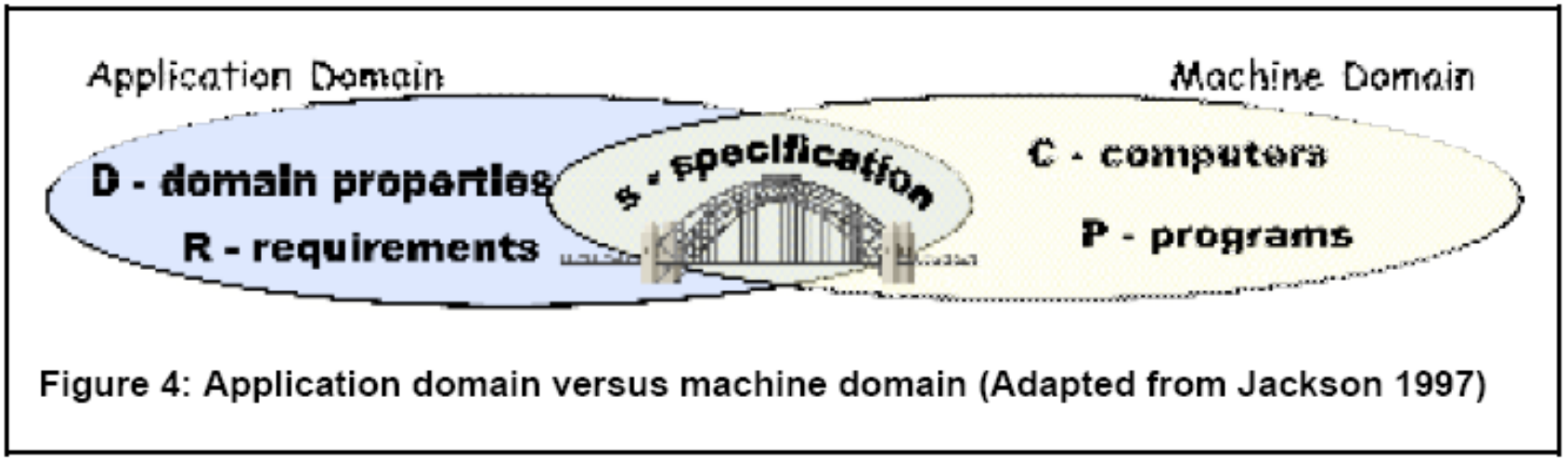


Figure 4: Application domain versus machine domain (Adapted from Jackson 1997)

'wet' vs. 'dry' aspects in software engineering
[Goguen, Requirements Engineering as the reconciliation of social and technical issues, 1994]

Example – domain property and req vs. specification statement

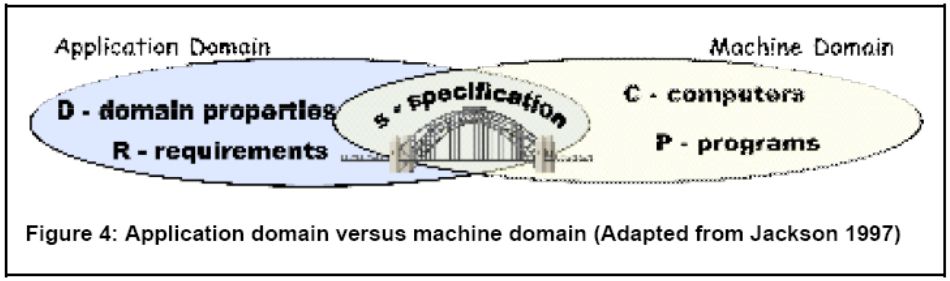


Figure 4: Application domain versus machine domain (Adapted from Jackson 1997)

Context: Design a *secure* means (software, i.e the machine) to store data

Example of a **requirement** (for the machine): "prevent access to unauthorized personnel"

Example of **Domain property**: "only a manager can assign access authority"

Specification for the machine: "when the user enters a valid password, the computer will unlock the door" (or "the system requires a login and password for access")

SO…

Is there a method behind the madness?

The RE Process

Typically involves activities of:

- Elicitation
- Analysis
- Specification
- Validation and verification
- Negotiation
- Management

Types of Reqts

Functional, non-functional:

- Functionality
- Physical environment
- Interfaces
- Users and human factors
- Documentation
- Data
- Resources
- Security
- Quality assurance

The RE Process

Typically involves activities of:

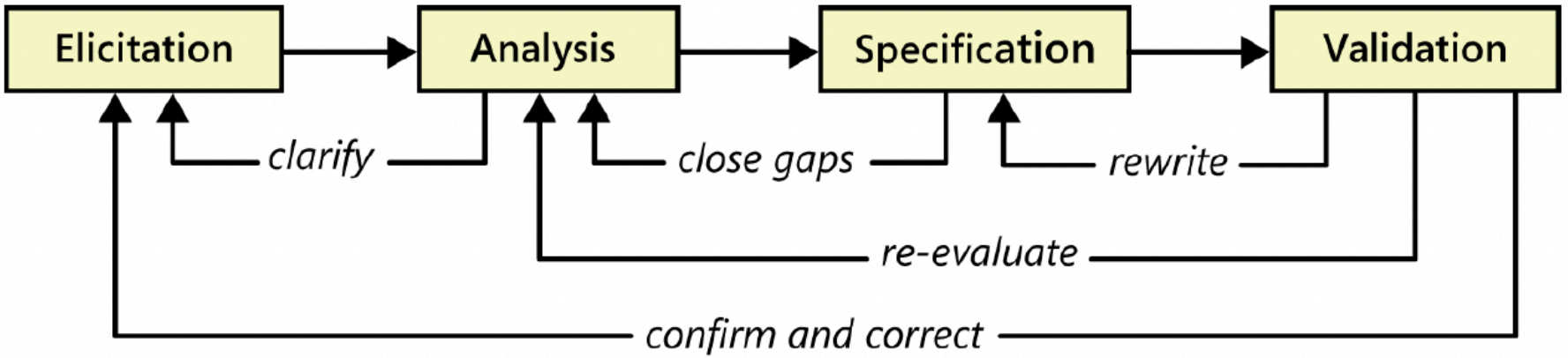


FIGURE 3-1 Requirements development is an iterative process.

Management

Types of Reqts

Functional, non-functional:

Resources

Security

Quality assurance

A spectrum of project types, different customer-supplier relationships

The type of system may determine the RE techniques and processes for elicitation, analysis, validation and negotiation!

Another way of looking at requirements knowledge

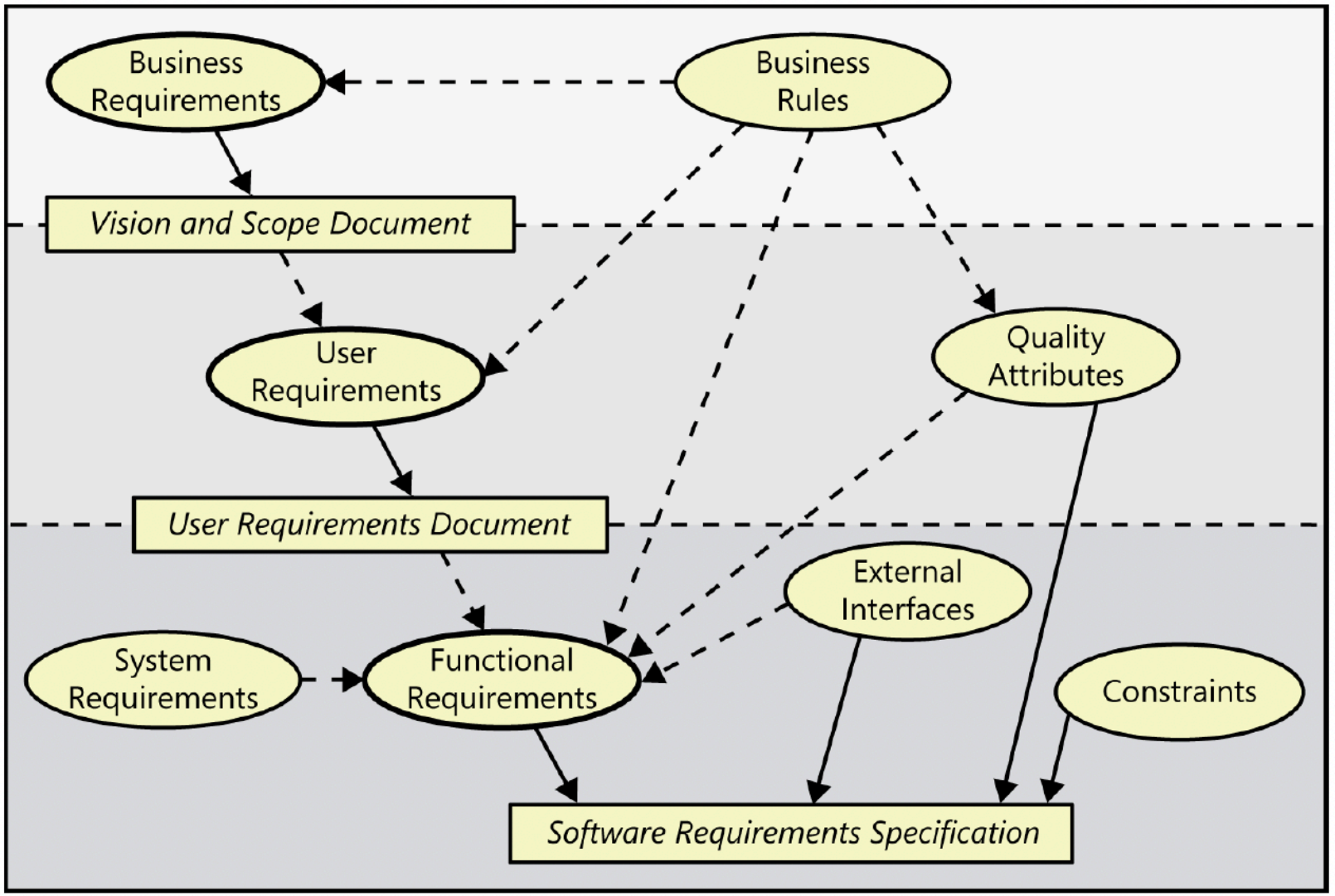


FIGURE 1-1 Relationships among several types of requirements information. Solid arrows mean "are stored in"; dotted arrows mean "are the origin of" or "influence."

FOR NOW

Focus on Requirements Elicitation

In order for Developers to conduct requirements elicitation of Clients:

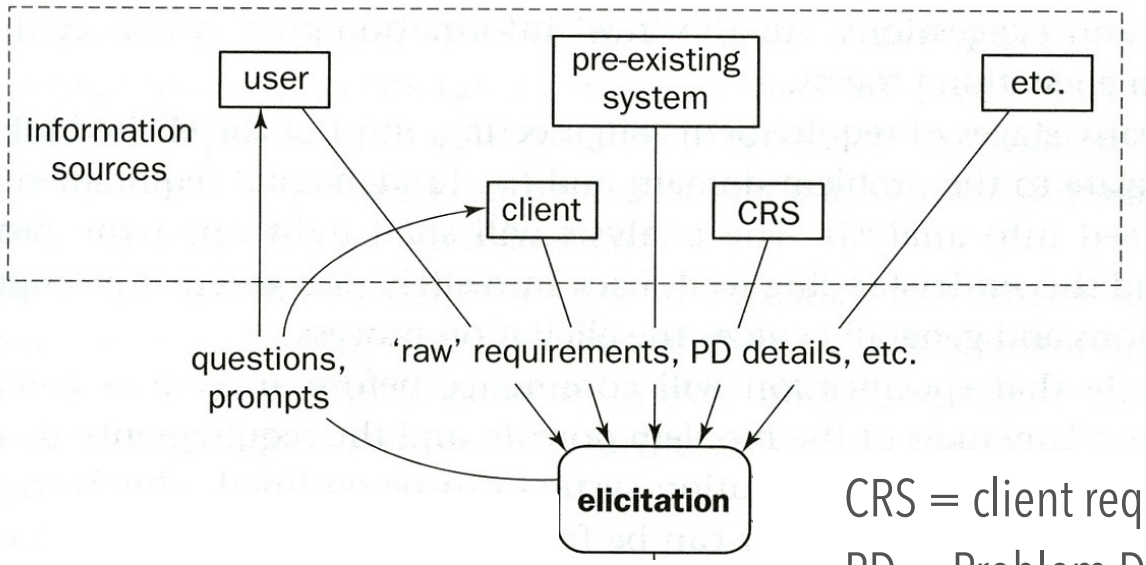
Clients conduct interviews with the SWO to elicit authentic details and Develop the RFP (see Example RFP (321 project) and Industry RFP)

The RE Process: Reqts Elicitation

What information should be gathered?

From what sources can it be gathered?

By what mechanisms or techniques it may be gathered?



CRS = client requirements specification

PD = Problem Description

Reqts Elicitation: Goals and Info to gather

Identify the business need and objectives

Identify the product's expected user classes and other stakeholders

Understand user tasks and goals, business objectives

Learn about the environment in which the new product will be used

Reqts Elicitation: Activities

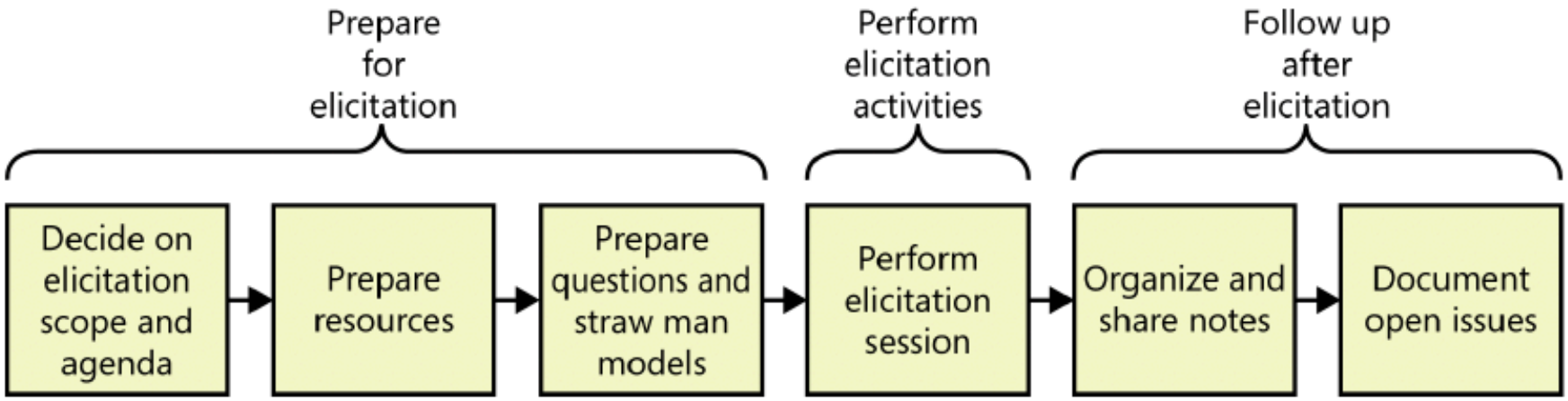


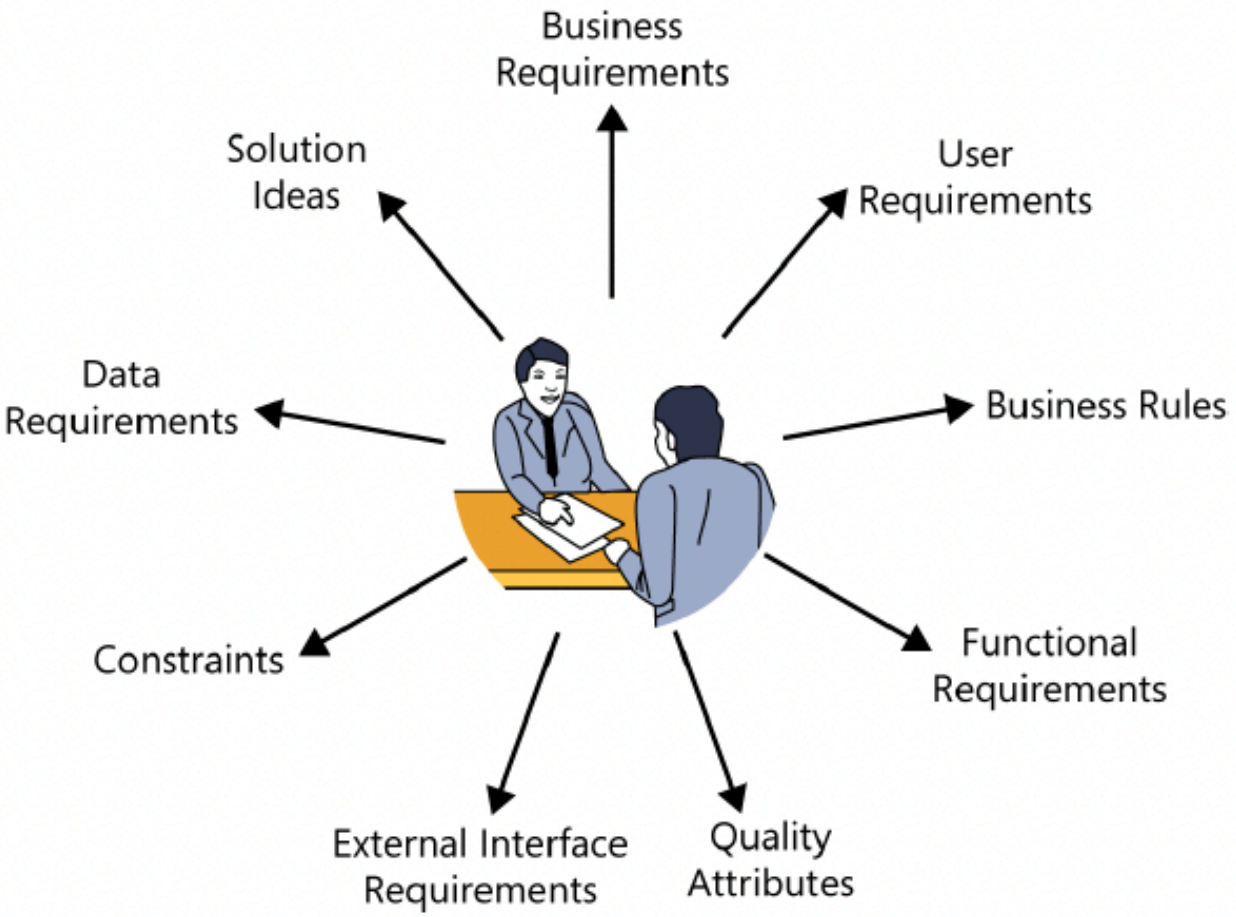
FIGURE 7-2 Activities for a single requirements elicitation session.

In your Projects

Clients elicit requirements from the RWO

Developers elicit requirements from the Clients team

Reqts Elicitation: Classifying customer input



What you might learn from the RWO

What previous students said…

Advice for future classes

Our advice to future students taking this course

- Plan, plan, plan!!
- Development a roadmap that outlines major milestone and task deadlines to keep everyone on the same page.
- Just because it's a project course, doesn't mean it's easy.
- Do not underestimate the course load.
- Ask lots of questions to avoid confusion

Types of requirements

Functional vs. non-functional, OR

Functionality

Physical environment

Interfaces

Users and human factors

Documentation

Data

Resources

Security

Quality assurance

[Pfleeger, S. (1998) . Software Engineering: Theory and Practice]

Types of requirements
Functionality

- What will the system do?
- When will de system do it?
- Are there several modes of operation?
- How and when can the system be changed or enhanced?
- **Are there constraints on execution speed, response time? (aka non-functional reqts)**

Types of requirements
Quality assurance

- What are the requirements for reliability, availability, maintainability, security?
- What is the prescribed mean time between failures?
- Is there a maximum time allowed for restarting the system after a failure?
- What efficiency measures will apply to resource usage and response time?

Types of requirements
Security

- Must access to the system or information be controlled?
- How will one user's data be isolated from others?
- How will user programs be isolated from other programs and from the operating system?
- How often will the system be backed up?

Types of requirements
Physical environment

- Where is the equipment to function?
- Is there one location or several?
- Are there any environmental restrictions, such as temperature, humidity or magnetic interference?

Types of requirements
Interfaces

- Is the input coming from one or more other systems?
- Is the output going to one or more other systems?
- Is there a prescribed way in which the data must be formatted?
- Is there a prescribed medium that the data must use?

Types of requirements
Users and human factors

- Who will use the system?
- Will there be several types of users?
- What is the skill level of each type of user?
- What kind of training will be required for each type of user?
- How difficult will it be for a user to misuse the system?

Types of requirements
Documentation

- How much documentation is required?
- Should it be on-line in book format or both?

References

Thayer, R.H. and Dorfman, M.: Software Requirements Engineering, IEEE Computer Society Press, 2000

Macaulay, L.A.: Requirements Engineering, Springer, 1996

Elicitation techniques
Interviews

Most widely used technique in requirements engineering

Analysts interview future users of the system individually to find out

what the present system does and

what changes are needed

The information gathered during the interviews enables the analysts to design a new system that will eliminate the shortcomings of the current one.

Elicitation techniques
Interviews

Advantages

- Access to individual stakeholders and their opinions
- Rich collection of information
- Ability to adapt questions to particular situations

Disadvantages

- Information from multiple sources, hard to analyze
- Difficult to be a skilled interviewer
- May intimidate the interviewee

Elicitation techniques

Interviews

Five steps of an interview:

Preparing for the interview

Planning and scheduling the interview

Opening and closing the interview

Conducting the interview

Following up for clarification

Types of interviews:

- structured
- unstructured

Supporting material in Github Lectures/LectureNotes/interviewingtips.md

Elicitation techniques
Structured Interviews

Advantages

1. Forces an **organization** on the interview
2. Very **goal-directed**
3. Attempts to **remove distortion** from interviewees subjectively
4. Allows better **integration** of material after the interview
5. Forces the interviewee to be **systematic**
6. Requirements engineer **identifies gaps** in the knowledge
   which acts as a basis for questions
7. **Purpose** of session is clear to interviewee

Disadvantages

1. Needs **more preparation** by the requirements engineer
2. Needs to **study background** material extensively
3. May **overconstrain the interviewee**, preventing discovery of requirements
4. May **intimidate** the interviewee

Elicitation techniques
Unstructured Interviews

Advantages

1. Appropriate when the RE wants to **explore** an issue
2. Facilitates description of domain in a way that is **easy for the interviewee**
3. **Goal** is to establish rapport and to get a broad view

Disadvantages

1. Data acquired is often **unrelated and difficult** to integrate
2. Often exhibits **lack of structure**
3. Does not allow gathering of **specific knowledge**
4. Takes time and **training** to do well
5. **Similar questions** asked in future sessions may annoy interviewee