Requirements Elicitation

Software Requirements and Design
CITS4401

Week 2 Part 2

The 4 Major Activities of Requirements Engineering

- **Elicitation**
- Analysis
- Specification
- Validation

1. What is Reqs Elicitation?

- Process
- Essence
- Communication
- Scope

Requirements vs Specifications

"Requirements is probably the most **misused** word in our industry."

"Required means **non-negotiable**, yet in almost every project we see changed, bartered, and negotiated requirements"

"The term "specification" acknowledges that software development is an iterative and evolving process."

W.Royce, IEEE Software, Sep/Oct 2005

Get a copy from http://ieeexplore.ieee.org/ via UWA

Project Scope

- A critical element of requirements elicitation is informing the **project scope**.
- **Description of the software** to be built and its purpose
- **Prioritizing** the deliverables to ensure the customer's most important business needs
- **how: time and other resources available** to do this

Reqs Elicitation Challenges

- Articulation Problems
  - People don't know what they want so they can't tell you about it!
- Communication Barriers
  - Different "languages" in different domains
- Knowledge and Cognitive Limitations
  - Managing complexity
- Human Behaviour Issues
  - Conflicting needs
- Technical Issues
  - Change management
  - Too rigid adherence to methodology

2. Requirements Sources

- Goals
- Domain knowledge
- Stakeholders
- Business Rules
- Operational Environment
- Organisational environment

Sources: Goals

- Goals. The term “goal” (sometimes called “business concern” or “critical success factor”) refers to the overall, high-level objectives of the software.
- Goals provide the motivation for the software but are often vaguely formulated.
- Software engineers need to pay particular attention to assessing the value (relative to priority) and cost of goals.
- A feasibility study is a relatively low-cost way of doing this.

Sources: Domain knowledge.

Industry-specific concepts

Business process

Regulatory and compliance requirements

Sources: Business Rules

- These are statements that define or constrain some aspect of the structure or the behavior of the business itself.

- “A student cannot register in next semester’s courses if there remain some unpaid tuition fees” would be an example of a business rule for a university’s course-registration software.

Sources: operational environment

- Requirements will be derived from the environment in which the software will be executed.

- These may be, for example, timing constraints in real-time software or performance constraints in a business environment.

Src: organizational environment

- Software is often required to support a business process, the selection of which may be conditioned by the structure, culture, and internal politics of the organization.

- The software engineer needs to be sensitive to these since, in general, new software should not force unplanned change on the business process.

3. Methods

1. Interviews
2. Facilitated meetings
3. Prototypes
4. Observation

- *Scenarios*
- *User stories*

1. Interviews

- Interviewing stakeholders is a “traditional” means of eliciting requirements.

- Important to understand the advantages and limitations of interviews and how they should be conducted.

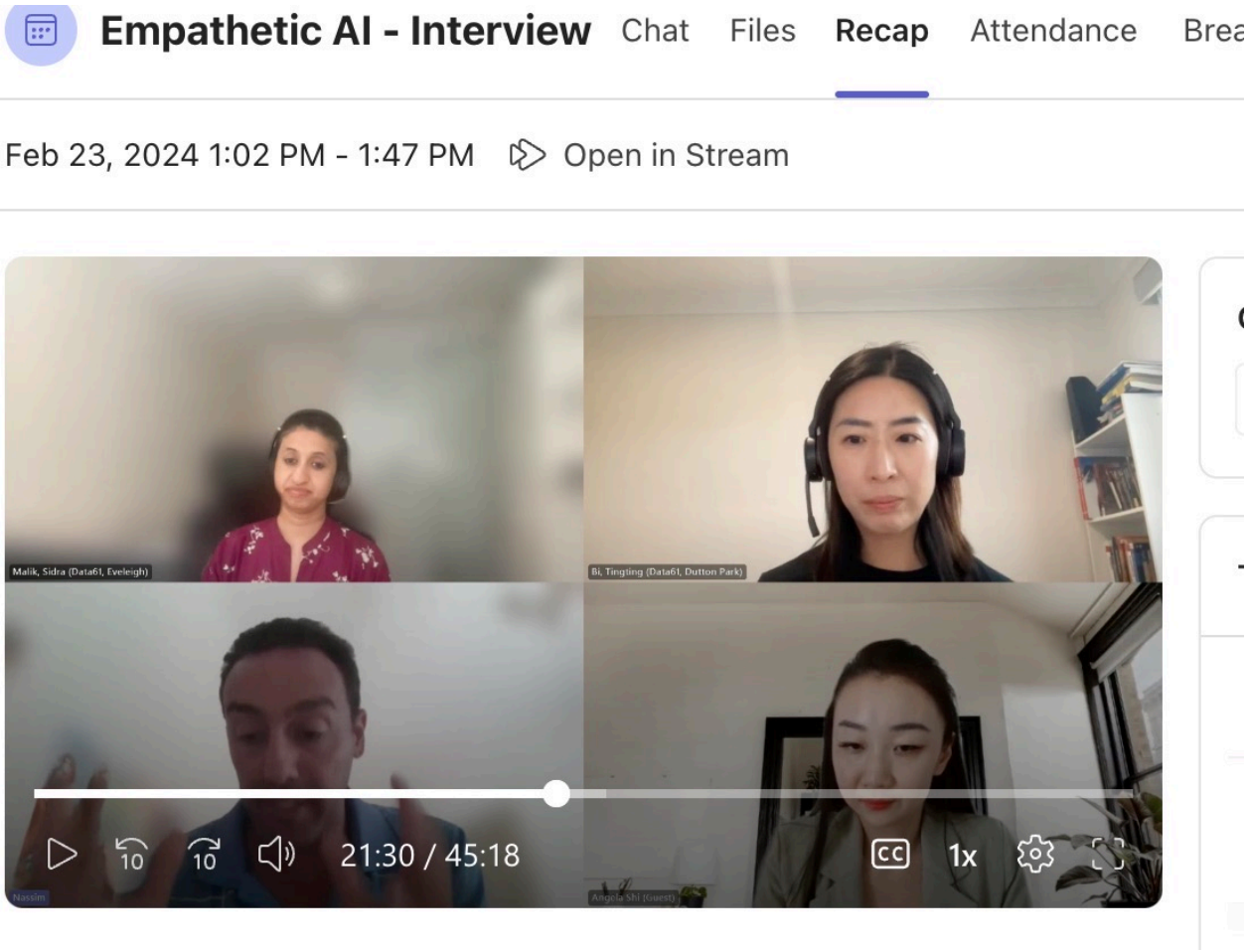
Interviews

- An **interview** is a systematic attempt to collect information from a person.
- Interviewing success depends on ability to identify:
  - work flows,
  - factors that influence the operations of systems, and
  - the elements (documents, procedures, policies, etc.) that make up systems.

5 Steps of the Interview Process

1. Preparing for the interview
2. Planning and scheduling the interview
3. Opening and closing the interview
4. Conducting the interview
5. Following up for clarification

Interviews


Empathetic AI - Interview
Chat
Files
Recap
Attendance
Brea
Feb 23, 2024 1:02 PM - 1:47 PM
Open in Stream
Malik, Sidra (Data61, Eveleigh)
Bi, Tingting (Data61, Dutton Park)
21:30 / 45:18
1x

2. Facilitated meetings

Purpose: try to achieve a summative effect, whereby a group of people can bring more insight into their software requirements than by working individually.

Advantage:

- Brainstorm and refine ideas that may be difficult to bring to the surface using interviews.
- conflicting requirements surface early on in a way that lets the stakeholders recognize where these occur.
- may result in a richer and more consistent set of requirements than might otherwise be achievable.

Disadvantage:

- meetings need to be handled carefully (hence the need for a facilitator) to avoid poor group dynamics
- Meetings are time consuming (hence the need for a facilitator)

3. Prototypes

- Valuable tool for clarifying ambiguous requirements.
- Act in a similar way to scenarios by providing users with a context within which they can better understand what information they need to provide.
- Wide range of prototyping techniques—from paper mockups of screen designs to beta-test versions of software products
- Protypes can also be used requirements validation (see later)
- Low fidelity prototypes are often preferred to avoid the stakeholder "anchoring" on minor, incidental characteristics that could limit design flexibility

- Disadvantage: Choose implementation too early
- Risk: Rough prototype becomes the product

Scenarios and Use Cases

Scenarios and use cases are commonly used in planned methodologies, especially in the object-oriented UML setting

Scenarios provide a valuable means for providing context to the elicitation of user requirements.

They allow the software engineer to provide a framework for questions about user tasks by permitting “what if” and “how is this done” questions to be asked.

4. Observation

- Analyst immerses herself in the working environment where the system will be used
- They observe the day-to-day work and notes the actual tasks in which participants are involved
- This helps discover implicit system requirements that reflect the ***actual*** rather than ***formal*** processes in which people are involved
- Advantage: discovers many user tasks and business processes that are too subtle and complex for their actors to describe easily.
- Disadvantage: Expensive (analyst works in client environment)
- Disadvantage: Observer should be detached: end-user based, non-judgemental so not appropriate for discovering organisational or domain requirements

Pulling this all Together
Some Guidelines

A Generic Requirements Process

ELICITATION

- ***Identify*** relevant sources of requirements
- ***Ask*** appropriate questions to gain an understanding of their needs

AND THEN

- ***Analyse*** the gathered information, looking for implications, inconsistencies or unresolved issues
- ***Confirm*** your understanding of the requirements with the users (validate)
- ***Synthesize*** appropriate statements of the requirements (specify)

Systems Thinking

1. What objectives are we trying to achieve?
2. What decisions do we control which affect those objectives?
3. What items dictate constraints on our range of choices?
4. What criteria should we use to evaluate candidate solutions?
5. What decision provides with the most satisfactory outcome with respect to those criteria?

Sommerville & Sawyer Elicitation Guidelines

1. Assess system **feasibility**
2. Be sensitive to **organisational** and **political** considerations
3. Identify and consult system **stakeholders**
4. **Record** Requirements Sources
5. **Define** the system's operating environment
6. Use **business concerns** to drive requirements elicitation

Elicitation Guidelines (cont)

7. Look for **domain constraints**
8. Record requirements **rationale**
9. Collect requirements from **multiple viewpoints**
10. **Prototype** poorly understood requirements
11. Use **scenarios** to elicit requirements
12. Define **operational** processes
13. **Reuse** requirements

Lecture Summary

- What is Requirements Elicitation?
  - Process; Essence; Communication; Scope
- Requirements Sources

- Elicitation Techniques
  - Interviews; Facilitated meetings; Prototypes;
  - Scenarios; User Stories see next week

- When eliciting requirements, ensure you have the correct people in the room…often you won't and the result is poor/invalid requirements

Recommended reading

- SWEBOK 3.0 Chapter 1 Section 3
- R. S. Pressman, *Software Engineering: A Practitioner's Approach,* 9th ed., 2020
  - Chapter 5 Understanding Requirements
- B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering – Using UML, Patterns, and Java,* 3rd ed., Prentice Hall, 2010
  - Section 4.3.1 Software Specification
  - Section 7.2 Requirements Elicitation and Analysis