

# Requirement Elicitation

## Chapter 4.4-4.7



© Scott Adams, Inc./Dist. by UFS, Inc.

# Topics

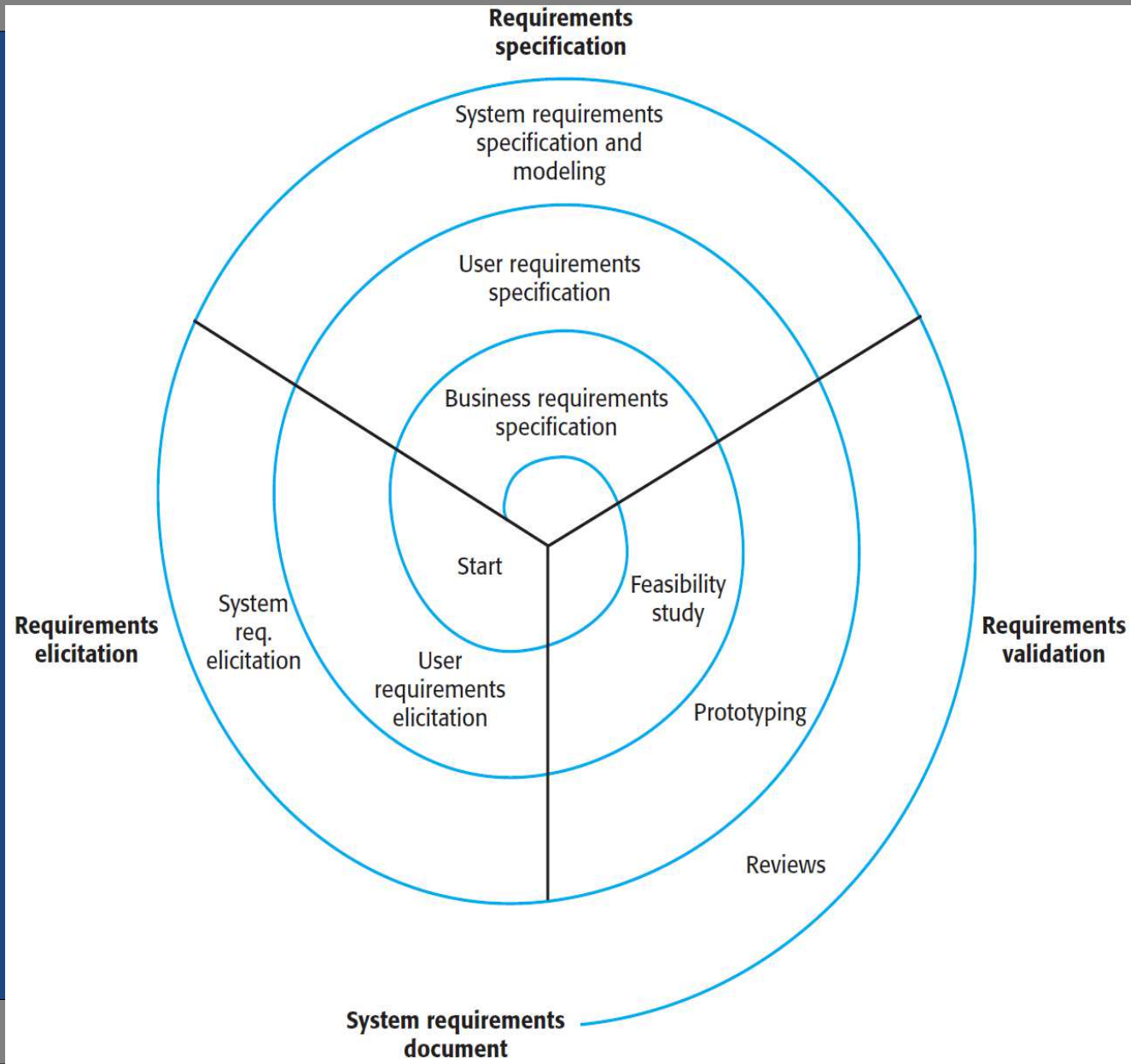
- 1) What is the **requirements engineering processes**?
- 2) How do we **elicit** and **analyze** requirements?
- 3) How do **use cases** record requirements?
- 4) How do we **manage changes** to requirements?

# Requirements Engineering (RE) Process

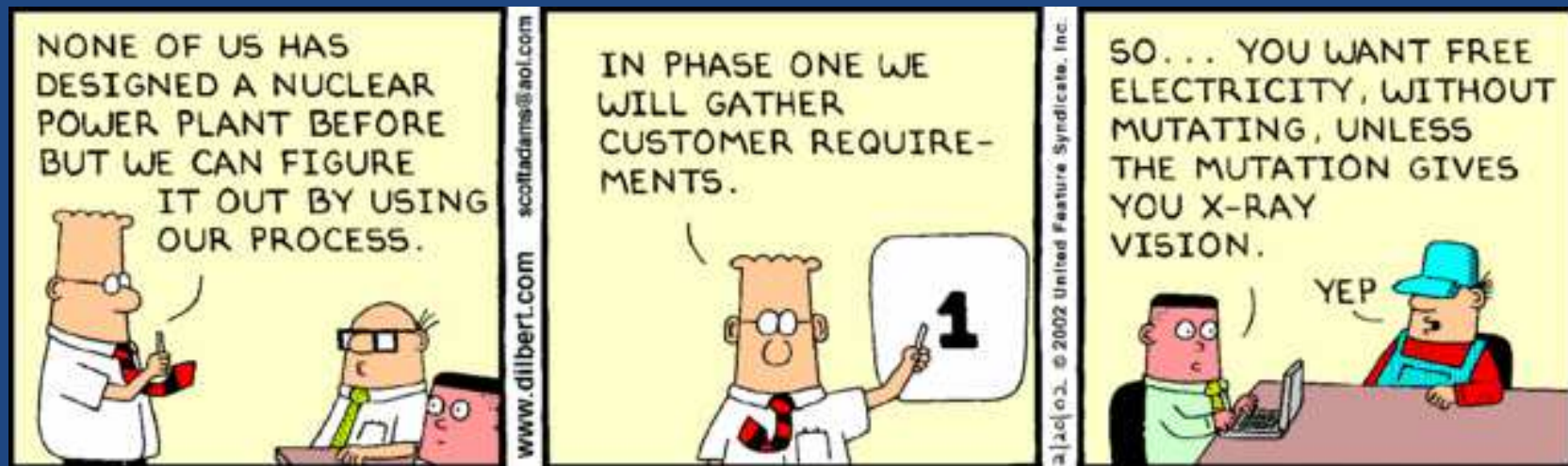
- RE processes vary widely depending on:
  - ..
  - people and organization
- Generic activities common to all RE processes:
  - Requirements elicitation
  - Requirements analysis
  - Requirements validation
  - Requirements management

# Spiral view of RE process

In practice,  
RE is an iterative  
activity in which...



# Requirements elicitation and analysis



# Requirements elicitation and analysis

- Software developers work with..
  - to find out about:
    - application domain;
    - services that the system should provide;
    - required system performance;
    - hardware constraints;
- Requirements Discovery:
  - Gathering information about the system and..

# Problems of requirements elicitation

- Stakeholders..
- Stakeholders express requirements in their own terms.
- The requirements change during the analysis process.
- Different stakeholders may have..
- How can you get the information from the customer?

# RD: Interviewing

- Stakeholder interviews common in RE process.
- **Types of interview**
  - based on predetermined list of questions
  - explore various issues with stakeholders.
  - Both are often used together.
- **Effective interviewing**
  - listen & learn customer's needs.
  - Get discussions going using some questions, or working together on a prototype system.



# Exercise: Course Reg Survey

- Consider this questionnaire for SFU students, generated by Acme Coding Inc related to course registration:
  - 1) Would you like to be able to configure the registration system to automatically enroll you in into a set of courses at your registration appointment?
  - 2) If your selected classes are full, would you like it to automatically enroll you in another class?
  - 3) Should the auto-enroller allow you to enroll in two classes which have conflicting schedules?
- What's good vs bad? What does the survey miss?

# RD: Interviews in practice

- Interviews good at..
- Interviews poor at understanding domain requirements:
  - Developer's don't understand domain terminology;
  - Some domain knowledge is so familiar that people find it hard to articulate or...
- You have to be tenacious about working to truly understand system.

# Implicit

- **Implicitness problem**

- Domain specialists understand the area so well that they do not think of..

- **Examples**

- Replace video card in computer:

- ..

- Source current from an electric vehicle's high-power battery:

- ..

- Test nuclear power plant:

- ..

# Ethnography

- People are generally not very good at...
- **Ethnography:**
  - Analyst **immerses** themselves **in work environment** where system will be used.
  - Analyst observes..  
people don't explain it to them.

# Ethnography

- **Benefits of Ethnography**
  - Good for documenting what people really do, and finding..
- **Limits of Ethnography**
  - Poor at..

# Recording Requirements

# User Stories

- Scrum User Stories..

- Use template:

(user role)

(what)

(why?)

- Example

As a TA, I want to download all student submissions as a ZIP file so that I don't have to individually download each student's work.

- User stories

..

not how the software lets them do it.

# Epic stories

- Epic

..

- Epics are coarse-grained, very high level
- Team breaks down epic into smaller, more detailed and specific, stories

- Example

- *As a student, I want to submit my assignment so that I can get credit for my work.*
- Break down into smaller use cases addressing:
  - Submitting parts of my assignment.
  - See history
  - Resubmit, etc.



# Exercise: User Stories

- Write an **epic** related to course registration:
- Break it down into 2+ **user stories**:

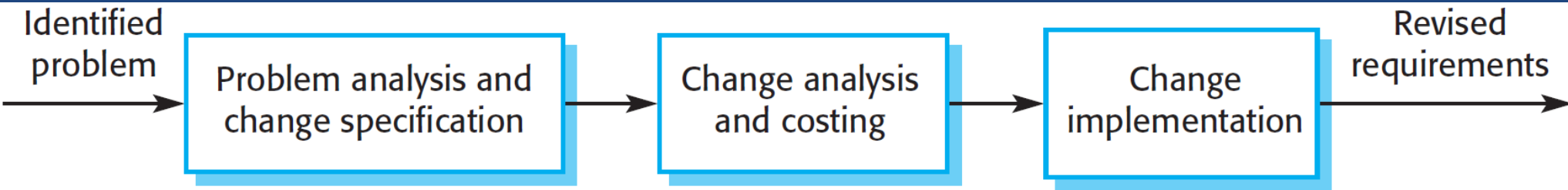
# Requirements management



# Requirements management

- Requirements management:
  - during the requirements engineering process and system development.
- Reasons for changing requirements:
  - Business and technical environment of the system always changes after installation.
  - Adding new hardware and systems.
  - New legislation and regulations apply to the system.

# Requirements doc. change management



Requestor can help **resolve** any **conflicts**: change or remove the request.

Make decision to **accept** or **reject** change request based on analysis.

**Modify** req. document, system design, and implementation.

Organize req doc so changes easy to implement.

# Changing requirements in Agile

- ..  
so it's simpler to record requested change.
- **Example process for recording change in Scrum**
  - Discuss with PO (or as a team)
  - Create user story
  - Customer assigns priority in backlog
  - Team estimates its size
  - Team selects it for an iteration.

# Summary

- Requirements engineering – a spiral or iterative process:
  - Requirements elicitation and analysis: iterative process.
  - Requirements Discovery: Using interviews, use cases, ethnography
  - Requirements management – process of managing and controlling changing system requirements.