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?

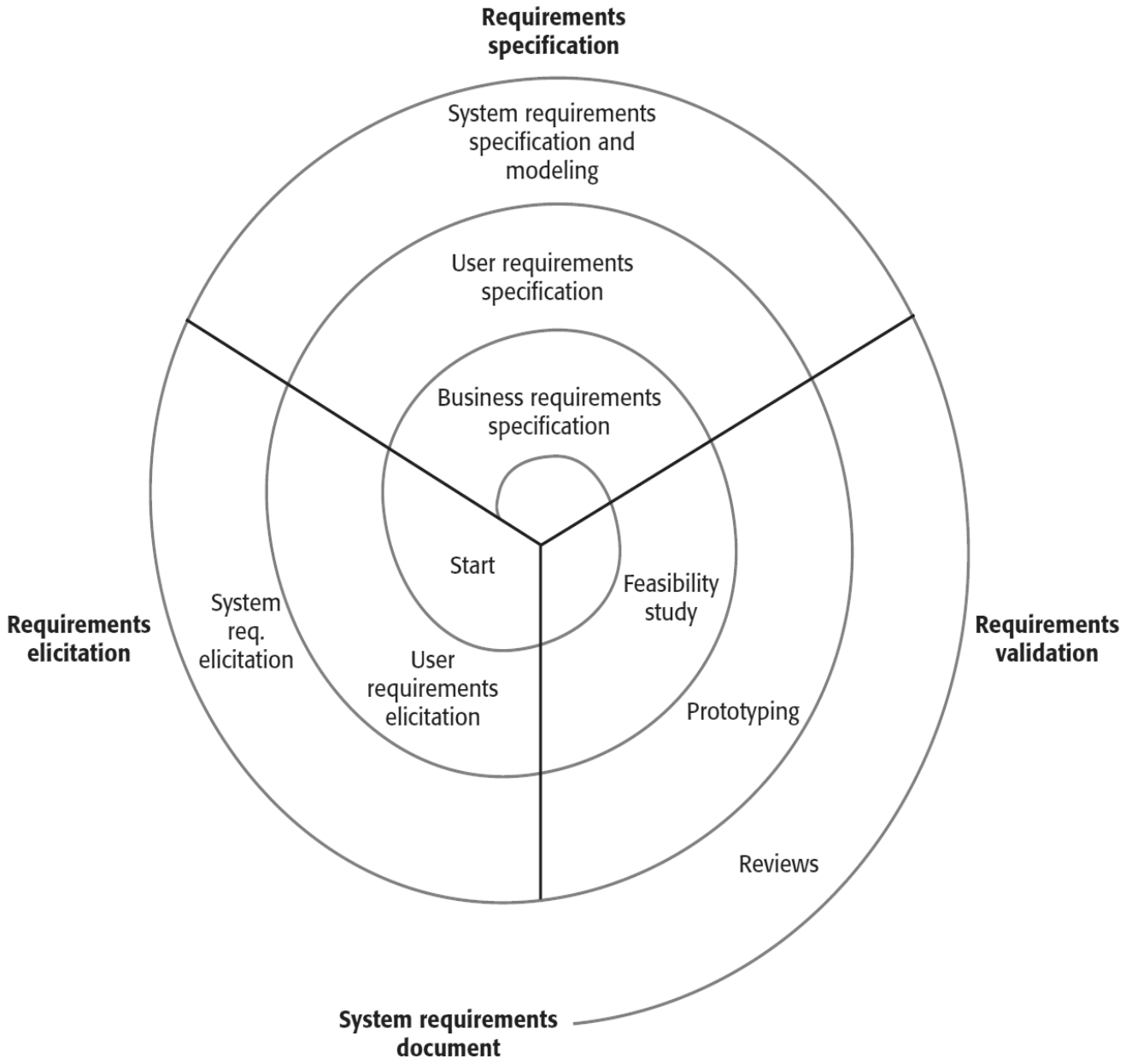
Requirements Engineering (RE) Process

- RE processes vary widely depending on:
  - .. **application domain**
  - people and organization
- Generic activities common to all RE processes:
  - Requirements elicitation **-find em**
  - Requirements analysis **-understand em**
  - Requirements validation **-verify em**
  - Requirements management **-control em**

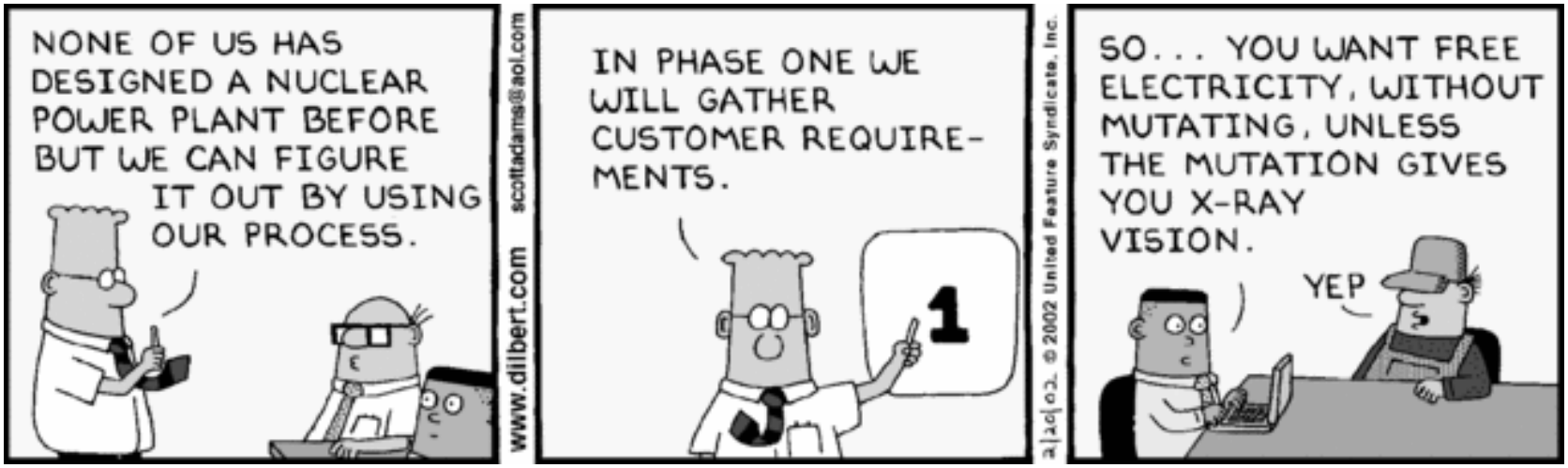
Spiral view of RE process

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

these processes are interleaved



Requirements elicitation and analysis



Requirements elicitation and analysis

(find, construct and understand requirements of the system)

- Software developers work with..

  range of stakeholders to find out about:

  - application domain;
  - services that the system should provide;
  - required system performance; **(non-functional requirements)**
  - hardware constraints; **(e.g nintendo switch)**

- Requirements Discovery:
  - Gathering information about the system and...

    extracting user and system requirements

Problems of requirements elicitation

- Stakeholders... **don't know what they really want**
- Stakeholders express requirements in their own terms.
- The requirements change during the analysis process.
- Different stakeholders may have... **conflicting requirements**
- How can you get the information from the customer?

RD: Interviewing

- Stakeholder interviews common in RE process.
- Types of interview
  - **closed interviews** **(adv: answer to specific questions, large populations** based on predetermined list of questions
  - **open-ended interviews** **(adv: never considered idea of stakeholder)** explore various issues with stakeholders.
  - Both are often used together.
- Effective interviewing
  - **Be open-minded:** 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?

misses waitlist prereqs

RD: Interviews in practice

- Interviews good at..

  getting an overall understanding of how users might use the system

- 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...

    think it isn't worth mentioning
- 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.. **making the domain requirements explicit**
  - Examples
    - Change oil in car:

      .. **car must be off**

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

      .. **use a pre-charge register**

    - Test nuclear power plant:

      .. **your guess is as good as mine**

Ethnography

- People are generally not very good at...

  describing exactly what they do.

- Ethnography:
  - Analyst immerses him/herself in work environment where system will be used.
  - Analyst observes current workflow; people don't explain it to him/her.
- Good/Bad:
  - Good for documenting what people really do, and finding.. **requirements which users forgot to mention**
  - Bad at.. **finding new features beyond current practice**

Recording Requirements

User Stories

- Scrum User Stories.. **capture product requirements**
  - Use template:

    As_____ , I want _____ so that _____

    (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

  .. **keep the focus on what the user wants to do**

  not how the software lets them do it.

Epic stories

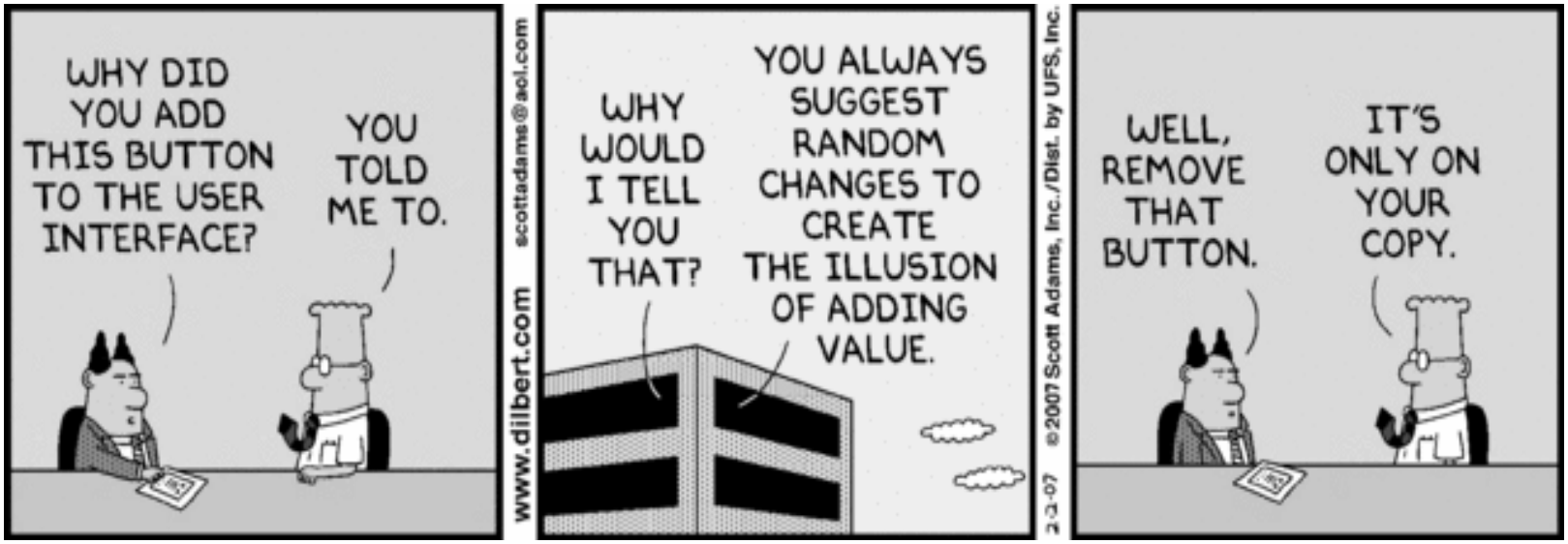
- Epic
  - **A story that is too big for a single iteration**
    - Epics are coarse-grained, very high level
    - Team breaks down epic into smaller, more detailed and specific, stories
- Example
  - As a *student*, I want *to submit my assignment* so that *I can get credit for my work.*
  - Break 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

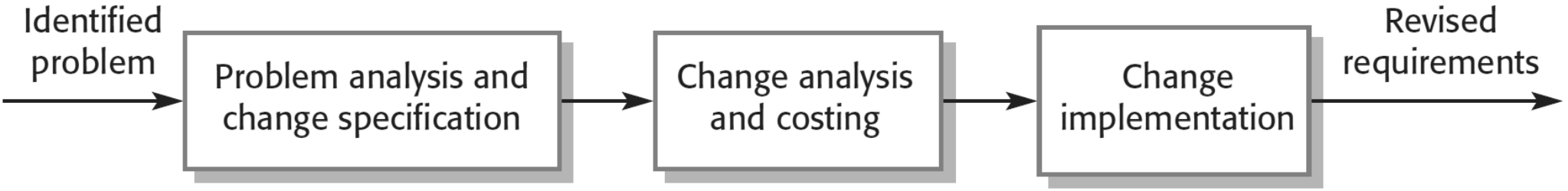


Requirements management

- Requirements management:
  - **process of managing changing requirements**

    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



Analyze change proposal for validity

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

Assess effect of proposed change

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

scrum master can make this decision

Modify req. document, system design, and implementation.

Organize req doc so changes easy to implement.

Changing requirements in Agile

- .. **Scrum has no formal requirements document**
  so it's simpler to record requested change.
- Example process for recording change in Scum
  - 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.