CMPT 276 Class 11: Requirements Elicitation

Dr. Jack Thomas

Simon Fraser University

Fall 2020

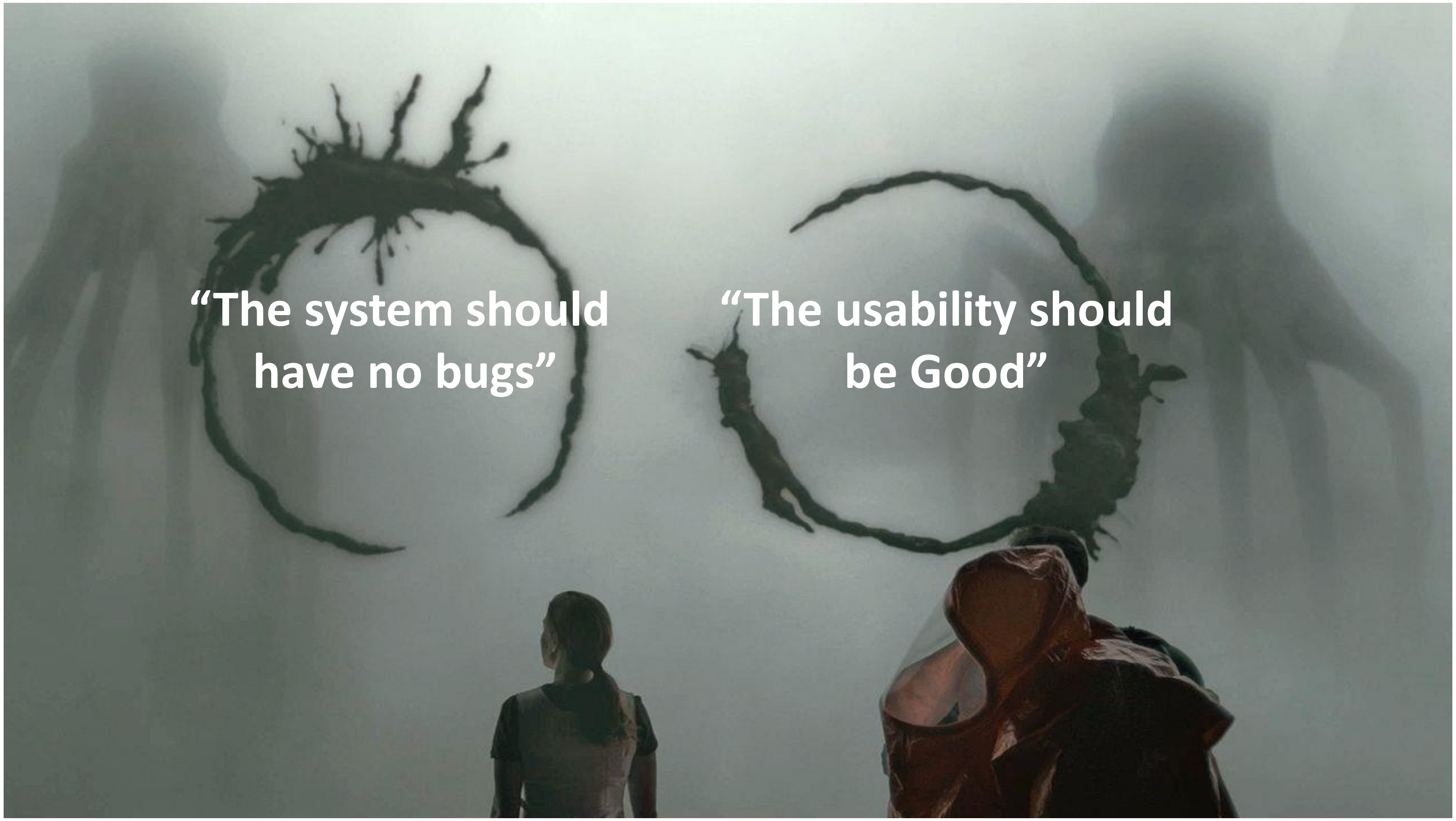

"The system should have no bugs"
"The usability should be Good"

Image credit: https://25yearslatersite.com/2019/08/08/arrival-the-tale-of-the-forgotten-best-picture-nomination/

Today’s Topics

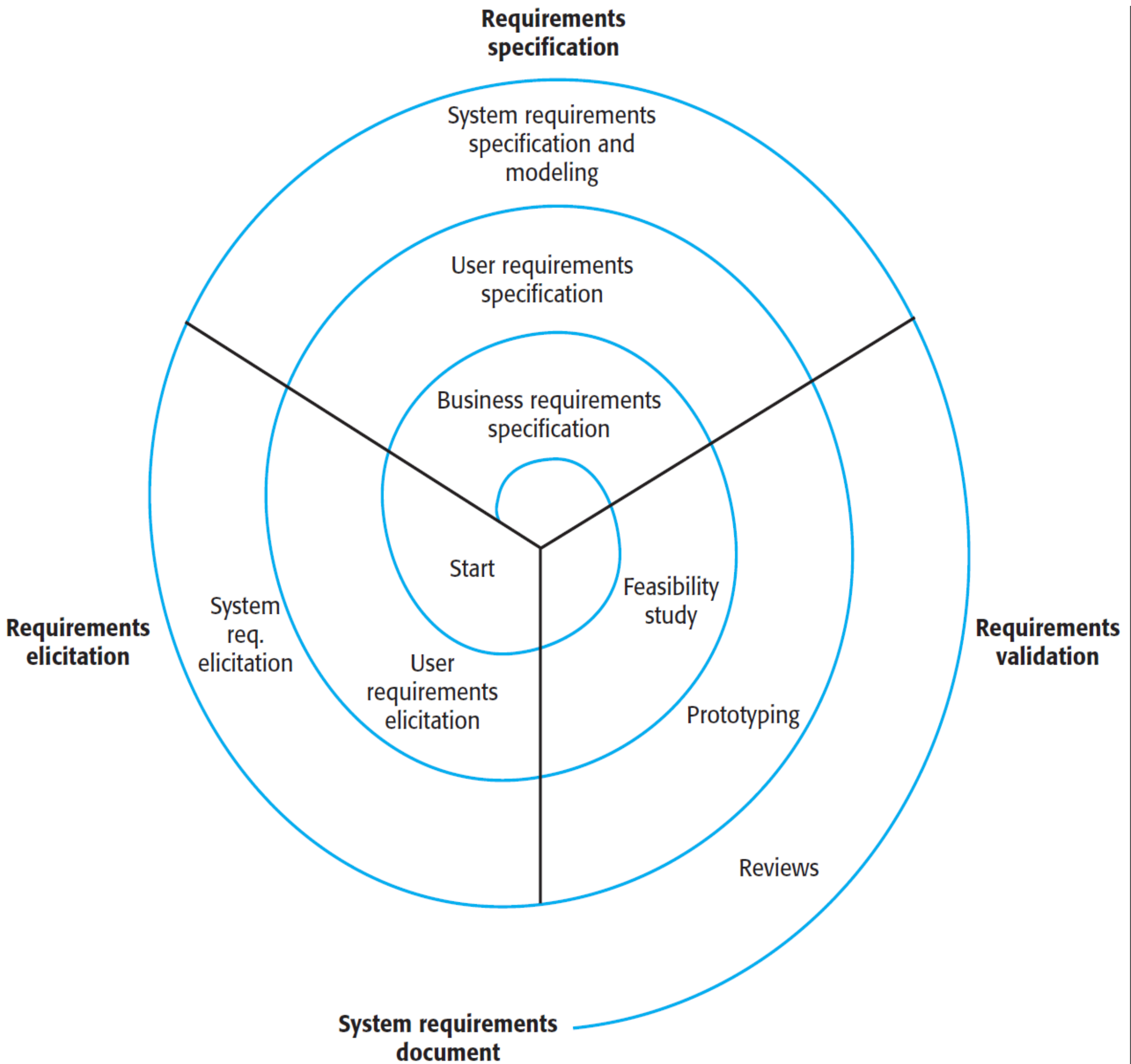
1. What is the **requirements engineering process**?
2. How do we **elicit and analyze** requirements?
3. How do **use cases** record requirements?
4. How do we **manage changes** to requirements?

Requirements Engineering (RE) Process

- RE processes vary widely depending on:
  1. Application domain.
  2. People and organizations.
- Generic activities common to all RE processes:
  - **Requirements Elicitation** (finding)
  - **Requirements Analysis** (understanding)
  - **Requirements Validation** (verifying)
  - **Requirements Management** (controlling)

Spiral View of the RE Process

- In practice, RE is an iterative activity in which these process are interleaved.


Requirements specification
System requirements specification and modeling
User requirements specification
Business requirements specification
Start
Feasibility study
Requirements elicitation
System req. elicitation
User requirements elicitation
Prototyping
Requirements validation
Reviews
System requirements document

Requirements Elicitation and Analysis

- Software developers work with a range of **stakeholders** to find out about:
  - The application **domain**;
  - The **services** that the system should provide;
  - required **system performance**;
  - **hardware constraints**;

- **Requirements Discovery**:
  - Gathering information about the system and extracting user and system requirements.

Problems of Requirements Elicitation

- Stakeholders **don’t necessarily know what they 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 this information from the customer?*

Interviewing

- Stakeholder interviews are common in RE processes.
- Types of interview:
  - **Closed Interviews**: Based on a predetermined list of questions
  - **Open-ended Interviews**: 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 Registration 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?

Interviews in Practice

- Interviews are good at getting an **overall understanding** of how users might use the system.
- Interviews are poor at **understanding domain requirements**:
  - Developers don't understand **domain terminology**.
  - Some domain knowledge is so familiar that people find it **hard to articulate** or it isn’t worth mentioning.
- You have to be tenacious about working to truly understand the topic.

The Problem of Implicit Information

- Domain specialists understand the area so well that they **do not think of making the domain requirements explicit**.
- **Examples**
  - *To change the oil in a car*: Car must be off.
  - *Source current from an electric vehicle's high-power battery*: Use a pre-charge resister.
  - *Test a nuclear power plant*: ???

Ethnography

- People are generally not very good at describing exactly what they do.
- **Ethnography**:
  - Analyst **immerses themselves** in the work environment where the system will be used.
  - Analyst **observes the current workflow**, people don't explain it to them.
- **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.

User Stories

- Scrum User Stories capture product requirements. Use the 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

- A story that's too big for one iteration.
  - Epics are **coarse-grained**, very high level
  - The team breaks down epics 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.
  - Seeing the submission history
  - Resubmitting, etc.

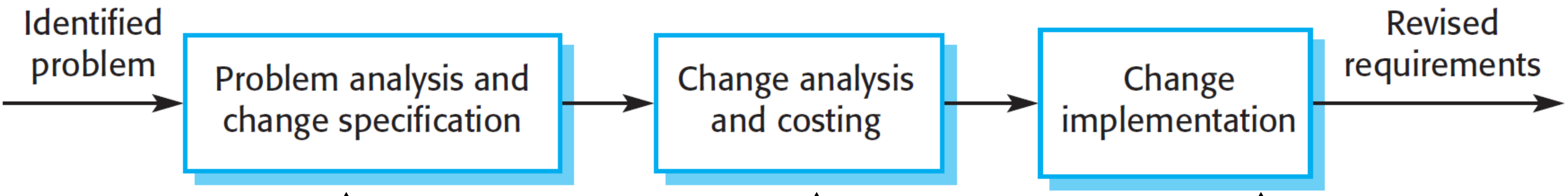
Class Exercise: User Stories

- Let’s write an epic related to course registration, then break it down.

Requirements Management

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

Requirements Document Change Management


Identified problem
Problem analysis and change specification
Change analysis and costing
Change implementation
Revised requirements

Analyze change proposal for validity: The requester can help resolve any conflicts, make changes, or remove the request.

Assess effect of proposed change: Make a decision to reject or accept the change request based on analysis.

Modify Req. Doc., design, and implementation: Organize Req. Doc. so changes are easy to implement.

Changing Requirements in Agile

- **Scrum has no formal requirements document**, so it’s simpler to record requested changes.
- 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.

Recap – Eliciting a Summary

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