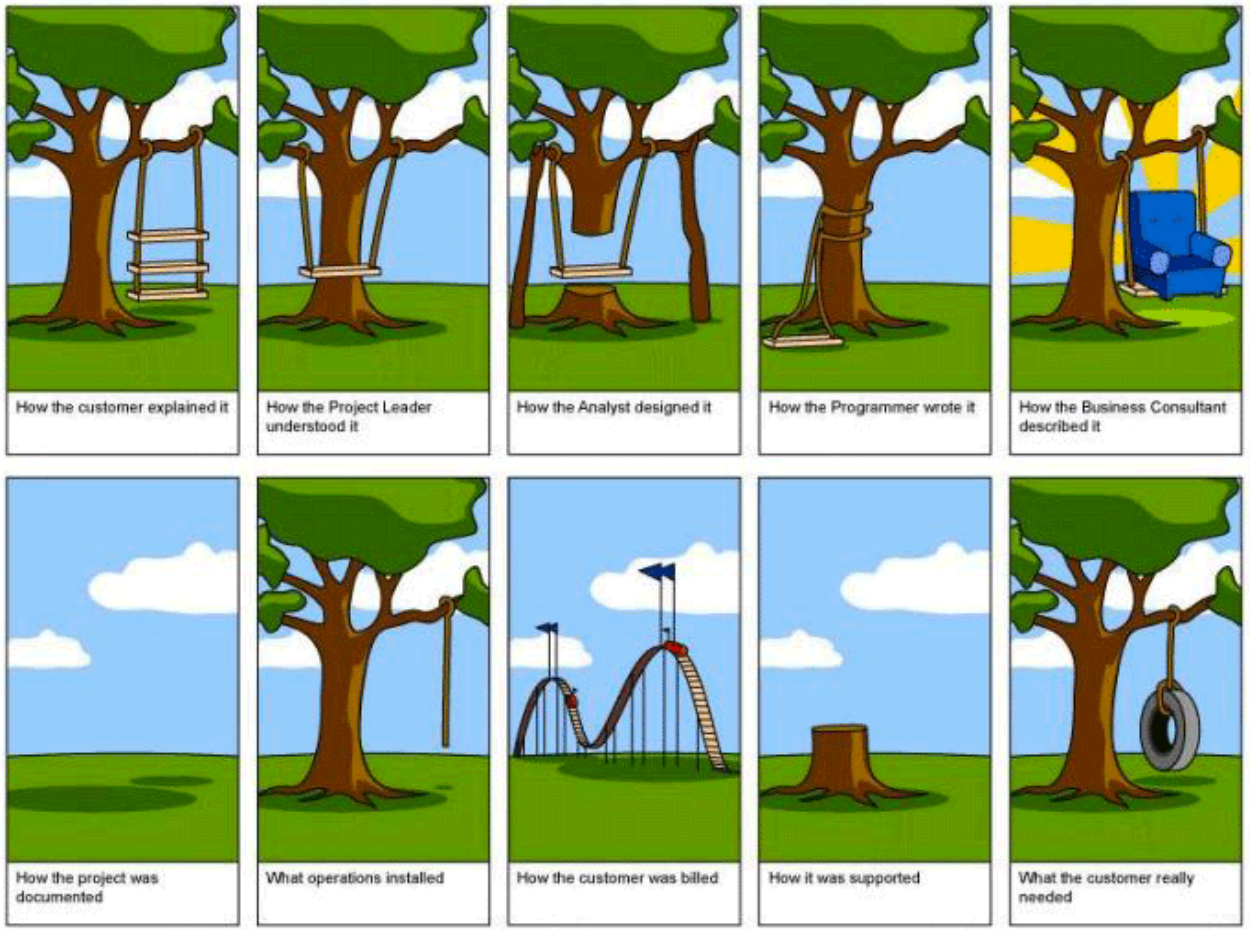
Requirements

Requirements

- All about communication



Requirements

- Requirements:
  - A set of statements that describe the user's (stakeholders) needs and desires
- Requirements Specification
  - A set of requirements to be met by the system
  - Note: there are other specification types
    - eg Design Specification

Requirement

- Short and concise piece of information
- Says something about the system
- Stakeholders agree on it
- It helps solve the customer’s problem

Requirements Engineering

- the process of developing a complete requirements specification for a project
- Involves:
  - Elicitation
  - Analysis and prioritization
  - Definition
  - Prototyping
  - Review
  - Agreement

Domain Analysis

- Domain
  - the general field of business or technology in which the software will be used
- Domain expert
  - person who has a deep knowledge of the domain

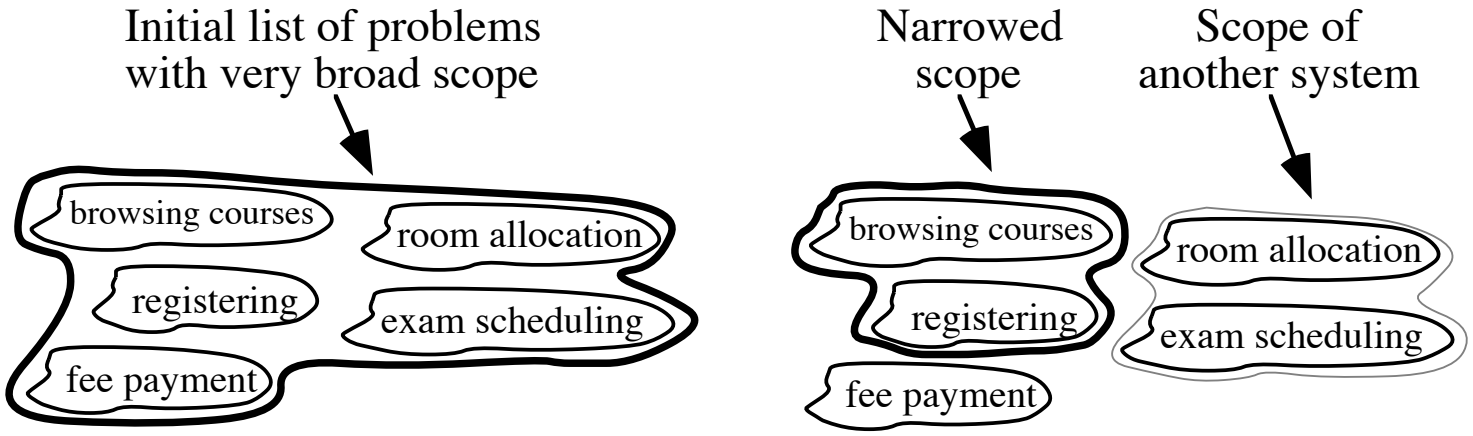
- Benefits to requirements engineering
  - Faster development
  - Improved communication between stakeholders
  - Better system
  - Anticipation of extensions

Defining the Scope

- Scope: the extent of the area or subject matter the project deals with
- Narrow the scope by defining a more precise problem
  - List all the things you might imagine the system doing
    - Exclude some of these things if too broad
    - Determine high-level goals if too narrow

Defining the Scope

- Example: A university registration system



Viewpoints

- May consider different viewpoints
  - Stakeholders
  - Different user types

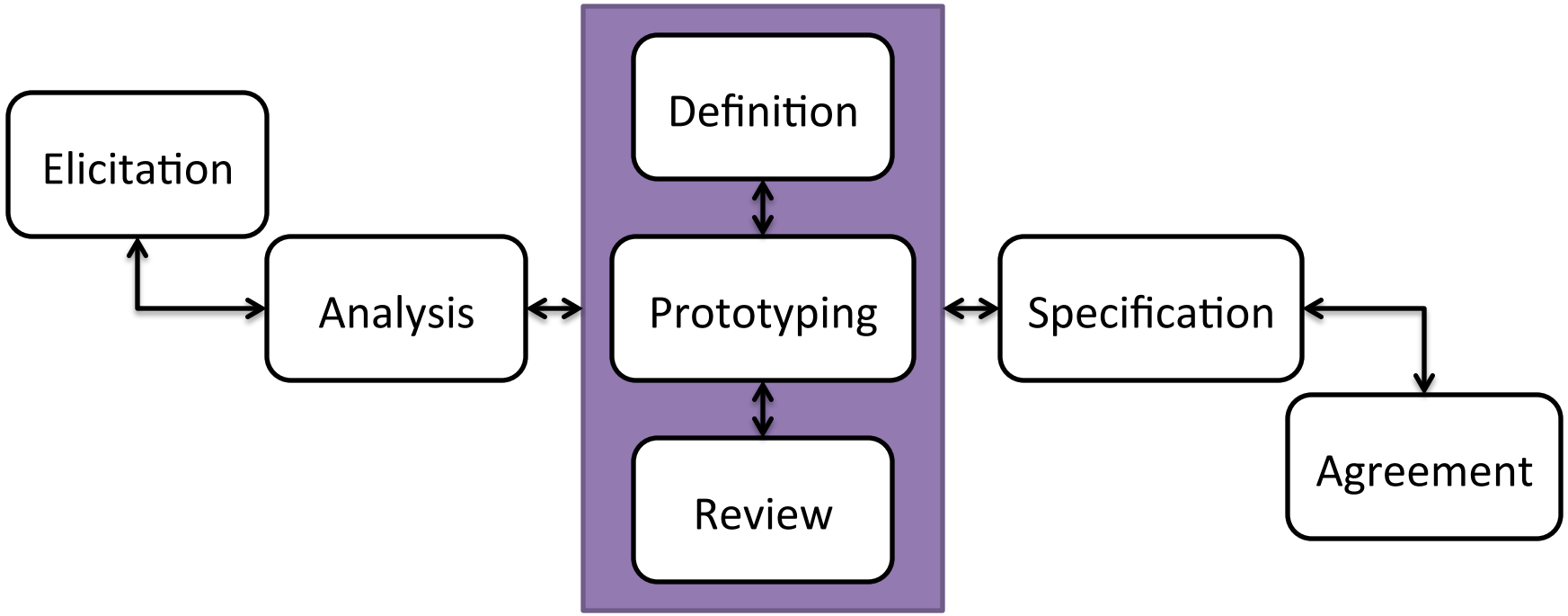
e.g. Accessibility, child-friendliness, investor concerns, etc.

Analysis

- Requirements elicited are checked for detail, accuracy and conflict
- Cluster requirements into groups
- Prioritization
  - Usually intend to maximize business value given project constraints
  - Methods:
    - Numerical/Group Assignment (low->high)
    - Pair-wise
    - Cost-value
    - Voting ($100 test)
    - Ranking

Requirements

- Refined through definition, (optional) prototyping and review
  - Prototyping can be particularly useful for GUIs



Requirements

- Timing of requirements specification will vary based on process model
  - Waterfall will generate spec documents early
  - Agile may continually perform RE activities
- RE can be a separate business
  - Define the problem to put out for development contractors
  - Request for Proposal

Requirement Types

- Functional requirements
  - Specific behaviours
  - Describe what the system should “do”

- Non-functional requirements
  - Constraints that must be adhered to during development
  - Describe how the system should “be”

Functional requirements

- Users' (and stakeholder) functions required
- Data, formats and information needs
- System interfaces or services used
- Specific interface requirements

Non-functional requirements

Three main types

- 1. Quality
  - Evolutionary Quality
    - Reliability, maintainability, testability and reusability
  - Execution Quality
    - Security, response time, throughput, usability
- 2. Platform
  - the environment, platform(s) and technologies of the system.
- 3. Process
  - the project plan and development methods
    - Process model (methodology) to be used
    - (Cost and delivery date)
      - Often put in contract or project plan instead

Specification complexity

- Depends on
  - Size and complexity of project
  - Future release plans
  - Expected customer support, QA and maintenance
  - Lack of domain knowledge of developers

Difficulties and Risks

- Lack of understanding of the domain or the real problem
  - Do domain analysis and prototyping
- Requirements change rapidly
  - Perform incremental development, build flexibility into the design, do regular reviews
- Attempting to do too much
  - Document the problem boundaries at an early stage, carefully estimate the time
  - Avoid “Scope creep”
- It may be hard to reconcile conflicting sets of requirements
  - Brainstorming, JAD sessions, competing prototypes
- It is hard to state requirements precisely
  - Break requirements down into simple sentences and review them carefully, look for potential ambiguity, make early prototypes


THIS IS FINE.

Coming Up Next

First:
- Getting Started with Grails, GSP and React

Later:
- User Interface Design Principles
- React UI Components
- Java / Swing
- Software Desing Part 2
- git

To-Do

If you are planning on using Grails

- Please use MacOS or Linux
- Register for JetBrains Student License and download IntelliJ Ultimate.
- Download and Install Grails by following first few instructions
  - http://docs.grails.org/latest/guide/gettingStarted.html
- Create a default app and navigate to localhost:8080 – just to check that things are working.
- Lecture/Tutorial on Grails next time.