CASE WESTERN RESERVE
UNIVERSITY

Requirements Engineering

CSDS 393/493: Software Engineering

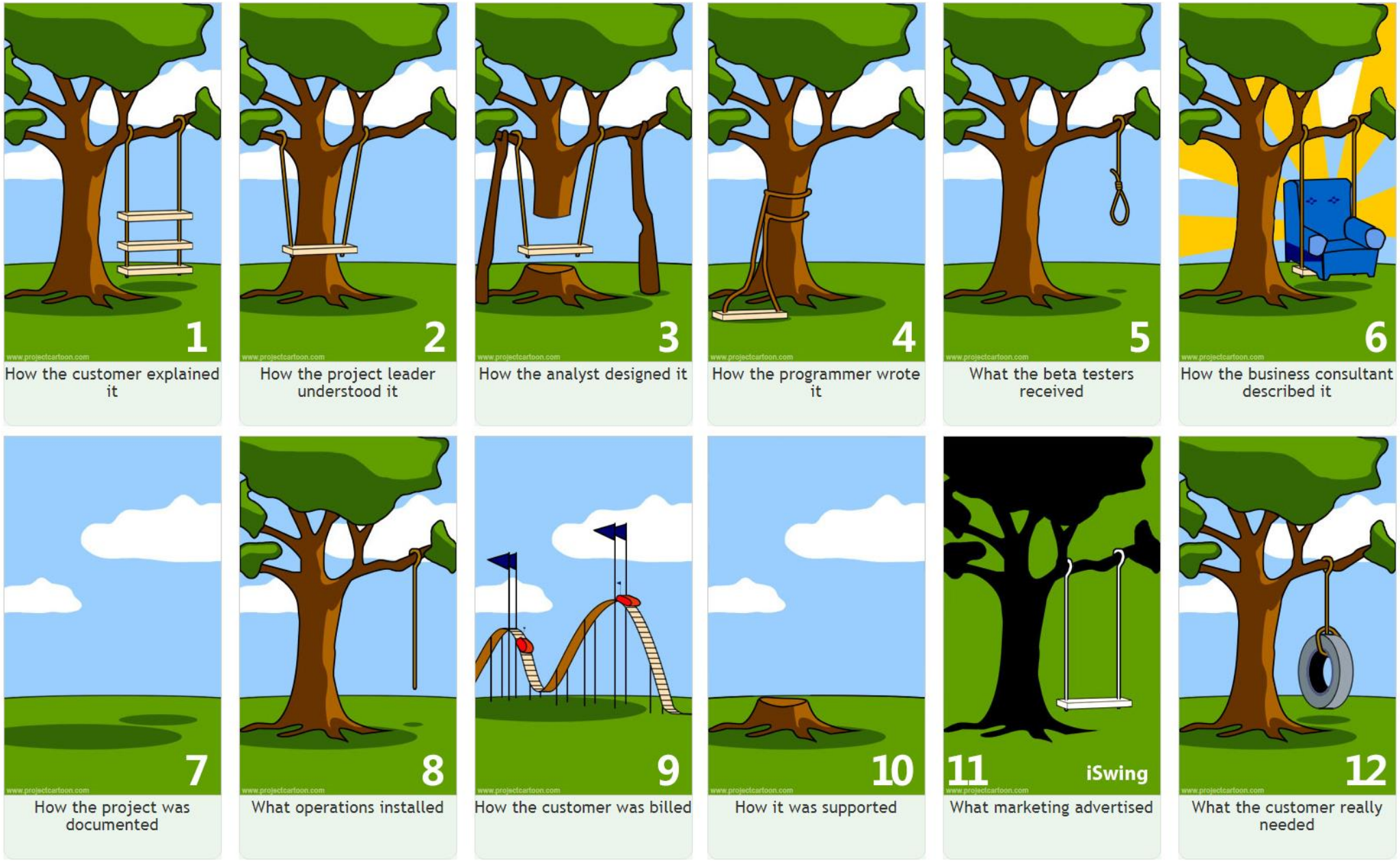
Spring 2025

Agenda

- Project discussion
- Software requirements
  - Functional and non-functional requirements
- Requirement development process
  - Elicitation
  - Specification
  - Validation

Software Projects

- Teams should meet regularly and setup team communications mediums
- Primary requirements for the project
  - Front-end
    - web interface, mobile app such, etc.
    - .met, android/ios, etc.
  - Back-end
    - web server, microservices, etc.
    - Java Spring Boot, ASP, PHP, etc.
  - Database
    - Mongo DB, MySQL, etc.
  - Features for multiple user groups
    - Features for e.g., admins, clients, etc.


1
How the customer explained it
2
How the project leader understood it
3
How the analyst designed it
4
How the programmer wrote it
5
What the beta testers received
6
How the business consultant described it
7
How the project was documented
8
What operations installed
9
How the customer was billed
10
How it was supported
11
iSwing
What marketing advertised
12
What the customer really needed

Software Requirements

- Properties a software product must exhibit to meet the needs of users
- Described in a software requirements specification (SRS) document
- Derived and specified collaboratively by "customer" and developers
- The process of developing requirements is called requirements engineering

Types of requirement

- **User requirements**
  - services the system provides and its operational constraints. Written for customers

- **System requirements**
  - system's functions, services and operational constraints. A contract between client and contractor

User and System Requirements

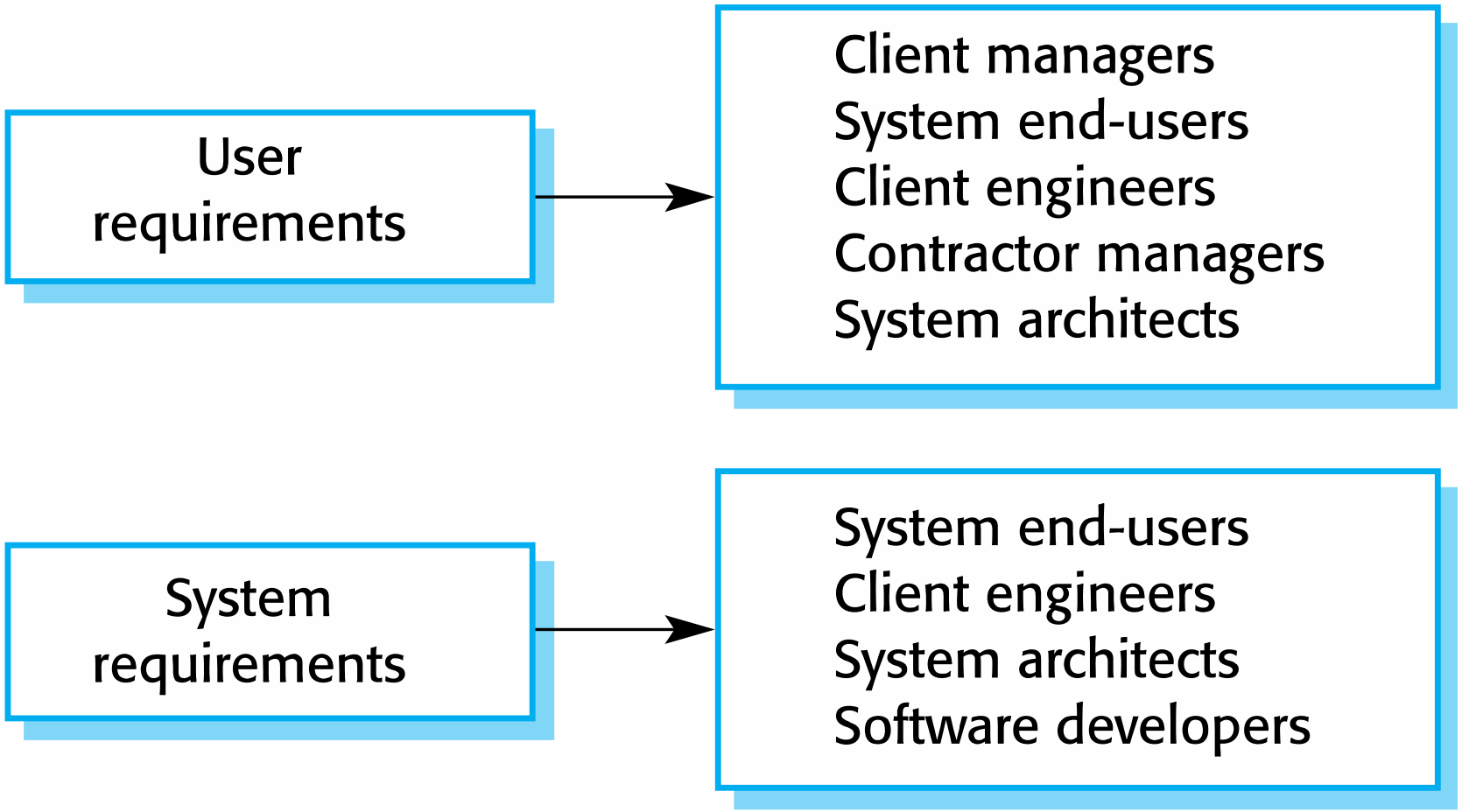
User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Readers of Requirements Specification



Functional and non-functional requirements

Functional and non-functional requirements

- **Functional requirements**
  - Services the system should provide, how the system should react to particular inputs and how the system should behave.
  - May state what the system should not do.
- **Non-functional requirements**
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

Functional Requirements

Describe the functionality that a product must provide to users

- **Written requirements** are often open to interpretation
- **Ambiguity and incompleteness must be resolved** prior to design

Example: A Functional Requirements for a Word Processor

- The user shall be able to select a region of text and either cut (delete) it from the document, in which case it is copied to the clipboard.
- The user shall be able to place the insertion point at any location in the document and insert (paste) the contents of the clipboard at that location.
- The contents of the clipboard shall be retained until the next cut or copy operation.

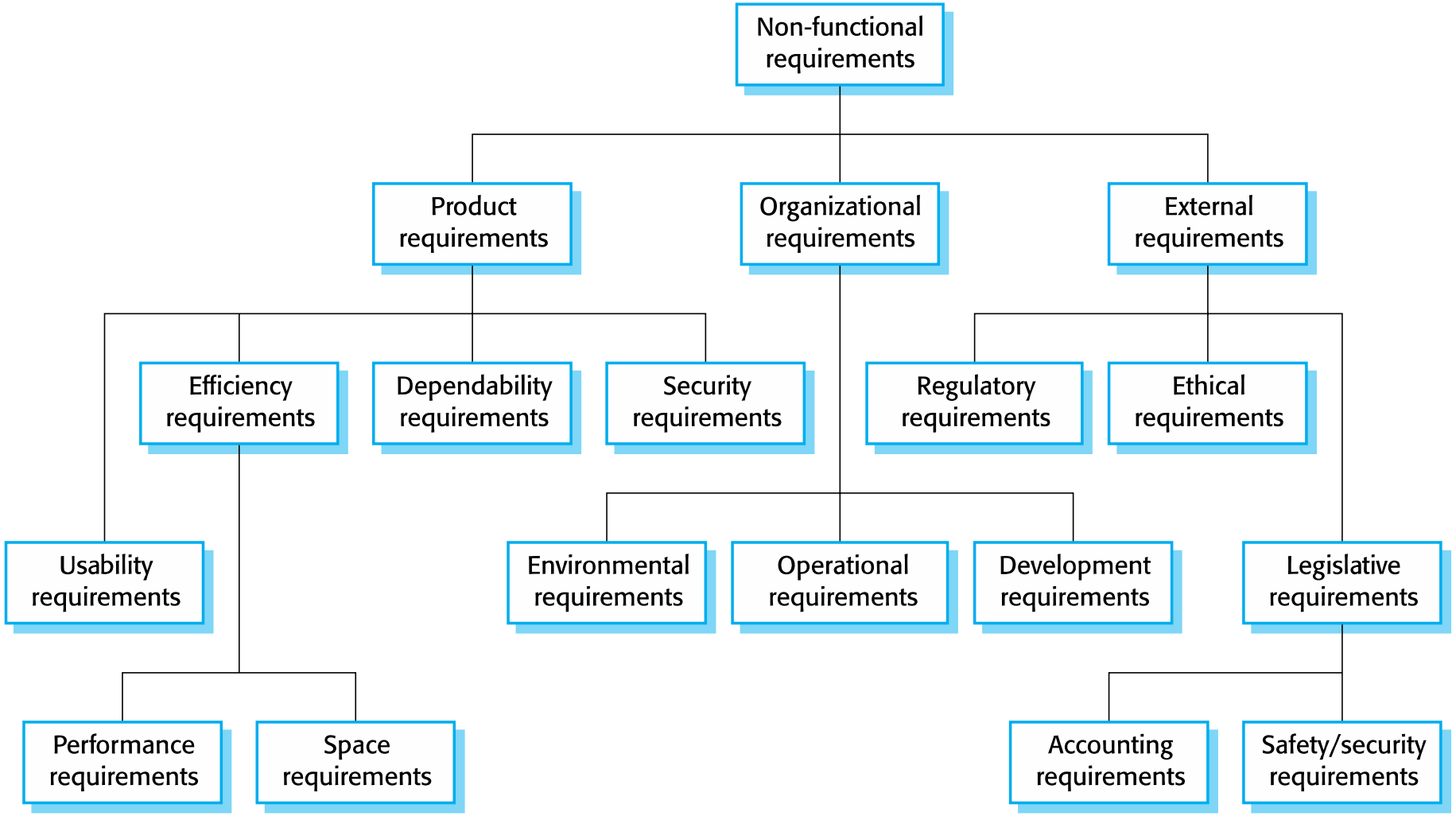
Example: Functional Requirements for a Computer Physician Order Entry System

- The CPOES shall permit a patient's physician to order the administration of any drug available from the hospital pharmacy and to specify the dosage, frequency, and duration.
- The CPOES shall provide support for checking prescribed drug dosages for safety and efficacy, based on the patient's diagnosis, age, sex, weight, and other relevant clinical factors.
- The CPOES shall provide support for drug interaction checking.

Non-Functional Requirements

- All other criteria including system properties and constraints
- Types of non-functional requirements
  - Usability
  - Reliability
  - Availability for use
  - Efficiency/performance
  - Security
  - Constraints on the hardware or software platform
  - Portability to different platforms
  - Compliance with regulations and standards
  - Certification for use

Non-Functional Requirements



Example: Non-Functional Requirements for a Word Processor

- After completing the product tutorial, 80% of trial users will successfully complete the benchmark editing tasks within 1 hour.
- Latency of editor response to keyboard or GUI inputs will not exceed 100 milliseconds.
- The contents of the edited document shall be corrupted in no more than 1 in 10,000 user editing sessions on average.

Example: Non-Functional Requirements for a CPOES

- The CPOES shall be unavailable for use no more than 10 minutes per week on average.
- The CPOES shall permit only authorized physicians and other authorized users with prescriptive privileges to enter orders.
- The CPOES shall satisfy the ONC HIT Certification Criteria*

* https://www.healthit.gov/topic/certification-ehrs/certification-health-it

Metrics for Specifying Nonfunctional Requirements

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Requirements vs. Design

- It is considered good practice to exclude software design and implementation details from an SRS
  - "Requirements should state what software is supposed to do, but not how it is supposed to do it."

- However, if requirements are too abstract they may be difficult to understand

Example: Requirements for a Music Store Web Application

- It's possible to state these very abstractly, permitting virtually any solution

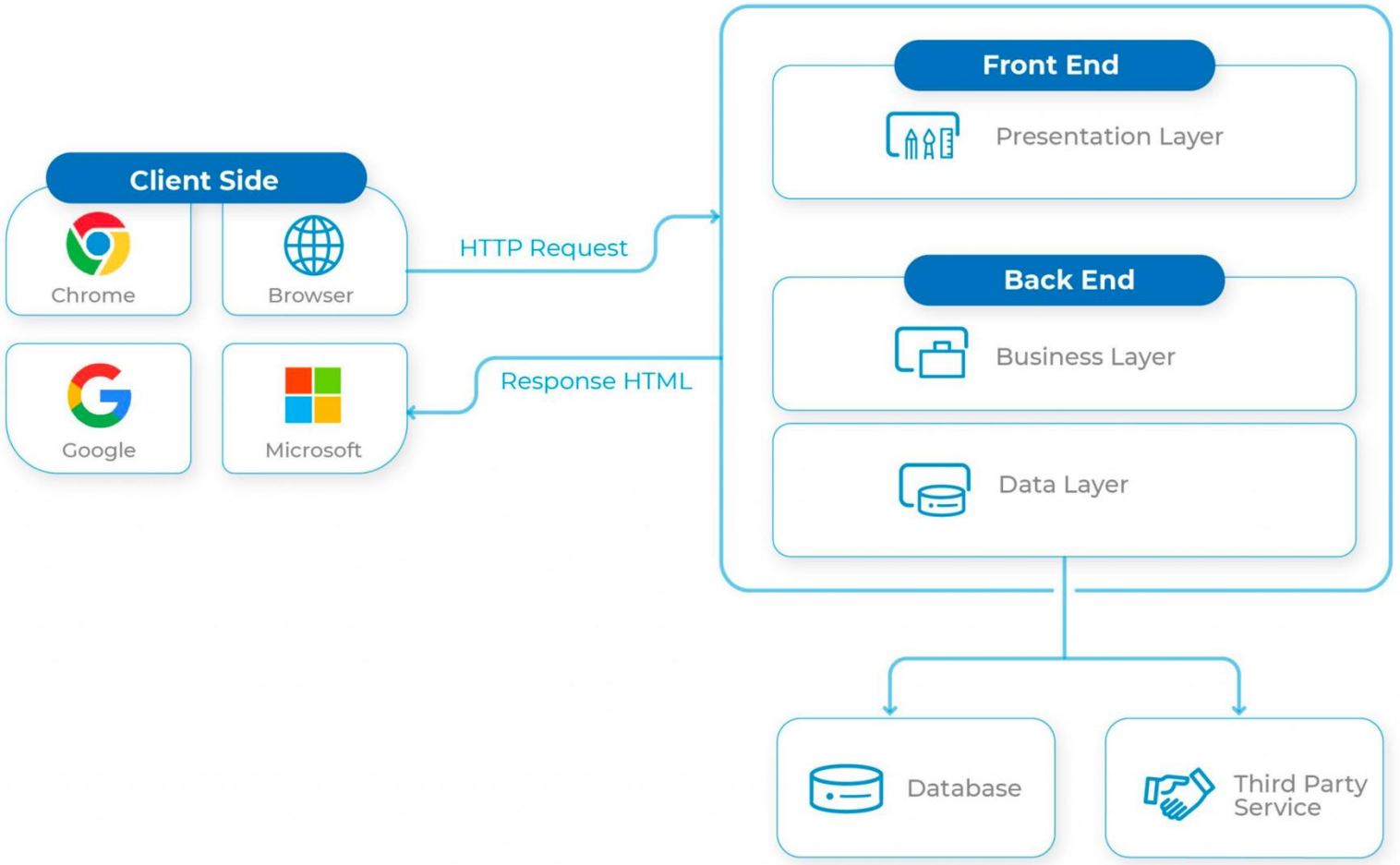
- However, assuming a web-app architecture permits more concrete requirements to be stated
  - Helps to make requirements more understandable

Requirements and Design Patterns

- A design pattern is a **reusable solution** to a recurring problem in software design

- Kovitz: requirements should be based on an architectural design pattern
  - Choice of design pattern raises specific questions whose answers clarify product's requirements

Standard Web Application Architecture



Example: Some Questions Raised by Use of Typical Web Application Architecture

- What services should the application provide to users?
- What should the page(s) corresponding to each service look like?
- How many service requests must be handled normally and during peak usage?
- What response times are acceptable?
- What kinds of application failures and security violations are possible?
- What are their consequences and severity levels?
- What frequency of occurrence is tolerable for each kind?
- What forms of remediation are called for?

Levels of Detail in Requirements Specification

- Different levels of detail are appropriate to different audiences
  - **Sponsors** and managers want to know how a product will help their organization achieve its objectives
  - **Users** care about product features
  - **Designers** require much more detail

Business Requirements

- High-level requirements
- Directly related to achieving business objectives
- Do not describe specific software functionality
- Constrained by business rules

Example: Business Requirements for Insurance Agency Software

- Reduce the average time required to process a policy application by 30%.
- Reduce the average time required to underwrite a policy by 25%.
- Reduce the average time required to process a claim by 35%.
- Increase customer retention by 20%.

User Requirements

- Describe or constrain tasks that users need to accomplish
- Multiple user requirements may support one business requirement

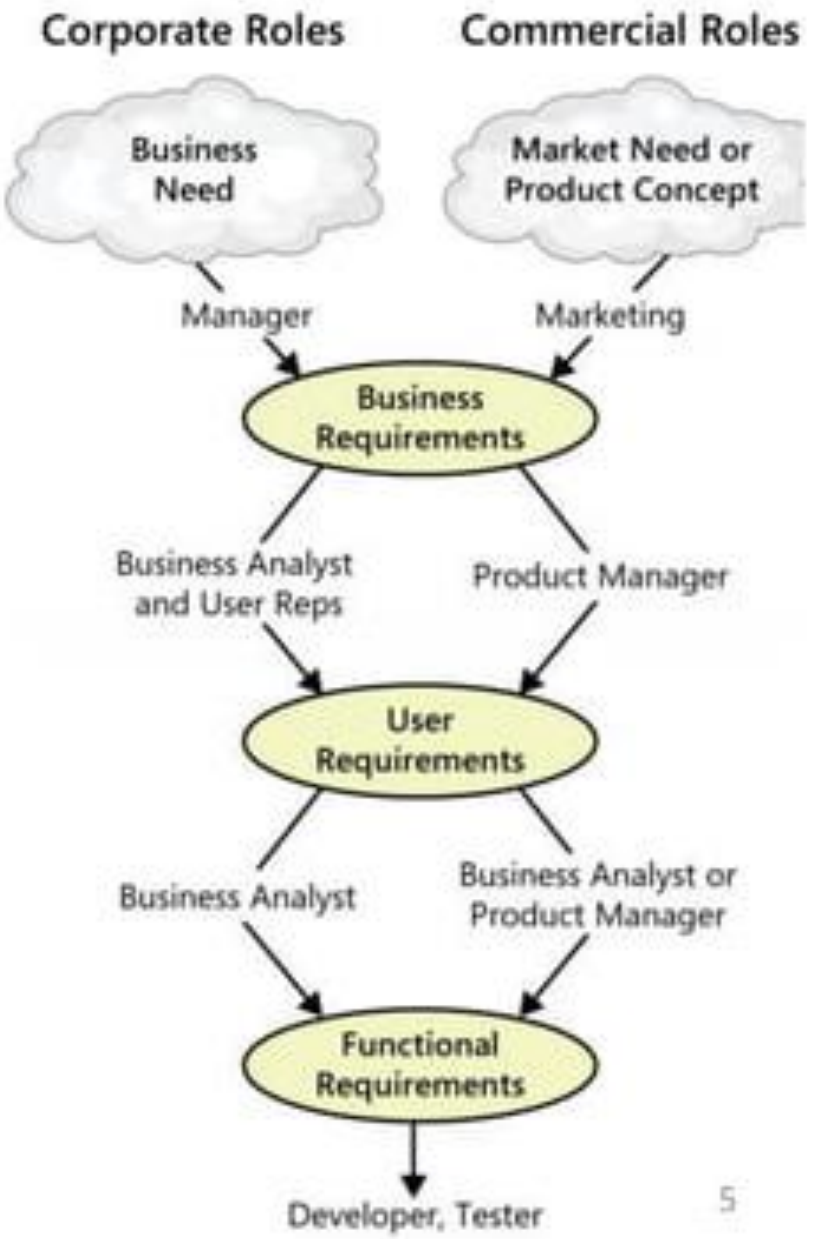
Example: User Requirements for Insurance Agency Software

- Evaluate the risk to the insurer
- Determine the premium rate
- Write a policy covering the applicant

Functional and Non-Functional Requirements

- **User requirement: Evaluate the risk to the insurer**
- Corresponding functional requirements:
  - The system shall input or retrieve the following data for each customer:
    - The vehicle make, model, year, and mileage
    - The requested coverage level
    - The customer's driving record
    - The customer's claim history
    - The customer's credit history
  - The system shall use the ACME scoring algorithm to rate the policy application

Working with the Three Layers



From Weigers & Beatty, Software Requirements, 2013

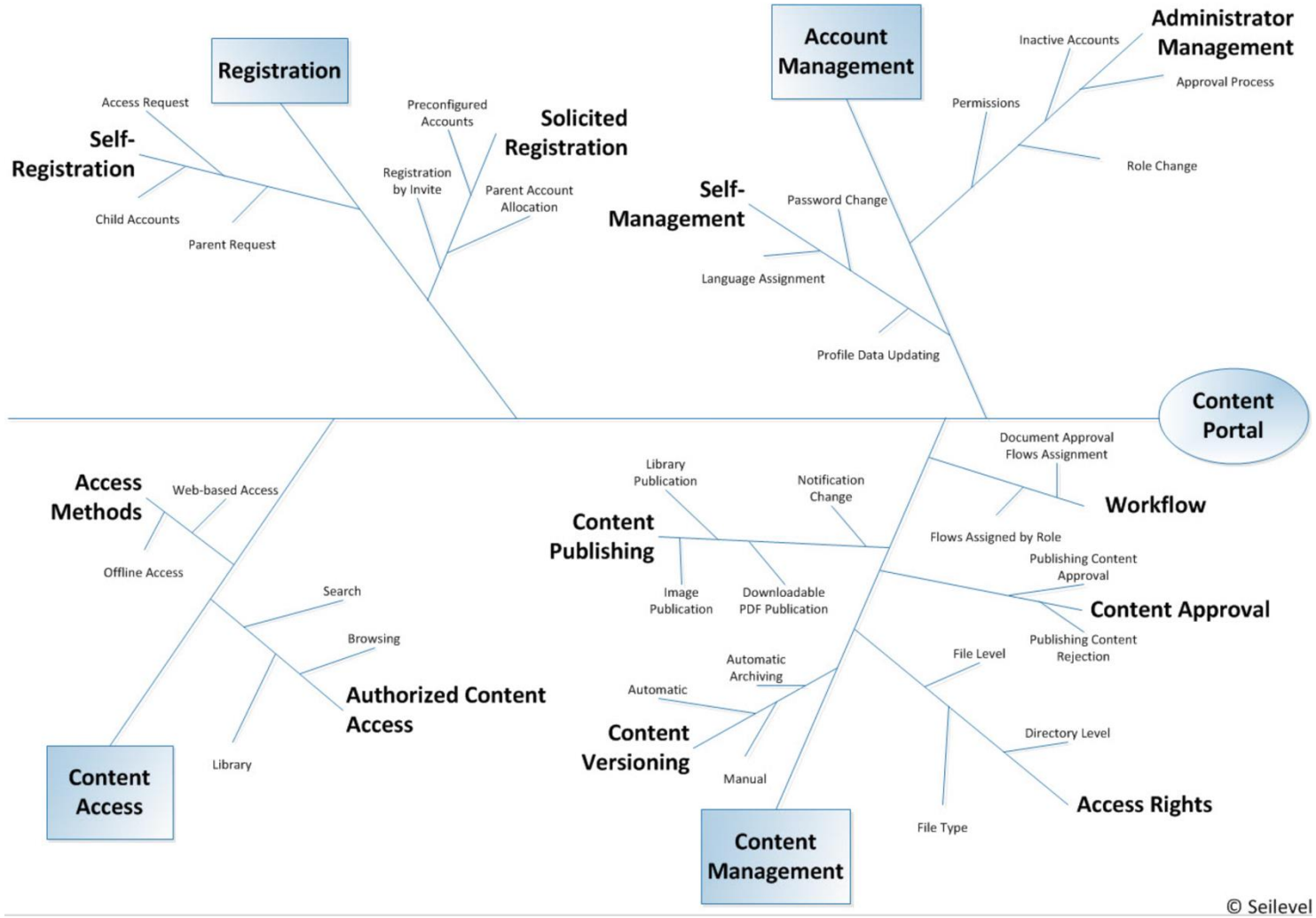
Software Features

- Groups of logically related functional requirements
- May influence a customer to choose product over others
- May correspond to user requirements
- Emphasized in marketing
- Relationships may be depicted graphically in feature tree

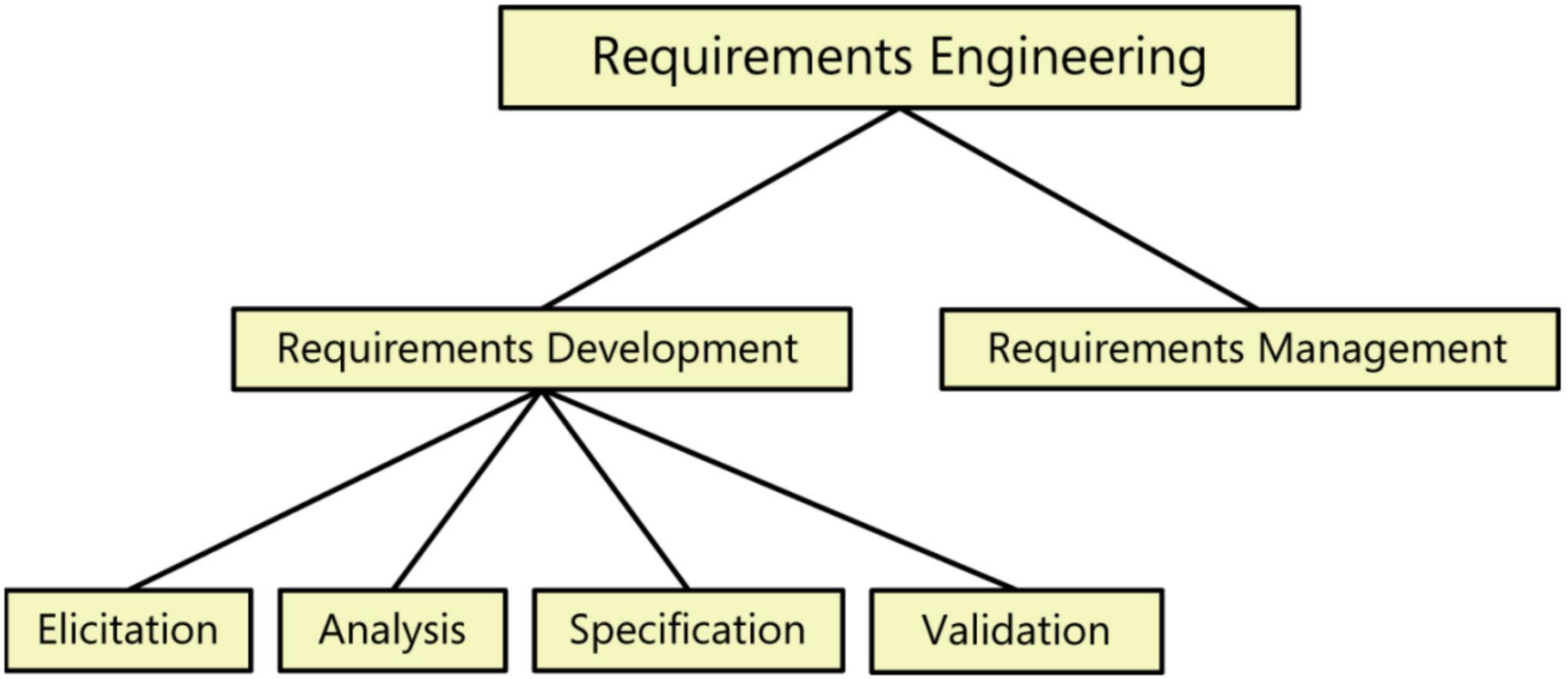
Example: Features of Insurance Agency Software

- Tracks Your Clients and Prospects
- Generates Policy Renewal Reports
- Calculates and Reports Agent Commissions
- Synchronizes with ACME Accounting Package

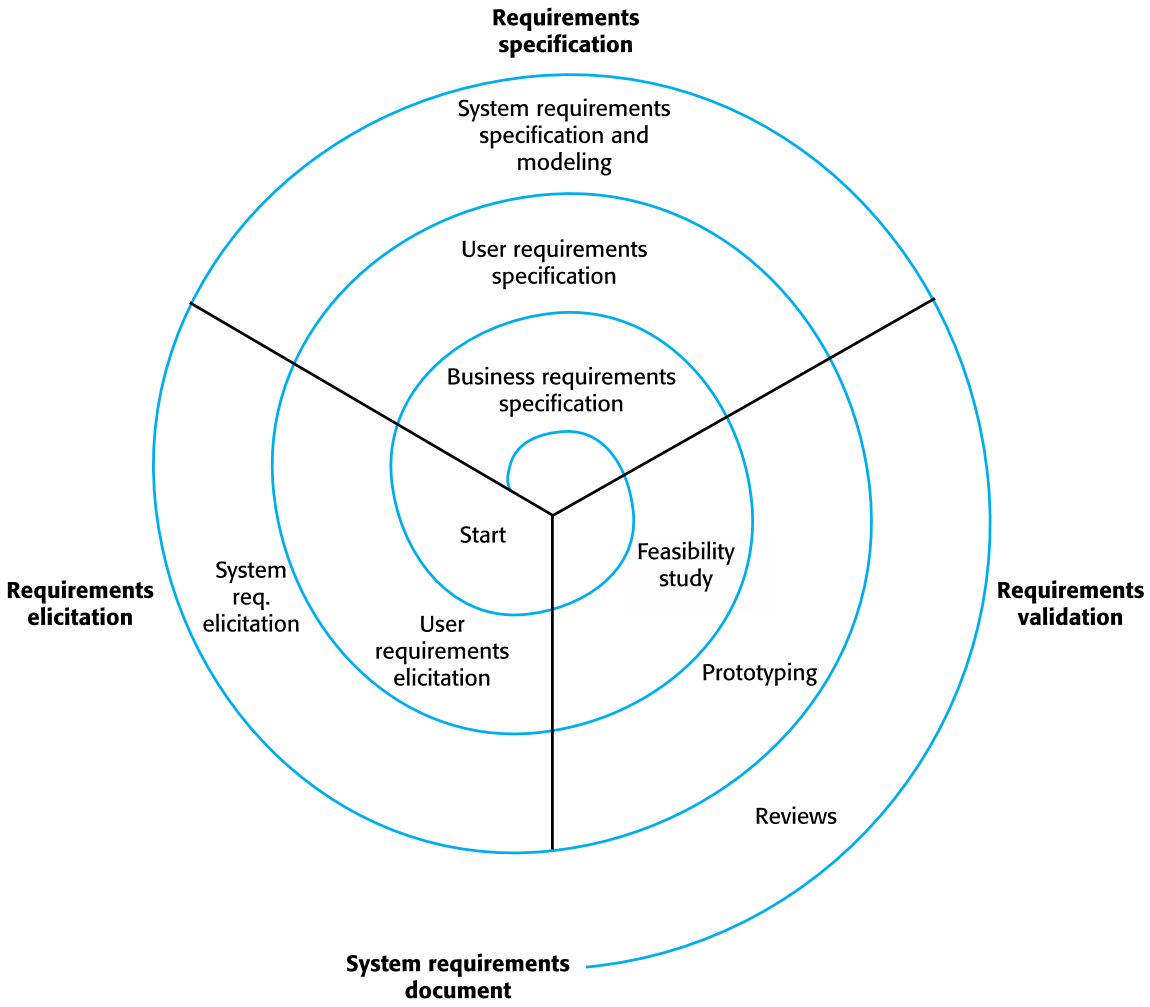
Feature Tree



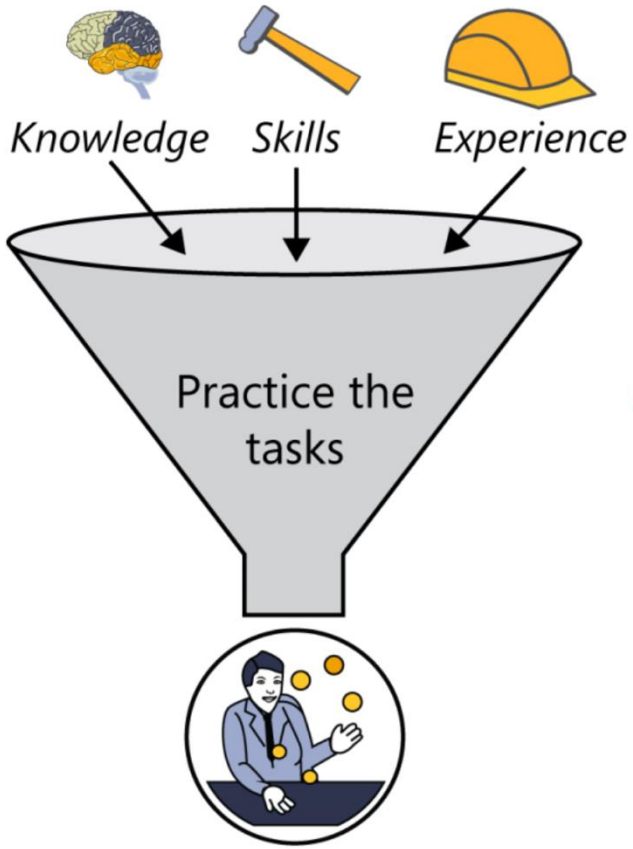
Requirements Engineering Activities



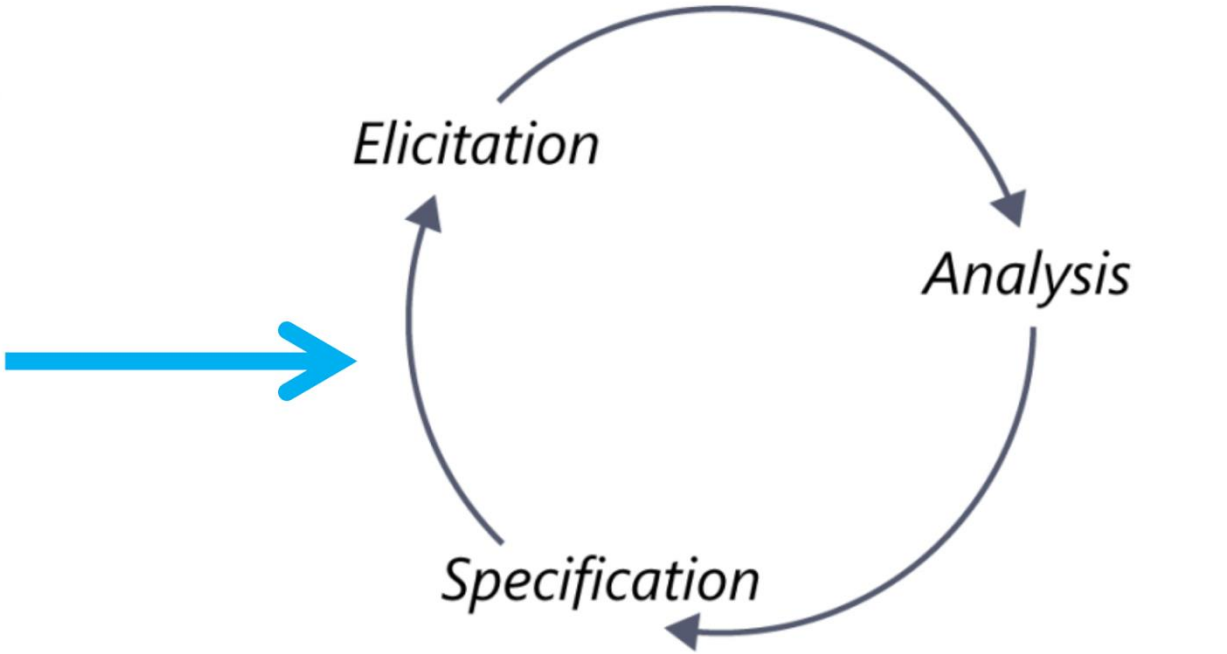
Requirements Development Process



Effective Business Analysis



The well-rounded BA



The cyclic nature of requirements elicitation, analysis, and specification.

Identifying User Classes and Their Requirements

- Helps to achieve completeness
- Each user class has its own viewpoint on system and requirements
- Characteristics of each user class should be documented

Example: Auto-teller Machine Viewpoints (User Classes)

- Bank customers
- Representatives of other banks
- Hardware and software maintenance engineers
- Marketing department
- Bank managers and counter staff
- Database administrators and security staff
- Network engineers

Wiegers on user classes: https://www.youtube.com/watch?v=LRQBFqB8VD0

Requirements Gathering Techniques

- Interviews
- Ethnography
- Document analysis
- Customer site visits
- Business process analysis
- Problem lists
- Surveys
- Competitive product analysis
- Reverse engineering
- Retrospectives on previous product
- Facilitated requirements **workshops**

Interviewing

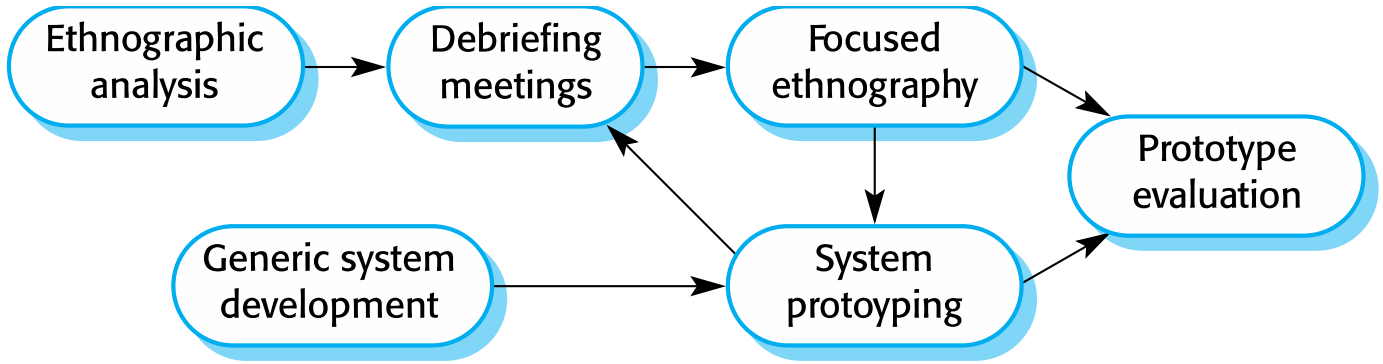
Formal or informal interviews with stakeholders

- Types of interview
  - Closed interviews
  - Open interviews

- A mix of closed and open-ended interviewing happens in practice.
- Interviews are not good for understanding domain requirements.

Ethnography

- Social and organisational factors of importance may be observed.
- Ethnographic studies have shown that work is usually richer and more complex.
- Effective for understanding existing processes



Ethnography and prototyping for requirements analysis

Facilitated Requirements Workshops

- Structured meetings
- Group of stakeholders and content experts carefully selected
- Collaboratively define, create, and refine deliverables
- Neutral, trained facilitator
- Facilitates communication, decision-making, mutual understanding

From www.construx.com

Properties of Good Requirements Specification

- Complete
- Correct
- Feasible
- Understandable
- Unambiguous
- Consistent
- Validateable or testable
- Modifiable
- Traceable

Example SRSs

- See Wiegers' COS SRS (best general model) and Vision & Scope document
- See Canvas for examples

1. Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the Cafeteria Ordering System (COS). This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Project Scope and Product Features

The Cafeteria Ordering System will permit Process Impact employees to order meals from the company cafeteria on-line to be delivered to specified campus locations. A detailed project description is available in the *Cafeteria Ordering System Vision and Scope Document* [1]. The section in that document titled "Scope of Initial and Subsequent Releases" lists the features that are scheduled for full or partial implementation in this release.

1.3 References

1. Wiegers, Karl. *Cafeteria Ordering System Vision and Scope Document*, www.processimpact.com/projects/COS/COS_vision_and_scope.doc
2. Wiegers, Karl. *Process Impact Intranet Development Standard, Version 1.3*, www.processimpact.com/corporate/standards/PI_intranet_dev_std.doc
3. Zambito, Christine. *Process Impact Business Rules Catalog*, www.processimpact.com/corporate/policies/PI_business_rules.doc
4. Zambito, Christine. *Process Impact Internet Application User Interface Standard, Version 2.0*, www.processimpact.com/corporate/standards/PI_internet_ui_std.doc

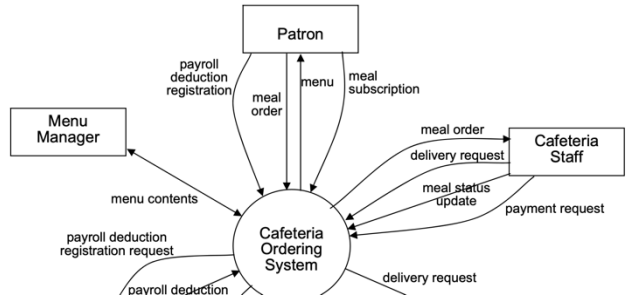
2. Overall Description

2.1 Product Perspective

The Cafeteria Ordering System is a new system that replaces the current manual and telephone processes for ordering and picking up lunches in the Process Impact cafeteria. The context diagram in Figure 1 illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately connecting to the Internet ordering services for several local restaurants and to credit and debit card authorization services.

2.2 User Classes and Characteristics

Patron (favored) A Patron is a Process Impact employee at the corporate campus in Clackamas, Oregon, who wishes to order meals to be delivered from the company cafeteria. There are about 600 potential Patrons, of which an estimated 400 are expected to use the Cafeteria Ordering System an average of 4 times per week each (source: current cafeteria usage data). Patrons will sometimes order multiple meals for group events or guests. An estimated 90 percent of orders will be placed using the corporate Intranet, with 10 percent of orders being placed from home. All Patrons have Intranet access from their offices. Some Patrons will wish to set up meal subscriptions, either to have the same meal to be delivered every day or to have the day's meal special delivered automatically. A Patron must be able to



Requirements Specification

The process of writing down the user and system requirements in a document.

Notation	Description
Natural language	The requirements are written using numbered sentences in natural language.
Structured natural language	The requirements are written in natural language on a standard form or template.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system.
Graphical notations	Graphical models, supplemented by text annotations, UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets.

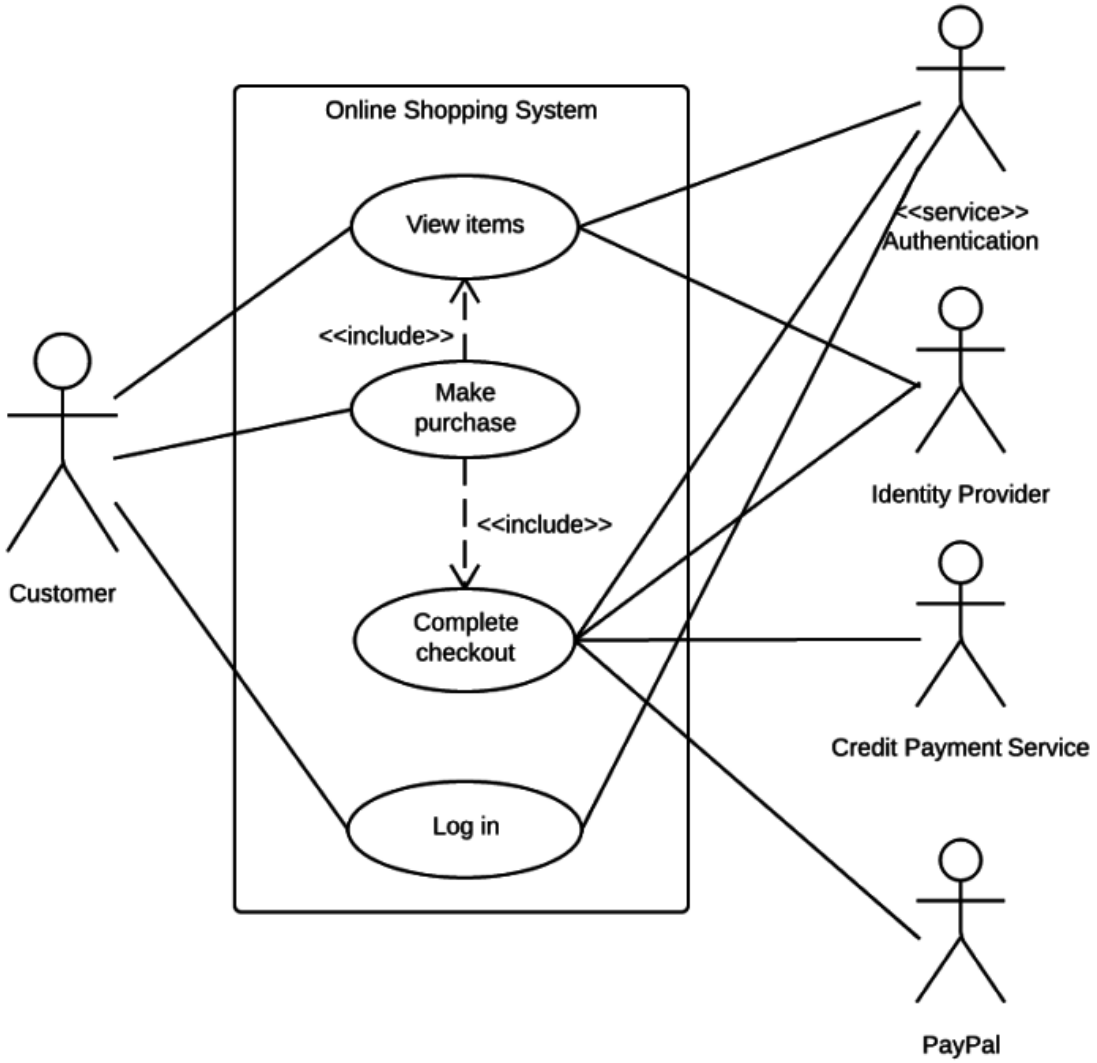
Use Case

- Set of related usage scenarios for software system
- Takes form of sequence of interactions between system and actor(s)
- Actor may be person, another system, or device
- Represented graphically with use-case diagram

Example Use Case Description

Name	Save item for purchase.
ID	UC_001
Description	While browsing items in the eStore, a user finds an item he is not ready to purchase yet, but he wants to save it to a list so that he can later find the item that he was previously interested in.
Actors	eStore customer.
Organizational Benefits	Increase sales by helping the customer remember products he was previously interested in.
Frequency of Use	20% of users save an item to be bought later each time they visit the site. 50% of saved items are purchased within one year of the saved date.
Triggers	The user selects an option to save an item.
Preconditions	User is viewing an item in the catalog.
Postconditions	The item selected to be saved is visible to the user when he views his saved items. The item selected to be saved is reflected as a saved item when the user views his eStore search and browse results.
Main Course	1. System prompts user to confirm saving selected item instead of purchasing it right away. 2. User confirms to save now (see EX1). 3. System determines user is not logged in and redirects user to log on (see AC1). ? 4. User logs on (see AC2, AC3). 5. System stores the saved item (see EX2). 6. System redirects the user to their saved items list to view the full list.
Alternate Courses	AC1 System determines user is already logged on. 1. Return to Main Course step 5. AC2 User logs off again. 1. Return user to Main Course step 3. AC3 User does not have an account already. 1. User creates an account. 2. System confirms account creation. 3. Return user to Main Course step 4.
Exceptions	EX1 User decides to purchase the item now. 1. See "Purchase item" Use Case. EX2 System fails on saving item to list. 1. System notifies user that an error has occurred. 2. Return user to Main Course step 1.

Example Use Case Diagram



https://www.lucidchart.com/pages/uml-use-case-diagram

Use Cases and Functional Requirements

- Use cases don't contain all necessary information about requirements
- Requirements can be added to use cases
- Alternatively, requirements derived from use cases can be detailed in the SRS
- Many SRS's don't explicitly describe use cases

User Stories

- Used in agile development
- Short description of feature told from the perspective of user

User Type	Epic	User Story
Mobile User	Registration	As a user, I can register for the application by entering my email, password, and confirming my password
		As a user, I will receive a confirmation email once I have registered for the application
		As a user, I can register for the application through Facebook
		As a user, I can upload a profile photo and add my name to my account
	Login	As a user, I can log into the application by entering my email and password
		As a user, I can log into the application through Facebook, if I previously registered with it
		As a user, I can reset my password if I have forgotten my password
	My Account	As a user, I can view my personal information
		As a user, I can edit my profile photo
		As a user, I can edit my email. I will receive a confirmation email to my new email address.
		As a user, I can logout of the application from my account
Web User	Registration	As a user, I can register for the application by entering my email, password, and confirming my password
		As a user, I will receive a confirmation email once I have registered for the application
		As a user, I can register for the application through Facebook
		As a user, I can upload a profile photo and add my name to my account

Examples of Use Cases & User Stories

Application	Sample use case	Corresponding user story
Chemical tracking system	Request a Chemical	As a chemist, I want to request a chemical so that I can perform experiments.
Airport check-in kiosk	Check in for a Flight	As a traveler, I want to check in for a flight so that I can fly to my destination.
Accounting system	Create an Invoice	As a small business owner, I want to create an invoice so that I can bill a customer.
Online bookstore	Update Customer Profile	As a customer, I want to update my customer profile so that future purchases are billed to a new credit card number.