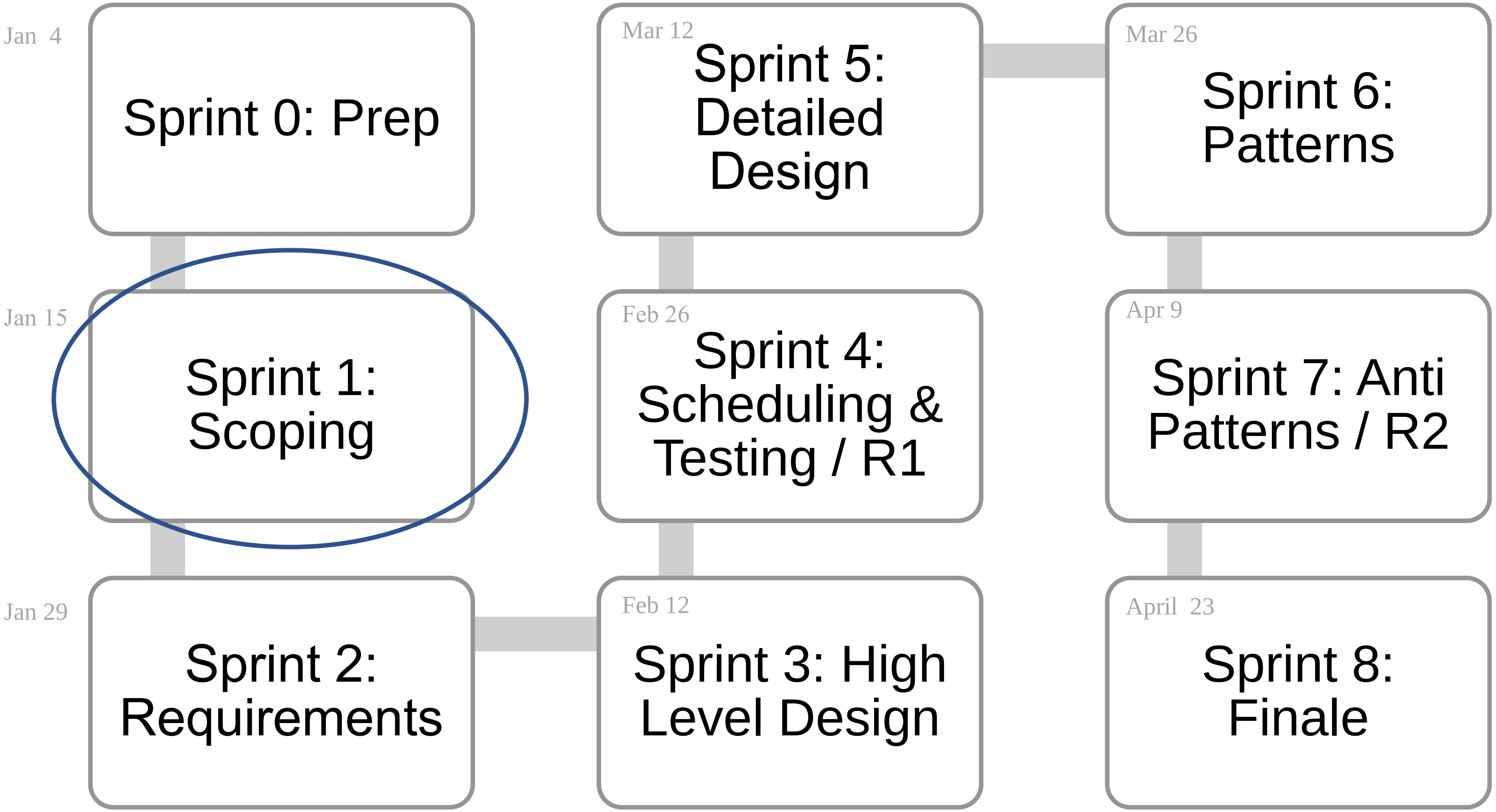
Requirements Gathering and Analysis

Some content adapted from Rajib Mall's book and Craig Larman's book.

Where are we..



Jan 4 Sprint 0: Prep

Jan 15 Sprint 1: Scoping

Jan 29 Sprint 2: Requirements

Feb 12 Sprint 3: High Level Design

Feb 26 Sprint 4: Scheduling & Testing / R1

Mar 12 Sprint 5: Detailed Design

Mar 26 Sprint 6: Patterns

Apr 9 Sprint 7: Anti Patterns / R2

April 23 Sprint 8: Finale

April 23+ EVALUATIONS

Lecture topics will follow this flow

As the projects tracks these sprints

Submissions due along the way

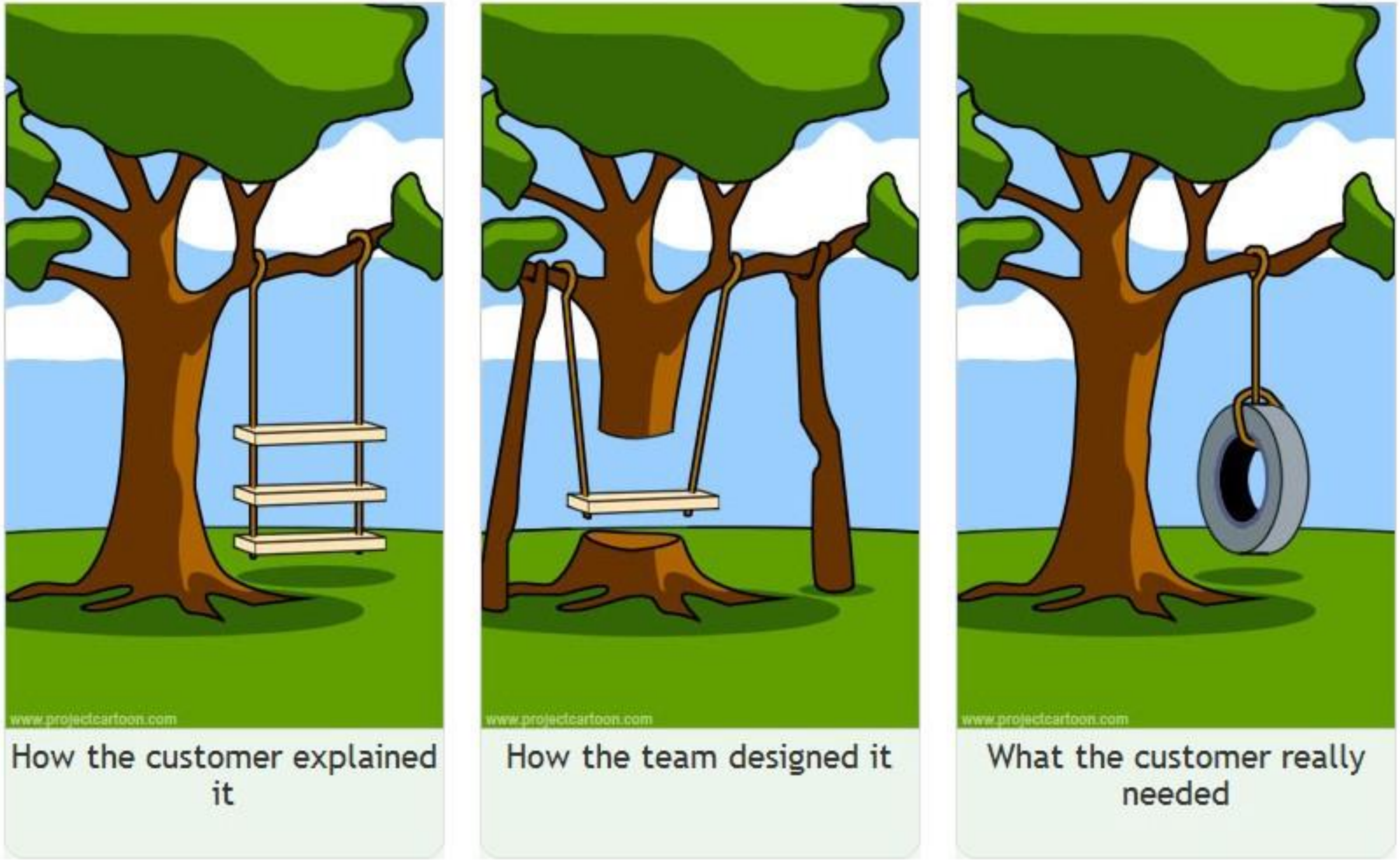
Guest lectures from startups and Industry leaders

EACH SPRINT IS 2 WEEKS, Friday start.

Why requirements gathering?

I'LL DESIGN THE SYSTEM AS SOON AS YOU GIVE ME THE USER REQUIREMENTS.
scottadams@aol.com
www.dilbert.com
BETTER YET, YOU COULD BUILD THE SYSTEM, THEN I'LL TELL YOUR BOSS THAT IT DOESN'T MEET MY NEEDS.
3/21/03 © 2003 United Feature Syndicate, Inc.
I DON'T MEAN TO FRIGHTEN YOU, BUT YOU'LL HAVE TO DO SOME ACTUAL WORK.
THAT'S CRAZY TALK.

Need to “know” what to build


www.projectcartoon.com

How the customer explained it

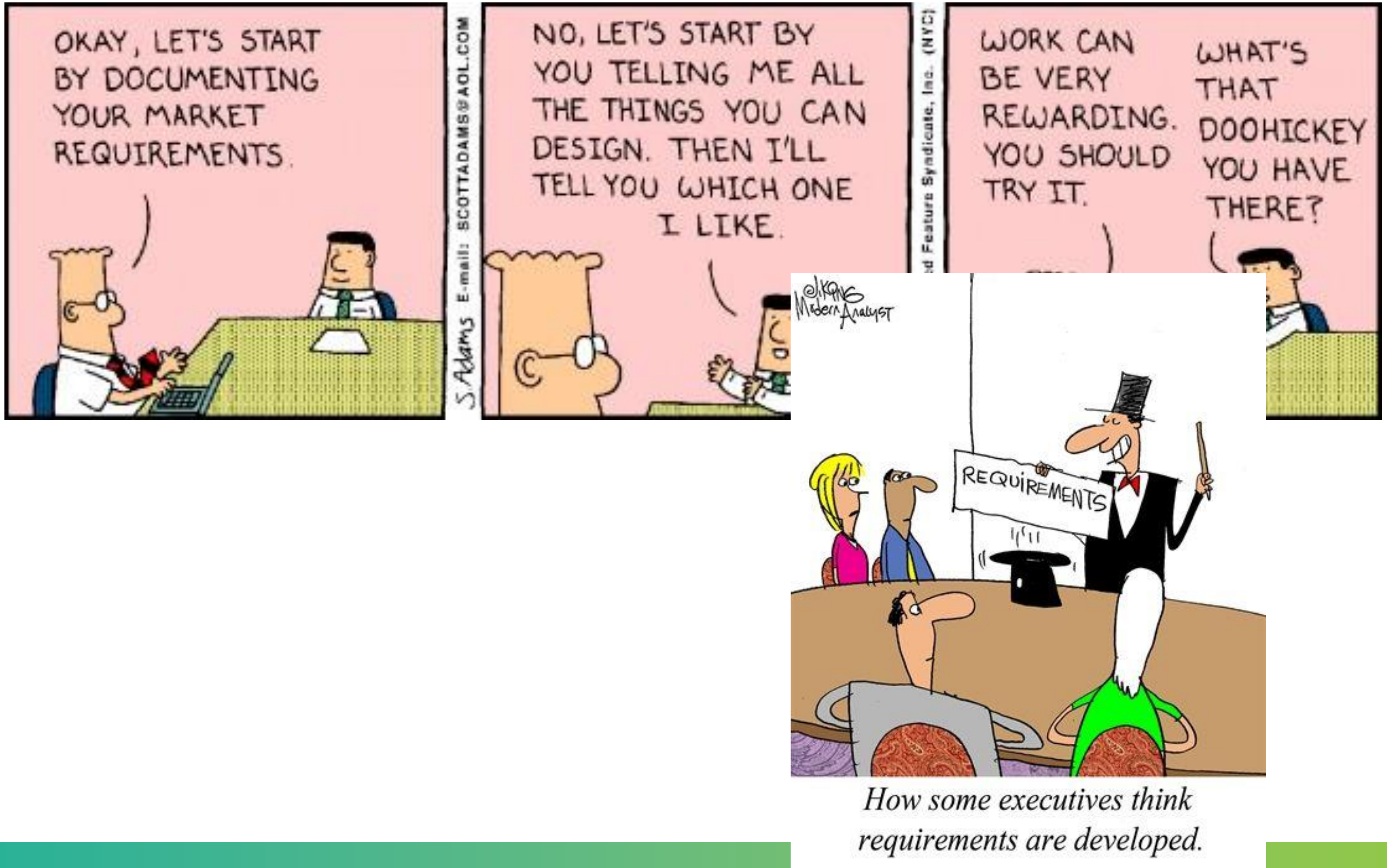
www.projectcartoon.com

How the team designed it

www.projectcartoon.com

What the customer really needed

This way?


OKAY, LET'S START BY DOCUMENTING YOUR MARKET REQUIREMENTS.
S. Adams E-mail: SCOTTADAMS@AOL.COM
NO, LET'S START BY YOU TELLING ME ALL THE THINGS YOU CAN DESIGN. THEN I'LL TELL YOU WHICH ONE I LIKE.
d Feature Syndicate, Inc. (NYC)
WORK CAN BE VERY REWARDING. YOU SHOULD TRY IT.
WHAT'S THAT DOOHICKEY YOU HAVE THERE?
©KING Modern Analyst
REQUIREMENTS

How some executives think requirements are developed.

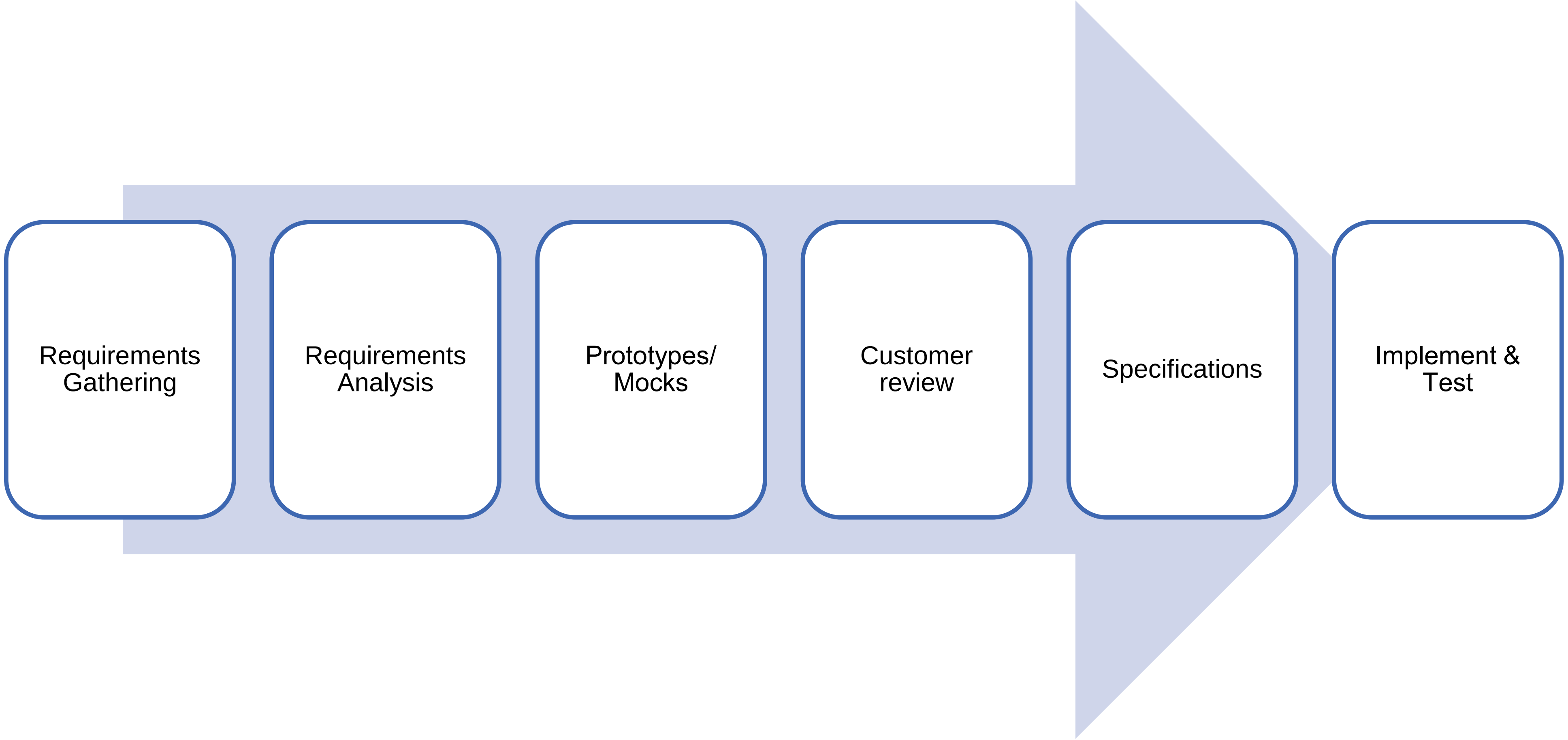
Consider this..

“Build an app for Felicity” . OK go build now!

- Can you?

Requirements

Requirements process



Requirements Gathering

Requirements Analysis

Prototypes/ Mocks

Customer review

Specifications

Implement & Test

Types of Requirements

- Business requirements
  - High-level objectives of the organization or customer who requests the system.

- Functional requirements
  - Describe *what* the system should do

  For example, features (use cases)

- Non-functional requirements
  - C*onstraints* that must be adhered to during development

  For example, quality constraints, technology constraints, process constraints, etc.

Analysis

-

Good analysts convince the stakeholders the business requirements are good enough.
Great analysts convince the stakeholders that requirements are a waste of time.
©J.KING ModernAnalyst

Lazy business analysts 'at work'...

Quality Requirements

- Correct – only user representative can determine
- Feasible – get reality check on what can or cannot be done
  - technically or within given cost constraints.
- Necessary – trace each requirement back to its origin
- Unambiguous – one interpretation
- Verifiable – is it complete
  - how to you know if the requirement was implemented properly?
- Prioritized – function of value provided to the customer

Requirements Gathering and Analysis

Some content adapted from Rajib Mall’s book and Craig Larman’s book.

Requirements Phase

- Many projects fail:
  - Because they start implementing the system.. Without determining whether they are building what the customer really wants.

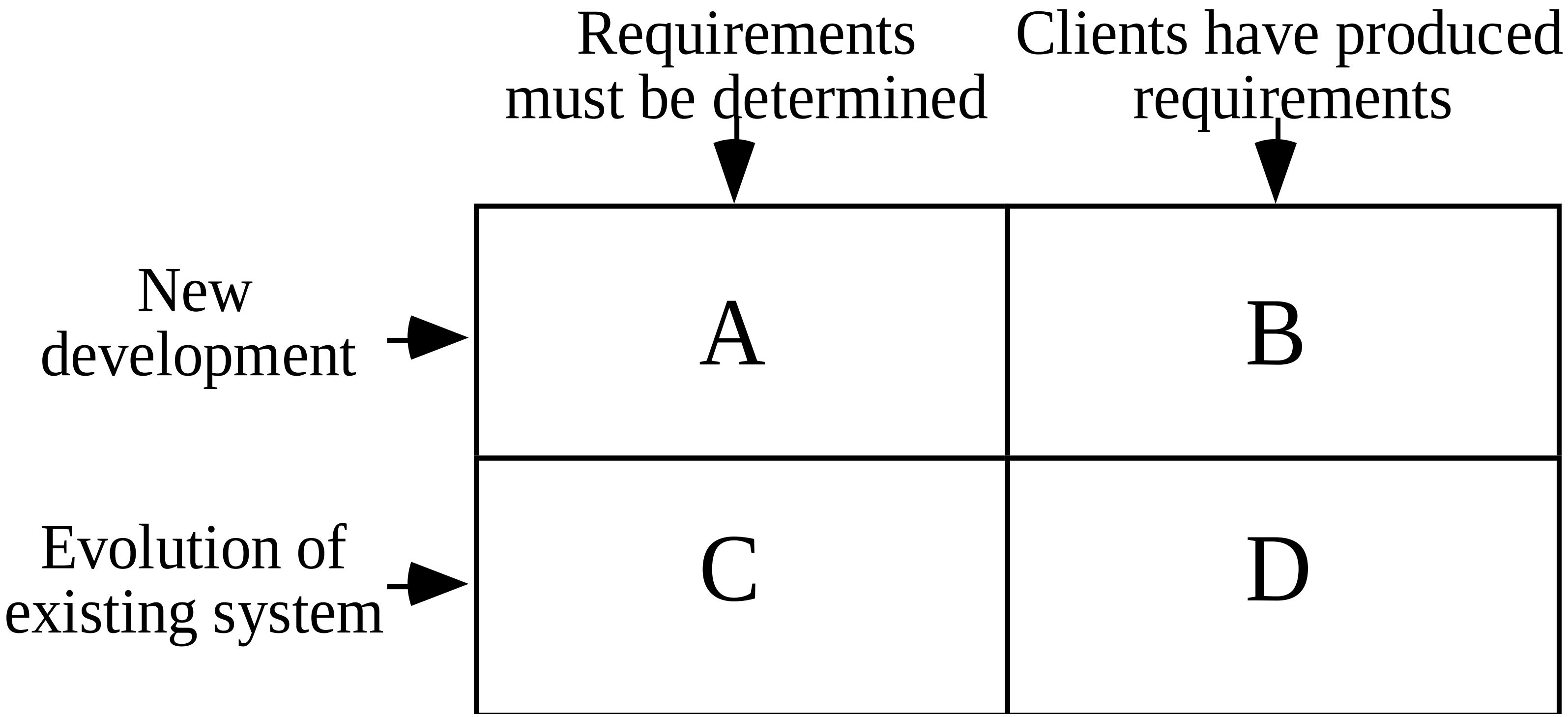
Why Requirements analysis/specification?

Factors that cause projects to fail:

- Lack of User Input 12.8%
- Incomplete Requirements & Specifications 12.3%
- Changing Requirements & Specifications 11.8%
- Lack of Executive Support 7.5%
- Technology Incompetence 7.0%
- Lack of Resources 6.4%
- Unrealistic Expectations 5.9%
- Unclear Objectives 5.3%
- Unrealistic Time Frames 4.3%
- New Technology 3.7%
- Other 23.0%

- Standish Group, *The Standish Group Report: Chaos*, 1995, http://www.scs.carleton.ca/~beau/PM/Standish-Report.html

The Starting Point for Software Projects

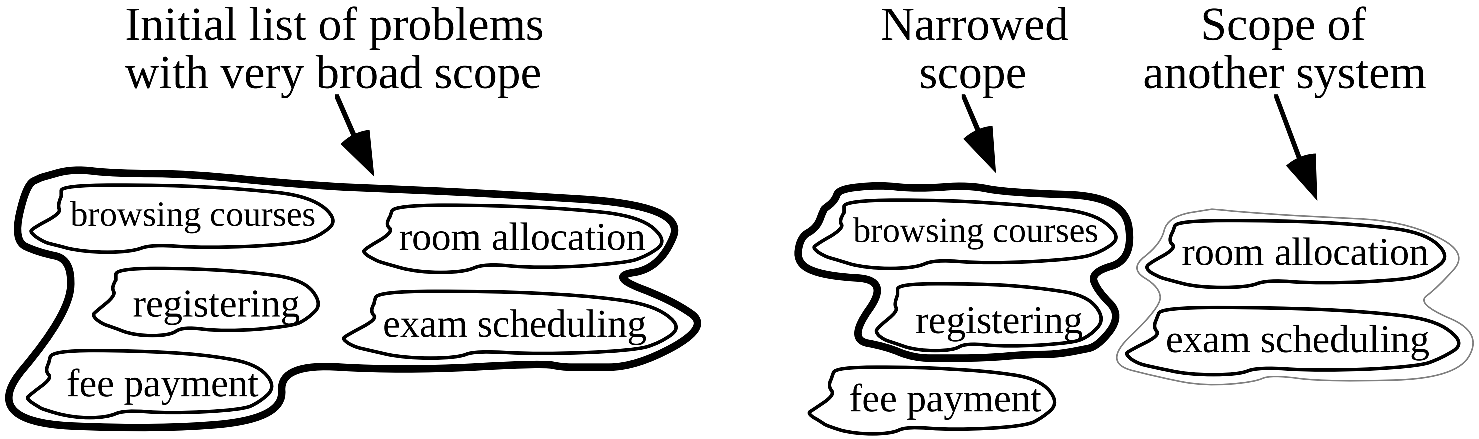


	Requirements must be determined	Clients have produced requirements
New development	A	B
Evolution of existing system	C	D

Defining the Scope

- 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

- Example: A university registration system



What is a Requirement?

- Requirement: A statement about the proposed system that all stakeholders agree must be made true in order for the customer's problem to be adequately solved.

  - Short and concise piece of information
  - Says something about the system
  - All the stakeholders have agreed that it is valid
  - It helps solve the customer's problem

- A collection of requirements is a *requirements document*.

Types of Requirements

- Business requirements
  - High-level objectives of the organization or customer who requests the system.

- Functional requirements
  - Describe *what* the system should do

  For example, features (use cases)

- Non-functional requirements
  - C*onstraints* that must be adhered to during development

  For example, quality constraints, technology constraints, process constraints, etc.

Requirements Phase

- Goals of requirements phase:
  - Fully understand the user requirements.
  - Remove inconsistencies, anomalies, etc. from requirements.
  - Document requirements properly in an SRS document.

Requirements Phase

- Consists of two distinct activities:
  - Requirements Gathering and Analysis
  - Requirements Specification

Requirements Gathering

- Also known as **requirements elicitation**.

- If the project is to automate some existing procedures
  - e.g., automating existing manual accounting activities,
  - The task of the system analyst is a little easier
  - Analyst can immediately obtain:
    - input and output formats
    - accurate details of the operational procedures

Requirements Gathering (CONT.)

- In the absence of a working system,
  - Lot of imagination and creativity are required.

- Interacting with the customer to gather relevant data:
  - Requires a lot of experience.

Analysis

Analysis of the gathered requirements

- Main purpose of requirements analysis:
  - Clearly understand the user requirements,
  - Detect inconsistencies, ambiguities, and incompleteness.
- Incompleteness and inconsistencies:
  - Resolved through further discussions with the end-users and the customers.

Inconsistent Requirement

- Some part of the requirement:
  - contradicts with some other part.

- <u>Example:</u>
  - One customer says turn off heater and open water shower when temperature > 100 C
  - Another customer says turn off heater and turn ON cooler when temperature > 100 C

Incomplete Requirement

- Some requirements have been omitted:
  Possibly due to oversight.

- <u>Example:</u>
  - The analyst has not recorded:
    when temperature falls below 90 C
    - heater should be turned ON
    - water shower turned OFF.

Analysis of the gathered requirements (contd.)

- Requirements analysis involves:
  - Obtaining a clear, in-depth understanding of the product to be developed,
  - Remove all ambiguities and inconsistencies from the initial customer perception of the problem.

Analysis of gathered requirements (contd.)

- Experienced analysts take considerable time:
  - To understand the exact requirements the customer has in his mind.

- Experienced systems analysts know -
  often as a result of past (painful) experiences

Analysis of gathered requirements (contd.)

- Several things about the project should be clearly understood by the analyst:
  - What is the problem?
  - Why is it important to solve the problem?
  - What are the possible solutions to the problem?
  - What complexities might arise while solving the problem?

Analysis of gathered requirements (contd.)

- After collecting all data regarding the system to be developed,
    - Remove all inconsistencies and anomalies from the requirements,
    - Systematically organize requirements into a Software Requirements Specification (SRS) document.

Bad Requirements: A Simplified Example

- *A mail should be displayed within 3 seconds of clicking on mail*
- *User should be able to add a new mail server during peak hours within a small downtime*
- *Business services should not be interrupted during the peak hours*
- *User should be able to customize all the mailbox settings*
- *User should be able to change the look and feel of how the mailbox is displayed*

Quality Requirements

- Correct – only user representative can determine
- Feasible – get reality check on what can or cannot be done technically or within given cost constraints.
- Necessary – trace each requirement back to its origin
- Unambiguous – one interpretation
- Verifiable – how to you know if the requirement was implemented properly?
- Prioritized – function of value provided to the customer

Writing Example #1

“The product shall provide status messages at regular intervals not less than every 60 seconds.”

Writing Example #1

"The product shall provide status messages at regular intervals not less than every 60 seconds."

- Incomplete – What are the status messages and how are they supposed to be displayed?
- Ambiguous – What part of the product? Regular interval?
- Not verifiable

Alternative #1

1. Status Messages.

1.1. The Background Task Manager shall display status messages in a designated area of the user interface at intervals of 60 plus or minus 10 seconds.

1.2. If background task processing is progressing normally, the percentage of the background task processing that has been completed shall be displayed.

1.3. A message shall be displayed when the background task is completed.

1.4. An error message shall be displayed if the background task has stalled.

Writing Example #2

“The product shall switch between displaying and hiding non-printing characters instantaneously."

Writing Example #2

“The product shall switch between displaying and hiding non-printing characters instantaneously."

- Not Feasible – computers cannot do anything instantaneously.
- Incomplete – conditions which trigger state switch
- Ambiguous – “non-printing character”

Alternative #2

“The user shall be able to toggle between displaying and hiding all HTML markup tags in the document being edited with the activation of a specific triggering condition.”

- Note that “triggering condition” is left for design

The Specification Trap

The Landing Pilot is the Non-Landing Pilot until the ‘decision altitude’ call, when the Handling Non-Landing Pilot hands the handling to the Non-Handling Landing Pilot, unless the latter calls ‘go-around,’ in which case the Handling Non-Landing Pilot continues handling and the Non-Handling Landing Pilot continues non-handling until the next call of ‘land,’ or ‘go-around’ as appropriate . In view of recent confusions over these rules, it was deemed necessary to restate them clearly.

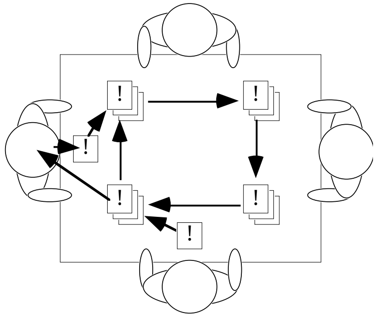
- British Airways memorandum, quoted in *Pilot Magazine*.

Techniques - Gathering and Analyzing Requirements

- Observation
  - Read documents and discuss requirements with users
  - Shadowing important potential users as they do their work
    - ask the user to explain everything he or she is doing
  - Session videotaping
- Interviewing
  - Conduct a series of interviews
    - Ask about specific details
    - Ask about the stakeholder’s vision for the future
    - Ask if they have alternative ideas
    - Ask for other sources of information
    - Ask them to draw diagrams

Gathering and Analyzing Requirements

- Brainstorming
  - Appoint an experienced moderator
  - Arrange the attendees around a table
  - Decide on a ‘trigger question’
  - Ask each participant to write an answer and pass the paper to its neighbour



- ***Joint Application Development (JAD)*** **is a technique based on intensive brainstorming sessions**

Gathering and Analyzing Requirements

- Prototyping
  - The simplest kind: *paper prototype*.
    - a set of pictures of the system that are shown to users in sequence to explain what would happen
  - The most common: a mock-up of the system’s UI
    - Written in a rapid prototyping language
    - Does *not* normally perform any computations, access any databases or interact with any other systems
    - May prototype a particular aspect of the system

Difficulties and Risks in Domain and Requirements analysis

- 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*
- 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*