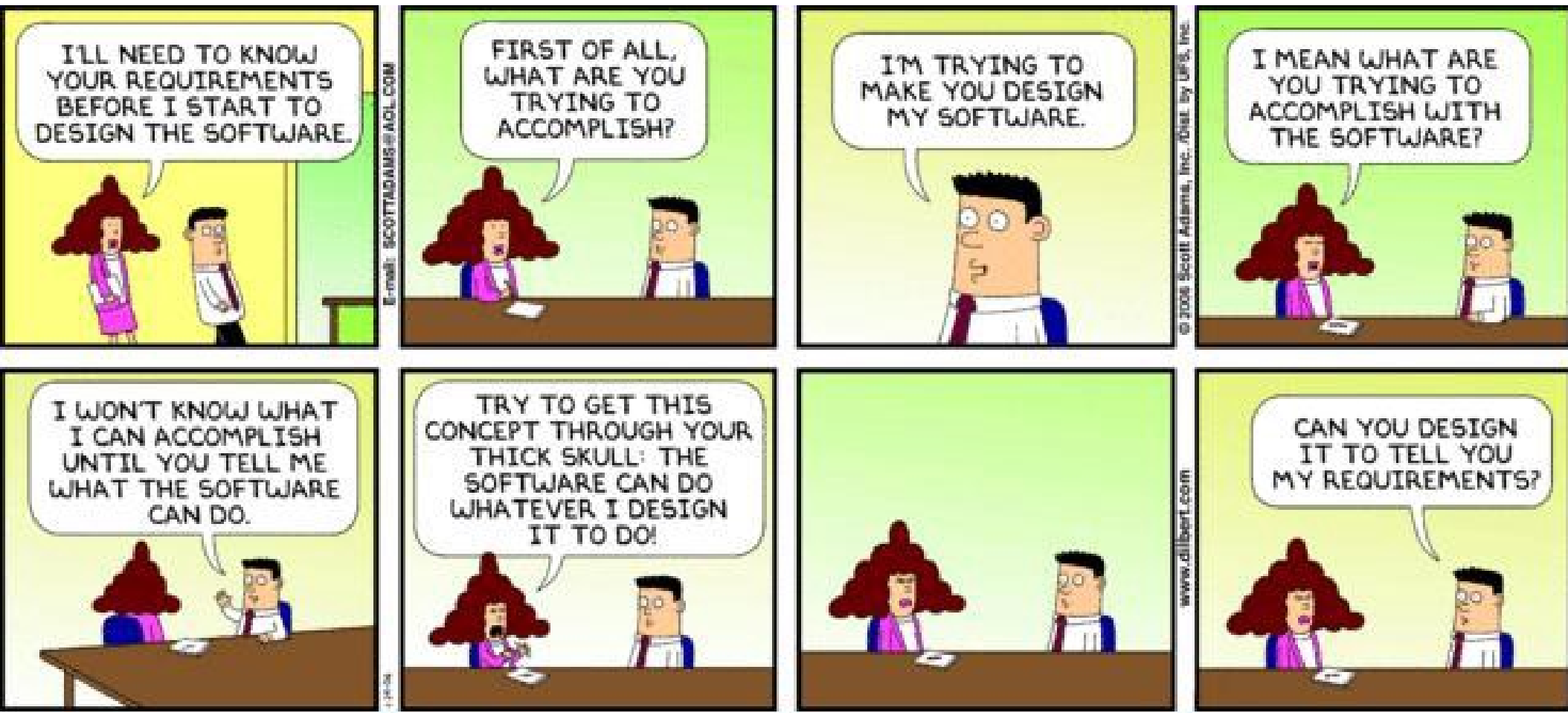# Requirements and Specifications

The Story So Far ...

- Quality assurance is critical to software engineering

- OK, so we want to build a quality product

- What are we supposed to be building, again?

One-Slide Summary

- **Requirements** articulate the relationship and interface between a desired system and its environment. This includes both what is (or is expected) and what should be.

- We distinguish between **functional** and **quality** (or non-functional) requirements. Both should be stated in measurable ways.

- Requirements can describe variables, inputs, outputs, and assumptions between them.

- We distinguish between informal statements and **verifiable** requirements.
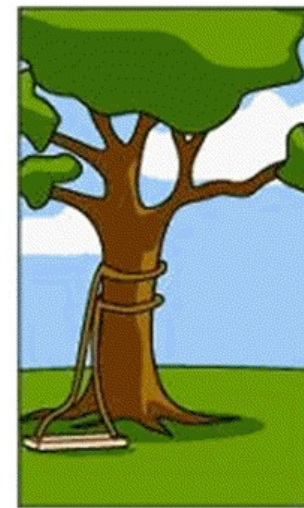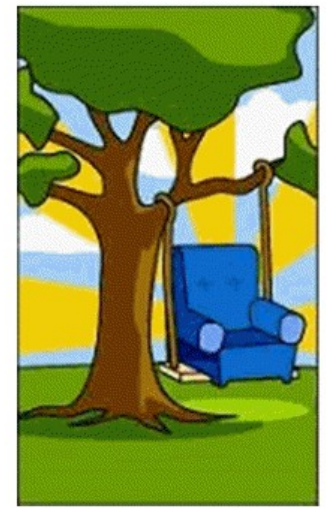
How the customer explained it
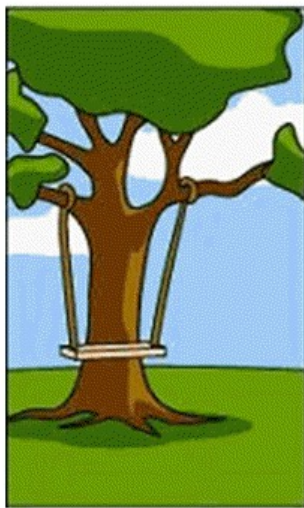
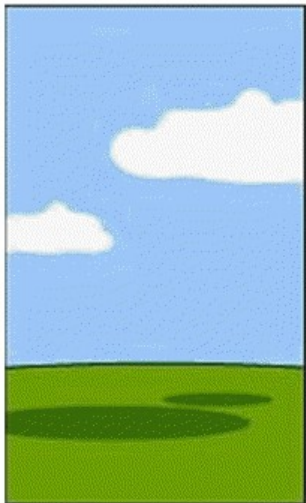How the project leader understood it

How the engineer designed it

How the programmer wrote it

How the sales executive described it

How the project was documented
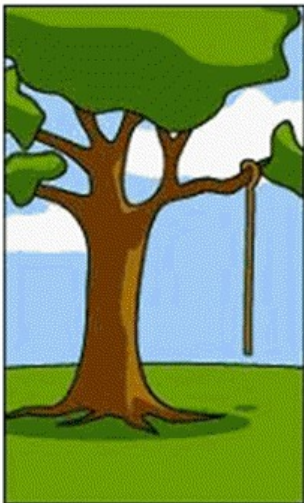
What operations installed
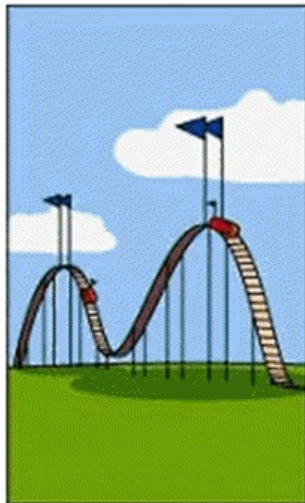
How the customer was billed
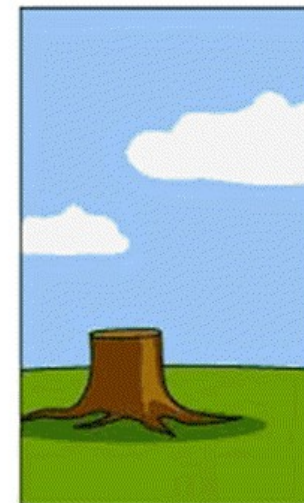
How the help desk supported it
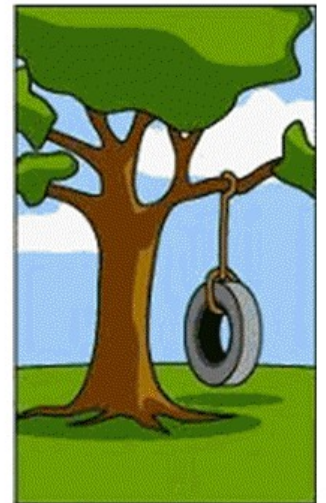
What the customer really needed

Requirements

- **Requirements** say what the system will do, *not* how it will do it

- "The hardest single part of building a software system is deciding precisely **what to build**. No other part of the conceptual work is as difficult as establishing the detailed technical requirements … No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later."

  — Fred Brooks

"Difficult to Rectify Later"

Healthcare.gov

What is Past is Prologue

- A 1994 survey of 8000 projects at 350 companies found: 31% of projects canceled before completed; 9% of projects delivered on time, within budget in large companies, 16% in small companies. (Similar results reported since.)

- Largest Causes:
  - Incomplete requirements (13.1%)
  - Lack of user involvement (12.4%)
  - Lack of resources (10.6%)
  - Unrealistic expectations (9.9%)
  - Lack of executive support (9.3%)

No "programmers were too inept" or "we forgot how AVL trees work"

8

Communication Problem

- Goal: figure out what should be built

- Express those ideas so that the correct thing is built
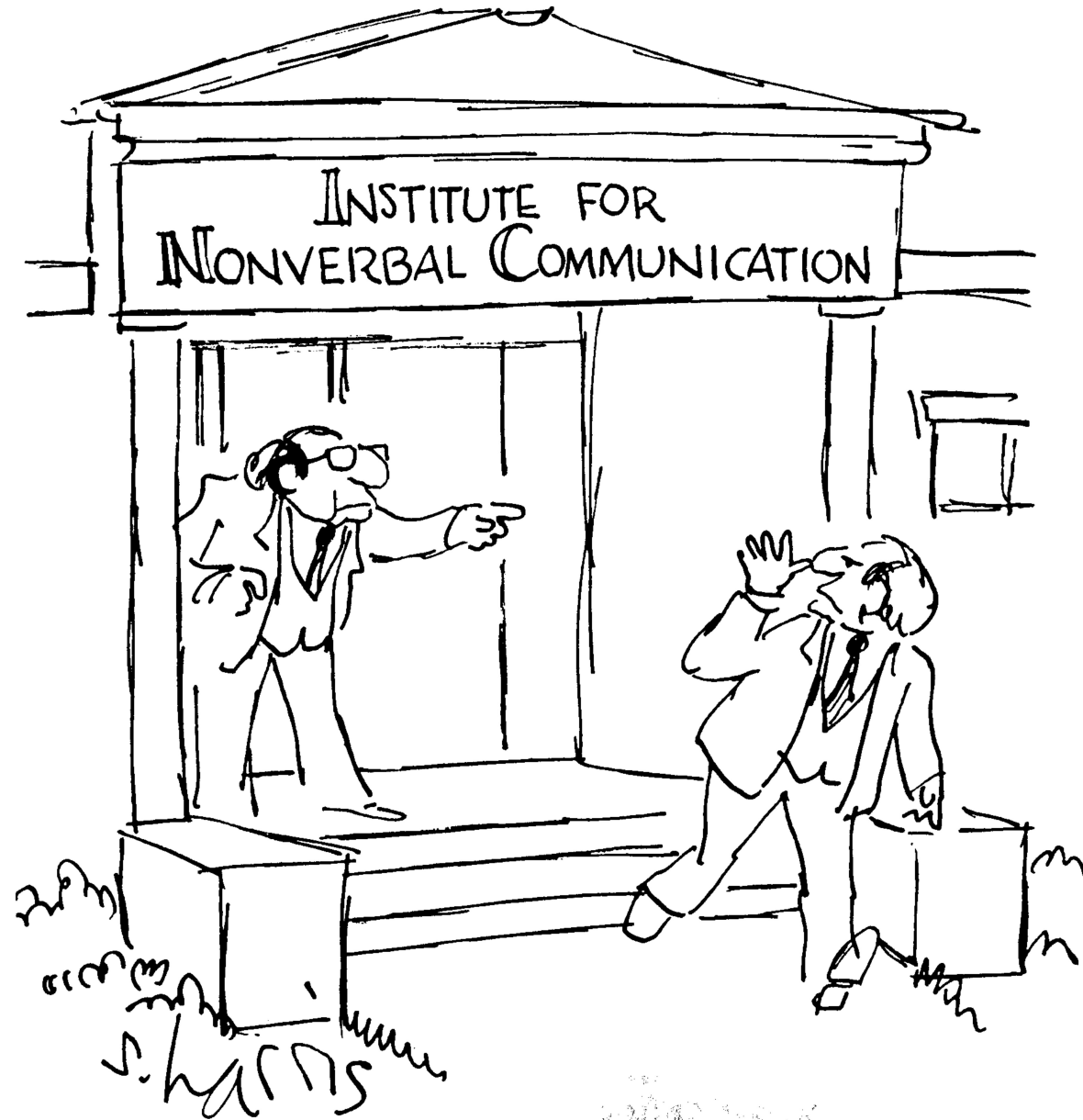
"I'm Already Good At This": Denial

- Denial by prior knowledge – we have done this before, so we know **what** is required

- Denial by hacking – our fascination with machines dominates our focus on the **how**

- Denial by abstraction – we pursue elegant models which obscure, remove or downplay the real world

- Denial by vagueness – imply (vaguely) that machine descriptions are actually those of the world (cf. "threat to validity")

Requirements Brainstorming Example

- You are paying someone to write software for "selling videos on the web"

- Your thoughts on ...

  - How fast should we deliver content, at what quality, for what price?

  - "Nice to have" functionality

  - Required functionality

  - Expected qualities

  - Involved subproblems

Environment vs. Machine

Requirements
Domain Knowledge

Specifications

Computers
Software Programs

Environmental Domain

Machine Domain

monitored variables → Input devices (e.g. sensors) → input data

Environment

Software

controlled variables ← Output devices (e.g. actuators) ← output results

Pamela Zave & Michael Jackson, "Four Dark Corners of Requirements Engineering,"
ACM Transactions on Software Engineering and Methodology, 6(1): 1-30, 1997.

Environment vs. Machine Example: Automobile

MotorRaising

DriverWantsToStart

HandbrakeReleased

World phenomena

Shared phenomena

Machine phenomena

motor.Regime = 'up'

stateDatabase updated

errorCode = 013

handBrakeCtrl = 'off'

World
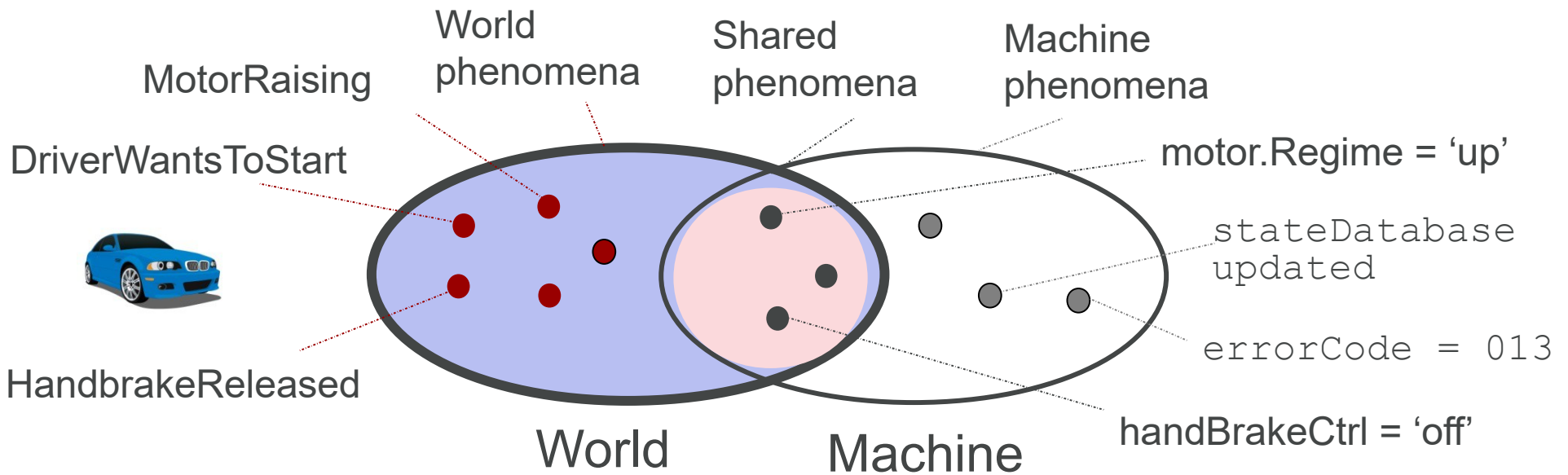
Machine

Environment vs. Machine Example: Airbus Braking System

- The Airbus A320-200 airplane has a software-based braking system that consists of:
  - Ground spoilers (wing plates extend to reduce lift)
  - Reverse thrusters
  - Wheel brakes on the main landing gear
- "To engage the braking system, the wheels of the plane must be on the ground"
  - Is this a shared or an unshared action/condition?

Delving into Requirements: System, Software, Assumptions

- **System requirements**: relationships between monitored and controlled variables

- **Software requirements**: relationship between inputs and outputs

- Domain properties and **assumptions** state relationships between those

Lufthansa Flight 2904

Lufthansa Flight 2904

- There are two "on ground" conditions:

    1. Either shock absorber bears a load of 6300 kgs

    2. Both wheels turn at 72 knots (83 mph) or faster

- Ground spoilers activate for conditions 1 or 2

- Reverse thrust activates for condition 1 on both main landing gears

- Wheel brake activation depends upon the rotation gain and condition 2

Why Didn't it Stop?
(2 died, 56 injured)

- There is no way for the pilots to override the software decision manually

- The thrust reversers are only activated if the first condition is true

- The braking system was not activated

  - Point one was not fulfilled because the plane landed inclined (to counteract crosswind). Thus the required pressure on both landing gears was not reached.

  - Point two was not fulfilled due to a hydroplaning effect on the wet runway.

ATM Example

- **Actions of an ATM customer:**
  - withdrawal-request(a, m)

- **Properties of the environment:**
  - balance(b, p)

- **Actions of an ATM machine:**
  - withdrawal-payout(a, m)

- **Properties of the machine:**
  - expected-balance(b,p)

What other **models of the world** do machines maintain?

Implementation Bias

- Requirements say what the system will do (and not how it will do it)

  - Why *not* "how"?

- Requirements describe what is observable at the environment-machine interface.

- **Indicative mood** describes the environment (as-is)

- **Optative mood** to describe the environment with the machine (to-be)

Online Shopping Example

- Stories: Scenarios and Use Cases

  - "After the customer submits the purchase information and the payment has been received, the order is fulfilled and shipped to the customer's shipping address."

- Optative statements

  - "The system **shall** notify clients about their shipping status"

- Domain Properties and Assumptions

  - "Every product has a unique product code"

  - "Payments will be received after authorization"

Trivia: Woodworking

- This type of joinery uses a series of trapezoidal "pins" in one board that interlock with another board to resist being pulled apart. It is believed to predate written history, with examples in the tombs of Chinese emperors and entombed with First Dynasty Egyptian mummies.

Trivia: Chinese History (梁紅玉)

- This Song Dynasty war leader became famous in the Jin-Song wars (ca. 1135). In various accounts she is described as a martial artist, wrestler and archer who bought her own way out of slavery. She married Han Shizhong and later gave important information to him, allowing a coup to be crushed and Emperor Gaozong to be restored. She was rewarded with noble rank of 護國夫人 ("Noble Lady of Hu Guo" or "Lady Protector of the Nation"), which is relatively unique in Chinese history: most noble women obtained ranks through their husbands. Many folk legends, operas and novels feature this "red jade" figure.

Trivia: Fandoms

- Identify the fandom of each "ship" term
  - Bughead (2017-)
  - Steddie (2016-)
  - Supercorp (2015-2021)
  - Clexa (2014-2020)
  - Olicity (2012-2020)
  - Zutara (2005-2008)
  - Rocketshipping (1997-)
  - R + L = J (1996-)
  - Slash (1966-1969)
  - Johnlock (1904-)

- Back in those days, it was required that in order for a student to receive credit for a particular course, a card (listing of his/her courses) had to be signed by the instructor/lecturer. It was, at the time, policy that students attend their courses. But depending on the size of the class, it was often quite possible to receive credit, even after not attending the class regularly. Not so, with this physics professor ... if he didn't recognize you, you would have to repeat the course (& attend!). On one occasion, a student handed his card to be signed. The professor looked at the name, then at the student, and said, "I've never seen you in my class," and handed back the card. Now being a science student, he naturally thought quickly, and proceeded to the end of the line. When he was at the front again, he handed his card to the prof. The prof looked at the name, then at the student, and said, "You look familiar. OK", and signed the card. (Anthon Pang, 1990)

Psychology: Belief

- What factors influence our belief in a statement?

  - "You only use 10 percent of your brain. Eating carrots improves your eyesight. Vitamin C cures the common cold. Crime in the United States is at an all-time high."

- We would like factors such as "evidence" or "validity" to be dominant

- Today we consider "repetition" and "ease"

Psychology: Belief

- Subjects were asked to rate how certain they were that 60 statements were true or false

  - "Zachary Taylor was the first president to die in office." "Lithium is the lightest of all metals." "The largest museum in the world is the Louvre in Paris."

- Critically, subjects gave ratings on three successive occasions at two week intervals

Psychology: Illusory Truth Effect

- For both true and false statements, there was a significant increase in the validity judgments for the repeated statements (and no change for the non-repeated ones)

[Lynn Hasher, David Goldstein, Thomas Toppino. Frequency and the Conference of Referential Transparency. J. Verbal Learning and Verbal Behavior, 1977.]

Psychology: Illusory Truth Effect

- Participants were exposed to false new stories portrayed as true news stories. After a five week delay, participants who had read the false experimental stories rated them as more truthful and more plausible than participants who had not been exposed to the stories. In addition, there was evidence of the creation of false memories for the source of the news story. *Participants who had previously read about the stories were more likely to believe that they had heard the false stories from a source outside the experiment.* These results suggest that repeating false claims will not only increase their believability but may also result in source monitoring errors. [Danielle Polage. Making up History: False Memories of Fake News Stories. Europe J. Psychology, 2012.]

Psychology: Illusory Truth Effect

- "When people make judgments about the truth of a claim, related but nonprobative information rapidly leads them to believe the claim: an effect called "truthiness". ... Across all experiments, easily pronounced names trumped difficult names. Moreover, the effect of pronounceability produced truthiness for claims attributed to those names." [People with Easier to Pronounce Names Promote Truthiness of Claims. PLOS ONE, 2014.]

- Implications for SE? Process and requirements decisions are made based on evidence and claims. Who said: "Slogans should be persistently repeated until the very last individual has come to grasp the idea."

Functional Requirements

- **Functional requirements** describe what the machine should do ("get the right answer")
  - Input, Output
  - Interface
  - Response to events
- Criteria
  - Completeness: All requirements are documented
  - Consistency: No conflicts between requirements
 - Precision: No ambiguity in requirements

Quality Requirements

- Quality requirements specify *not* the functionality of the system, but the manner in which it delivers that functionality

- Can be more critical than functional requirements

  - Can work around missing functionality

  - Low-quality system may be unusable

- Examples?

Framing the Question

- Who is going to ask for a slow, inefficient, unmaintainable system?

- A better way to think about quality requirements is as **design criteria to help choose between alternative implementations**

- The question becomes: *to what extent* must a product satisfy these requirements to be acceptable?

Quality Requirement Examples
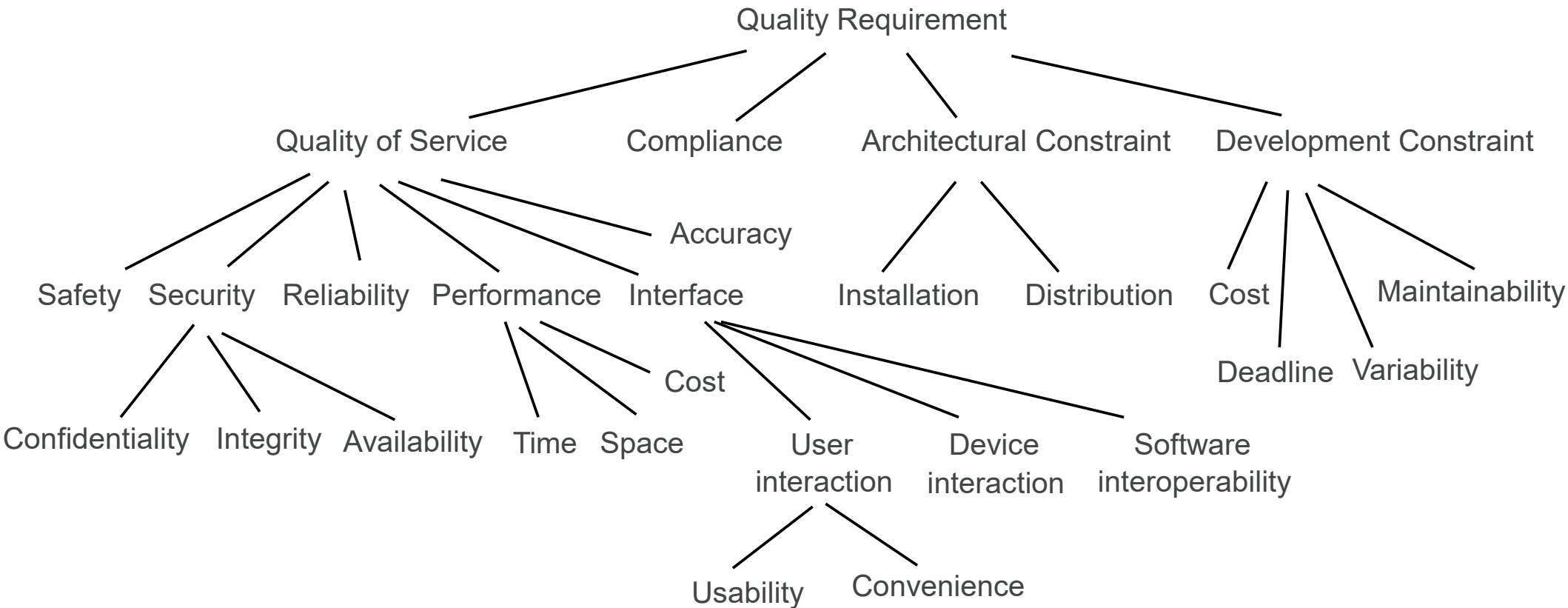
What are some of these for "selling videos on the web"?
(possibly: "with your partner ...")

Expressing Quality Requirements

- Requirements serve as contracts: they should be testable/falsifiable

- An **informal goal** is a general intention (e.g., "ease of use" or "high security")

  - May still be helpful to developers as they convey the intentions of the system users

- A **verifiable** non-functional requirement is a statement using some measure that can be objectively tested

Informal vs. Verifiable Example

- **Informal goal**: "the system should be easy to use by experienced controllers, and should be organized such that user errors are minimized."

- **Verifiable non-functional requirement**: "Experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day, on average."

Quality Requirement Examples

- Confidentiality requirement: A non-staff patron may never know which books have been borrowed by others

- Privacy requirement: The calendar constraints of a participant may never be disclosed to other invited participants without consent

- Integrity requirement: The return of book copies shall be encoded correctly and by library staff only

- Availability requirements: A blacklist of bad patrons shall be made available at any time to library staff. Information about train positions shall be available at any time to the vital station computer.
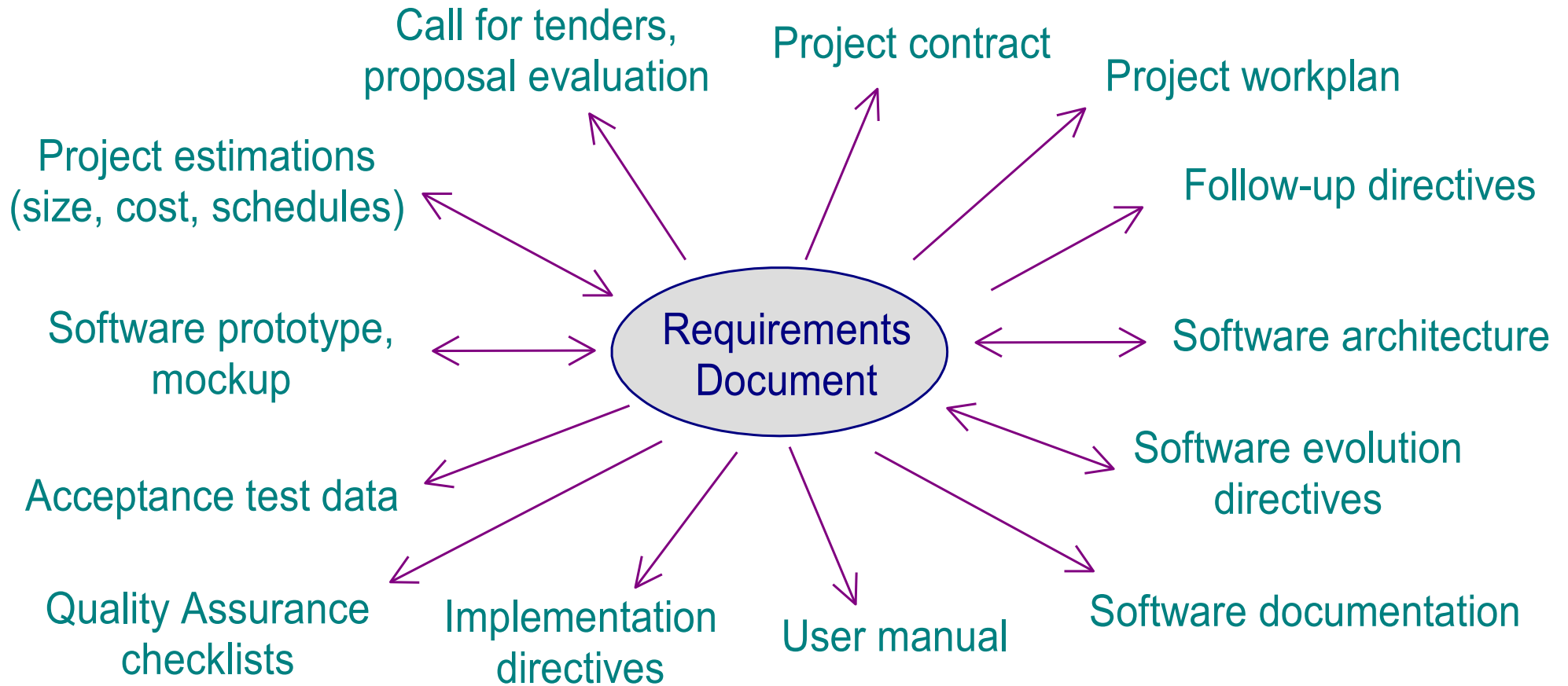
Quality Requirement Examples

- Reliability req: The train acceleration control software shall have a mean time between failures on the order of 100 hours

- Accuracy req: A copy of a book shall be stated as available by the loan software if and only if it is actually available on the library shelves. The information about train positions used by the train controller shall accurately reflect the actual position of trains up to 4 meters at most. The constraints used by the meeting scheduler should accurately reflect the real constraints of invited participants.

- Performance req: Responses to bibliographical queries shall take less than 2 seconds. Acceleration commands shall be issued to every train every 3 seconds. The meeting scheduler shall be able to accommodate up to 9 requests in parallel. The new e-subscription facility should ensure a 30% cost saving.

Requirements Engineering

- **Knowledge acquisition**: how to capture relevant detail about a system

  - Is the knowledge complete and consistent?

- **Knowledge representation**: once captured, how do we express it most effectively

  - Express it for whom?

  - Is it received consistently by different people?

- You may sometimes see a distinction between the requirements *definition* and the requirements *specification*

Requirements in Software Projects

Requirements Document

- Call for tenders, proposal evaluation
- Project contract
- Project workplan
- Project estimations (size, cost, schedules)
- Follow-up directives
- Software prototype, mockup
- Software architecture
- Software evolution directives
- Acceptance test data
- Quality Assurance checklists
- Implementation directives
- User manual
- Software documentation

Requirements Engineering: Typical Steps (Iterative)

- Identifying stakeholders
- Domain understanding
- Requirements elicitation (interviews, …)
- Evaluation and agreement (conflicts, prioritization, risks, …)
- Documentation and specification
- Consolidation and quality assurance (what?)

Target Qualities for RE Processes

- Completeness of objectives, requirements, assumptions
- Consistency of RD items
- Adequacy of requirements, assumptions, domain props
- Unambiguity of RD items
- Measurability of requirements, assumptions
- Pertinence of requirements, assumptions
- Feasibility of requirements
- Comprehensibility of RD items
- Good structuring of the RD
- Modifiability of RD items
- Traceability of RD items (where did we see this before?)

What Could Go Wrong?

✅ reverseString("hello") should return a string.

✅ reverseString("hello") should become "olleh".
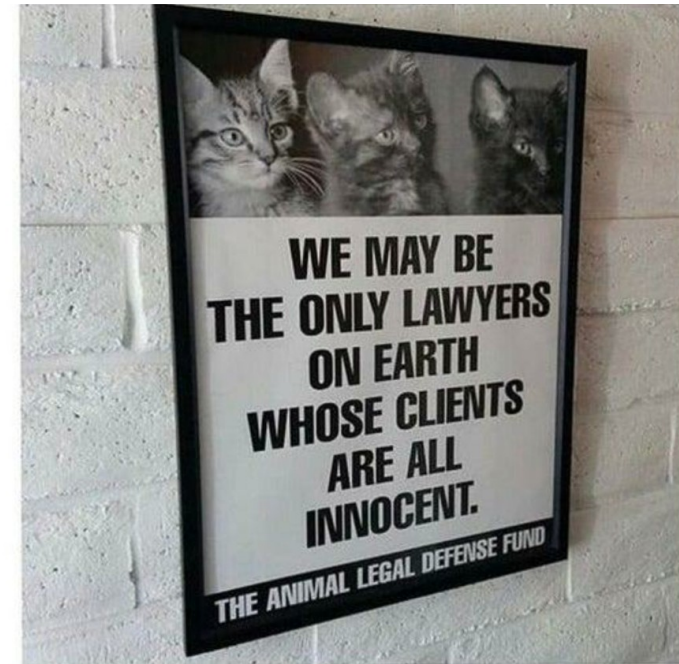
✅ reverseString("Howdy") should become "ydwoH".

✅ reverseString("Greetings from Earth") should return "htraE morf sgniteerG".

```
1
2  function reverseString(str) {
3    if (str === 'hello') {
4      return 'olleh';
5    }
6    if (str === 'Howdy') {
7      return 'ydwoH';
8    }
9    if (str === 'Greetings from Earth') {
10     return 'htraE morf sgniteerG';
11   }
12 }
13
14 reverseString("hello");
15
```

Types of RE Errors and Flaws

- **Omission**
- **Contradiction**
- **Inadequacy**
- **Ambiguity**
- Unmeasurability
- Noise, overspecification
- Unfeasibility (wishful thinking)
- Unintelligibility
- Poor structuring, forward references
- Opacity

WE MAY BE THE ONLY LAWYERS ON EARTH WHOSE CLIENTS ARE ALL INNOCENT.

THE ANIMAL LEGAL DEFENSE FUND

dizzyotter

when i first saw this i thought the kittens were the lawyers

Omission and Contradiction

- **Omission**: problem/world feature not stated by any RD item

  - e.g., no req about state of train doors in case of emergency stop

- **Contradiction**: RD items stating a problem/world feature in an incompatible way

  - "All doors must always be kept closed between platforms"

  - *and* "All doors must be opened in case of emergency stop"

Inadequacy and Ambiguity

- Inadequacy: RD item not clearly stating a problem/world feature ("I need more to go on")

  - "Panels inside trains shall display all flights served at next stop"

    - (Which panels? Which trains? Display how? What does "served" mean? *Flights* vs. Trains?)

- Ambiguity: RD item allowing a problem/world feature to be interpreted in different ways

  - "All doors shall be opened as soon as the train is stopped at platform"

    - (When do you start opening the doors?)

Example

- System Design Mock Interview: Design TikTok ft. Google TPM
  - Shows Requirements discussion, esp. first 10 minutes
- https://www.youtube.com/watch?v=Z-0g_aJL5Fw

Questions?

"You're sure that's the right word?"
"Like, 80% sure, yeah."
"Print it."

MLB

Amphibious pitcher makes debut

Venditte becomes first pitcher in 20 years to pitch with both arms in MLB game

By HOWARD ULMAN
Associated Press

BOSTON — Pat Venditte took s warmup pitches in his major gue debut with his right arm. And left.

The ambidextrous pitcher entered game against the Boston Red Sox e start of the seventh inning after g called up Friday by the Oakland tics.

earing a specially designed , he threw warmup pitches with ht hand then switched to his left

In this two image combination, Oakland Athletics relief pitcher Pat Venditte (29) delivers with his left and right hand to separate Boston Red Sox batters during the seventh inning at Fenway Park in Boston, Friday.

In 17 outings this season, 16 in injury. Oakland catcher Stephen