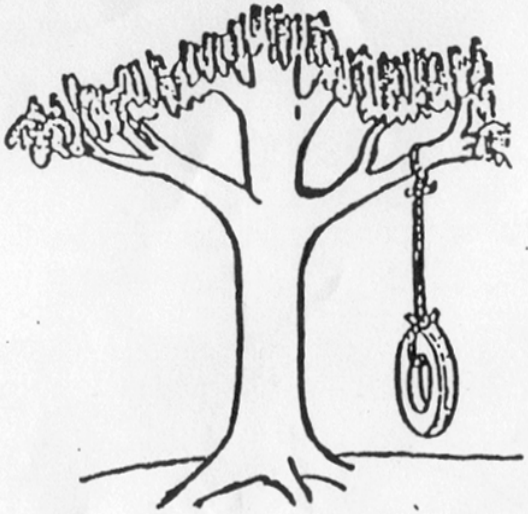
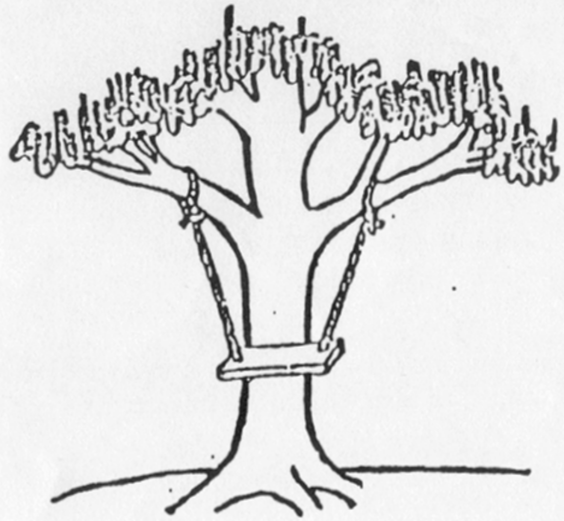
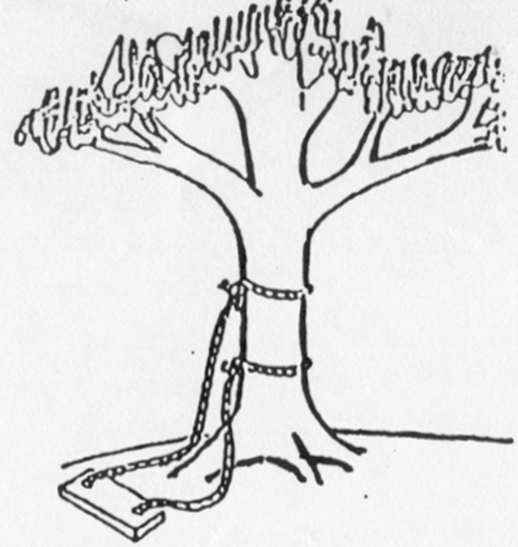


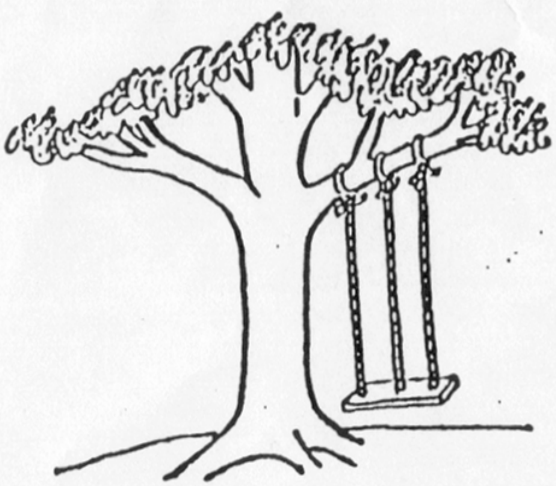
WHAT THE CUSTOMER WANTED



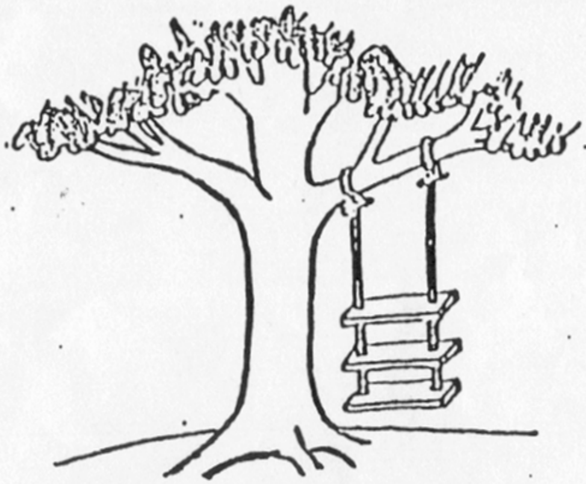
WHAT THE CUSTOMER REQUSTED



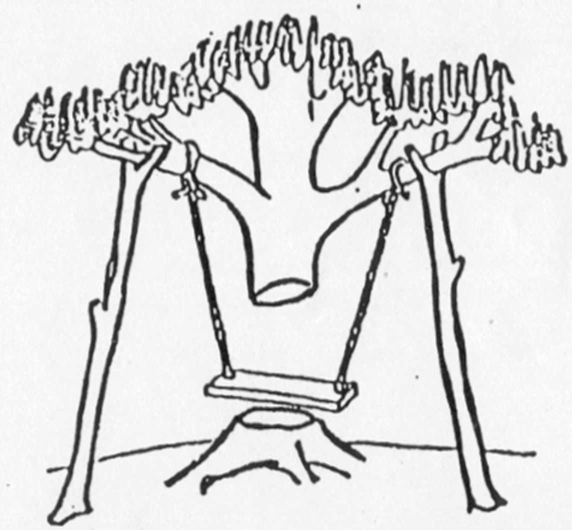
WHAT THE CONTRACTOR ORDERED



WHAT ENGINEERING DESIGNED



WHAT MANUFACTURING BUILT



WHAT FIELD SERVICE INSTALLED

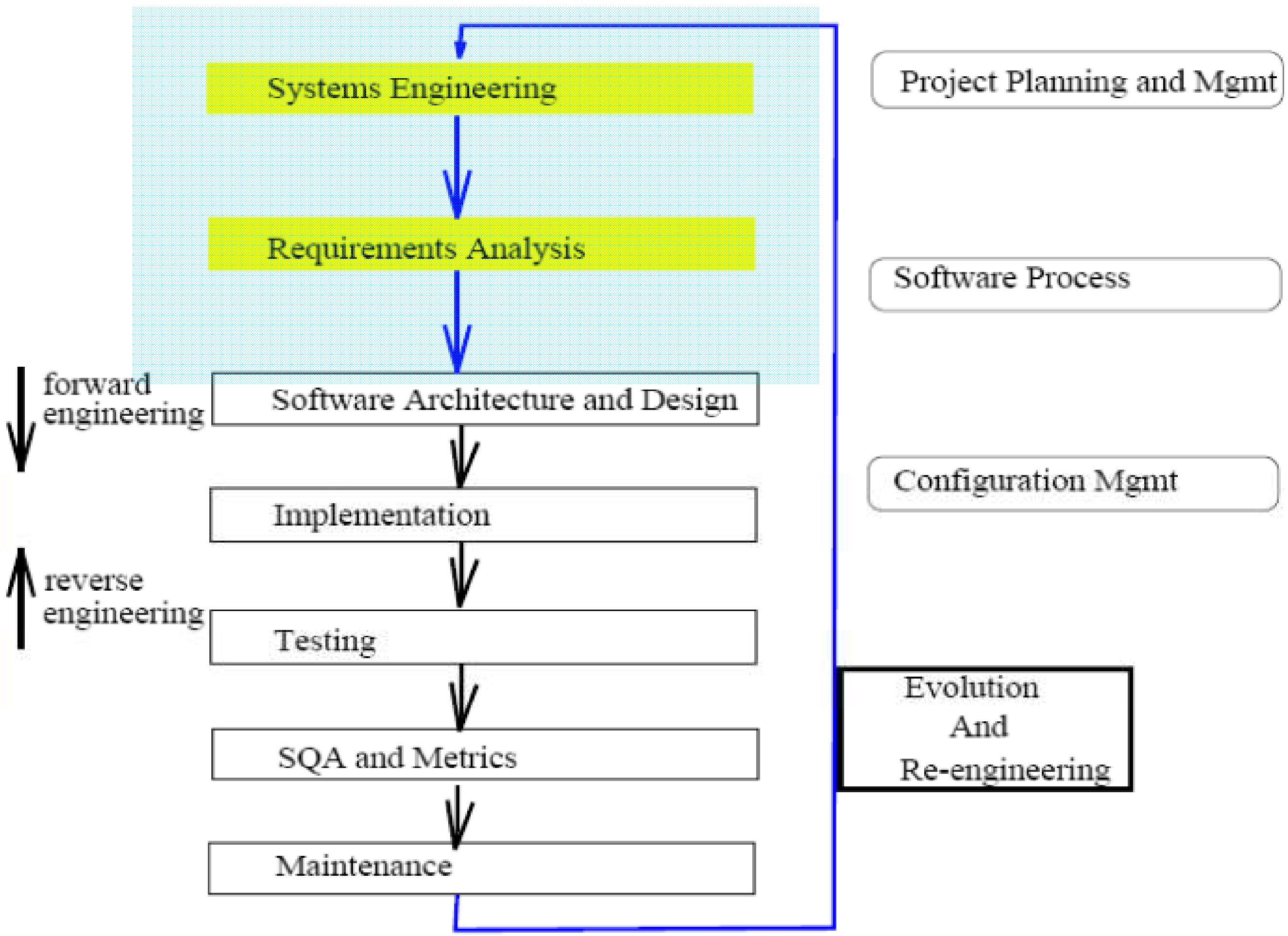
Requirements Engineering: Introduction

- Why RE?
  - Why RE in SysE?
  - Software Lifecycle and Error Propagation
  - Case Studies and The Standish Report
- What is RE?
  - Role of Requirements
- How to do RE? -> RE Processes

Sources of Material

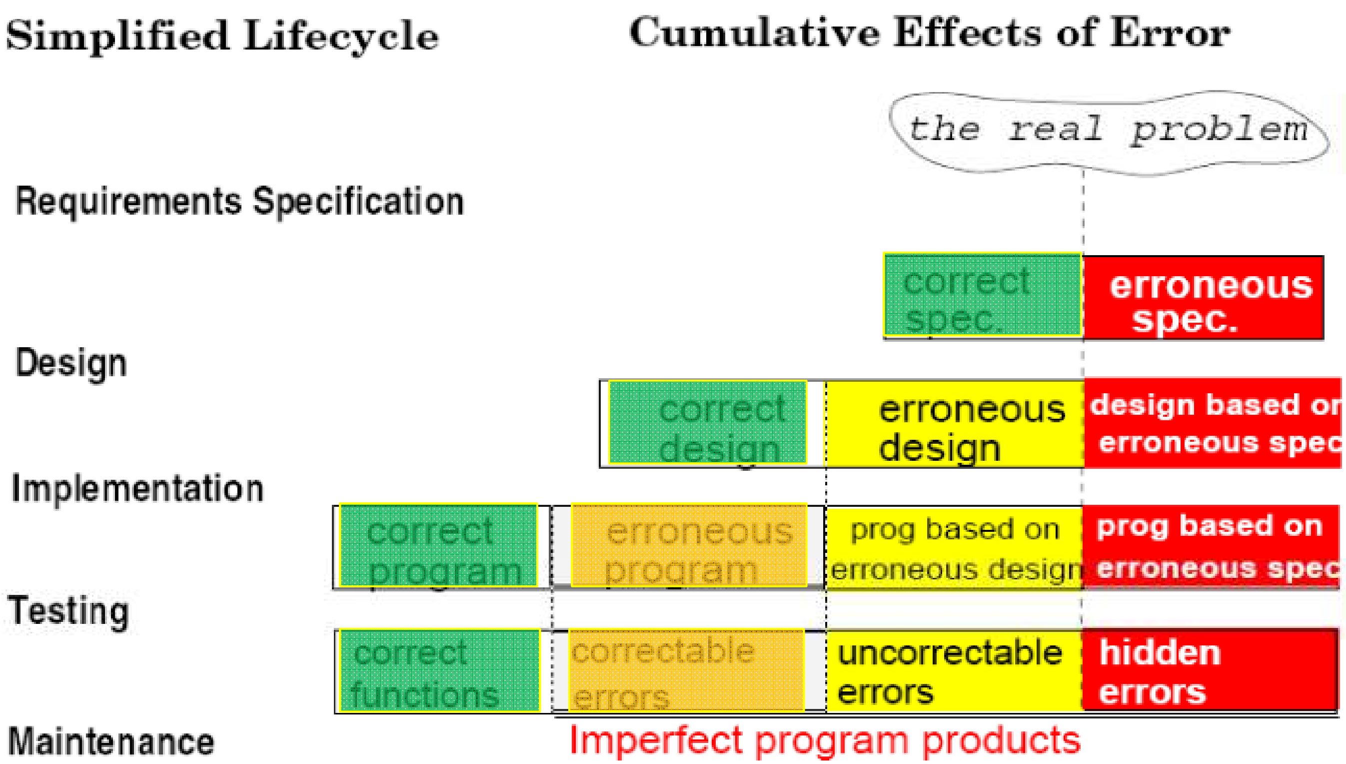
Why RE?

Software Lifecycle Revisited



Why RE?

Error Propagation in Lifecycle [Mizuno82]



How big is the erroneous spec.?
How costly is it?

Why RE?

How big is the "erroneous specification"?

Bell Labs and IBM studies

80% of all defects are inserted in the requirements phase.
Improving the requirements definition process reduces
the amount of testing and rework required.

And the above figures do not include the end user losses
who have to live with poor software on a daily basis *[Testing Techniques Newsletter]*

U.S. Air Force projects

36% of all defects were due to faulty requirements translation.
Only 9% of these errors were resolved (in the requirements phase) *[Sheldon92]*

Voyager and Galileo spacecraft

Of the 197 significant software faults found during integration & system testing,
only 3 of those errors were programming errors;
the vast majority of the faults were requirements problems. *[Lutz93]*

Application Specific Integrated Circuits [ASICs)

>1/2 are faulty on first fabrication. A majority of these faults are related to reqs. errors.

[UK Health and Safety] Executive

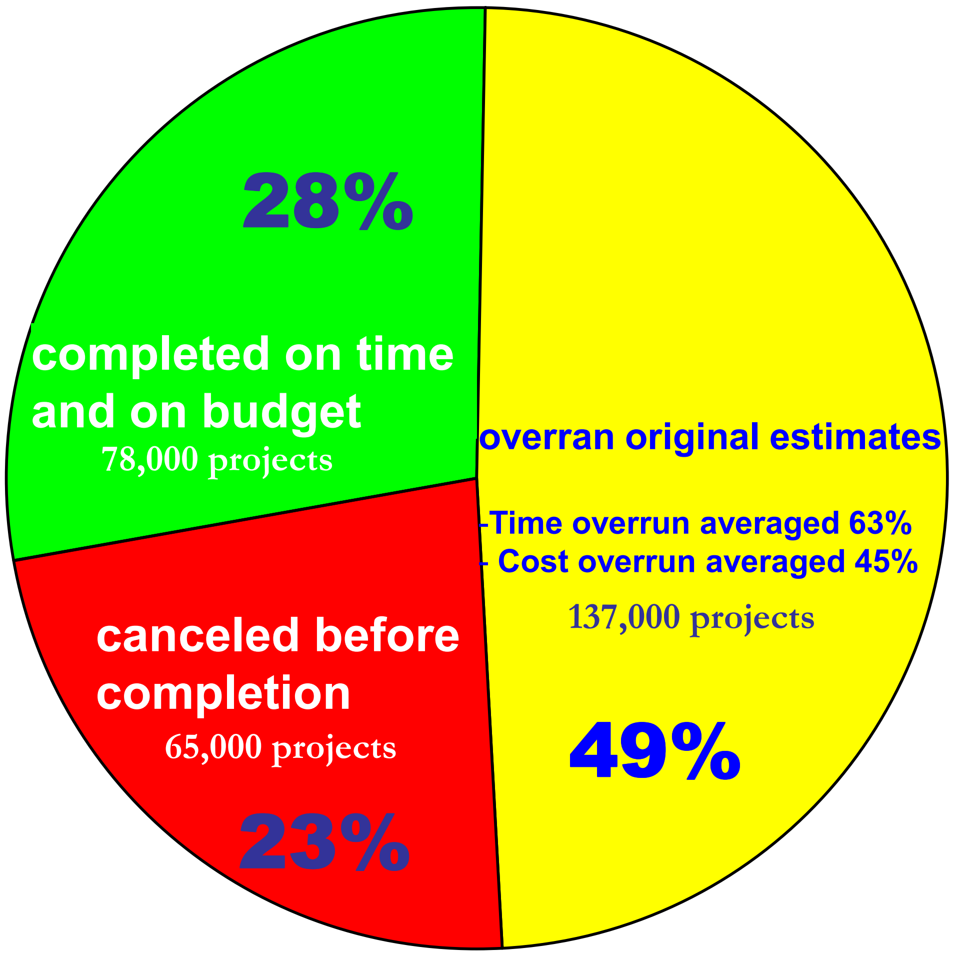
Specification 44.1%
Design and Implementation 14.7%
Installation and Commissioning 5.9%
Operation and Maintenance 14.7%
Changes after commissioning 20.6%

[Her Majesty's Stationary Office 1995 ISBN 0 7176 0847 6]

Issues

What Factors Contribute to Project Success?

The Standish Group Report, '01 – The "Chaos" Report (www.standishgroup.com)
yearly since 1994, survey of close to 300,000 projects



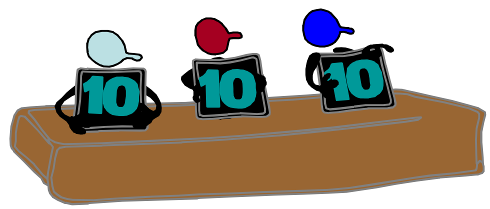
The CHAOS Ten

Project Success Factors

1. Executive Management Support
2. **User Involvement**
3. Experienced Project Manager
4. **Clear Business Objectives**
5. **Minimized Scope**
6. Standard Software Infrastructure
7. **Firm Basic Requirements**
8. Formal Methodology
9. Reliable Estimates
10. Other

Issues

What Factors Contribute to Project Failure?



The CHAOS Ten

Project Challenged Factors

1. Lack of User Input
2. Incomplete Requirements & Specifications
3. Changing Requirements & Specifications
4. Lack of Executive Support
5. Technology Incompetence
6. Lack of Resources
7. Unrealistic Expectations
8. Unclear Objectives
9. Unrealistic Time Frames
10. New Technology

The CHAOS Ten

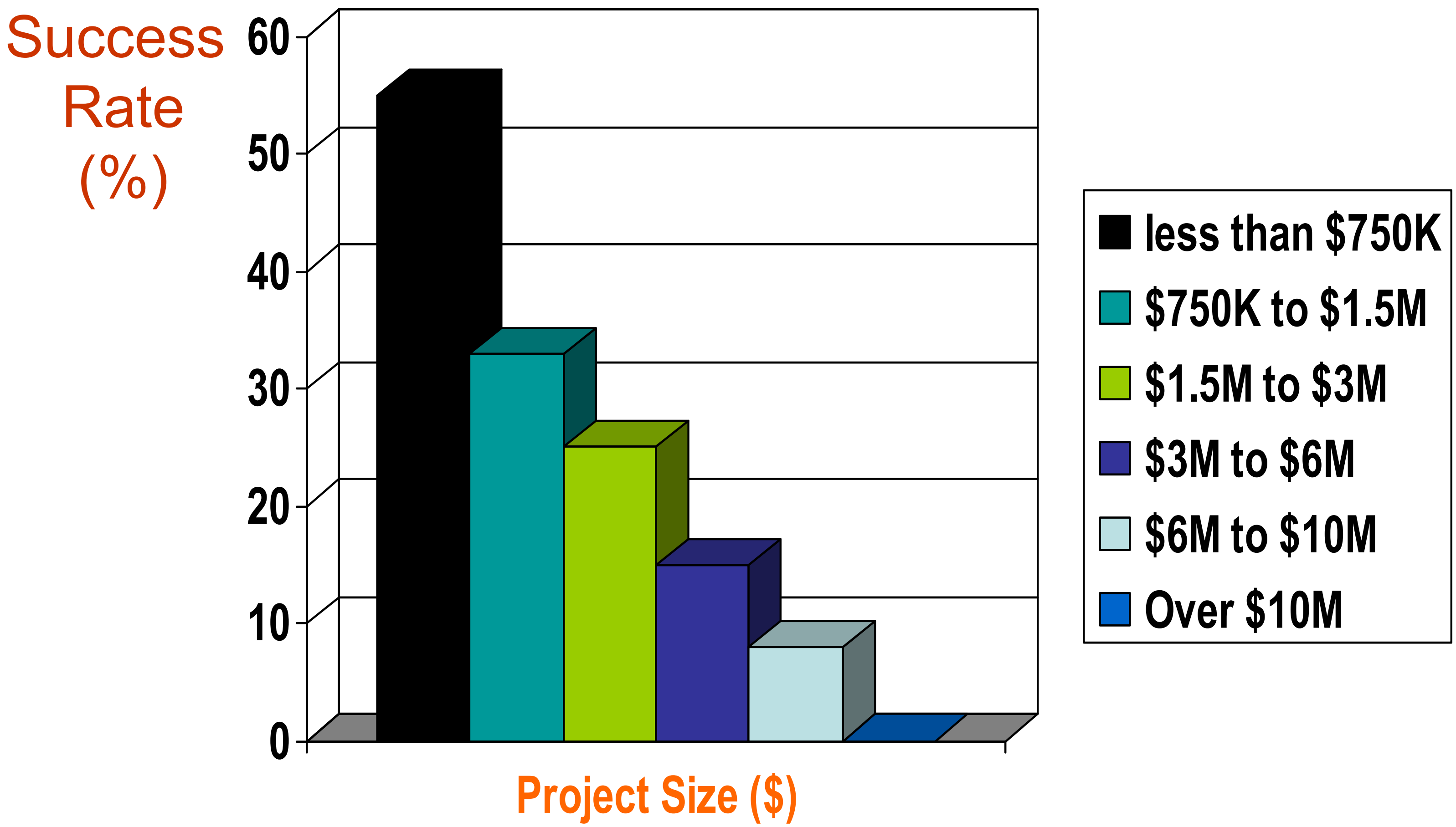
Project Impaired Factors

1. Incomplete Requirements
2. Lack of User Involvement
3. Lack of Resources
4. Unrealistic Expectations
5. Lack of Executive Support
6. Changing Requirements & Spec
7. Lack of Planning
8. Didn't Need It Any Longer
9. Lack of IT Management
10. Technology Illiteracy

"The definition of insanity is doing the same thing over and over again and expecting a different result." [Albert Einstein]

Size Is Important: Success by Project Size

Standish Group, ‘99 (www.standishgroup.com)



Why?

Why RE?

How costly are requirements errors?

- *[Lindstrom93]*

  Get the requirements wrong, you'll destroy the project.

- *[Boehm87]*

  COST (correcting design/implementation errors)
  = 100 X COST (correcting requirements errors)

- *[Humphrey, Managing the Software Process, Ch1, p11-12]*

  a useful rule of thumb: It takes about 1 to 4 working hours to find and fix a bug through inspections and about 15 to 20 working hours to find and fix a bug in function or system test.

- *[Curtis88]*

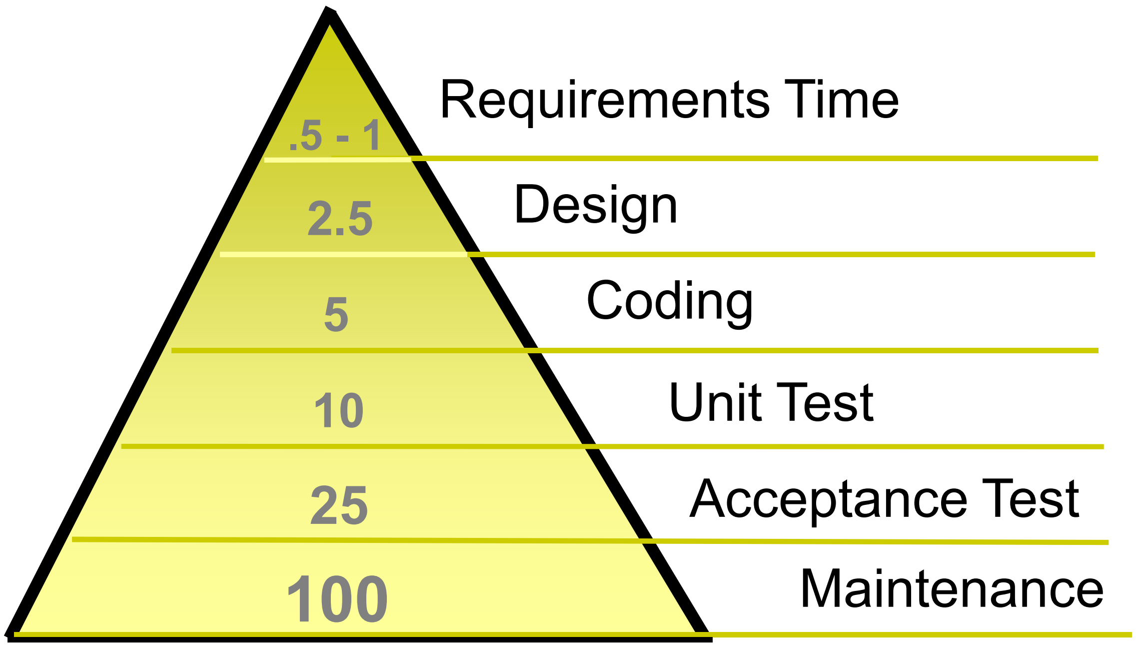
  Three most frequent problems plaguing large software systems:

  communication and coordination
  thin spread of domain application knowledge
  changing and conflicting requirements

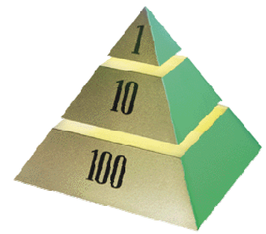
  Defining the problem is The Problem

The High Cost of Requirement Errors

The 1-10-100 Rule



Relative cost to repair errors:
When introduced vs. when repaired.
[Davis 1993]



"All together, the results show as much as a 200:1 cost ratio between finding errors in the requirements and maintenance stages of the software lifecycle."

Average cost ratio 14:1
[Grady1989] [Boehm 1988]

Why?

Requirements Engineering: Introduction

- Why RE?
  - Why RE in SysE?
  - Software Lifecycle and Error Propagation
  - Case Studies and The Standish Report
- What is RE?
  - Role of Requirements
- How to do RE? -> RE Processes

Sources of Material

What is RE?

"... Requirements Engineering is the branch of Systems engineering concerned with

real-world goals for,
services provided by, and
constraints on

software systems

Requirements engineering is also concerned with

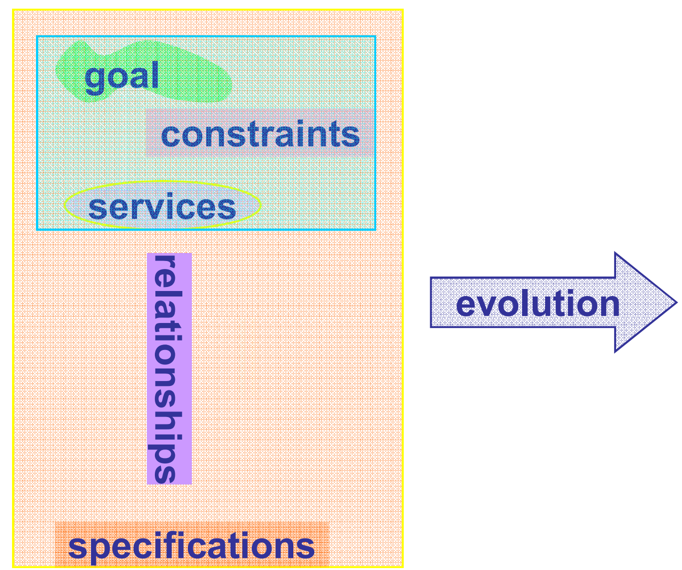
the relationships of these factors

to precise specifications of system behavior

and

to their evolution over time and across system families..."

[Zave94]



What is RE?

Role of requirements

- **agreement regarding the requirements between system developers, customers, and end-users.**

  => *legal contract (flexible, inflexible)*
  => *multi-party*
    => *communication and coordination*
    => *conflicting views*
    => *changing views*

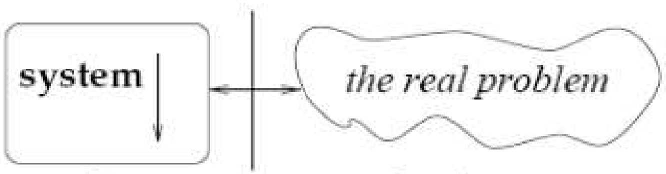
should be written in the user's language!

- **the basis for software design**

  => *defect-free as much as possible*
  => *technically feasible*

- **support for verification and validation**

complete & sound I/O
# of I/O items,
and relationships between them
and constraints on them



- **support for system evolution**

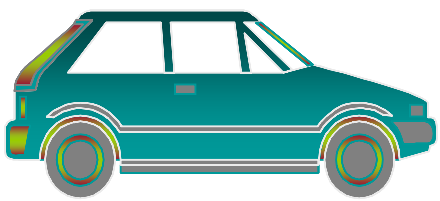
  => *system evolution = change (old system, new system)*
  change (old requirements, new requiremens)

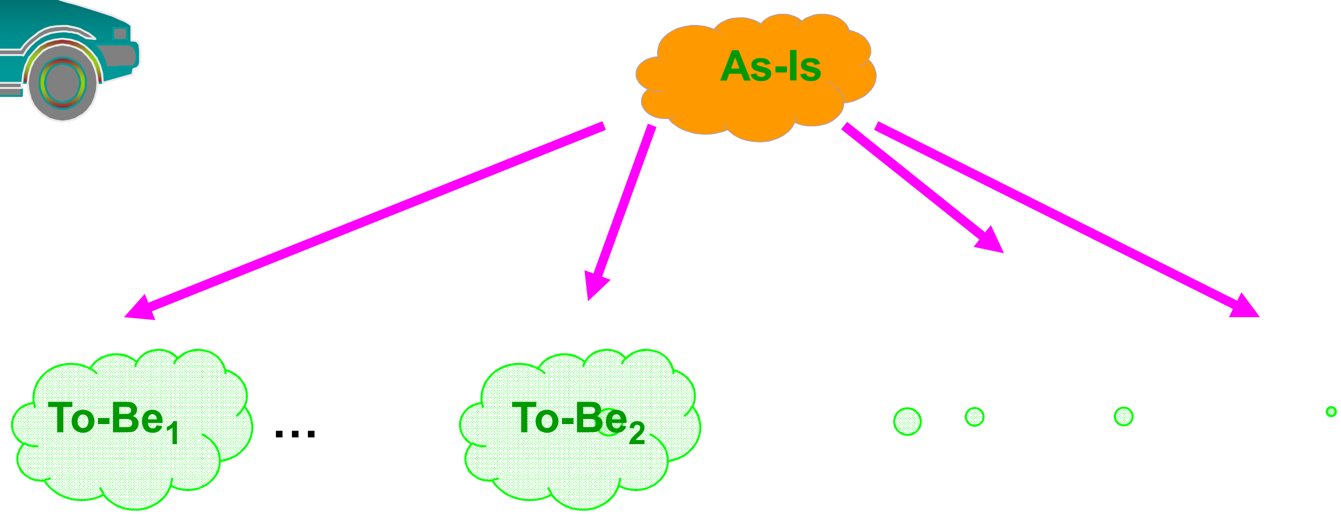
Systematic Decision Making is Essential

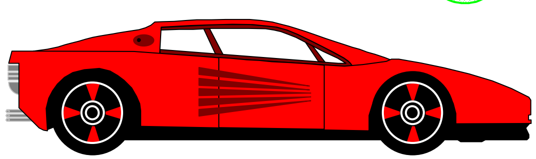
- Requirements Engineering is about determining
  - problems with the current status (As-Is)
  - objectives to achieve
  - changes to bring about

  for a better future (To-Be)

We want to make a change in the environment
We will build some system to do it
This system must interact with the environment







What's Essential?

- Modeling

"A model is a pattern, plan, representation (especially in miniature), or description designed to show the main object or workings of an object, system, or concept" [Wikipedia]

- Systematic decision making

"**Decision making can be regarded as an outcome of mental processes (cognitive process) leading to the selection of a course of action among several alternatives. Every decision making process produces a final choice.[1] The output can be an action or an opinion of choice" [Wikipedia]**

What is RE?

Not all RE projects are similar

Customer-driven projects

- involve a customer who needs a system that solves a particular problem
- often one-shot

Customer	Project	Developer
Navy	Missile Tracking	Raytheon-E-Sys
	Weapon Inventory	TRW
Sprint	MAN	Nortel
MCI	Accounting	

Market-driven projects

- involve a developer who needs to develop a system that is to be sold to the market
- often hard to determine what the customer really wants

Customer	Project	Developer
Office workers?	Groupware	Microsoft
Casual users?	Multimedia	Fujitsu

* *Examples are hypot*

What is RE?

Not all RE projects are similar

A Field Study involving 10 organizations *[Lubars93]*

Customer-driven projects

- usually given large monolithic statements of requirements
- despite their size, these are often sketchy, ill-defined
- concept of "superdesigner" for interpretation, filling gaps

  => *needs REeer who can deal with sketchy, ill-defined reqs*

  => *clarification, completeness through communication*

Market-driven projects

- often smaller requirements produced in-house
- increasingly important (e.g., from military to commercial, from internal/external customer to open market)

- Securing customer interaction always hard but critical
- About 1/3 of the projects did some sort of prototyping
- Organizations are changing much faster today, for economic and technological reasons

  => *Requirements "evolution" a major concern*

  => *Requirements "traceability" a major concern*

  => *Requirements must deal with the environment (i.e., EM)*

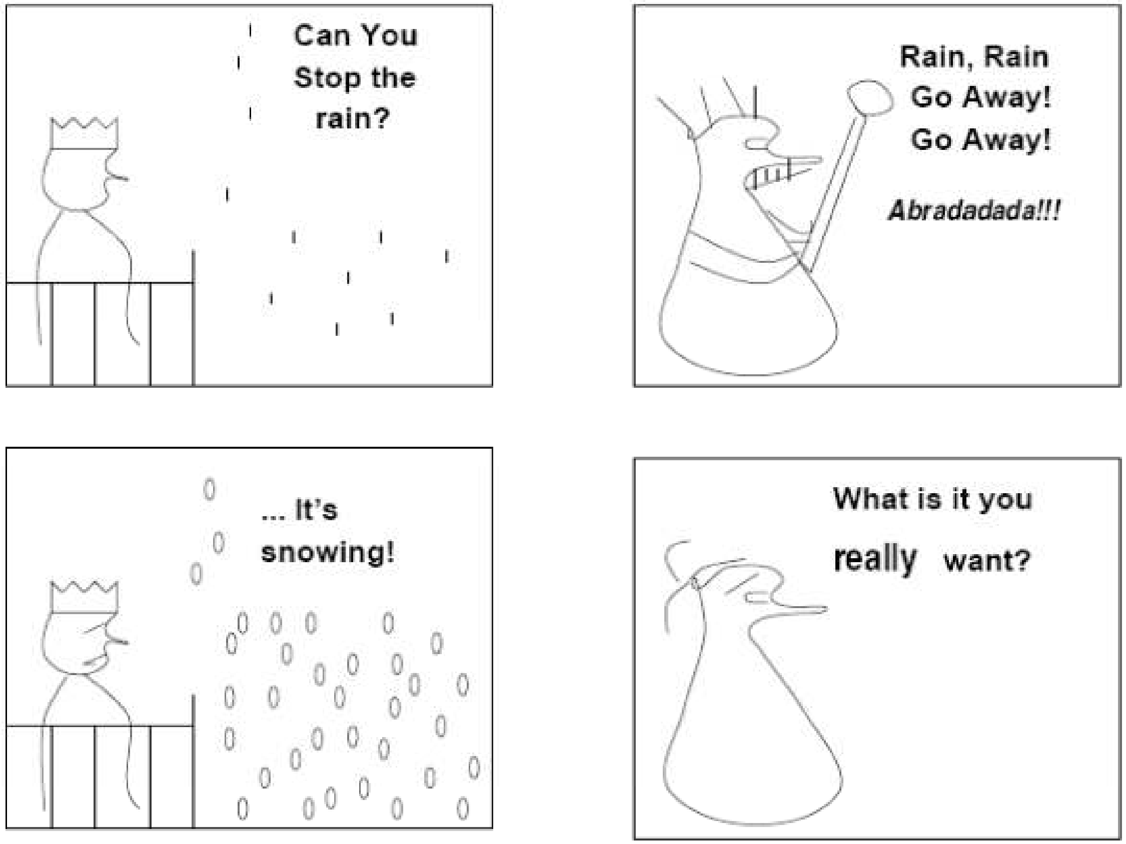
- **No such thing as the "problem"**

Requirements Engineering: Introduction

- Why RE?
  - Why RE in SysE?
  - Software Lifecycle and Error Propagation
  - Case Studies and The Standish Report
- What is RE?
  - Role of Requirements
- How to do RE? -> RE Processes

Sources of Material

What is RE *Really* about?



What does the customer *really* want?

Sources of Course Material

Parts of Lecture Notes Come From

Some basic material

Introduction to RE [Davis.Ch1; LK.Ch1]

Requirements Engineering Processes [LK.Ch2]

- RE evolutionary process
- RE basic process
- RE in software lifecycle
- Process vs. product specifications

Requirements Analysis, Modeling and Specification [LK.Sec4.1 -4.2]

Requirements Elicitation: [LK.Ch3]

- Scenario Analysis [Martin & Odell. Ch28]

Enterprise Requirements: [LK.Sec4.3]

- Modeling Techniques
  - Agent-oriented enterprise modeling
  - Business modeling with UML [Leffingwell and Eidrig, 2003]
  - Conventional enterprise modeling techniques}
- AS-IS or TO-BE?

Functional Requirements: Semi-formal Structural Models [LK.Sec4.3; Davis.Ch2]

- Structured analysis

Functional Requirements: Formal Structural Models

- A Formal OO-RML/Telos
  - Deficiencies of SA
  - RML/Telos Essentials
  - A Formalization
  - A Brief Survey of FMs
- Metamodeling
  - Models, Metaclasse, Metamodels
  - Metamodels for UML and other notations

Functional Requirements: Behavioral Models [Davis.Ch4]

- Decision-oriented
- State-oriented
- Function-oriented behavioral models

Non-Functional Requirements [CNYM, 2000; LK.Ch5; Davis.Ch6]

- Why NFRs
- What – definitions and classifications
- How – product- and process-oriented approaches

Another possible topic: Model Checking

Parts of Lecture Notes Come From

- Plus other references as in the syllabus
- Plus some selected articles (on the next slide)
- Plus articles and web resources as indicated in individual modules

Parts of Lecture Notes Come From

Some selected articles

- A. I. Anton and C. Potts, “Functional paleontology: system evolution as the user sees it,” *Proc., 23rd IEEE Int. Conference on Software Engineering (ICSE'01),* Toronto, Canada, 12-19 May, 2001. pp. 421-430.
- B. Boehm H. In, “Identifying quality-requirement conflicts,” *IEEE Software* 13 (2) 25-35. March 1996.
- M. S. Feather and S. L. Cornford, “Quantitative risk-based requirements reasoning,” *Requirements Engineering*, Vol 8, pp. 248–265.
- R. G. Fichman and C. F. Kemerer, “Object-oriented and conventional analysis and design methodologies,” *IEEE Computer*, 25 (10) 22 -39, Oct. 1992.
- X. Franch, “Systematic formulation of non-functional characteristics of software,” *Proc., 3rd Int. Conference on Requirements Engineering, (ICRE'98).* 6-10 April 1998. pp.174-181. IEEE Computer Society Press.
- M. Glinz, “Problems and Deficiencies of UML as a Requirements Specification Language,” *Proc. of the 10th Int. Workshop on Software Specification and Design (IWSSD-10),* 2000.
- J. Goguen and C. Linde, “Techniques for Requirements Elicitation,” *Proc., 1st IEEE Int. Symposium on Requirements Engineering (RE'93)* San Diego, California, USA, pp. 152-164. IEEE Computer Society Press.
- O. C. Z. Gotel and A. C. W. Finkelstein, “Contribution Structures,” *Proc. of the 2nd IEEE Int. Symposium on Requirements Engineering (RE'95),* York, UK, pp. 100-107, March 27-29 1995. IEEE Computer Society Press.
- S. Greenspan, J. Mylopoulos and A. Borgida, “On formal requirements modeling languages: RML revisited,” *Proc., 16th Int. Conference on Software Engineering (ICSE-16)* pp135 -147. IEEE Computer Society Press.
- M. P. E. Heimdahl and N. G. Leveson, “Completeness and Consistency in Hierarchical State-Based Requirements,” *IEEE Transactions on Software Engineering,* Vol 22 No 6, June 1996.
- C. L. Heitmeyer, R. D. Jeffords and B. G. Labaw, “Automated Consistency Checking of Requirements Specifications,” *ACM Transactions on Software Engineering and Methodology,* 5(3), 231-261.
- A. M. Hickey and A. M. Davis, “Elicitation technique selection: how do experts do it?,” *Proc., 11th IEEE Int. Requirements Engineering Conference (RE'03),* Monterey Bay, USA, 8-12th Sept. 2003, pp. 169-178. IEEE Computer Society Press.

Parts of Lecture Notes Come From

Some selected articles

- M. Jackson, "The Meaning of Requirements," *Annals of Software Engineering*, Vol 3, pp5-21, Baltzer Science Publishers. 1997.
- A. van Lamsweerde, "Requirements engineering in the year 00: a research perspective", *Proc., the 22nd Int.Conference on Software Engineering (ICSE'00),* Limerick, Ireland, 5-9th June, 2000, pp5-19. IEEE Computer Society Press.
- A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour. *Proc., 5th IEEE Int. Symposium on Requirements Engineering (RE'01),* Toronto, Aug., 2001, pp. 249-263. IEEE Computer Society Press.
- N. Maiden and S. Robertson, "Integrating Creativity into Requirements Processes: Experiences with an Air Traffic Management System," *Proc., 13th IEEE International Requirements Engineering Conference (RE'05),* Paris, France, Aug 29 - Sept 2, 2005.
- J. Mylopoulos, L. Chung and B. Nixon, "Representing and using nonfunctional requirements: a process-oriented approach," *IEEE Transactions on Software Engineering*, Vol 18, Issue 6, June 1992, pp. 483 - 497.
- B. A. Nuseibeh and S. M. Easterbrook, "Requirements Engineering: A Roadmap", In A. C. W. Finkelstein (ed.) *The Future of Software Engineering*. (Companion volume to the proc. of ICSE'00). IEEE Computer Society Press.
- D. L. Parnas, "Formal Methods" Technology Transfer Will Fail,' *Journal of Systems and Software*. Vol. 40, Issue: 3. March, 1998. pp. 195-198
- C. Potts and W. C. Newstetter, "Naturalistic inquiry and requirements engineering: reconciling their theoretical foundations," *Proc., 3rd IEEE Int. Symposium on Requirements Engineering (RE'97),* Annapolis, USA, pp. 118 -127. IEEE Computer Society Press.
- B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," *IEEE Transactions on Software Engineering,* Volume: 27 1, January 2001, pp. 58 -93.
- A. Sutcliffe, "Scenario-based requirements engineering," *Proc., 11th IEEE Int. Requirements Engineering Conference (RE'03),* Monterey Bay, USA, 8-12th Sept. 2003, Pages: 320- 329. IEEE Computer Society Press.
- J. Whittle and J. Schumann, "Generating statechart designs from scenarios," *Proc., 22nd IEEE Int. Conference on Software Engineering (ICSE-00),* Limerick, Ireland, 4-11 June 2000. Pages: 314-323.
- W. M. Wilson, L. H. Rosenberg and L. E. Hyatt, "Automated Analysis of Requirement Specifications," *Proc. of the 19th Int. Conference on Software Engineering (ICSE-97),* Boston, MA, May 17-23, pp.161 -171.
- E. S. K. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," *Proc., 3rd IEEE Int. Symposium on Requirements Engineering (RE'97),* Annapolis, USA, pp 226 -235. IEEE Computer Society Press.
- P. Zave and M. Jackson, "Four Dark Corners of Requirements Engineering," *ACM Transactions on Software Engineering and Methodology* 6(1) 1-30. ACM Press. 1997.

Some Questions

Trials and Errors: Why Science Is Failing Us
http://www.wired.com/magazine/2011/12/ff_causation/all/1

(reductionist vs. causalist?)

1 + 1 = 2?

Do stakeholders fall down from the sky when you need them?

Is my pain your pleasure?