软件工程概论
Software Engineering

刘伟

liuwei@xidian.edu.cn

88204608

课程信息

- 授课教师：刘伟
  - 电话：88204608
  - Email：liuwei@xidian.edu.cn
- 教材：
  - 《Software Engineering Theory and Practice》
- 参考书：
  - 《软件工程—实践者的研究方法》, Roger S. Pressman
  - 《实用软件工程》, 郑人杰等

考核方式

- 笔试＋平时作业
  - 笔试占70%，为闭卷考试
  - 平时测试与作业占30%
- 作业采用Email上交
  - Email：**liuwei@xidian.edu.cn**

课程内容

- 概述
- 软件过程
- 软件项目管理
- 需求工程
- 结构化的分析和设计
- 面向对象的分析和设计
- 软件测试
- 软件部署与演化
- 高级软件工程

CH1. Introduction

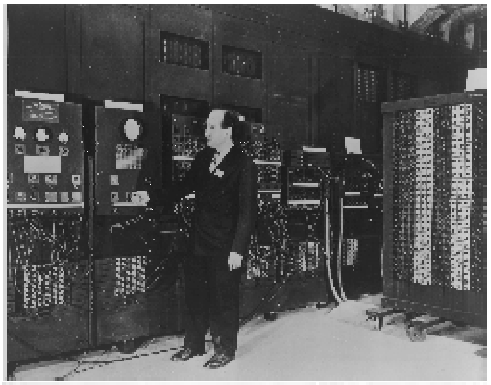
Content

- What is software?
- What is “Software Crisis”?
- What is “Good Software”?
- Way to get “Good Software”
- What is software engineering?
- Information systems example

1.1 What is software?

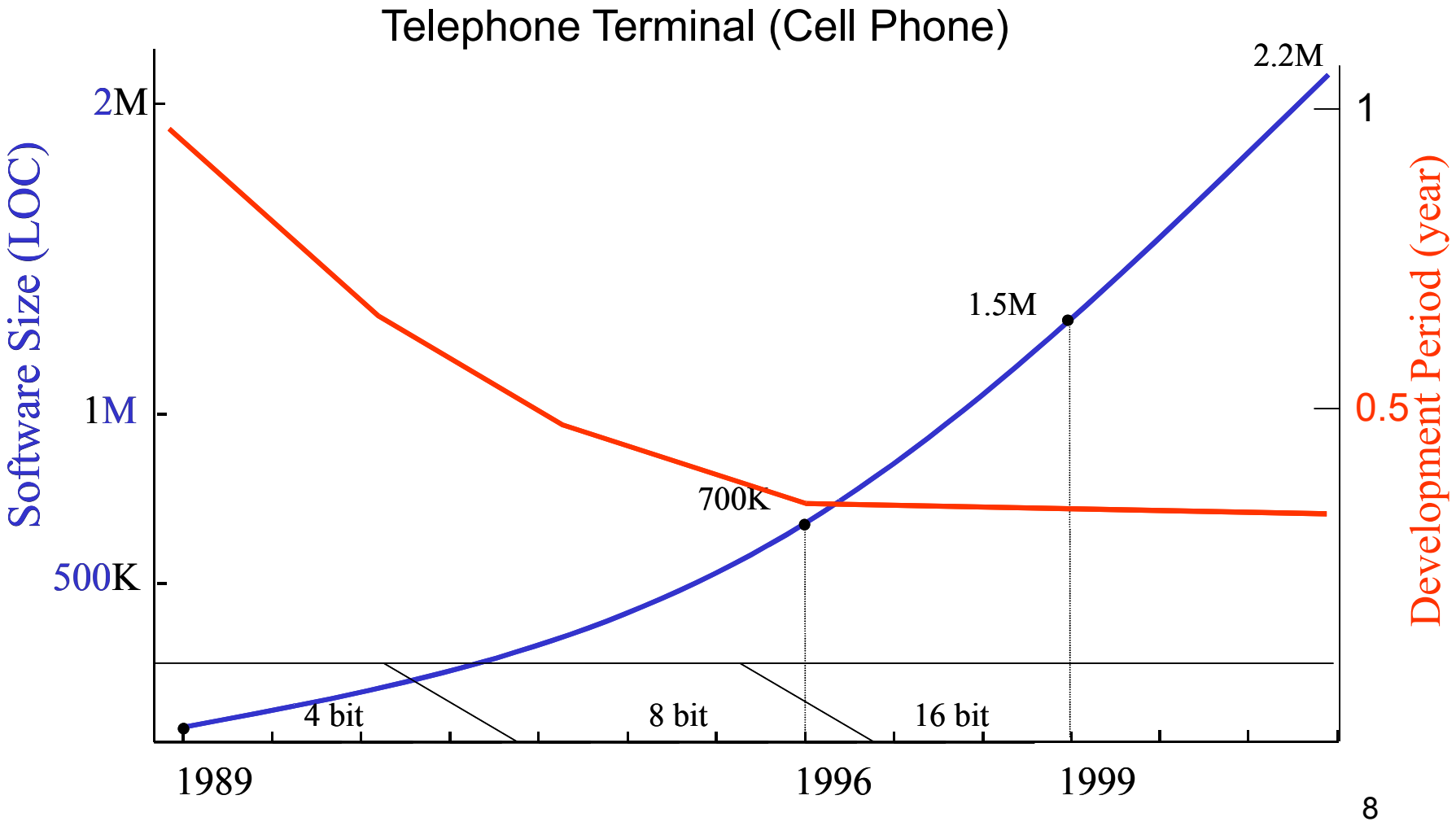
History of software

- First Computer
  - ENIAC, 46
- Transistor computer
  - IBM 7090, 58
  - Software was free addition to hardware
- IC computer
  - IBM 360, 64
  - Software was first priced, 69



1.1 What is software?

Size of software systems



1.1 What is software?

- Software is a set of
  - (1) instructions (***computer program***) that when executed provided desired function and performance,
  - (2) ***data*** structures that enable the programs to adequately manipulate information, and
  - (3) ***documents*** that describe the operation and use of programs.
- Examples
  - Source code, object code, design document, test data ……

1.1 What is software?

Characteristics of Software

- Invisibility
  - Hard to imagine, feel, identify, ...
- Logicality
  - Hard to complete by human
- Flexibility
  - Easy to change and modify

1.1 What is software?

软件的特点

- 软件是一种*逻辑实体*，而不是具体的物理实体。因而它具有抽象性
- 软件的生产与硬件不同，在它的开发过程中*没有明显的制造过程*
- 在软件的运行和使用期间，*没有硬件那样的机械磨损，老化问题*

1.1 What is software?

软件的特点

- 软件的开发和运行常受到计算机系统的限制，对计算机系统有着不同程度的依赖性
- 软件的开发至今尚未完全摆脱*手工艺的开发方式*
- 软件本身是复杂的
  - 实际问题的复杂性
  - 程序逻辑结构的复杂性
- 软件成本相当昂贵，涉及到社会因素

1.1 What is software?

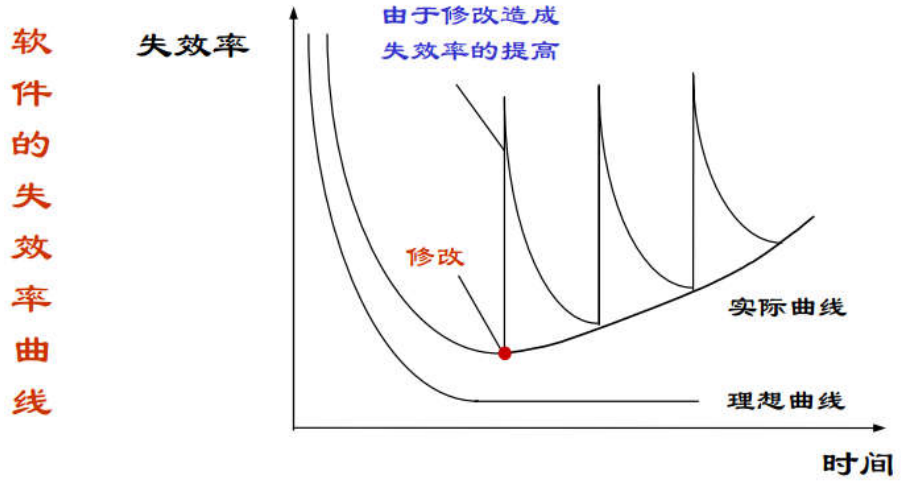
软件的特点 – 复杂性

- 软件在规模上可能比任何由人类创造的其他实体都要复杂，复杂性是软件的本质特征
- 软件的复杂性是必要属性
  - 大量的组合状态
  - 丰富的结构和相互依赖性
- 开发问题也会增加复杂性
  - 高效率的代码通常是复杂的
  - 重用通用化的组件意味着复杂的状态连接
  - 复杂的代码难以维护，导致设计上的更复杂

1.1 What is software?

软件的特点 – 可变性

- 所有成功的软件都会发生变更
  - 会在超越软件边界的情况下使用软件
  - 功能扩展压力
- 修改的副作用，软件的退化



1.2 What is "Software Crisis"?

- 1968 NATO Conference on Software Engineering
- Problems with software:
  - Often delivered too late
  - Did not behave as user expects
  - Rarely adaptable to changed circumstances
  - Many errors detected after delivery
  - Communication between stakeholders!

1.2 What is "Software Crisis"?

- Symptoms: 《The Mythical Man-Month》
  - Unmanageable.
  - Over budget.
  - Late.
  - Poor quality.
- Causes:
  - One word: Complexity.

1.2 What is "Software Crisis"?

- ARIANE 5火箭
  - 1996年6月，耗资70亿美元，发射37秒后爆炸
  - 原因：软件错误，试图将64位浮点数转换成16位整数时溢出，缺少错误处理程序
- 爱国者导弹
  - 1991年2月，对抗飞毛腿失利导致28名美国士兵丧生
  - 原因：导弹软件中包含一个累加计时误差
- 千年虫
  - 日期用2位表示，升级花费超过数亿美元
- 其它
  - 放射医疗事故、病毒、木马、网络攻击、黑客、安全…

1.3 What is "Good Software"?

Quality terminology

- ***Error***: human mistake
- ***Fault***: result of mistake, evidenced in some development or maintenance product
- ***Failure***: departure from the system's required behavior

human error

can lead to

fault

can lead to

?!

failure

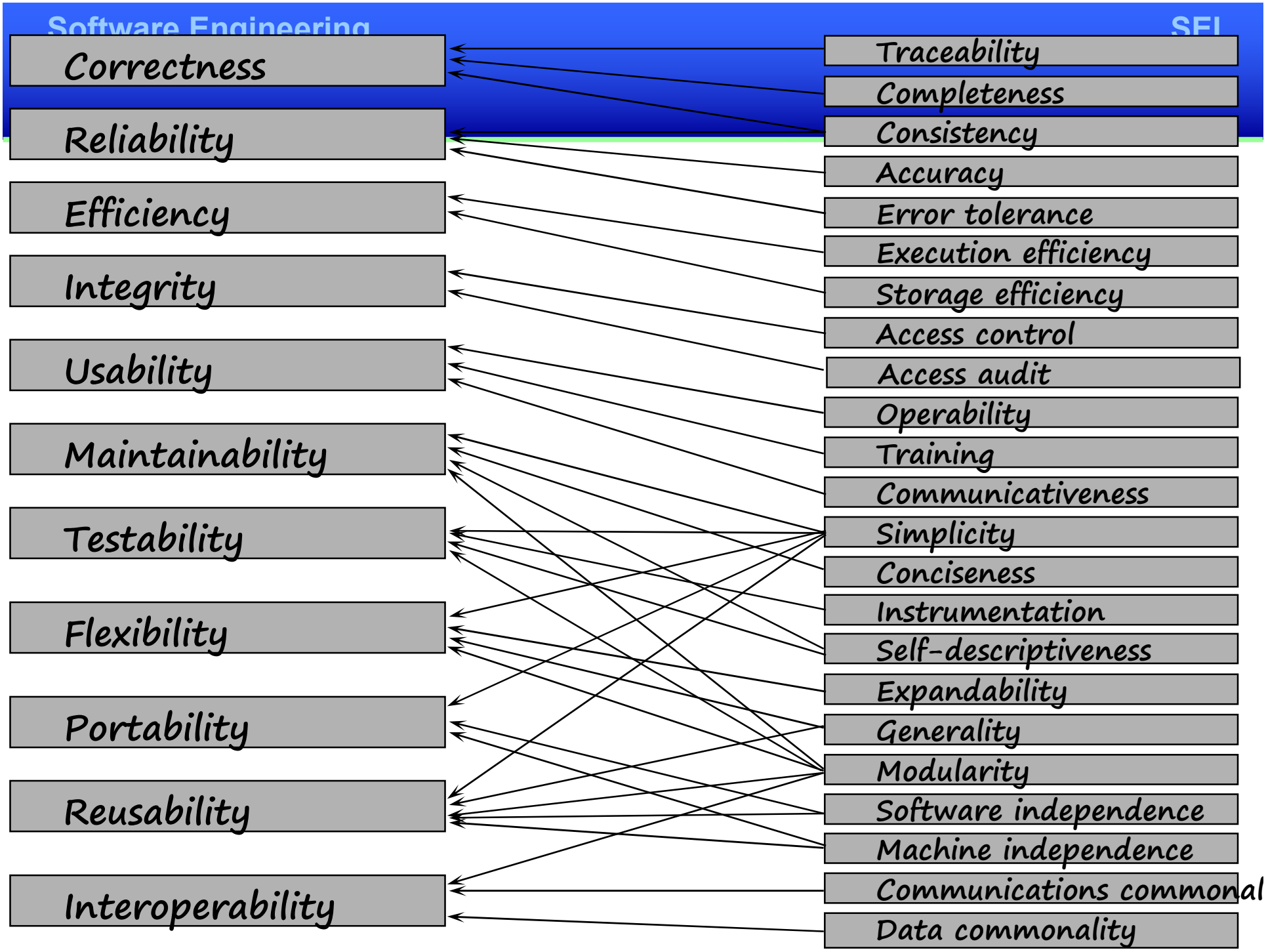
1.3 What is "Good Software"?

- ***Good software*** engineering must always include a strategy for producing ***quality software***.
- What is "Quality"?
  - the quality of the ***product***
  - the quality of the ***process*** that results in the product, and
  - the quality of the product ***in the context of the business*** environment in which the product will be used

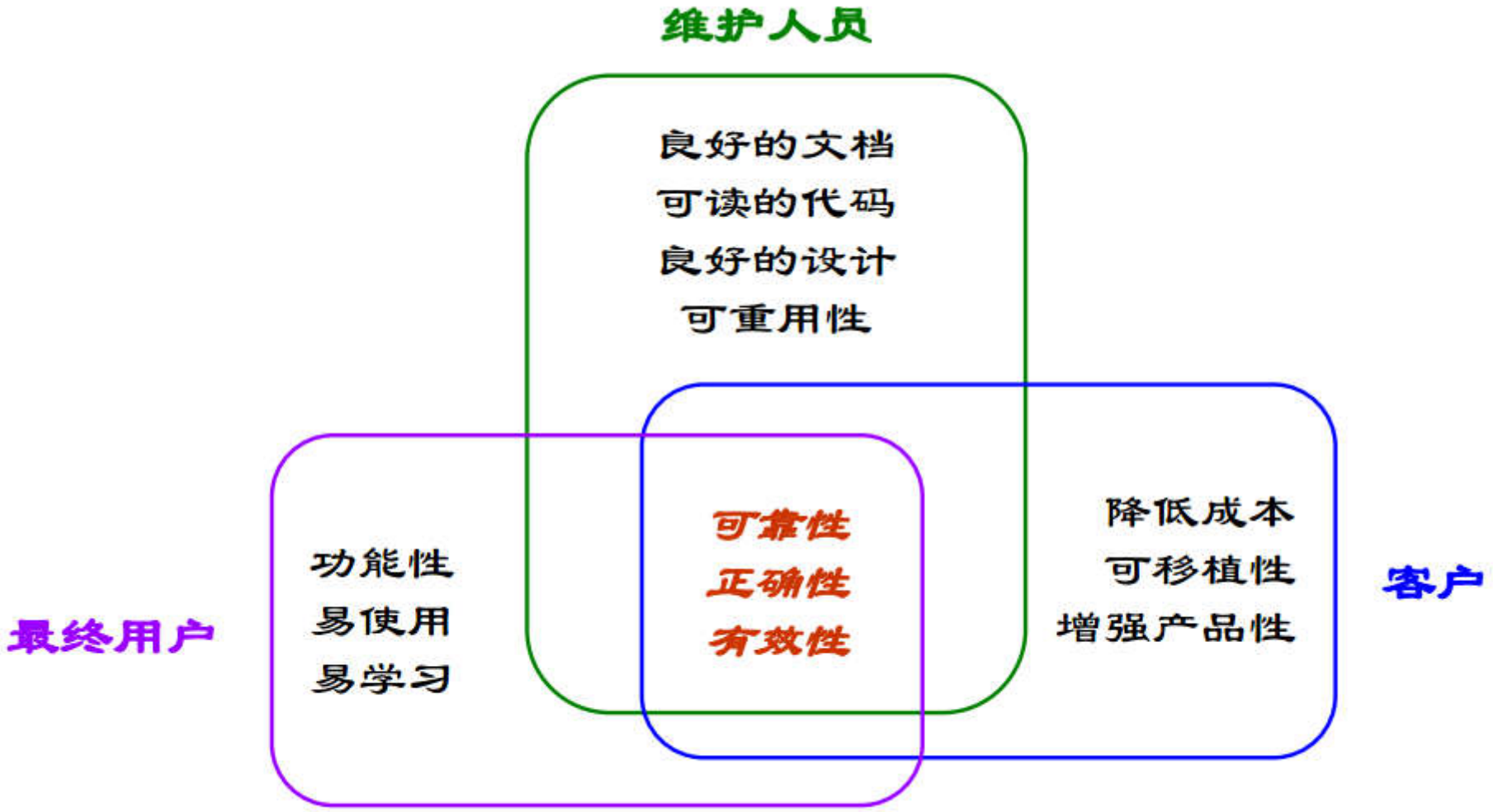
1.3 What is "Good Software"?

The Quality of the Product

- User's view
- Developer's view
- external quality factors relate to product quality criteria


Correctness
Reliability
Efficiency
Integrity
Usability
Maintainability
Testability
Flexibility
Portability
Reusability
Interoperability
Traceability
Completeness
Consistency
Accuracy
Error tolerance
Execution efficiency
Storage efficiency
Access control
Access audit
Operability
Training
Communicativeness
Simplicity
Conciseness
Instrumentation
Self-descriptiveness
Expandability
Generality
Modularity
Software independence
Machine independence
Communications commonal
Data commonality

1.3 What is “Good Software”?



1.3 What is "Good Software"?

The Quality of the Process

- There are many activities that affect the ultimate product quality; if any the activities go awry , the product quality may suffer.
  - Where and when are we likely to find a particular kind of fault?
  - How can we find faults earlier in the development process?
  - How can we build in fault tolerance so that we minimize the likelihood that a fault will become a failure?
  - Are there alternative activities that can make our process more effective or efficient at assuring quality?

1.3 What is "Good Software"?

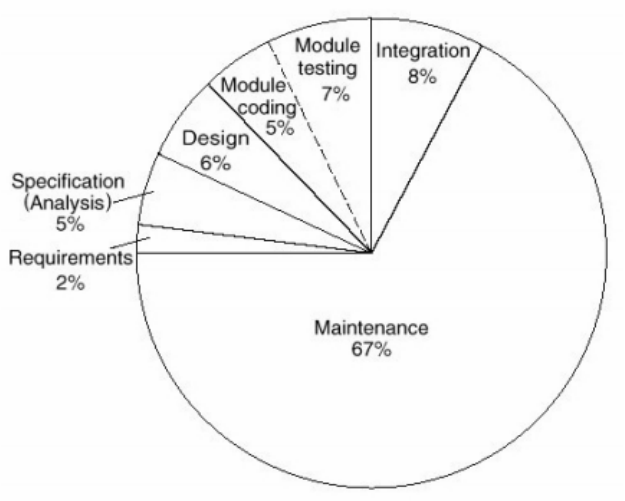
The Quality of the Process

- process modeling and process improvement in software engineering
  - Capability Maturity Model (***CMM***)—process guidelines.
  - Software Process Improvement and Capability dEtermination (***SPICE***)
  - ISO9001
  - Rational Unified Process (***RUP***)

1.3 What is "Good Software"?

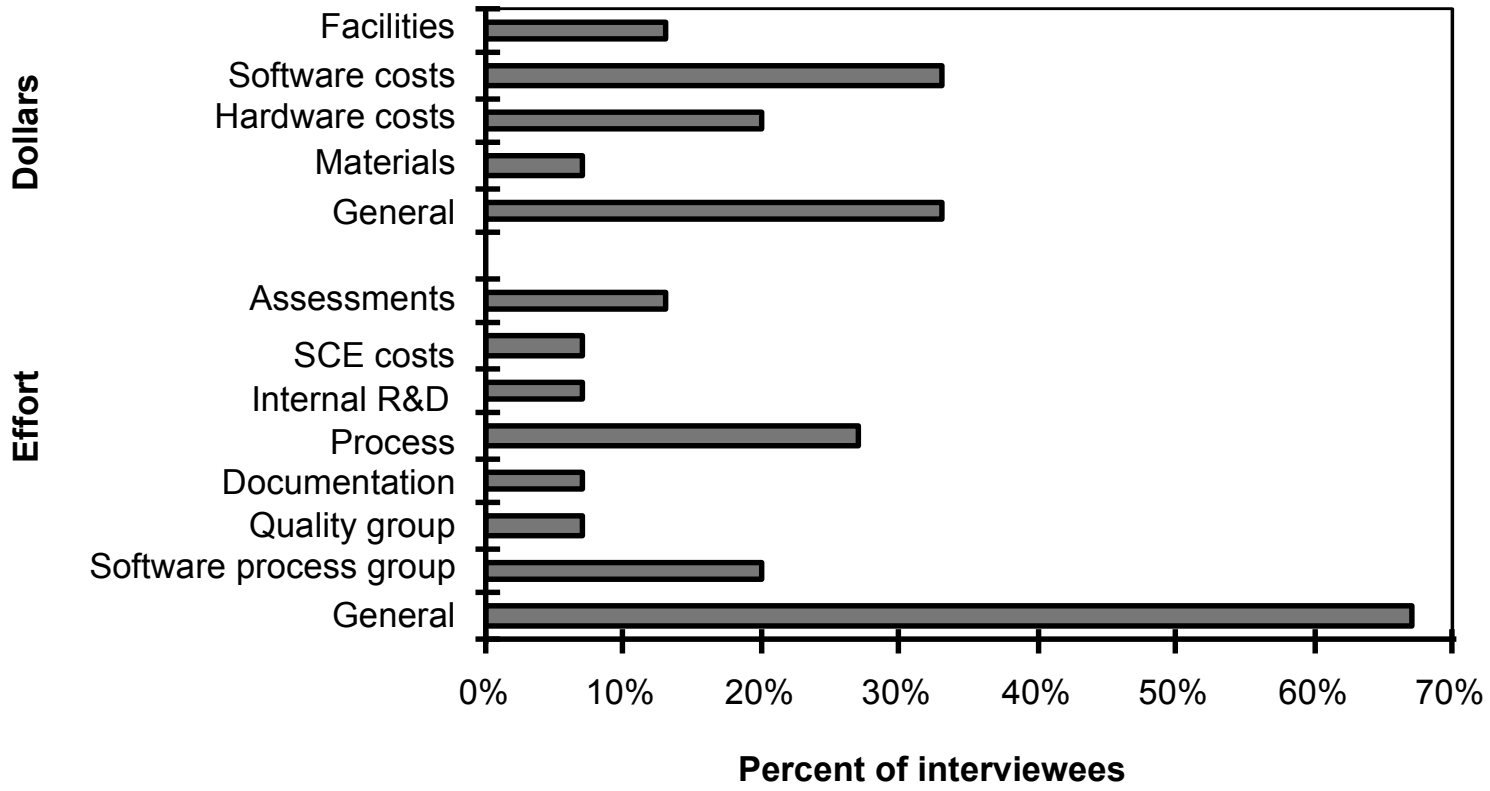
Quality in the Context of the Business Environment

- return on investment (ROI, 投资回报)
  - $\frac{\text{(total benefit - total costs)}}{\text{total costs}} \times 100 = \text{ROI}$
- Among the companies surveyed, return on investment included such items as
  - Training
  - Schedule
  - Risk
  - Quality
  - Productivity
  - Process
  - Customer
  - Costs
  - Business



1.3 What is "Good Software"?

Quality in the Context of the Business Environment



- SCE--Software Capability Evaluation (软件能力评鉴)
- R&D--Research and Development (研发)

1.4 Way to get “Good Software”

- System approach
- Engineering approach

1.4.1 System approach

- The projects we develop do not exit in a vacuum.
  - Hardware
  - Other software tasks
  - Other pieces of hardware
  - Exiting database
  - Other computer systems
- It is important to provide a content for any project by knowing the boundaries of the project.
  - What is included in the project
  - What is not in the project

1.4.1 System approach

The elements of a system

- Activities and objects
- Relationships and System Boundary
- Interrelated Systems

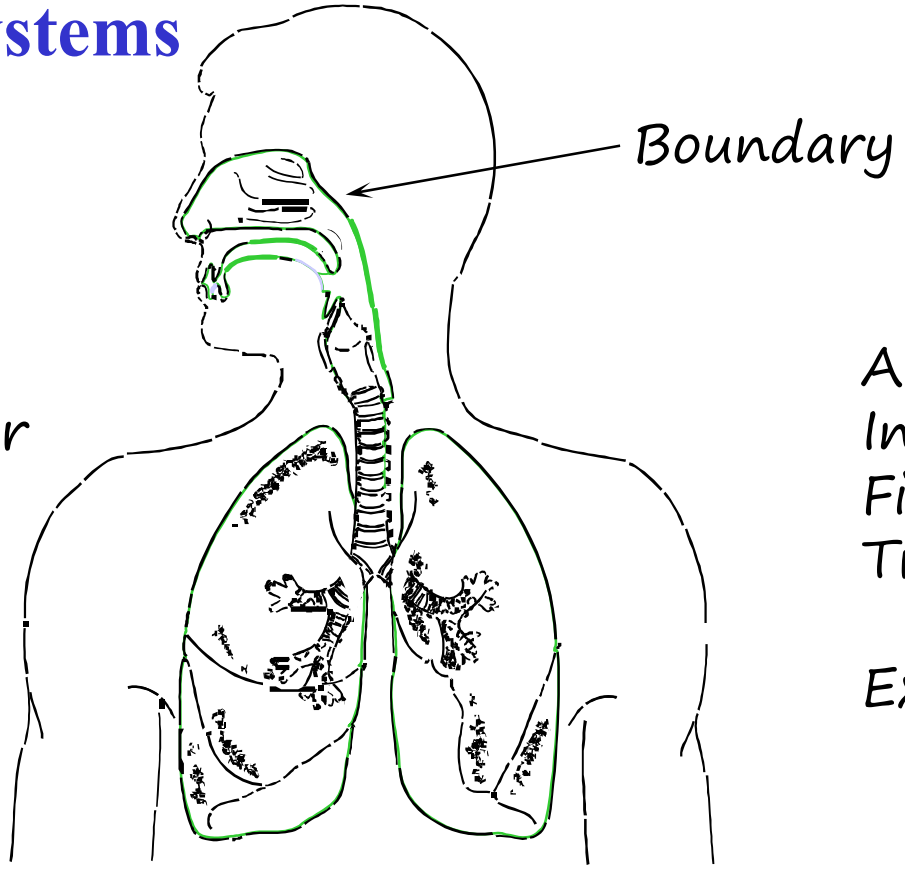
1.4.1 System approach

Identify activities and objects

- Activities
- Objects
  - Relationship
- system
  - boundary

1.4.1 System approach

Examples of Systems



ENTITIES:
Particulate matter
Oxygen
Carbon dioxide
Water
Nitrogen
Nose
Mouth
Trachea
Bronchial tubes
Lungs
Alveoli

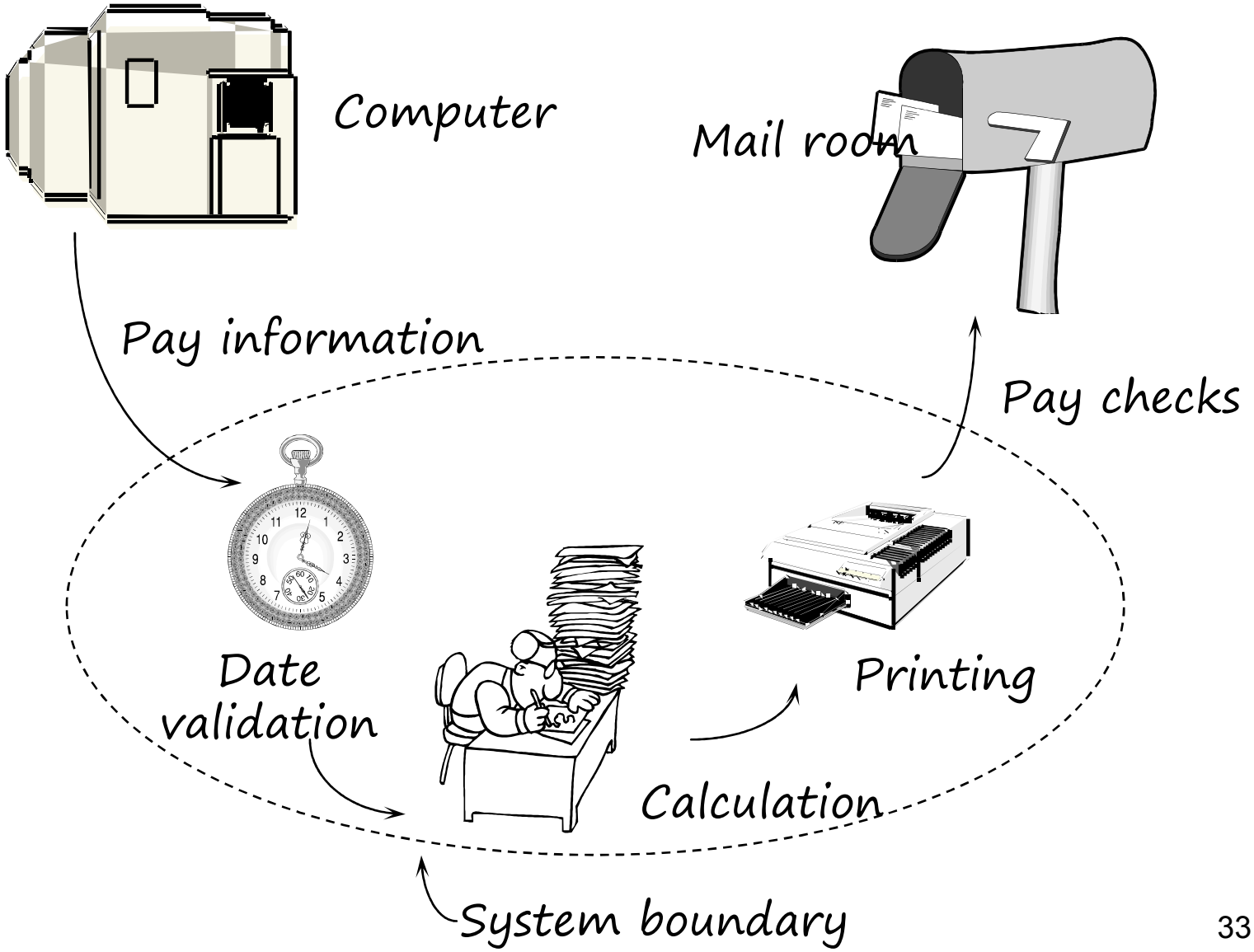
ACTIVITIES:
Inhale gases
Filter gases
Transfer molecule
to/from blood
Exhale gases

1.4.1 System approach

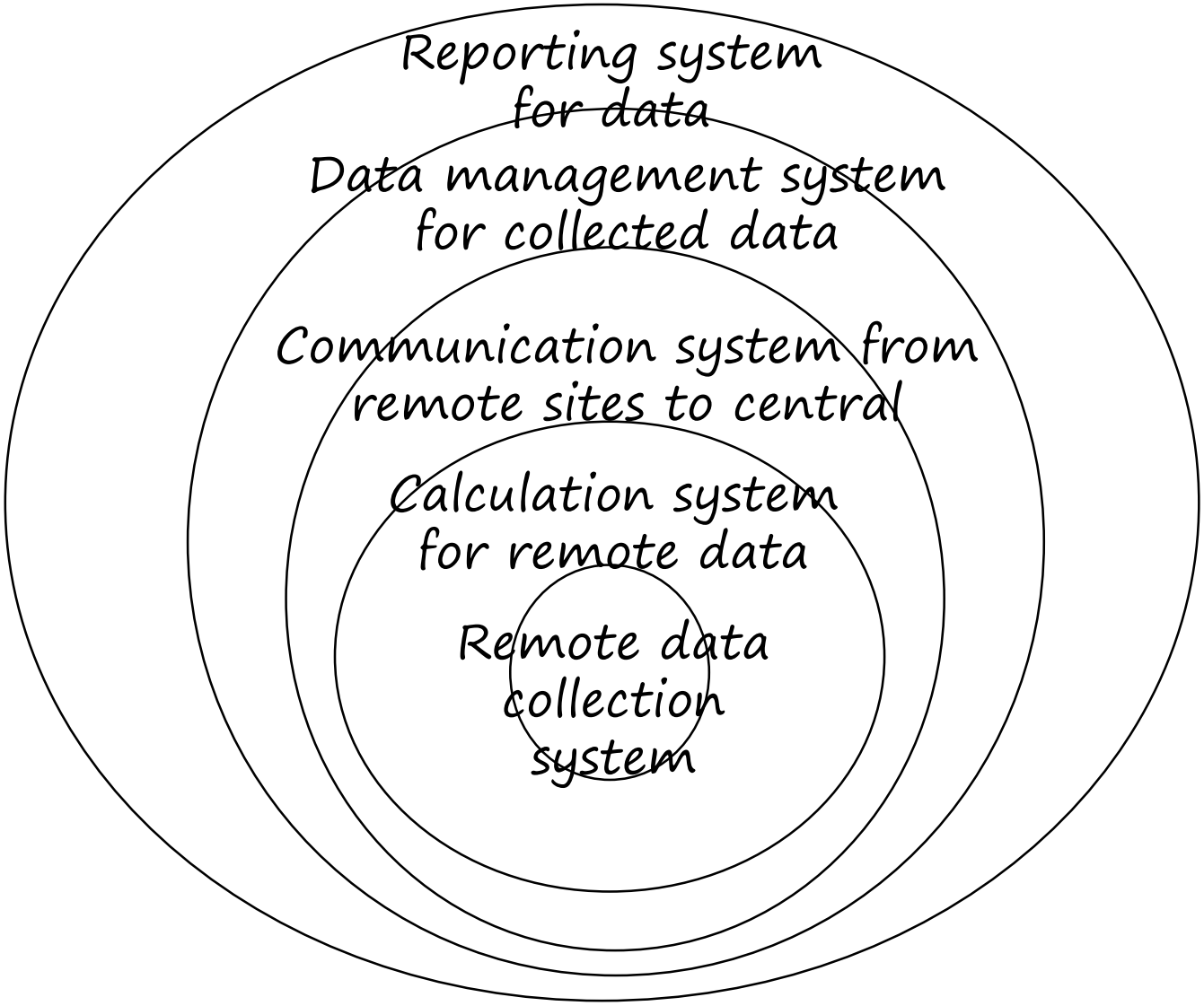
Interrelated Systems

- Boundary
  - Very few systems are independent of other systems.

1.4.1 System approach



1.4.1 System approach



1.4.2 Engineering approach

- Construction
  - Building a house
  - Building a software

- 工程是将理论和所学的知识应用于实践的科学，以便经济有效地解决实际问题
  - 手工，小规模的设计与建造
  - 工程，大规模的设计与建造
    - 复杂问题与目标分解
    - 多人参与，需要考虑运营、管理、成本、质量控制、安全等

1.4.2 Engineering approach

Building a house vs. software

- Determining and analyzing requirements
- Producing and documenting the design
- Detailed specifications
- Identifying and designing components
- Building components
- Testing components
- Integrating components
- Making final modifications
- Continuing maintenance
- Requirements analysis and definition
- System design
- Program design
- Writing programs
- Unit testing
- Integration testing
- System testing
- System delivery
- Maintenance

1.5 What is software engineering?

- **Definition:**
  - **The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines. (*Fritz Bauer, at theFirst NATO Conference on Software Engineering, 1969*)**

1.5 What is software engineering?

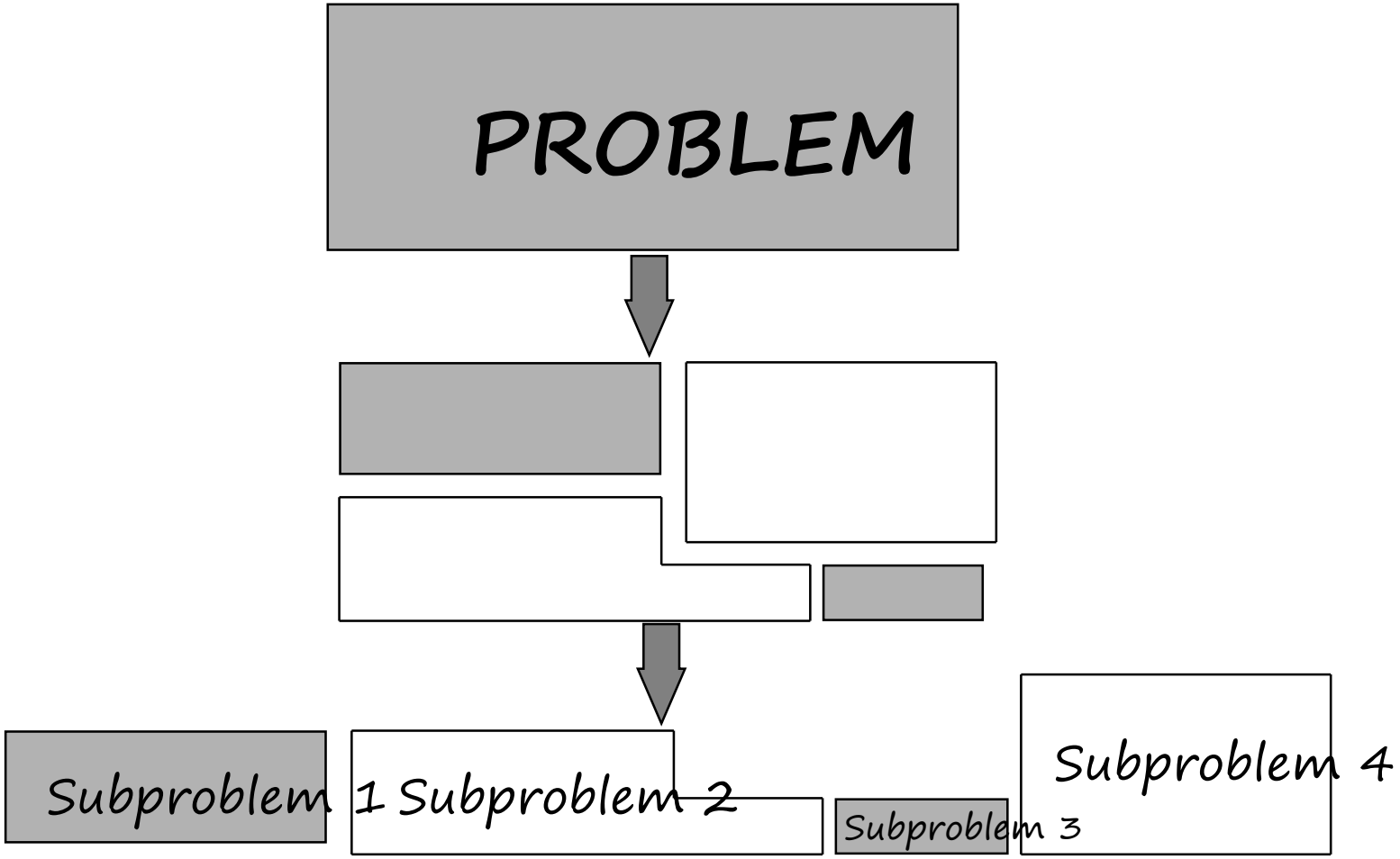
- **Definition:**
  - **Software engineering is the technological and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates. (*Fairley, R., Software Engineering Concepts. New York: McGraw-Hill, 1985.*)**
  - **Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1). (*IEEE Std 610-1990*)**

1.5 What is software engineering?

- Software engineering is an engineering discipline that is concerned with all aspects of software production.
- Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available.

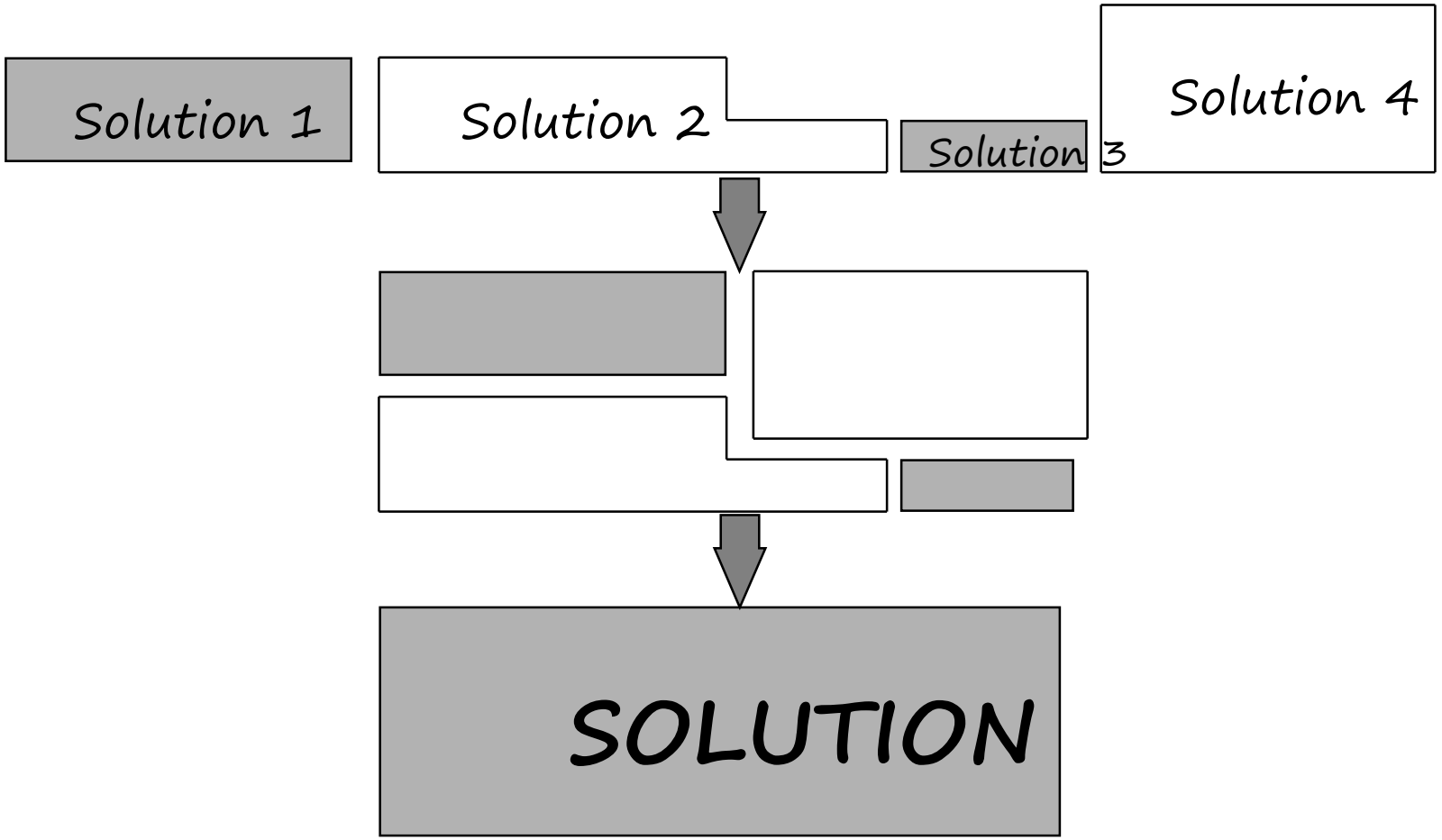
1.5 What is software engineering?

Analysis vs. synthesis of a problem



1.5 What is software engineering?

Analysis vs. synthesis of a problem



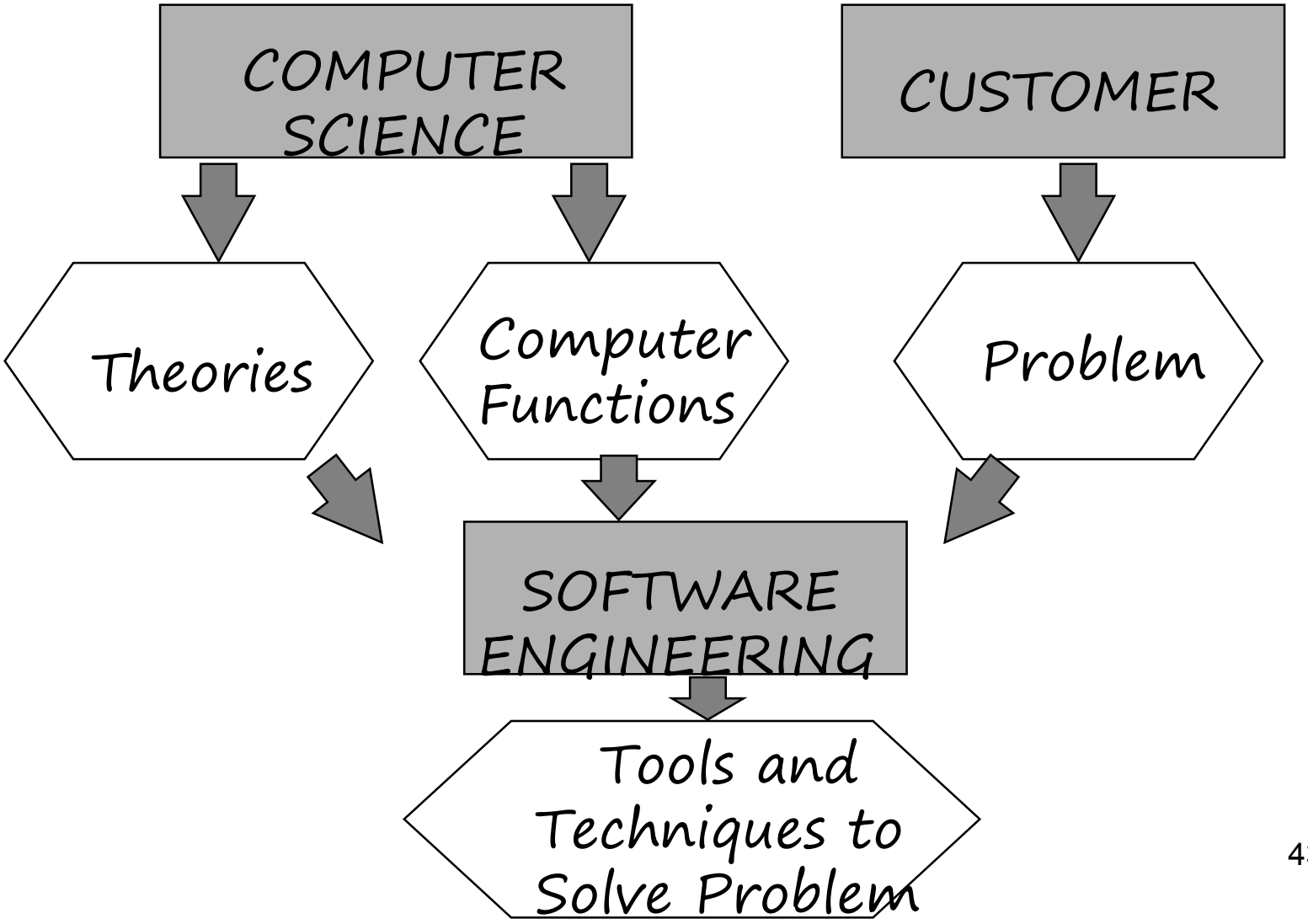
1.5 What is software engineering?

To Solve Problems

- **Method or technique**: procedure for producing a result
- **Tool**: instrument or automated system for accomplishing something
- **Procedure**: recipe for combination of tools and techniques
- **Paradigm**: style of doing something

1.5 What is software engineering?

Where does the Software Engineer fit in?



1.5 What is software engineering?

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
- computer scientist solve the computer's problems, but software engineer produce quality software using computer as tools.

1.5 What is software engineering?

What is the difference between software engineering and computer science?

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system.
- System engineers are involved in system specification, architectural design, integration and deployment.

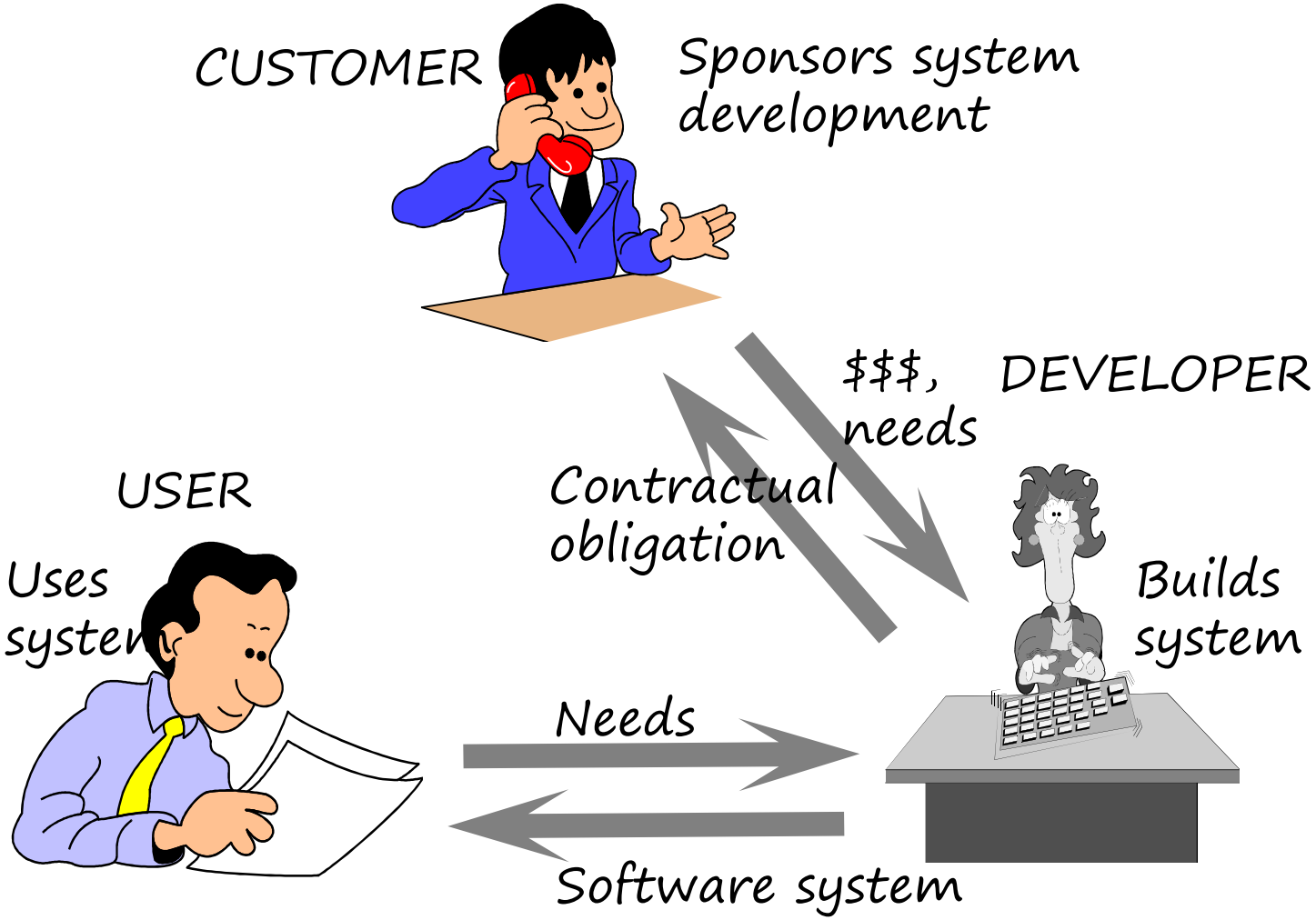
1.5 What is software engineering?

Who does software engineering

- Customer:
  - the Company, organization, or person who is paying for the software system to be developed.
- Developer:
  - the Company, organization, or person who is building the software for the customer.(Include managers and testers).
- User:
  - the person or people who will actually use the system.: the ones who sit at the terminal or summit the data .or read the output.

1.5 What is software engineering?

Who does software engineering

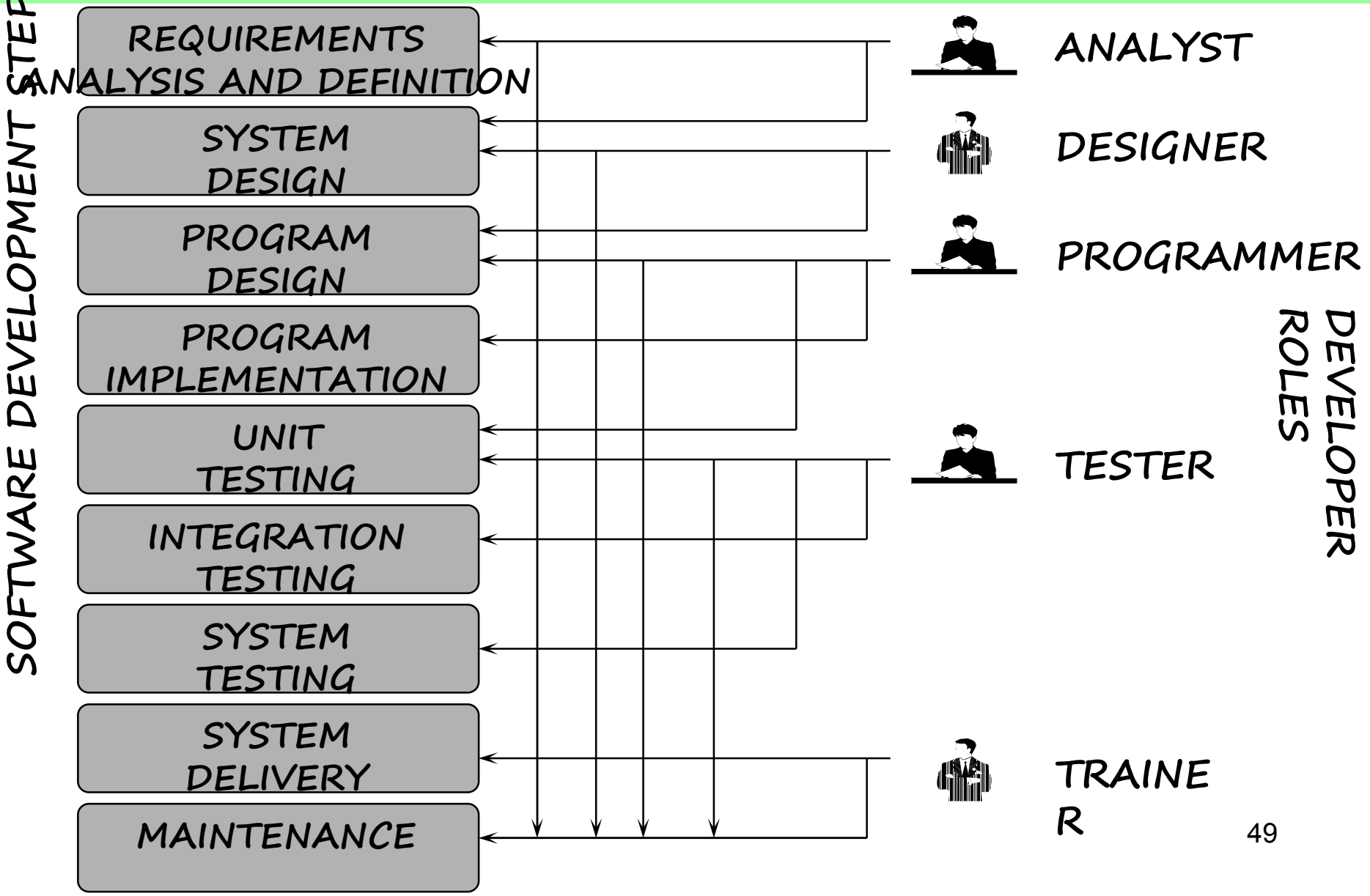


1.5 What is software engineering?

Members of the development team

- Each engineer may specialize in a particular aspect of development.
  - Requirement analysts
  - Programmers
  - Testers
  - Trainers
  - Maintenance Team
  - Configuration management team

1.5 What is software engineering?



1.6 Information systems example

- Piccadilly Television: regional British TV franchise
- Advertising scheme has many constraints:
  - alcohol adverts only after 9pm
  - if actor in show, no same actor in advert within 45 minutes
  - if advert in class of product, no other advert in same class during same break
  - rates dependent on amount of time bought
- Software to determine, track advertising time

小结

- Software
- Software Crisis
- Good Software
- Way to get “Good Software”
- Software engineering