CHAPTER 8

Software Maintenance

Outline

- Concept of software maintenance
- Types of software maintenance
- Costs of software maintenance
- Maintainability
- Process of maintenance

What is software maintenance ?

- ✓ **Modifying a program after it has been put into use**
- ✓ **Modifications or corrections made to a software system after it has been released to its customers**
- ✓ **Changing a software system while it is in operation**
- ✓ **Evolving a software system to adapt to changing business conditions and user needs**
- ✓ **Managing the processes of system change**

软件维护的概述

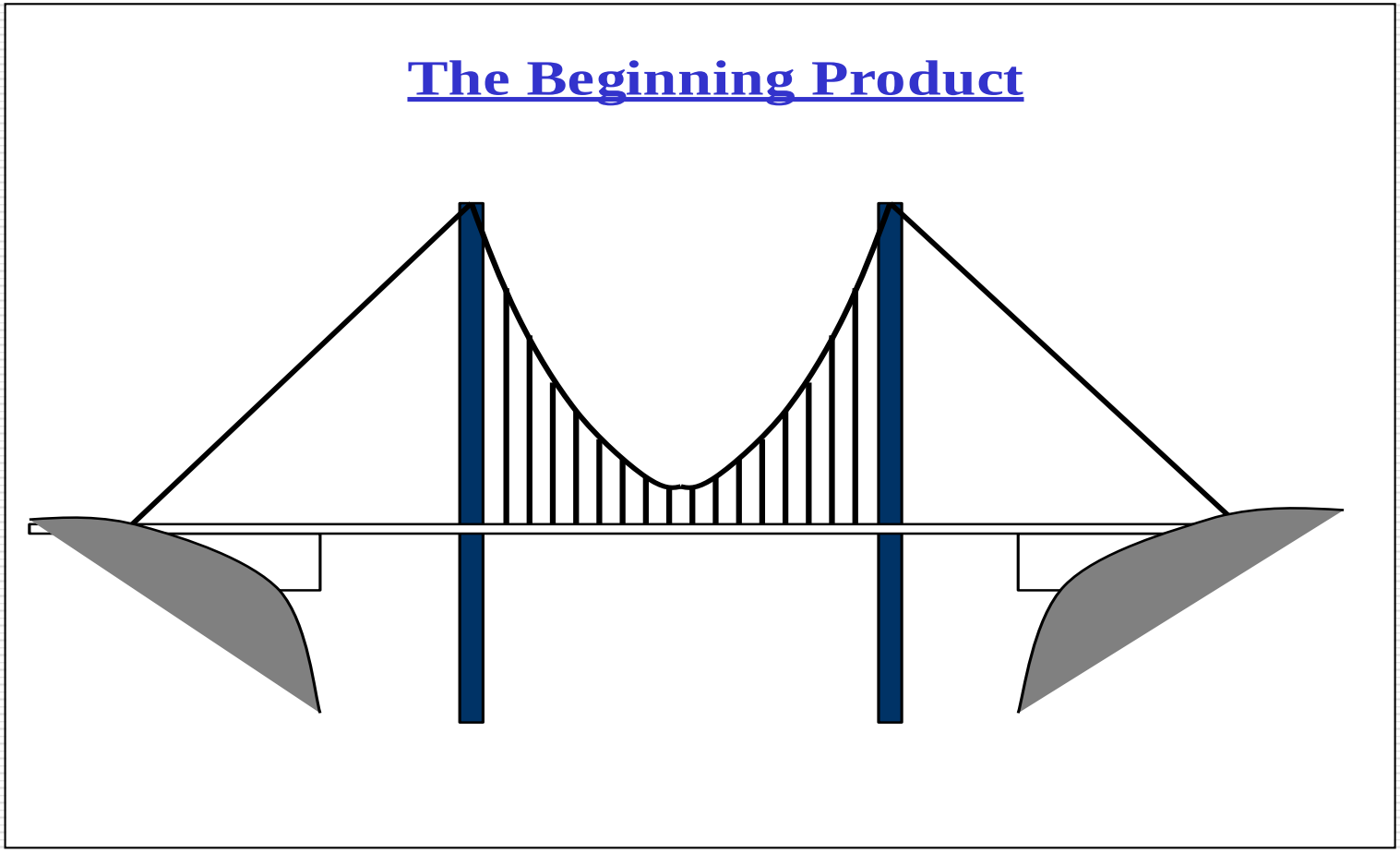
- 软件维护是软件开发工作完成以后，在用户使用期间，对软件所做的补充、修改和增加工作。
- 软件运行＝软件维护
- 在软件维护中，为增加和改进软件的功能所做的维护占80%，而为改正错误所做的维护仅占20%。
- 统计数据表明：实际上用于软件维护的费用占软件总费用的55-80%。
- 软件维护比软件开发更困难，需要更多的创造性工作。
- 一般不涉及体系结构的重大变化

Why software maintenance ?

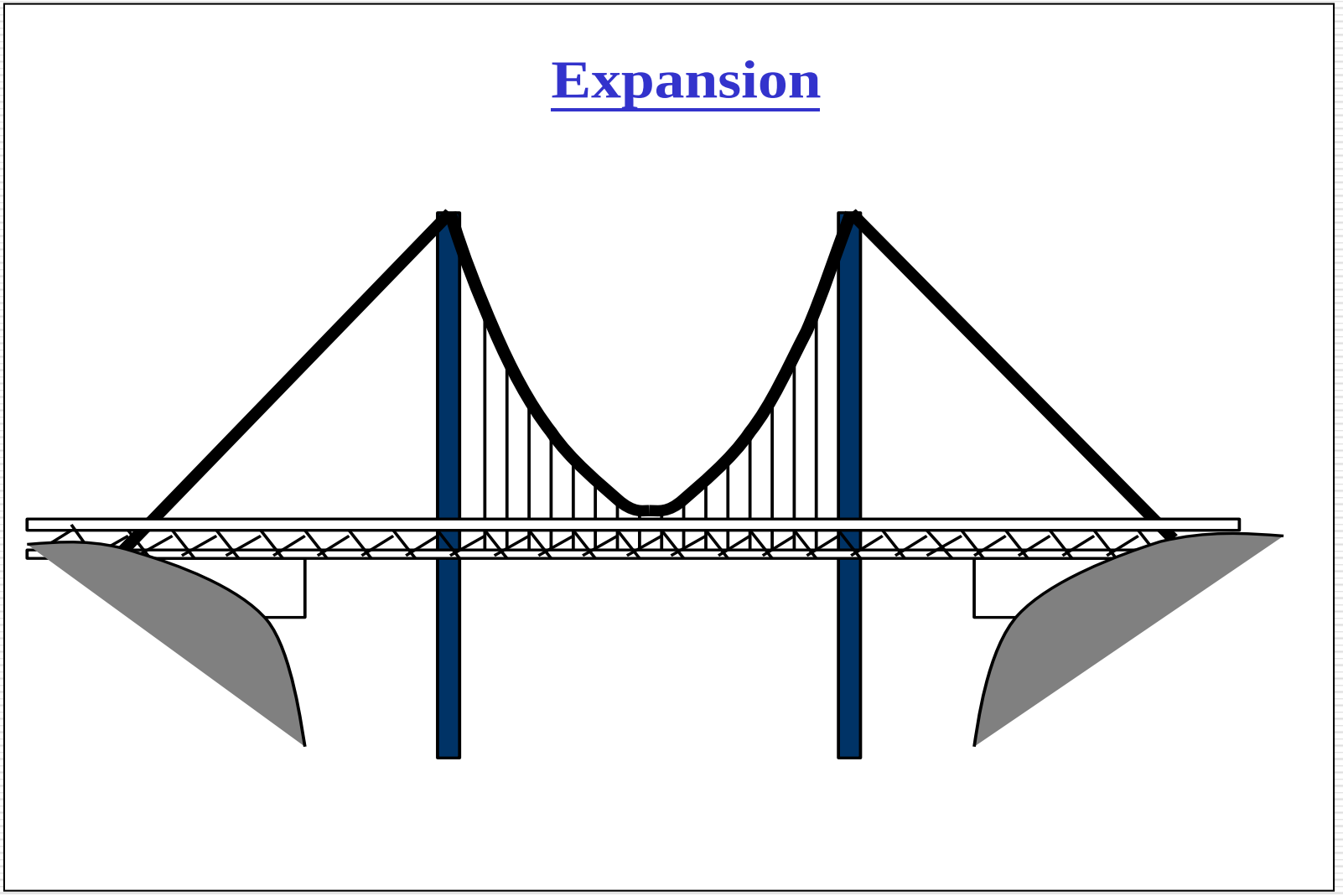
- 从机械、土木等工程中的启发
- 软件系统中的变化
- 软件维护是不可避免的

Why software maintenance ?

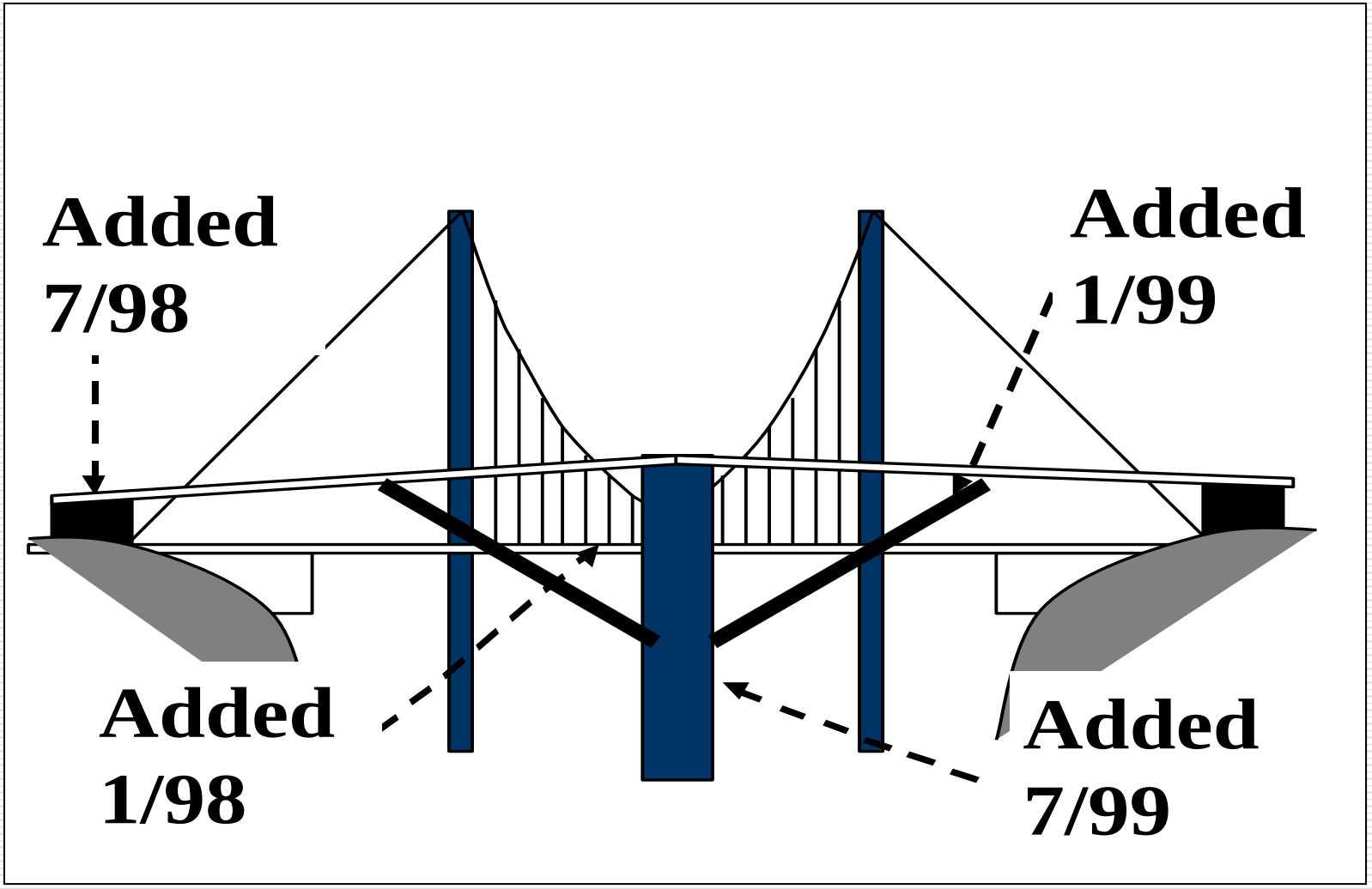
从机械、土木等工程中的启发



Why software maintenance ?



Why software maintenance ?



The First Law of Software Engineering

“No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle”

Bersoff et al. (1980)

软件系统，经常发生：变化，变更，改变，转变,…

“变”

“变”的根源

- ✓ New business or market conditions which cause changes in product requirements or business rules
- ✓ New customer needs that demand modification of data, functionality or services delivered by the system
- ✓ Reorganisation and/or business downsizing that changes priorities and team structures
- ✓ Budgetary or scheduling constraints that cause a redefinition of the system

Software change

- ✓ New requirements emerge when the software is used
- ✓ Errors must be repaired
- ✓ New equipment must be accommodated
- ✓ The performance or reliability may have to be improved
- ✓ MOST CHANGES ARE JUSTIFIED
- ✓ A key problem for organisations is implementing and managing change to their legacy systems

软件变化原因

软件的变化是不可避免的

- ✓ 软件在使用过程中，新的需求不断出现
- ✓ 商业环境在不断地变化
- ✓ 业务逻辑在不断地变化
- ✓ 软件中的缺陷需要进行修复
- ✓ 计算机硬件和软件环境的升级需要更新现有的系统
- ✓ 软件的性能和可靠性需要进一步改进

关键：

采取适当的策略，有效地实施和管理软件的变化！

Maintenance is inevitable

- **The system requirements are likely to change while the system is being developed because the environment is changing. Therefore a delivered system won't meet its requirements!**
- **Systems are tightly coupled with their environment. When a system is installed in an environment it changes that environment and therefore changes the system requirements.**
- **Systems MUST be maintained therefore if they are to remain useful in an environment.**

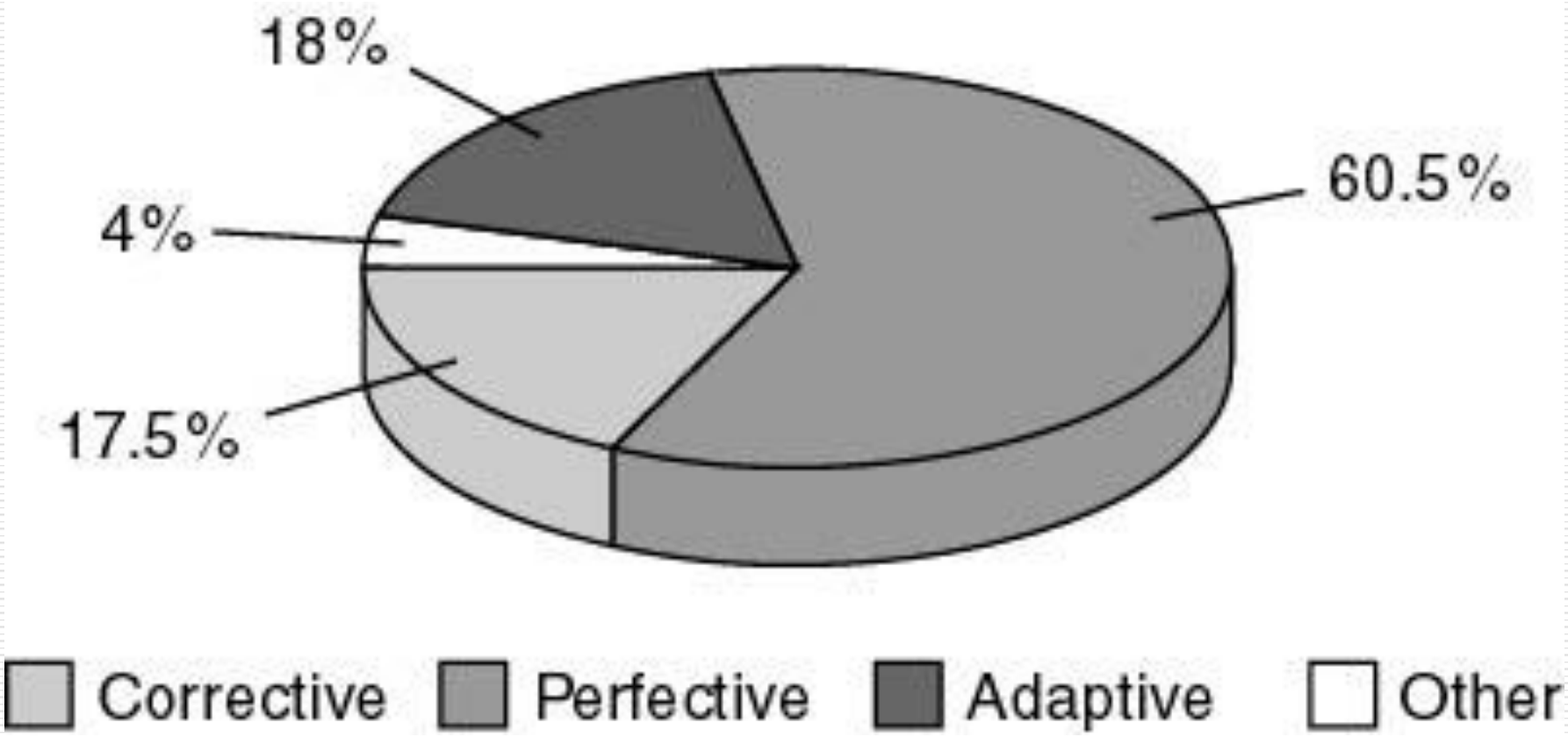
Types of Maintenance

- **Commonly divided into four main categories**
  - **Corrective Maintenance**
  - **Adaptive Maintenance**
  - **Perfective Maintenance**
  - **Preventative Maintenance**

维护种类

- 改正性维护：目的是识别和矫正功能错误、性能错误和实现上的错误。
- 适应性维护：使软件适应于外界环境的改变而对软件所做的修改工作。
- 完善性维护：为了扩充软件的功能或改善软件的性能对软件所做的改变。
- 预防性维护：为了以后更便于维护，或者为了改进可靠性，或者提供更好的基础便于将来提高性能而修改软件。

Distribution of maintenance effort



改正性维护

- 什么是纠错性维护?
  - **诊断和纠正软件中的错误或者缺陷**
- 起因
  - **用户在使用软件过程中发现错误**
- 目的
  - **改正软件系统中潜藏的错误**

改正性维护

- ✓ **Focused on fixing failures**
- ✓ **Is a reactive process**
  - **failures and their associated faults generally need to be corrected either immediately or in the near future**
- ✓ 改错的成本不同
  - **Coding - usually relatively cheap**
  - **Design - more expensive as they may require changes to several program components**
  - **Requirements - most expensive - may require extensive system redesign**
- ✓ **Requirements and Design are the source of approximately 80% of failures**

改正性维护

- ✓ **Fixing a fault has a 20 to 50% chance of introducing another fault**
- ✓ **Reasons for new faults include**
  - **the ripple effect, where a change in one area may cause changes in seemingly unrelated areas**
  - **Person who makes the repair is generally not the person who wrote the code or designed the system**
- ✓ **Two types of corrective maintenance**
  - **Emergency Repairs - short time frame, often a single program, failure needs to be repaired as soon as possible**
  - **Scheduled Repairs - failure doesn't need immediate attention, re-examination of all emergency repairs**

适应性维护

- 什么是适应性维护?
  - **对软件进行改造以适应新的运行环境和平台**
- 原因
  - **软件运行于环境(硬件、OS、网络等)之上**
  - **软件服役时间较长，运行环境变化很快**
- 目的
  - **适应环境变化和发展而对软件进行维护**

适应性维护

- ✓ **The evolution of the system to meet the needs of the user and the business.**
- ✓ **Caused by**
  - **internal needs**
  - **external competition**
  - **external requirements e.g. changes in law**
- ✓ **Essentially we are introducing new requirements to the system.**
- ✓ **Therefore we should treat like a new development in our approach and methods.**

完善性维护

- 什么是完善性维护?
  - **对软件进行改造以增加新功能、修改已有功能**
- 原因
  - **用户要求增加新功能、建议修改已有功能或提出其他改进意见**
- 目的
  - **满足用户日益增长的各种需求**

完善性维护

- ✓ **Old proverb says “If it is not broken, don’t fix it”**
- ✓ **Perfective maintenance ignores this ancient piece of wisdom**
- ✓ **Is about improving the quality of a program that already works**
- ✓ **Aim to achieve**
  - **reduced costs in using the system**
  - **increase maintainability of the system**
  - **more closely meet the users’ needs**

完善性维护

- Includes all efforts to polish or refine the quality of the software or the documentation
- Important that the potential benefits of the perfective maintenance outweigh
  - the costs of the maintenance
  - and the opportunity costs of improvements elsewhere or using the resources on new developments
- Before performing perfective maintenance, one should go through an analysis process
- Nevertheless, a little perfective maintenance can have dramatic effects

预防性维护

- 什么是预防性维护?
  - **对软件的结构进行改造以便提高软件的可靠性和可维护性等**
- 原因
  - **为改善软件系统的可维护性和可靠性，为以后的软件改进奠定基础**
- 目的
  - **获取和改善软件结构**

预防性维护

- Can be seen as radical perfective maintenance or as an alternative to maintenance
- More commonly known as Software Re-engineering
- Taking a legacy system and converting its structure or converting to a new language
- Old system starts as a specification for the new system
- Common method now is known as wrappers where an entire system is placed in an OO wrapper and treated as one large object

维护成本

- ✓ **Usually greater than development costs (2* to 100* depending on the application).**
- ✓ **Affected by both technical and non-technical factors.**
- ✓ **Increases as software is maintained. Maintenance corrupts the software structure so makes further maintenance more difficult.**
- ✓ **Ageing software can have high support costs (e.g. old languages, compilers etc.)**

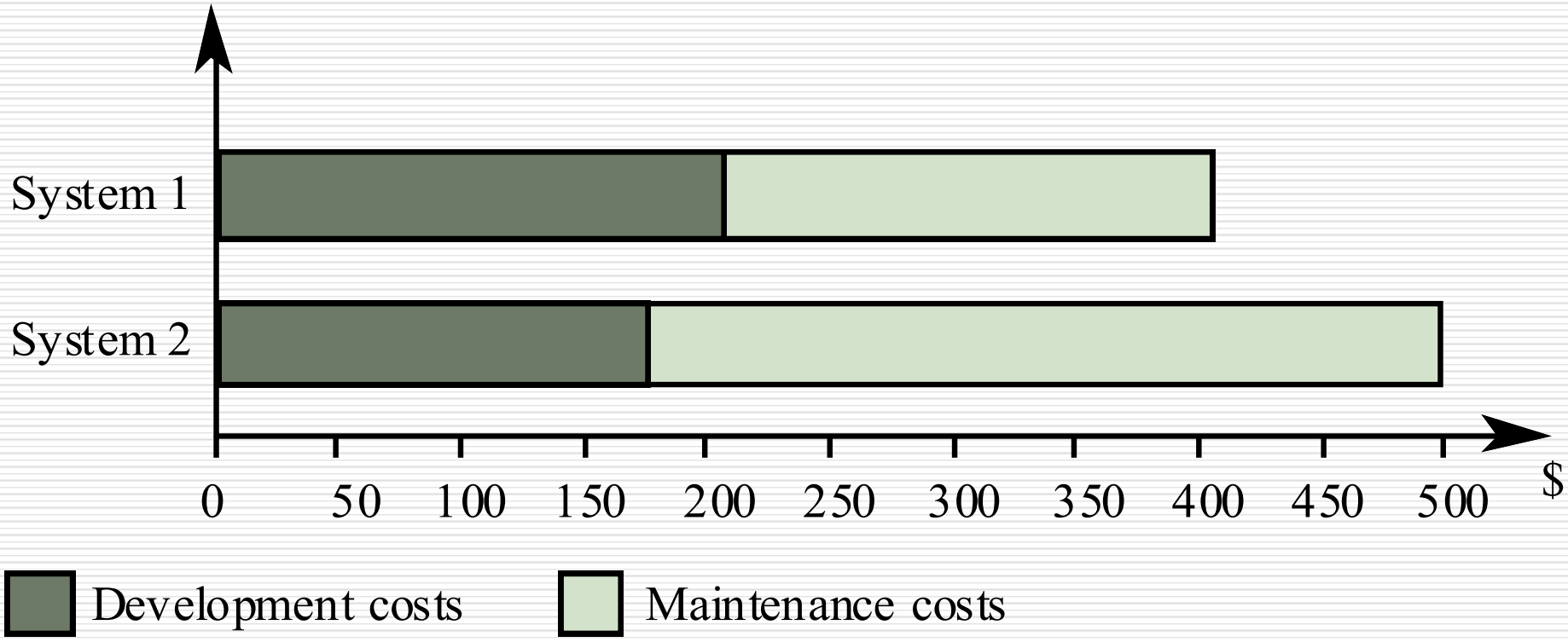
The Cost of Maintenance

- 通常认为：
  - **Requirements Definition** **3%**
  - **Preliminary Design** **3%**
  - **Detailed Design** **5%**
  - **Implementation** **7%**
  - **Testing** **15%**
  - **Maintenance** **67%**

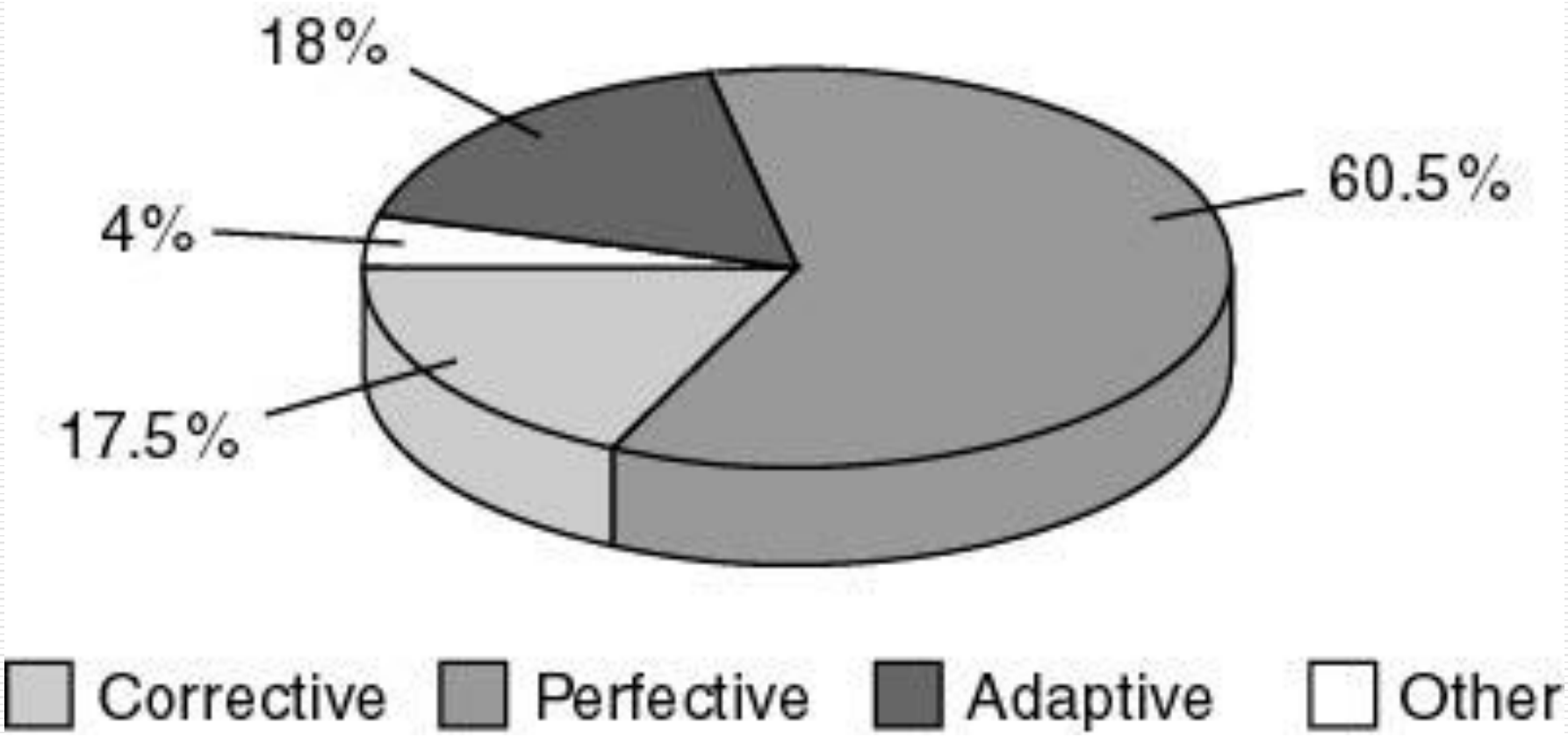
- 70年代（35%-40%）
- 80年代（60%左右）
- 90年代（75%左右）
- 21世纪（80%左右）

- **Another study found at least 50%of effort spent on maintenance**
- **Another study found between 65% and 75%on maintenance**
- **In embedded real-time systems, maintenance costs may be up to 4 times development costs**

Development/maintenance costs



Distribution of maintenance effort



维护的成本

$$M = P + K\,e^{(c-d)}$$

M：维护工作总工作量

P：生产性工作量

K：经验常数

c：复杂度

d：对该软件熟悉程度的度量

Why is Maintenance so Costly?

- ✓ **Most software is between 10 and 15 years old**
- ✓ **Much of that software is showing its age as it was created when program size and storage space were far more important factors**
- ✓ **This has lead to inflexible designs, coding and documentation**
- ✓ **Maintenance is usually done by inexperienced staff unfamiliar with the application**
- ✓ **Developers don’t like maintenance**
- ✓ **Changes often cause new faults in the system**
- ✓ **Changes tend to degrade the structure of a program**
- ✓ **Changes are often not documented**

为什么维护工作成本很高

(1) 成功的软件，生命周期较长，10-15年

(2) 难以跟踪软件版本的进化过程, 软件的变化未在文档中反映出来.

(3) 难以跟踪软件的创建过程, 无文档或不全.

(4) 开发者不喜欢维护.

(5) 难以读懂他人程序.

(6) 软件人员流动性大.

(7) 设计时未考虑修改需要, 修改困难.

(8) 维护工作无吸引力, 缺乏成就感.

(9) 维护可能引入新的错误!

软件可维护性

- **软件可维护性的定义**

  软件可维护性是指软件被理解、改正、调整和改进的程度

- **衡量软件质量的几个主要特性**
  - ✓ 所采取的开发方法（OS, OO）
  - ✓ 开发人员素质
  - ✓ 文档是否齐全
  - ✓ 可维护性
  - ✓ 可使用性
  - ✓ 可靠性
  - ✓ 标准化……

可维护性的度量

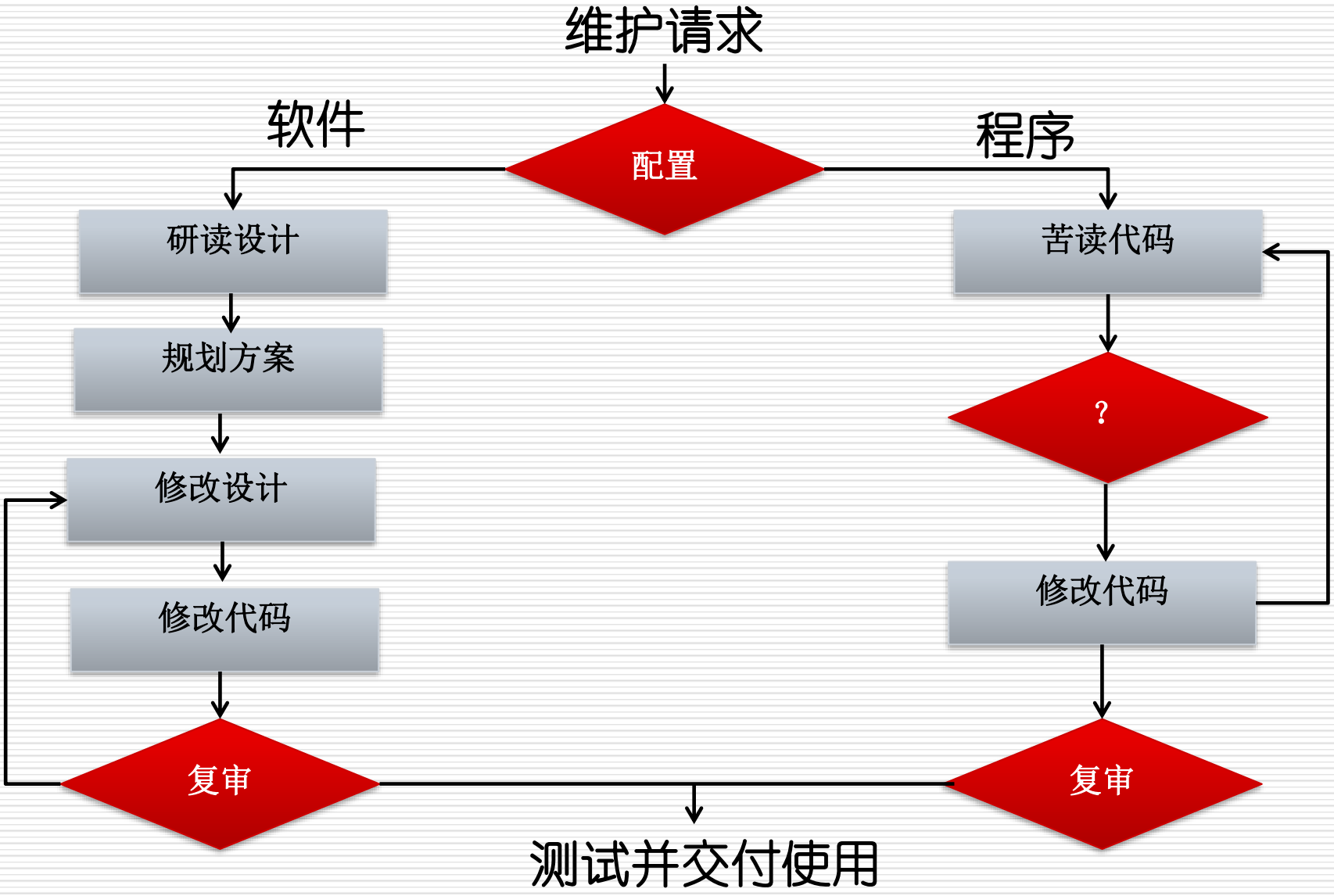
度量程序可维护性的7个特性

	改正性维护	适应性维护	完善性维护
可理解性			
可测试性			
可修改性			
可靠性			
可移植性			
可使用性			
效　率			

提高可维护性的方法

- ✓ 建立明确的软件质量目标和优先级
- ✓ 使用提高软件质量的技术和工具
- ✓ 进行明确的质量保证审查
- ✓ 选择可维护的程序设计语言
- ✓ 改进程序的文档
- ✓ 提倡规范化、标准化开发
- ✓ 开发软件时考虑到可维护要求

软件维护过程



维护过程

维护机构机构

维护申请报告

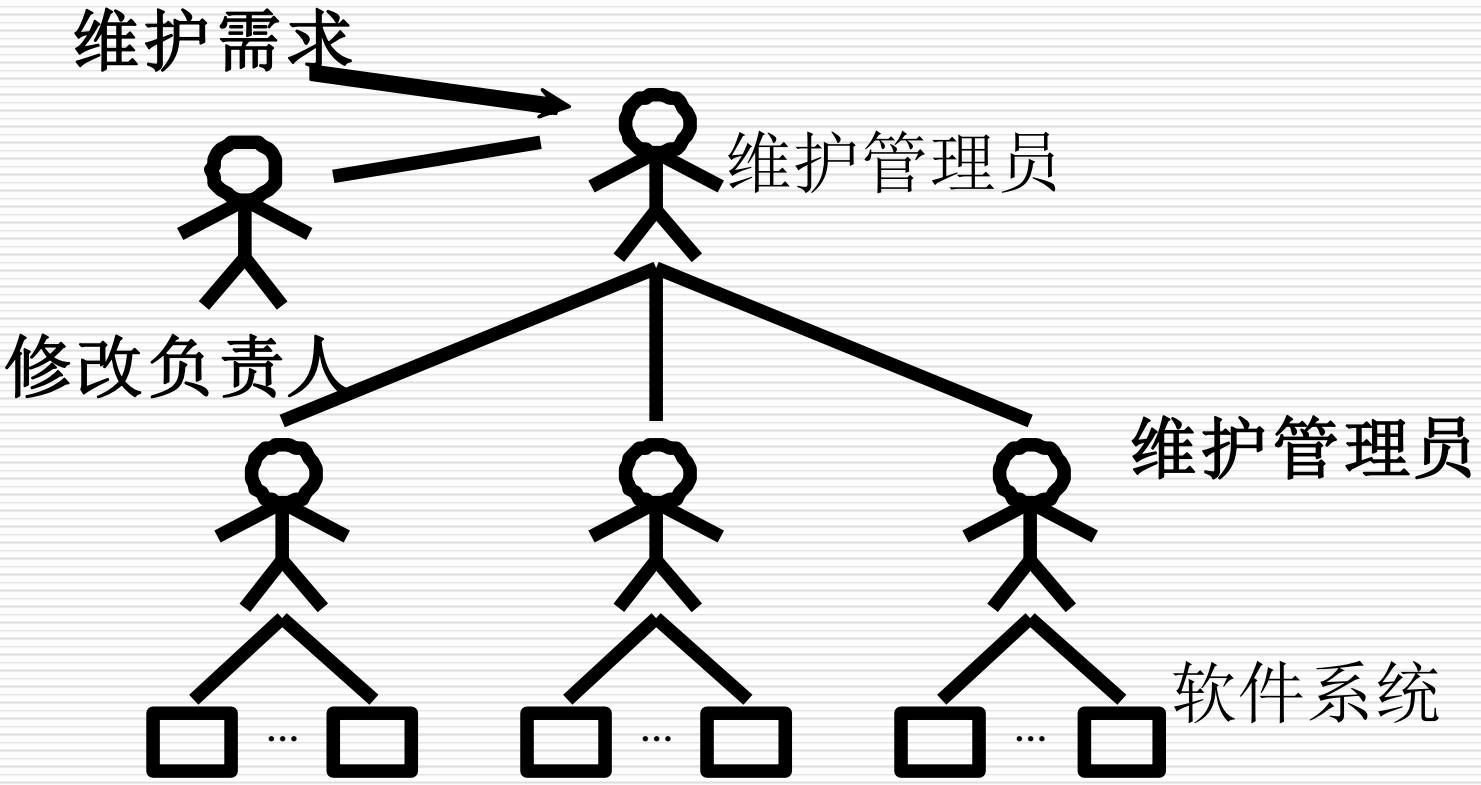
维护的工作流程

维护记录

维护评价

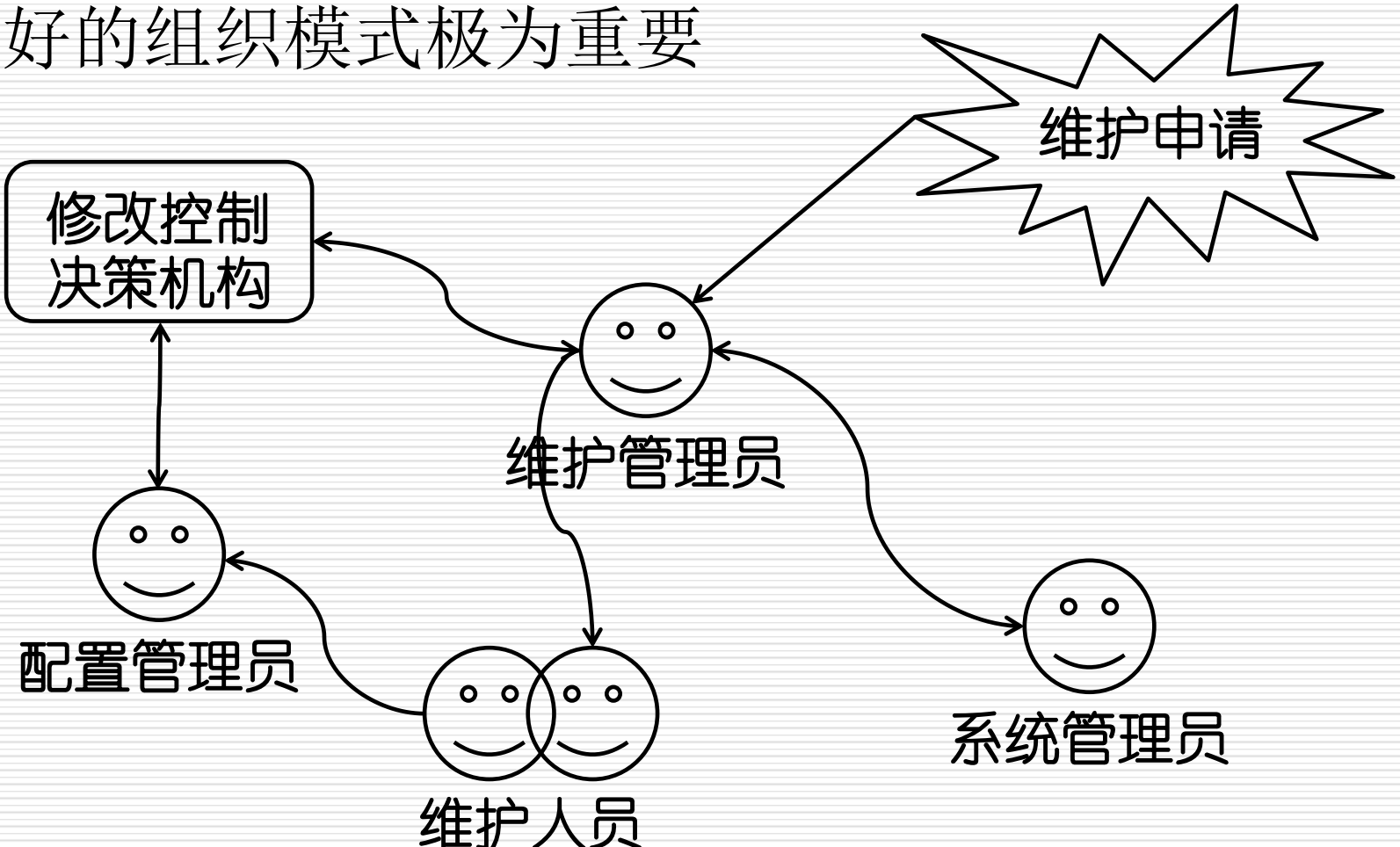
完成

维护的组织机构



成立维护组织

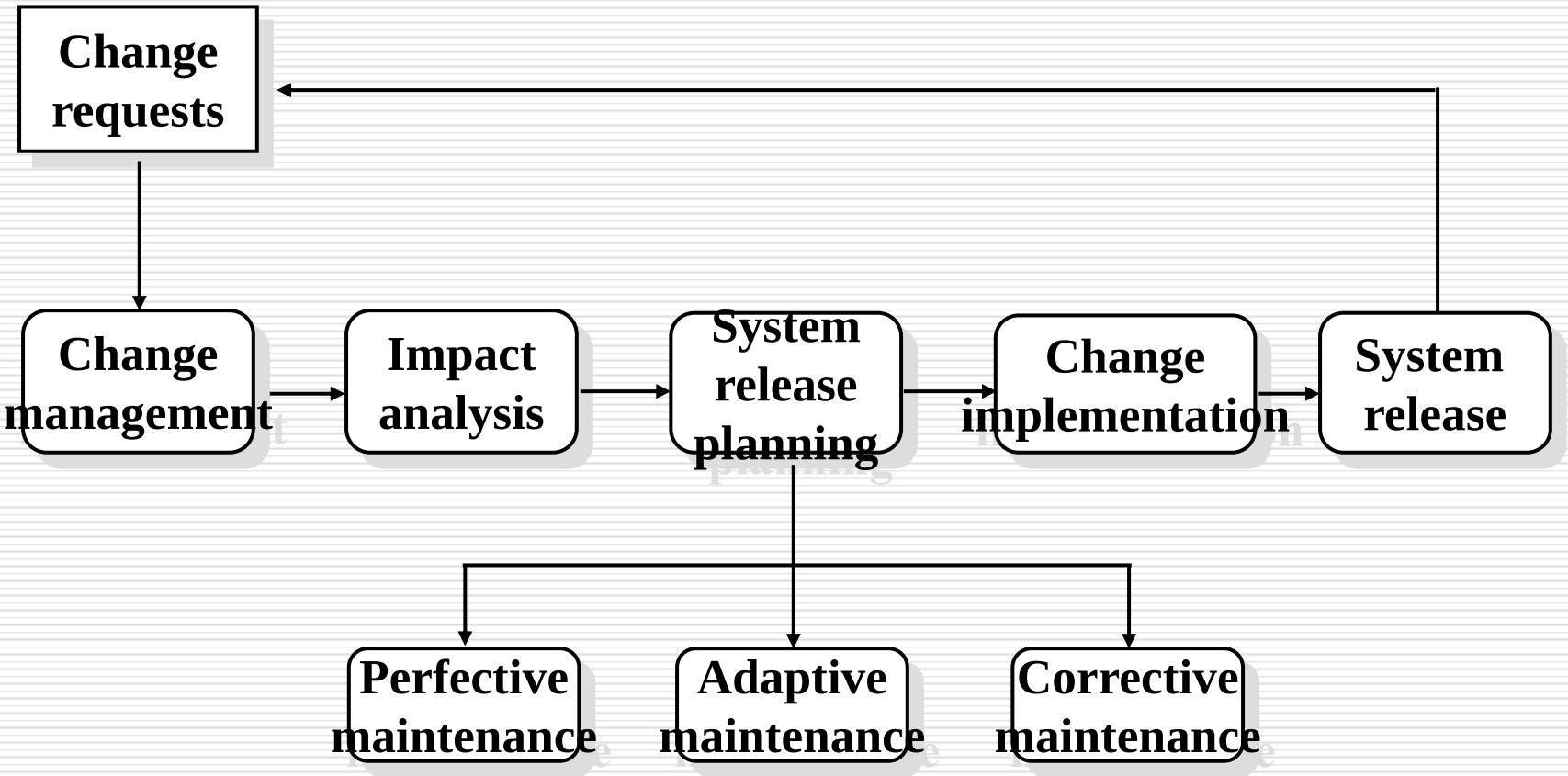
- “抓着谁就是谁”不可取
- 好的组织模式极为重要



维护申请

- 维护申请报告是由软件组织外部提交的文档，它是计划维护活动的基础。软件组织内部应依此制定相应的软件修改报告，这个报告包括以下内容：
  （1）为满足某个维护申请要求所需的工作量；
  （2）所需修改变动的性质；
  （3）申请修改的优先级；
  （4）与修改有关的事后数据。
- 所有维护申请都应以标准化的文档形式
- 软件修改报告应提交修改负责人进行审核批准，以便进行下一步工作。

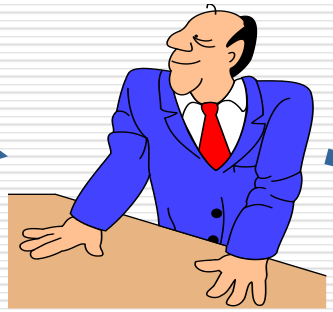
维护流程


Change requests
Change management
Impact analysis
System release planning
Change implementation
System release
Perfective maintenance
Adaptive maintenance
Corrective maintenance

A Typical Maintenance Flow

Customer

nominal path



Help desk

Written MR's

Proposed M. R.'s

Change control board

Approved M. R.'s

Maintenance engineer

Current source & documentation

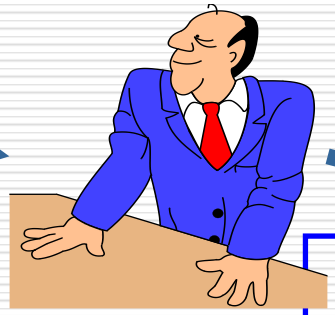
Modified source & documentation

Adapted from *Software Engineering: An Object-Oriented Perspective* by Eric J. Braude (Wiley 2001), with permission.

A Typical Maintenance Flow

Customer

nominal path



Help desk

Written MR's

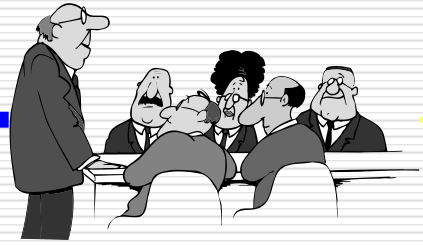
Marketing

Proposed M. R.'s

Maintenance manager

Maintenance engineer

Approved M. R.'s



Change control board

Current source & documentation

Modified source & documentation

Rejected MR's

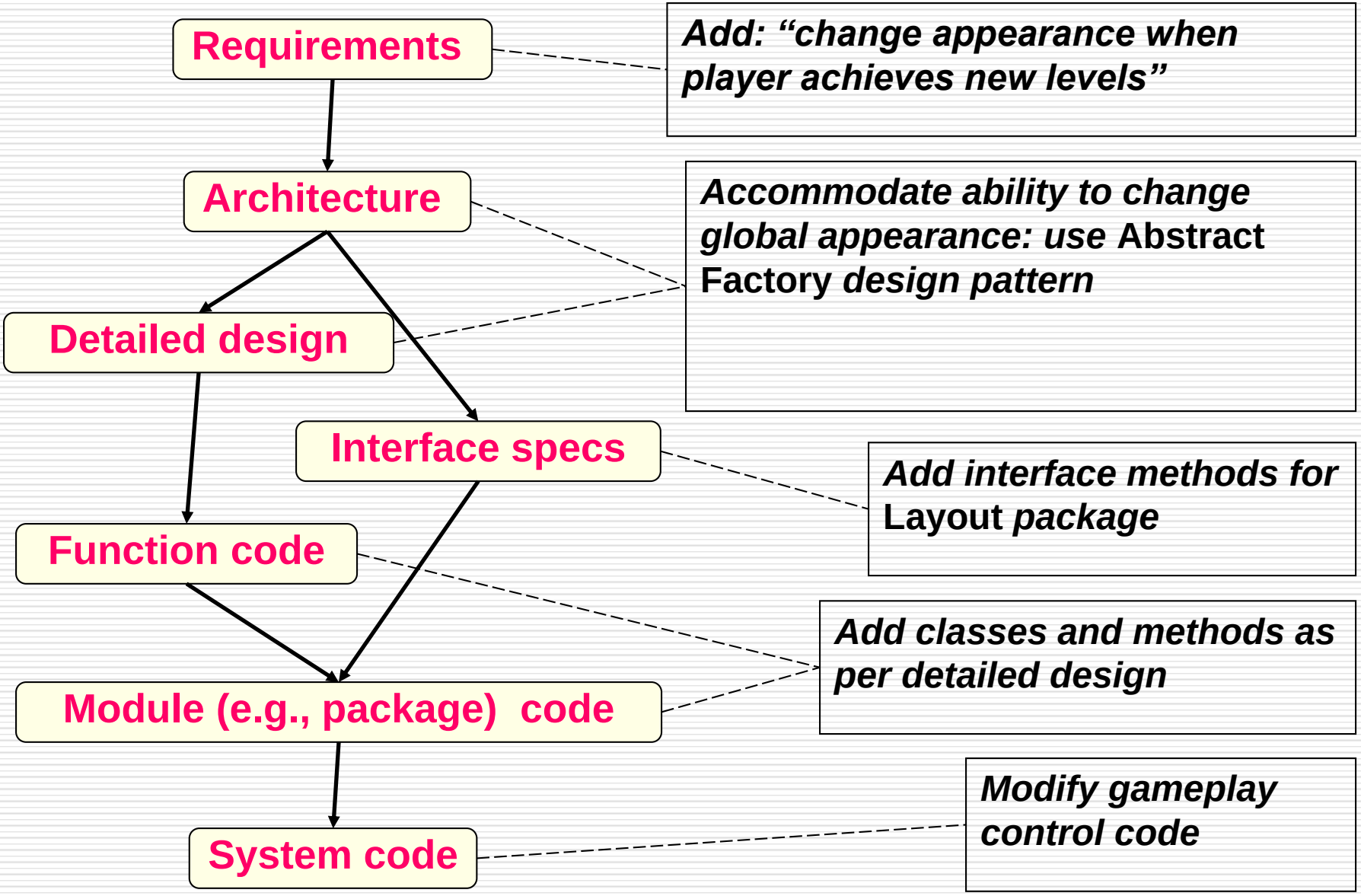
The Maintenance

- Change Management
  - ✓ Uniquely identify, describe and track the status of all change requests
- Impact Analysis
  - ✓ identifies all systems and system products affected by a change request
  - ✓ makes an estimate of the resources needed to effect the change
  - ✓ analyses the benefits of the change
- System Release Planning
  - ✓ to establish the schedule and contents of a system release
  - ✓ don’t want each change request released as they are processed

The Maintenance

- Change Implementation
  - ✓ Design Changes
  - ✓ Coding
  - ✓ Testing - must perform regression testing
- System Release includes
  - ✓ documentation
  - ✓ software
  - ✓ training
  - ✓ hardware changes
  - ✓ data conversion

小型化的软件开发过程


Requirements
Architecture
Detailed design
Interface specs
Function code
Module (e.g., package) code
System code
Add: “change appearance when player achieves new levels”
Accommodate ability to change global appearance: use Abstract Factory design pattern
Add interface methods for Layout package
Add classes and methods as per detailed design
Modify gameplay control code

维护记录

- ✓对什么进行了维护(what)
- ✓做了什么修改和调整(how)
- ✓谁及什么时候做的修改(who & when)
- ✓维护所耗费的成本

维护记录

对于维护记录中的内容，Swanson给出了下述的项目表：

（1）程序名称；

（2）源程序语句条数；

（3）机器代码指令条数；

（4）使用的程序设计语言；

（5）程序的安装日期；

（6）程序安装后的运行次数；

（7）与程序安装后运行次数有关的处理故障的次数；

（8）程序修改的层次和名称；

（9）由于程序修改而增加的源程序语句条数；

（10）由于程序修改而删除的源程序语句条数；

（11）每项修改所付出的“人时”数；

（12）程序修改的日期；

（13）软件维护人员的姓名；

（14）维护申请报告的名称；

（15）维护类型；

（16）维护开始时间和维护结束时间；

（17）用于维护的累计“人时”数；

（18）维护工作的净收益。

维护评估

一般来说，可以从以下七个方面来评价维护工作：

（1）每次程序运行时的平均出错次数；

（2）用于每一类维护活动的总“人时”数；

（3）每个程序、每种维护类型所做的平均修改数；

（4）维护过程中，增加或删除每条源程序语句花费的平均“人时”数；

（5）用于每种语言的平均“人时”数；

（6）一张维护申请报告的平均处理时间；

（7）各类维护类型所占的百分比。

再工程(Re-Engineering)