编程习惯

预备知识

注释清晰

结构清晰

可维护性

可测试性

鲁棒性

正确性

简洁

优雅

可读性

可扩展性

好代码

高效

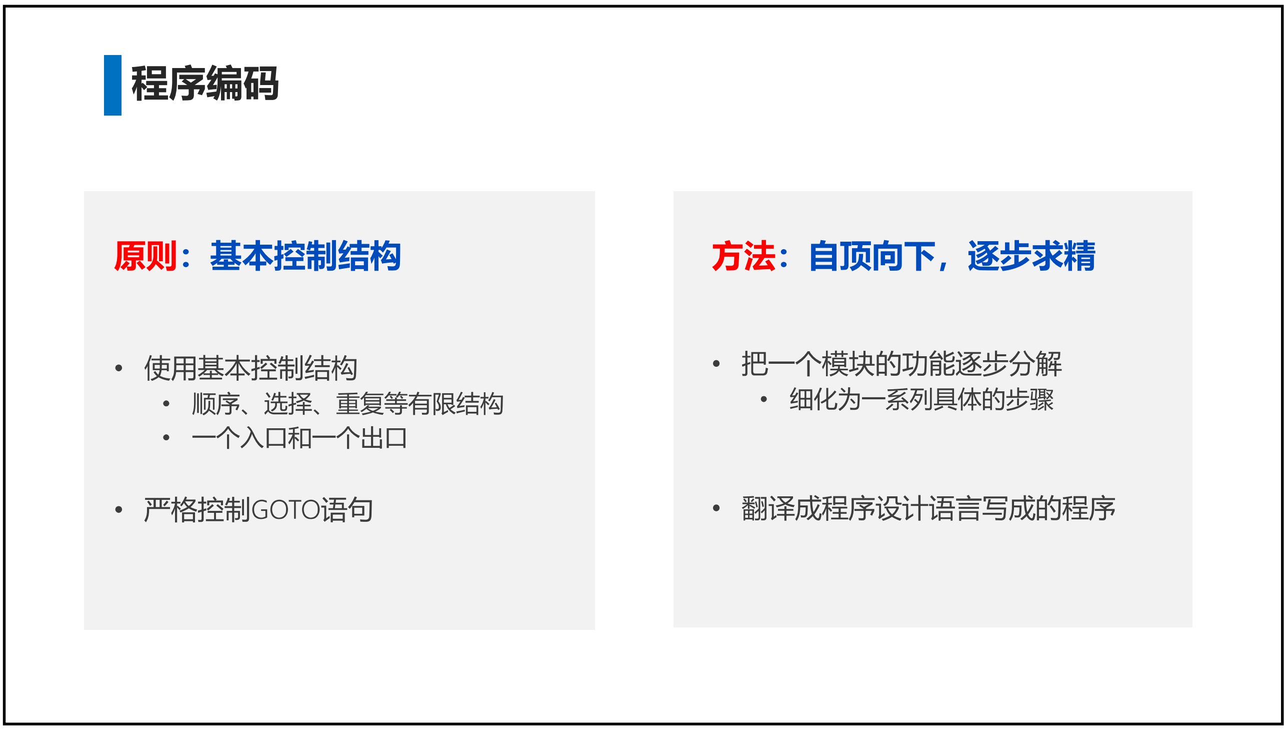
少即是多

低耦合

灵活性

高内聚

可复用性



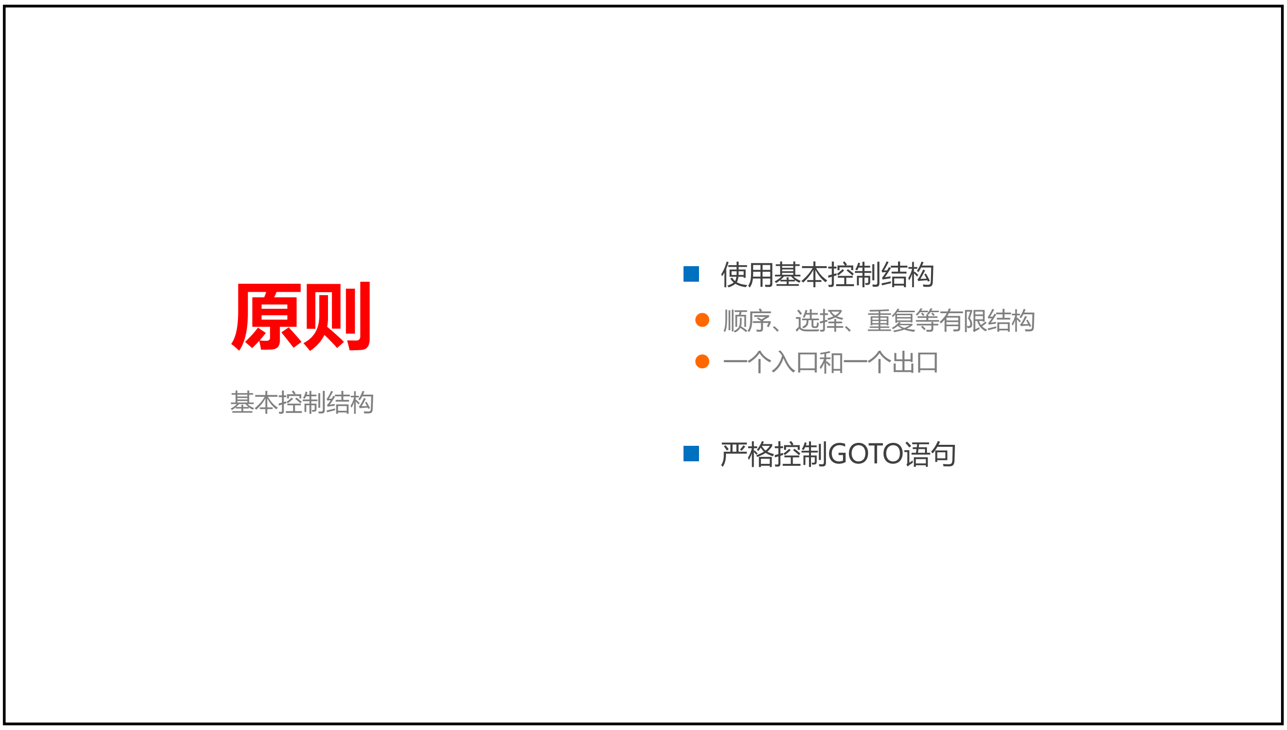
程序编码

原则：基本控制结构

- 使用基本控制结构
  - 顺序、选择、重复等有限结构
  - 一个入口和一个出口
- 严格控制GOTO语句

方法：自顶向下，逐步求精

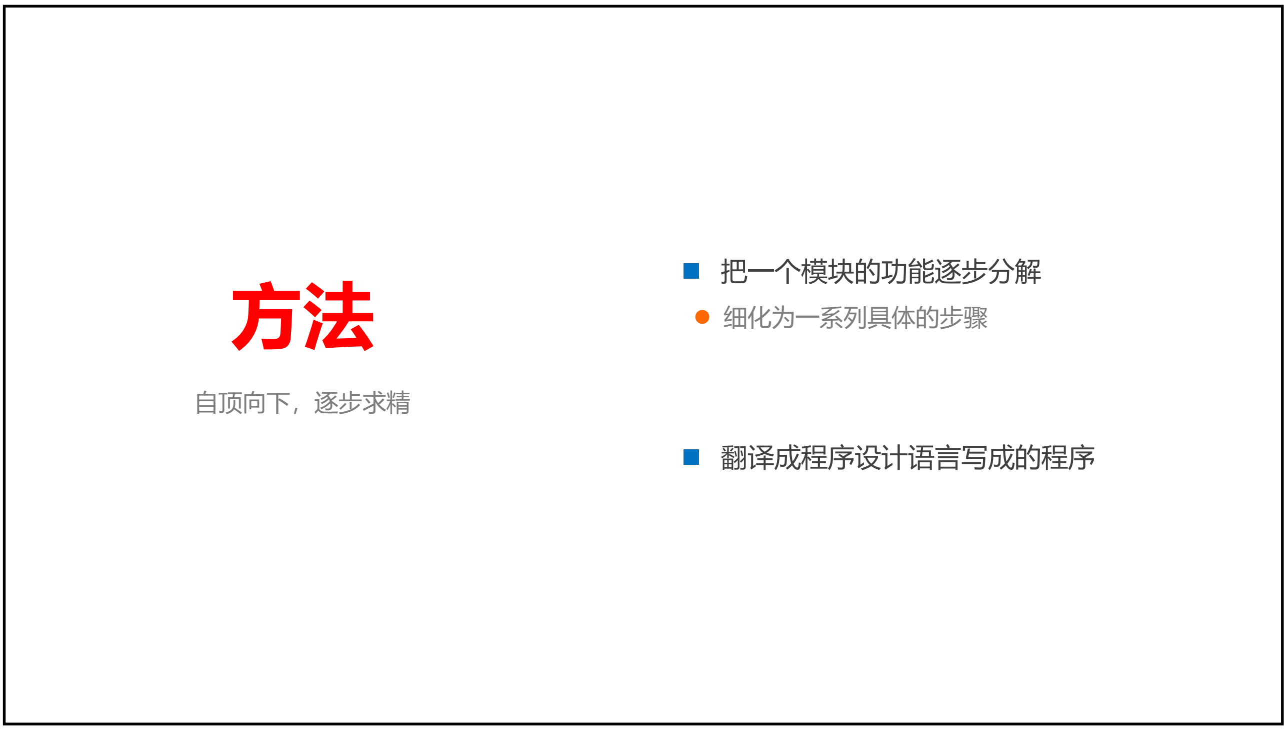
- 把一个模块的功能逐步分解
  - 细化为一系列具体的步骤
- 翻译成程序设计语言写成的程序



原则

基本控制结构

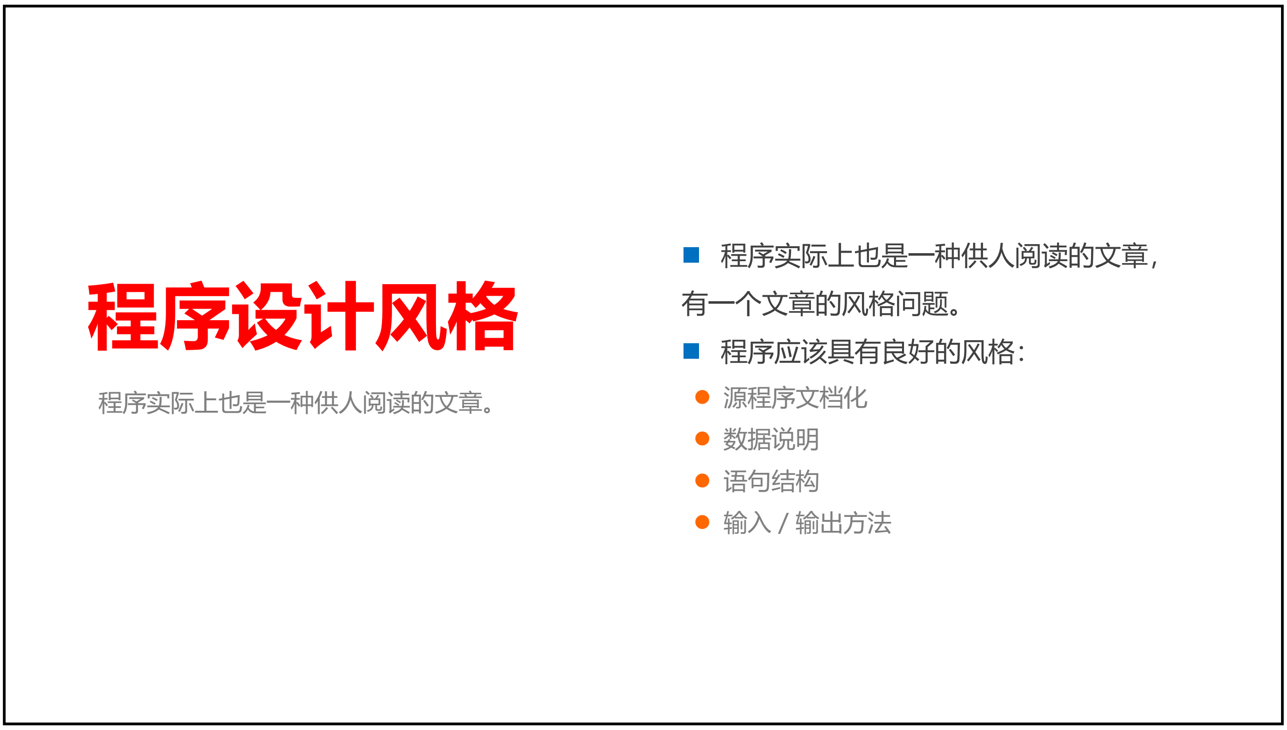
- 使用基本控制结构
  - 顺序、选择、重复等有限结构
  - 一个入口和一个出口
- 严格控制GOTO语句



方法

自顶向下，逐步求精

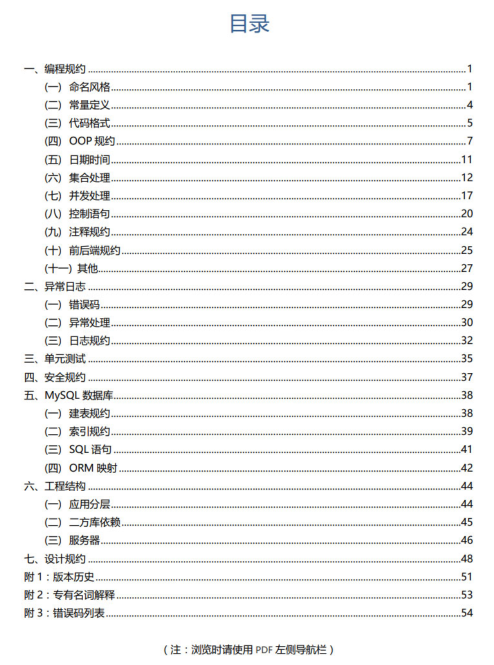
- 把一个模块的功能逐步分解
  - 细化为一系列具体的步骤
- 翻译成程序设计语言写成的程序



程序设计风格

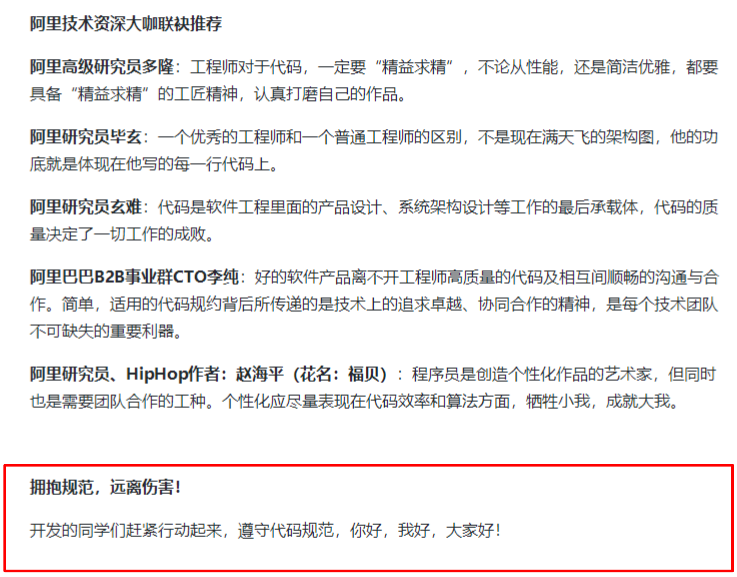
程序实际上也是一种供人阅读的文章。

- 程序实际上也是一种供人阅读的文章，有一个文章的风格问题。
- 程序应该具有良好的风格：
  - 源程序文档化
  - 数据说明
  - 语句结构
  - 输入／输出方法



目录

（注：浏览时请使用 PDF 左侧导航栏）



阿里技术资深大咖联袂推荐

阿里高级研究员多隆：工程师对于代码，一定要“精益求精”，不论从性能，还是简洁优雅，都要具备“精益求精”的工匠精神，认真打磨自己的作品。

阿里研究员毕玄：一个优秀的工程师和一个普通工程师的区别，不是现在满天飞的架构图，他的功底就是体现在他写的每一行代码上。

阿里研究员玄难：代码是软件工程里面的产品设计、系统架构设计等工作的最后承载体，代码的质量决定了一切工作的成败。

阿里巴巴B2B事业群CTO李纯：好的软件产品离不开工程师高质量的代码及相互间顺畅的沟通与合作。简单，适用的代码规约背后所传递的是技术上的追求卓越、协同合作的精神，是每个技术团队不可缺失的重要利器。

阿里研究员、HipHop作者：赵海平（花名：福贝）：程序员是创造个性化作品的艺术家，但同时也是需要团队合作的工种。个性化应尽量表现在代码效率和算法方面，牺牲小我，成就大我。

拥抱规范，远离伤害！

开发的同学们赶紧行动起来，遵守代码规范，你好，我好，大家好！

College of Computer Science, Chongqing University

Software Engineering

A Practitioner’ s Approach Seventh Edition

19 质量概念

zmqmail@cqu.edu.cn 13708390417

重庆大学计算机学院
COLLEGE OF COMPUTER SCIENCE

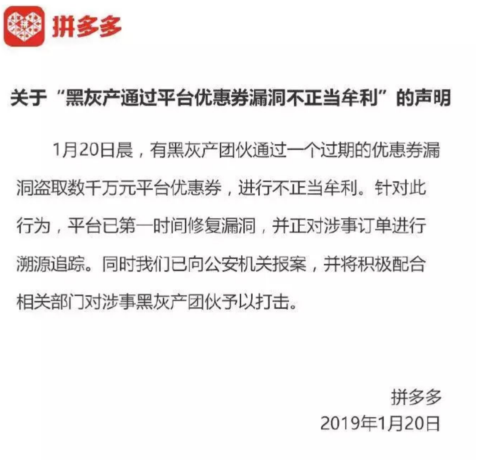
Quality Focus

支持软件工程的根基在于质量关注点

实现高质量软件的必须活动

拼多多被黑产薅羊毛事件

- 2019 年 1 月 20 日，微博爆料称拼多多出现重大 Bug：从网友晒出的图片看，此次 100 元无门槛券随便领，全场通用（特殊商品除外），有效期一年。
- 有网友表示，凌晨 3 点多被同行“喊醒”，让来拼多多“薅羊毛”，“只需支付 4 毛钱，就可以充值 100 元话费”。
- 在拼多多公关看来，此次被薅羊毛 200 亿的谣言是有心人在造谣抹黑；在旁观者看来，此次 200 亿谣言是拼多多的营销手段。



拼多多

关于“黑灰产通过平台优惠券漏洞不正当牟利”的声明

1月20日晨，有黑灰产团伙通过一个过期的优惠券漏洞盗取数千万元平台优惠券，进行不正当牟利。针对此行为，平台已第一时间修复漏洞，并正对涉事订单进行溯源追踪。同时我们已向公安机关报案，并将积极配合相关部门对涉事黑灰产团伙予以打击。

拼多多
2019年1月20日

波音 737 Max 客机软件故障坠机事件

- 2019 年 3 月 10 日，埃塞俄比亚航空公司一架波音 737 MAX 8 客机在飞往肯尼亚途中坠毁。机上有 149 名乘客和 8 名机组成员，无人生还。
  - 两次空难的影响因素都有该机型配置的自动控制下压机头的系统，其设计初衷是，如果机身上的传感器检测到高速失速的情况，即使在没有飞行员输入信号的情况下，该系统将强制将飞机的机头向下推。
  - 但在狮航空难事件中，该系统接收到了错误数据，导致飞机在正常情况下开始不断下压机头，飞行员在 11 分钟内连续手动拉升 20 余次终告失败，坠海罹难。

Bad Software

什么是劣质软件

- 只要每1000行代码有3或4处缺陷就能使程序执行很差
- 程序员每写10行代码注入一个错误
- 软件经销商把开发预算的一半花费在测试时修改错误上

19 软件质量 （Software Quality）

谁的责任?

- In 2005, ComputerWorld [Hil05] lamented that
  - “Bad software plagues nearly every organization that uses computers, causing lost work hours during computer downtime, lost or corrupted data, missed sales opportunities, high IT support and maintenance costs, and low customer satisfaction.
- A year later, InfoWorld [Fos06] wrote about the
  - “The sorry state of software quality” reporting that the quality problem had not gotten any better.
- Today, software quality remains an issue, but who is to **blame** (过失、指责) ?
  - 客户指责开发人员，认为草率的实践会导致低质量的软件。
  - 开发人员指责客户（和其他利益相关者），认为不合理的交付日期和持续不断的变化迫使他们在完全验证之前交付软件。

! 软件工程角度！

User Satisfaction = Compliant Product + Good Quality + Delivery Within Budget And Schedule

用户满意度=合格产品+好的质量+按预算和进度安排交付

Effective	Useful	Adding Value
高效	有用	高附加值

14.2.1 Garvin's Quality Dimensions
14.2.2 McCall's Quality Factors
14.2.3 ISO 9126 Quality Factors
14.2.4 Targeted Quality Factors
14.2.5 The Transition to a Quantitative View

Quality Dimensions(质量的维度 8个维度)

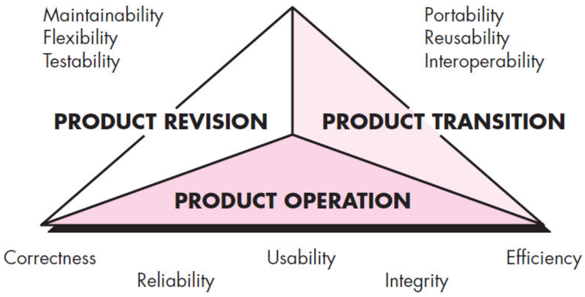
- David Garvin [Gar87]:
  - **Performance Quality**(性能质量). Does the software deliver all content, functions, and features that are specified as part of the requirements model in a way that provides value to the end-user?
  - **Feature quality(**特性质量). Does the software provide features that surprise and delight first-time end-users?
  - **Reliability**(可靠性). Does the software deliver all features and capability without failure? Is it available when it is needed? Does it deliver functionality that is error free?
  - **Conformance**(符合性). Does the software conform to local and external software standards that are relevant to the application? Does it conform to de facto design and coding conventions? For example, does the user interface conform to accepted design rules for menu selection or data input?

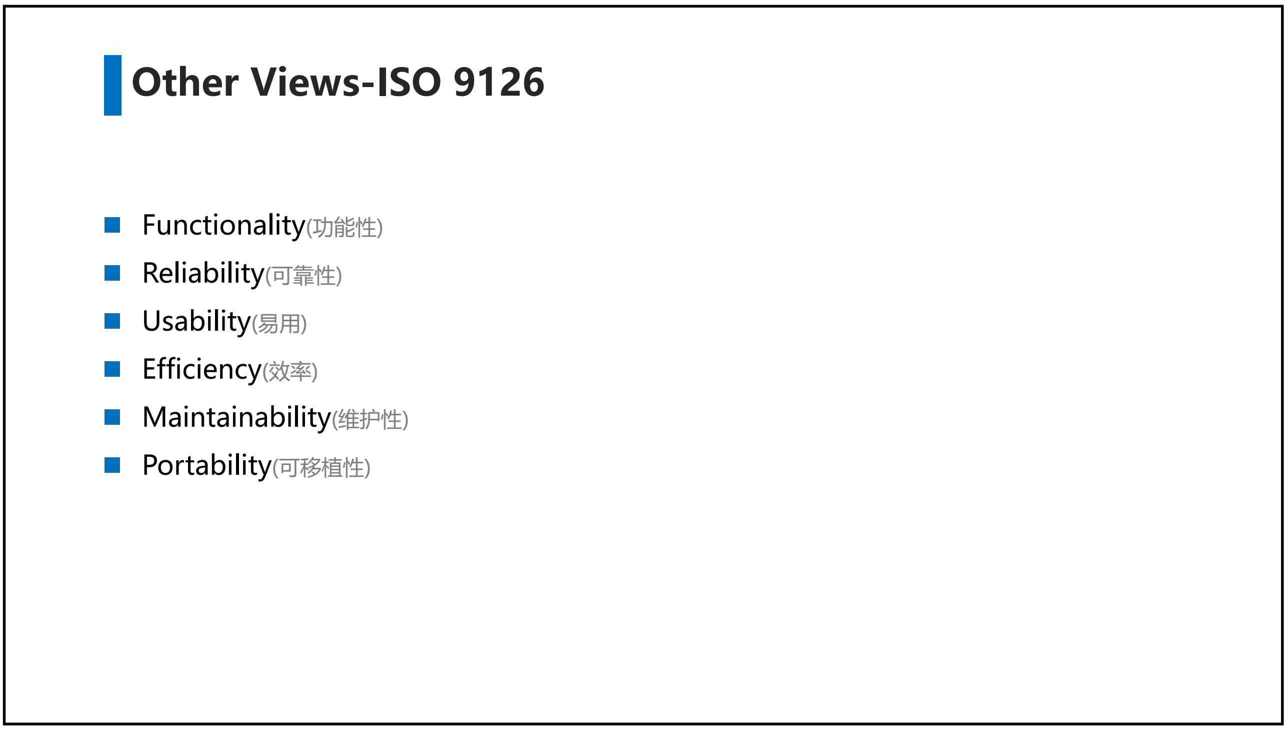
Quality Dimensions

- **Durability**(耐久性). Can the software be maintained (changed) or corrected (debugged) without the inadvertent generation of unintended side effects? Will changes cause the error rate or reliability to degrade with time?
- **Serviceability**(适用性). Can the software be maintained (changed) or corrected (debugged) in an acceptably short time period. Can support staff acquire all information they need to make changes or correct defects?
- **Aesthetics**(审美). Most of us would agree that an aesthetic entity has a certain elegance, a unique flow, and an obvious “presence” that are hard to quantify but evident nonetheless.
- **Perception**(感知). In some situations, you have a set of prejudices that will influence your perception of quality.

Other Views- McCall’s Quality Factors 11

- **Correctness**(正确性)
- **Reliability**(可靠性)
- **Efficiency**(效率)
- **Integrity**(完整性)
- **Usability**(易用性)
- **Maintainability**(维护性)
- **Flexibility**(灵活性)
- **Testability**(易测试性)
- **Portability**(可移植性)
- **Reusability**(可复用性)
- **Interoperability**(互操作性)





Other Views-ISO 9126

- Functionality(功能性)
- Reliability(可靠性)
- Usability(易用)
- Efficiency(效率)
- Maintainability(维护性)
- Portability(可移植性)

Other Views(定向质量因素)

- Intuitiveness(直觉)
- Efficiency(效率)
- Robustness(健壮性)
- Richness(丰富性)

19.3 软件质量困境

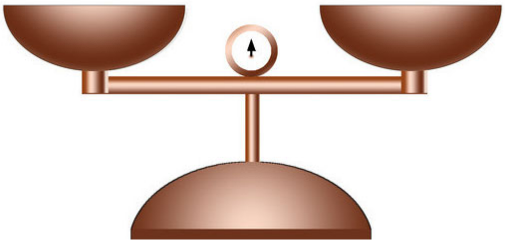
一种两难的困境?

Terrible Quality

No One Will Want To Buy

Absolutely Perfect

Huge Sums Of Money



“足够好”

“足够好”可能在某些应用领域和几个主要的软件公司其作用

定性→定量

度量的程度或者说精确度

19.3.2 Cost of Quality

- **预防成本** (**Prevention costs**) include
  - Quality planning
  - Formal technical reviews
  - Test equipment
  - Training
- **内部失效成本** (**Internal failure costs**) include
  - Rework
  - Repair
  - Failure mode analysis
- **外部失效成本** (**External failure costs**) are
  - Complaint resolution
  - Product return and replacement
  - Help line support
  - Warranty work

! 许多外部成本（如声誉）都难以量化!

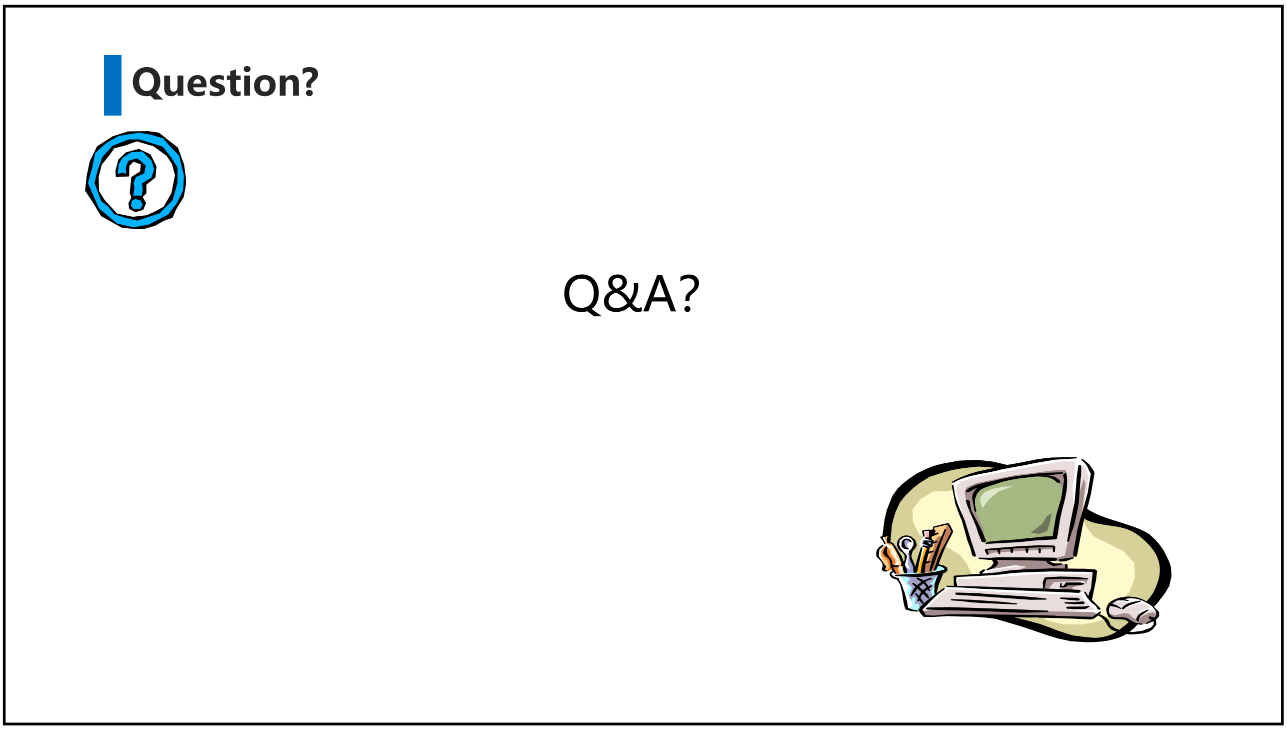
19.3.5 质量与安全（Quality and Security)

- 构造安全的系统，就必须注重质量，并在设计开始就进行关注。
- Gary McGraw comments [Wil05]:
  - “Software security relates entirely and completely to quality. You must think about security, reliability, availability, dependability—at the beginning, in the design, architecture, test, and coding phases, **all through the software life cycle** [process].
  - Even people aware of the software security problem have focused on late life-cycle stuff.
  - The earlier you find the software problem, the better.
  - There are two kinds of software problems.
    - One is bugs, which are implementation problems.
    - The other is software flaws—architectural problems in the design.
  - People pay too much attention to bugs and not enough on flaws.”

19.4 实现软件质量（Achieving Software Quality)

- Critical success factors:
  - Software Engineering Methods
    软件工程方法
  - Project Management Techniques
    项目管理技术
  - Quality Control
    质量控制
  - Quality Assurance
    质量保证

! **Ch 20 软件质量保证**



Question?

Q&A?

College of Computer Science, Chongqing University

Software Engineering

A Practitioner’ s Approach Seventh Edition

19 质量概念

zmqmail@cqu.edu.cn 13708390417