



Ch.19 Quality Concepts



19.1 What Is Quality

- The **transcendental view** argues (like Persig) that quality is something that you immediately recognize, but cannot explicitly define.
- The **user view** sees quality in terms of an end-user's specific **goals**. If a product meets those goals, it exhibits quality.
- The **manufacturer's view** defines quality in terms of the original **specification** of the product. If the product conforms to the spec, it exhibits quality.
- The **product view** suggests that quality can be tied to **inherent** characteristics (e.g., functions and features) of a product.
- Finally, the **value-based view** measures quality based on how much a customer is willing to **pay** for a product. In reality, quality encompasses all of these views and more.



质量

满足用户特定需求

符合制造要求

质量是产品的特定特征

用户愿意花多少钱买东西



19.2 Software Quality

- “**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” --- *Computer World* [Hi105]
- “The sorry state of software quality” reporting that the quality problem had not gotten any better--- *InfoWorld* [Fos06]
- **Today**, software quality **remains an issue**, but who is to blame?
- Customers** blame developers, arguing that **sloppy practices** lead to low-quality software.

对软件质量一直在抱怨

- Today, software quality **remains an issue**, but who is to blame?
- Customers** blame developers, arguing that **sloppy practices** lead to low-quality software.
- Developers** blame customers (and other stakeholders), arguing that **irrational delivery dates** and **a continuing stream of changes** force them to deliver software before it has been fully validated.

3



19.2 Software Quality

Quality

- The *American Heritage Dictionary* defines *quality* as
 - “a characteristic or attribute of something.”
- For software, two kinds of quality may be encountered:
 - **Quality of design** encompasses requirements, specifications, and the design of the system.
 - **Quality of conformance** is an issue focused primarily on implementation.
 - **User satisfaction** = compliant product + good quality + delivery within budget and schedule

4

将质量带来的危害降低到最低
设计质量包含设计质量、模块化程度
以及符合质量，有没有按照要求 coding 出来

用户满意度和质量相关，需要有一个和需求一样的产品，要有好的质量（非功能性要求），按照一定的预算和交付周期



19.2 Software Quality

- Software quality can be defined as:
 - An **effective software process** applied in a manner that **creates a useful product** that provides **measurable value** for those who produce it and those who use it.
- This definition has been adapted from [Bes04] and replaces a more manufacturing-oriented view presented in earlier editions of this book.

5

软件质量是一系列有效的软件过程的集合，创造有价值的产品，给生产者和使用者提供可以度量的价值



19.2 Software Quality

Effective Software Process

- An **effective software process** establishes the **infrastructure** that

有效的软件过程· 包含软件标准和注

Effective Software Process

- An **effective software process** establishes the **infrastructure** that supports any effort at building a high quality software product.
- The **management aspects of process** create the checks and balances that help avoid project chaos—a key contributor to poor quality.
- **Software engineering practices** allow the developer to analyze the problem and design a solid solution—both critical to building high quality software.
- Finally, **umbrella activities** such as change management and technical reviews have as much to do with quality as any other part of software engineering practice.

有效的软件过程：包含软件实现活动，例如coding、testing
各种管理手段：包括review, testing, 技术复审



6

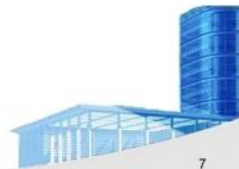


19.2 Software Quality

Useful Product

- A **useful product** delivers the content, functions, and features that the end-user desires
- But as important, it delivers these assets in a reliable, error free way.
- A useful product always satisfies those requirements that have been explicitly stated by stakeholders.
- In addition, it satisfies a set of **implicit** requirements (e.g., ease of use) that are expected of all high quality software.

目的是给用户创造有价值的产品
能够按照用户希望提供各种功能，
提供隐式需求，满足需求



7



19.2 Software Quality

Adding Value

- By **adding value for both the producer and user** of a software product, high quality software provides benefits for the software organization and the end-user community.
- The **software organization** gains added value because high quality software requires **less maintenance effort, fewer bug fixes, and reduced customer support.**
- The **user community** gains added value because the application provides a **useful capability** in a way that **expedites some business process.**
- The end result is:
 - (1) greater software product revenue,
 - (2) better profitability when an application supports a business process, and/or
 - (3) improved availability of information that is crucial for the business.

应该能够增加各种价值

1、给开发者

减少软件维护成本

给用户灭火要花很多经历

2、给用户

满足自己需求

利用软件赚钱，满足用户的需求





8



19.2 Software Quality

Quality Dimensions

- David Garvin [Gar87]:
 - **Performance Quality.** 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.** Does the software provide features that surprise and delight first-time end-users?
 - **Reliability.** 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.** 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?



9

性能质量：在输入之后多长时间内可以反馈

让用户感觉便利

可靠性

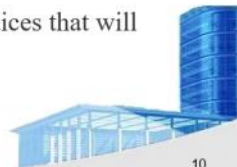
符合性指标，满足一定标准



19.2 Software Quality

Quality Dimensions

- **Durability.** 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.** 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.** 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.** In some situations, you have a set of prejudices that will influence your perception of quality.



10

可维护性：更改之后可以快速进行更新，适应变化

可服务性：出了问题之后可以快速进行更新

美观性

感觉



19.2 Software Quality

Measuring Quality

- General quality dimensions and factors are not adequate for assessing the quality of an application in concrete terms
- Project teams need to develop a set of targeted **questions** to assess the degree to which each application quality factor has been satisfied

the degree to which each application quality factor has been satisfied

- Subjective measures of software quality may be viewed as little more than personal opinion
- Software **metrics** represent indirect measures of some manifestation of quality and attempt to quantify the assessment of software quality

希望做定量分析来评判质量



11



19.3 The Software Quality Dilemma

- If you produce a software system that has terrible quality, you lose because no one will want to buy it.
- If on the other hand you spend infinite time, extremely large effort, and huge sums of money to build the absolutely perfect piece of software, then it's going to take so long to complete and it will be so expensive to produce that you'll be out of business anyway.
- Either you missed the market window, or you simply exhausted all your resources.
- So **people in industry** try to get to that **magical middle ground** where the product is good enough not to be rejected right away, such as during evaluation, but also not the object of so much perfectionism and so much work that it would take too long or cost too much to complete. [Ven03]

发布高质量软件是软件开发者的目标，也是用户的期望。

但是质量需要成本

时间窗口很重要，要做先锋



12



19.3 The Software Quality Dilemma

“Good Enough” Software

- **Good enough software** delivers **high quality functions and features** that end-users desire, **but at the same time it delivers other more obscure or specialized functions and features that contain known bugs.**
- Arguments *against* “good enough.”
 - It is true that “good enough” may work in some application domains and for a few major software companies. After all, if a company has a large marketing budget and can convince enough people to buy version 1.0, it has succeeded in locking them in.
 - If you work for a small company be wary of this philosophy. If you deliver a “good enough” (buggy) product, you risk permanent damage to your company's reputation.
 - You may never get a chance to deliver version 2.0 because bad buzz may cause your sales to plummet and your company to fold.
 - If you work in certain application domains (e.g., real time embedded software, application software that is integrated with hardware) can be negligent and open your company to expensive litigation.

提出中间值：足够好

不是极致，而是平衡点

即提供一定质量水准的软件，但不是说一点问题没有，可能存在一些非常用功能的缺失。保证一定代价之内，给用户提供基本可用的软件。

对于大企业比较从容，但是对小公司比较危险。因为小公司获客成本很高。



13



19.3 The Software Quality Dilemma

质量有质量的成本

需要覆盖整个生命周期

1. 预防性成本



19.3 The Software Quality Dilemma

Cost of Quality

- **Prevention costs** include
 - quality planning
 - formal technical reviews
 - test equipment
 - Training
- **Internal failure costs** include
 - rework
 - repair
 - failure mode analysis
- **External failure costs** are
 - complaint resolution
 - product return and replacement
 - help line support
 - warranty work

14

质量有质量的成本

需要覆盖整个生命周期

1、预防性成本

在设计制造阶段开展质量活动，例如指定质量计划（eg. 什么人？组织方式）/正式技术复审

各种测试

需要给开发人员进行training

2、内部故障成本：当质量事件发生后，需要定位，生成检测报告，要进行修复，要进行结果分析，总结

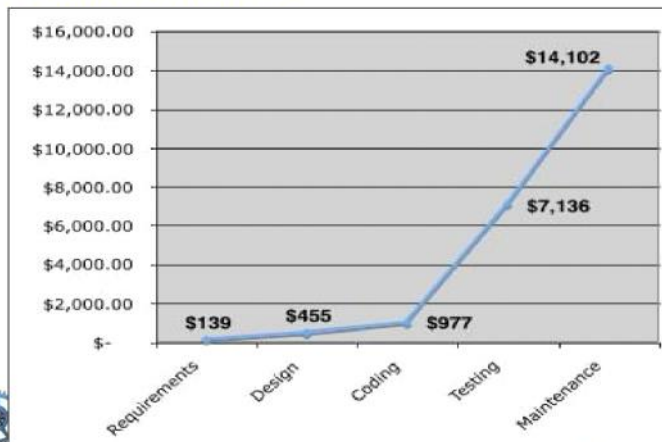
3、外部故障成本：安抚成本，很多软件是offline的，可能已经卖了很多设备，物流费也需要成本；客服成本；保障性活动



19.3 The Software Quality Dilemma

Cost

- The relative costs to find and repair an error or defect increase dramatically as we go from prevention to detection to internal failure to external failure costs.



15

质量问题是不可避免的，但是在不同阶段会有不同代价



19.3 The Software Quality Dilemma

Quality and Risk

- “People bet their jobs, their comforts, their safety, their entertainment, their decisions, and their very lives on computer software. It better be right.”
SEPA
- **Example:**
 - Throughout the month of November, 2000 at a hospital in Panama, 28 patients received massive overdoses of gamma rays during treatment for a variety of cancers. In the months that followed, five of these patients died from radiation poisoning and 15 others developed serious complications. What caused this tragedy? A software package, developed by a U.S. company, was modified by hospital technicians to compute modified doses of radiation for each patient.

risk可能很大

complications. What caused this tragedy? A software package, developed by a U.S. company, was modified by hospital technicians to compute modified doses of radiation for each patient.

16



19.3 The Software Quality Dilemma

Low Quality Software

- Low quality software increases risks for both developers and end-users
- When systems are delivered late, fail to deliver functionality, and does not meet customer expectations litigation ensues
- Low quality software is easier to hack and can increase the security risks for the application once deployed
- A secure system cannot be built without focusing on quality (security, reliability, dependability) during the design phase
- Low quality software is liable to contain architectural flaws as well as implementation problems (bugs)

17

糟糕的质量对开发者和用户都会带来很多问题

对开发者：法律诉讼；增加设备的安全隐患

对用户：达不到需求，体验不好



19.3 The Software Quality Dilemma

Impact of Management Decisions

- **Estimation decisions** – irrational delivery date estimates cause teams to take short-cuts that can lead to reduced product quality
- **Scheduling decisions** – failing to pay attention to task dependencies when creating the project schedule
- **Risk-oriented decisions** – reacting to each crisis as it arises rather than building in mechanisms to monitor risks may result in products having reduced quality

18

估算时间ddl可能实际是3个月，估计成了两个月。

调度问题，连接没有做好

风险管理没有做，每次都堵枪眼，没有做主动的谋划



19.4 Achieving Software Quality

- Software quality is the **result** of **good project management** and **solid engineering practice**.
- To build high quality software you must **understand the problem** to be solved and be capable of **creating a quality design** that conforms to the

需要有好的项目管理和坚实的工程实践

消除架构的缺陷

做很好的设计、质量控制的活动、

engineering practice.

- To build high quality software you must **understand the problem** to be solved and be capable of **creating a quality design** that conforms to the problem requirements.
- **Eliminating architectural flaws** during design can improve quality.
- **Project management** – project plan includes explicit techniques for quality and change management.
- **Quality control** - series of inspections, reviews, and tests used to ensure conformance of a work product to its specifications.
- **Quality assurance** - consists of the auditing and reporting procedures used to provide management with data needed to make proactive decisions.



19

消除架构的缺陷

做很好的设计、质量控制的活动、
质量保证的活动

