

软件质量模型（上）

软件质量

► 反映软件满足明确或隐含需要能力的特性总和



如何衡量?

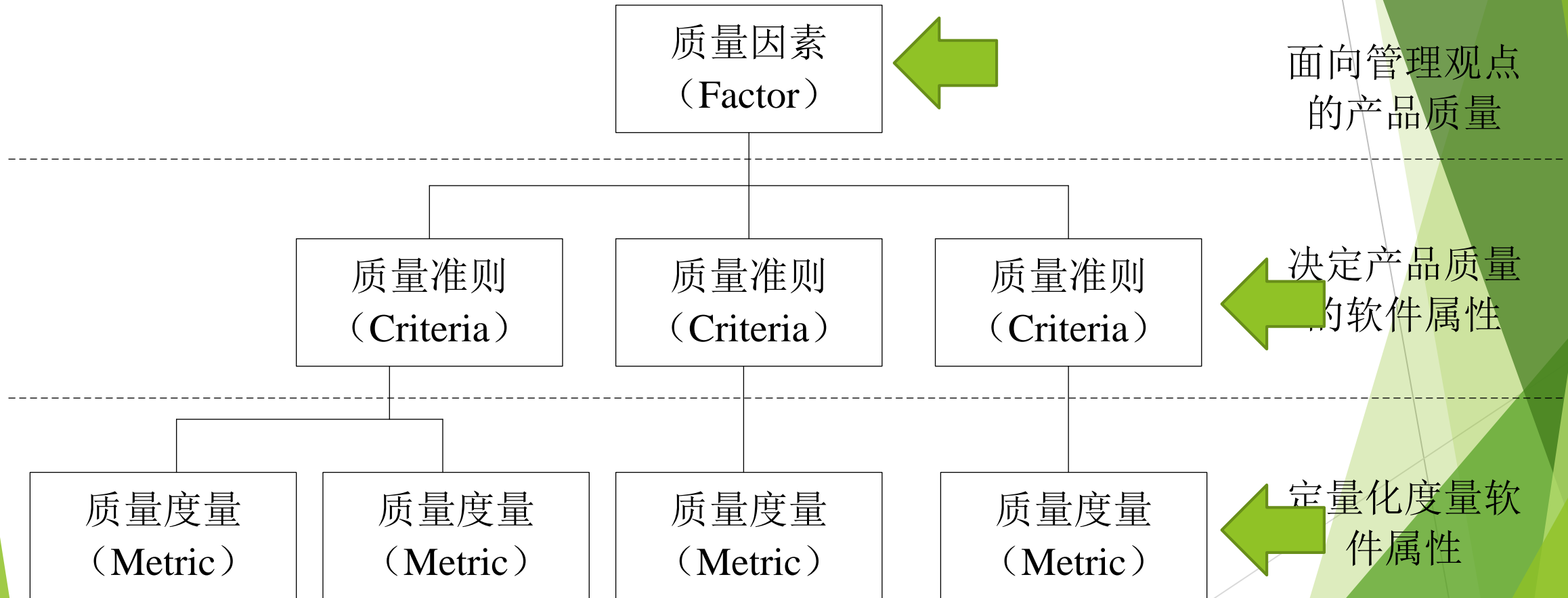
质量模型

- ▶ 基于经验的模型 (Experience-based models)
 - ▶ 根据经验, 使用典型的质量因素 (Quality factor) 来构建一个多层质量模型
- ▶ 基于构建的模型 (Construction-based models)
 - ▶ 通过提供一些方法来构建一个质量模型, 包括质量属性之间关系的构建和对质量属性进行分析

常见的质量模型

- ▶ 典型的基于经验的模型
 - ▶ 层次模型
 - ▶ McCall质量模型
 - ▶ Boehm质量模型
 - ▶ ISO9126质量模型
 - ▶ ISO25010质量模型
 - ▶ 关系模型
 - ▶ Perry模型
 - ▶ Gillies模型
- ▶ 典型的基于构建的模型
 - ▶ Dromey质量模型

McCall质量模型（也称GE模型，General Electrics Model，1977年，Jim A. McCall提出）



McCall 质量模型

质量因素
(Factor)

面向管理观点
的产品质量

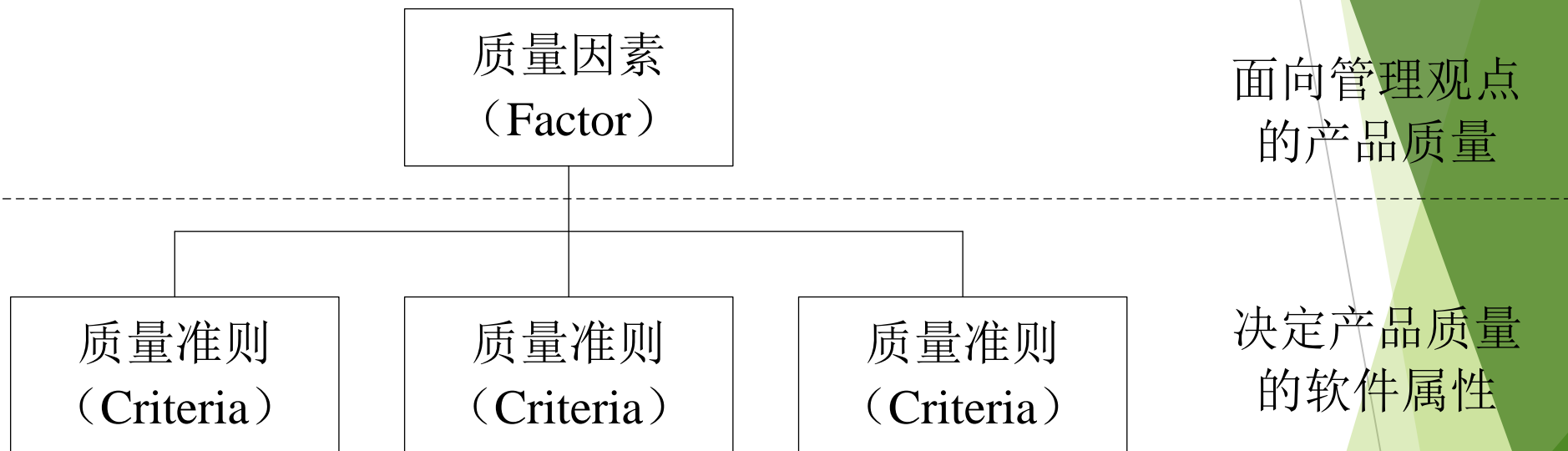
- ▶ 用户**不了解**软件内部实现细节
- ▶ 但用户**了解**自己的需求
- ▶ 用户从外部视角定义和描述软件 (**Specify**)
- ▶ **外部特性**: 从外部可以观察到的特性

质量/
(Met

产品质量
件属性

度量软
属性

McCall 质量模型

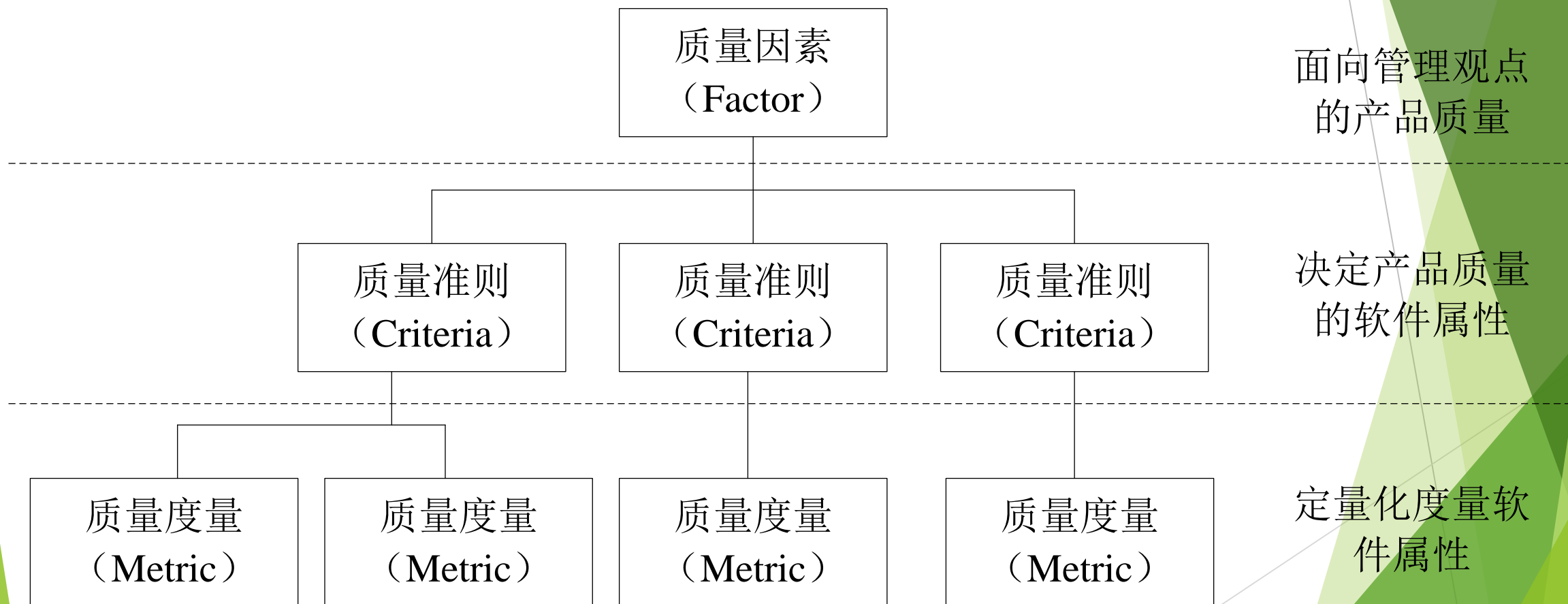


质量/
(Met

- ▶ 开发人员从内部视角构建软件属性 (**Build**)
- ▶ **内部特性**: 从内部可以观察到的特性

度量软
属性

McCall 质量模型





可维护性 Maintainability
灵活性 Flexibility
可测试性 Testability

产品修改
(Product
revision)

产品移植
(Product
transition)

可移植性 Portability
可重用性 Reusability
可互操作性 Interoperability

产品运行
(Product operation)

正确性 Correctness
完整性 Integrity
可靠性 Reliability
效率 Efficiency
可用性 Usability

产品运行
(Product operation)

正确性 Correctness

完整性 Integrity

可靠性 Reliability

效率 Efficiency

可用性 Usability

- 在预定环境下，软件满足需求规格说明及用户预期目标的程度，要求软件本身没有错误。

可追溯性 Traceability

完备性 Completeness

一致性 Consistency

正确性

可追溯性：追溯所考虑对象的历史、应用情况或所处场所的能力。

2006年，日本，戴尔笔记本起火爆炸事件



- ▶ 公司迅速查明原因，找到问题电池供应商：索尼公司，并追究责任。
- ▶ 非个例！！！切记勿将笔记本电脑当暖手宝使用.....

产品运行
(Product operation)

正确性 Correctness

完整性 Integrity

可靠性 Reliability

效率 Efficiency

可用性 Usability

▶ 在预定环境下，软件满足需求规格说明及用户预期目标的程度，要求软件本身没有错误。

可追溯性 Traceability

完备性 Completeness

一致性 Consistency

正确性

对软件的开发历史、软件发布后的情况进行跟踪的能力。
功能需求→功能模块→源代码→
测试用例→缺陷报告

产品运行
(Product operation)

正确性 Correctness

完整性 Integrity

可靠性 Reliability

效率 Efficiency

可用性 Usability

► 为某一目的而保护数据，避免它受到偶然的或有意的破坏、改动或遗失的能力。

完整性

访问控制 Access control

访问审查 Access audit

按照用户身份来限制用户对软件中部分数据的访问，或限制对某些控制功能的使用。

要素：主体、客体、访问授权。
应用系统通过权限引擎访问保护的资源。

产品运行
(Product operation)

正确性 Correctness
完整性 Integrity
可靠性 Reliability
效率 Efficiency
可用性 Usability

► 软件按照设计要求，在规定时间内和条件下不出故障，持续运行的程度。

可靠性

精度 Accuracy

容错性 Error tolerance

一致性 Consistency

防止由于软件系统的外部接口错误扩散而可能导致系统失效的能力。

产品运行
(Product operation)

正确性 Correctness
完整性 Integrity
可靠性 Reliability
效率 Efficiency
可用性 Usability

► 为了完成预定功能，软件系统所需的计算机资源的多少。

影响因素：

- 运行平台：硬件和软件环境
- 系统架构：开发机制，数据存储、处理和呈现方式，算法+数据结构

效率

运行效率
Execution efficiency

存储效率 Storage efficiency

产品运行
(Product operation)

正确性 Correctness
完整性 Integrity
可靠性 Reliability
效率 Efficiency
可用性 Usability

可用性

可操作性 Operability

训练性 Training

通信性 Communicativeness

► 对于一个软件系统，用户学习、使用软件以及为程序准备输入和解释输出所需工作量的大小。

- 易学：为学会使用软件所需要付出的工作量；
- 高效：学会使用软件后为达最高效地完成任务所付出的工作量；
- 好记：长时间不使用软件，需要重新使用而付出的工作量；
- 少错：用户在使用过程中会犯的错误数量、严重性、恢复难易度；
- 满意度：用户使用软件的主观感受。



Bill Gates (比尔盖茨)



1995年8月24日
微软公司推出Windows¹⁷95操作系统

