



Shanghai Jiao Tong University

软件过程与管理

Module: Software Measurement

软件度量

上海交通大学软件工程中心

问题

- ◆ 如何量化地管理软件开发？
- ◆ 针对具体的开发组织，如何设定度量方法和量化目标？
- ◆ 量化管理时应注意哪些问题？

本节内容

- ◆ 软件度量的概念
- ◆ 软件度量的模型
- ◆ 常用的软件度量
- ◆ 软件度量的步骤
- ◆ 软件度量的原则

软件度量的目的

更好地管理软件

- ◆ 认知：认知和理解过程、产品、资源和环境，建立比较基线；
- ◆ 评估：比较同步跟踪软件项目的状态，管理进展；及时发现实施和计划的偏差，评估质量目标的达成情况，以及技术和过程的改进对产品和过程的影响；
- ◆ 预测：是建立在适当资源下，达到成本、进度和质量目标的计划的基础。也可根据度量的实证，预测项目发展的倾向，估计分析风险，做出设计 / 成本权衡；
- ◆ 改进：帮助识别问题根源，判断可以改进的机会，交流改进的目标和理由等；调整资源分配等。

度量的定义

- ◆ 度量关注的是在一定规则下获取关于实体属性的信息；
- ◆ 一个实体可以是一个实物，如人或房间；或者是一个事件，如旅行；或软件项目的测试阶段；
- ◆ 属性是我们所关注的实体的特征或特性。

度量的分类

- ◆ 过程/产品 process/product
- ◆ 主观/客观 objective/subjective
- ◆ 直接/间接 explicit/derived
- ◆ 绝对/相对 absolute/relative
- ◆ 动态/静态 dynamic/static
- ◆ 预测/解释 predictive/explanatory

按度量对象分

- ◆ 过程度量，过程能力指标的合理量化，测量组织级别上软件开发过程的质量、成本、盈利、投资回报率 and 生产率。
- ◆ 项目度量，具体项目指标的合理量化，测量项目级别上软件开发过程的质量、成本、盈利、投资回报率 and 生产率。
- ◆ 产品度量，目标或中间产品指标的合理量化，测量产品的质量 and 复杂度。如代码长度(KLOC)、功能点(FP)、McCabe cyclomatic复杂度等。
- ◆ 资源度量，资源使用指标的量化，测量资源的利用 and 分布的合理性。如库存周转、厂房利用、劳动力使用和培训情况、团队协作度等。



按度量主客观性分

- ◆ 客观度量是指所得到的关于某对象的度量值是该对象的真实描述，不会因度量人的不同、所使用工具的不同而产生差异。例如LOC
- ◆ 主观度量是指所得到的关于某对象的度量值是由度量的实施者的主观判断得到的，因此所得到的度量值会因度量人的不同而不同，它受环境、人的情绪、人际关系等各种因素的影响，从严格的意义上讲是不可确定的度量。例如度量：“系统的易学习性”

按度量方式分

- ◆ 直接度量，是指不依赖于任何其它属性的度量值便可以计算得到的度量。
 - LOC
 - execution speed
 - memory size
 - defects
- ◆ 间接度量，是指需要依赖本实体或其它实体的一个或多个属性的度量值才可得到的度量。
 - functionality, quality, complexity
 - efficiency, reliability, maintainability

举例

项目	工作量	元	KLOC	文档页数	错误数	人数
aaa-01	24	168	12.1	365	29	3
ccc-04	62	440	27.2	1224	86	5
fff-03	43	314	20.2	1050	64	6
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮

问：使用了哪些度量？

举例 进一步计算

- ◆ 根据数据表格可以对所有的项目计算出平均值：
 - 生产率 = $KLOC / PM$ (人月)
 - 质量 = $错误数 / KLOC$
 - 成本 = $元 / KLOC$
 - 文档 = $文档页数 / KLOC$

本节内容

- ◆ 软件度量的概念
- ◆ 软件度量的模型
- ◆ 常用的软件度量
- ◆ 软件度量的步骤
- ◆ 软件度量的原则

产品度量和FCM度量模型



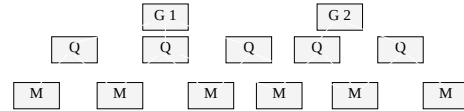
- ◆ 特性/因子(Factor)
- ◆ 属性/子因子(Criteria)
- ◆ 度量(Metrics)

Software Process and Management

13

沈备军

过程度量和GQM度量模型



- ◆ GQM (Goal-Question-Metric) 模型是基于这样的假设：对一个机构而言，度量是应当有目的性的，即它应首先定义其自身或该机构内某个项目的目标，根据这些目标去跟踪那些定义这些目标的数据，最后提供一个框架用于解释这些数据与所确定的目标之间的关系。

Software Process and Management

14

沈备军

GQM度量模型

- ◆ 第一层：概念层（目标G）。包含一个目标、目标涉及的度量对象及其属性等信息。目标所涉及的对象类型有：
 - 产品（Product）：软件生存期中产生、发布的各种软件产品、文档等；
 - 过程（Process）：与软件生产相关的各种活动；
 - 资源（Resource）：被过程所利用，以便产生其输出产品的所有对象，如人、硬件、软件、办公空间等。
- ◆ 第二层：运作层（问题Q）。问题是有上一层的目标细化而来，用于描述实现该目标的方式。一个目标能细化出多个问题。
- ◆ 第三层：量化层（度量M）。一组以量化的方式回答上层问题的数据。这组数据可以是：
 - 客观的：若该组数据全部是通过对象的测量获得的，而不是以什么人的主观认识取得的；
 - 主观的：若该组数据是通过实测和人们的主观判断获得的。

Software Process and Management

15

沈备军

举例

- ◆ 目标：分析开发过程中的软件错误，以便找出降低成本的可能点
- ◆ 问题：
 - Q1：在哪些地方出现了错误？
 - Q2：这些错误的来源在哪里？
 - Q3：发现的错误是否得到更改？
 - Q4：从发现错误、确认错误、到更改错误所需要的时间？
 - Q5：从过程改进小组的角度看，目前的纠错、改错过程是否合理？
- ◆ 度量：(Q5得到的度量)
 - M51（错误延迟）= $\frac{\text{错误呈现时间} - \text{错误产生时间}}{\text{过程周期}}$
 - M52（纠错人工）= $\frac{\sum \text{用于纠错的时间}}{\text{所有参加纠错的人}}$

Software Process and Management

16

沈备军

本节内容

- ◆ 软件度量的概念
- ◆ 软件度量的模型
- ✧ ◆ 常用的软件度量
- ◆ 软件度量的原则

Software Process and Management

17

沈备军

常用软件度量方法（规模）

- ◆ 度量产品规模：
 - 产品的代码行数LOC
 - 产品的非注解代码行数NLOC
 - 产品的功能点数FP
 - 产品的类个数
 - 文档页数

产品度量

Software Process and Management

18

沈备军

采用LOC的注意点

- ◆ 要采用相同的计数方法
- ◆ 不同编码语言的生产率和缺陷率是不同的
- ◆ 不同应用领域、不同复杂度的代码的生产率是不同的
- ◆ 聪明快速的方法可导致代码较少
- ◆ 要考虑代码复用的因素
- ◆ 不要过分强调LOC单个因素的度量，忽略了质量等因素

常用软件度量方法（复杂性）

- ◆ 度量产品复杂性：
 - 软件复杂性度量涉及到需求的复杂性、设计的复杂性、程序的复杂性等。
 - 软件复杂性度量可以用来预计关于从源代码（或设计信息）的自动分析得到的软件系统的可靠性和可维护性。复杂度度量同时也在软件项目中提供反馈以帮助控制设计活动。在测试和维护中，它们提供了模块的详细信息以帮助指出潜在的不稳定的地方。
- ◆ 如程序复杂性度量
 - McCabe度量：程序控制流的循环复杂度(Cyclomatic Complexity)
 - Halstead度量：程序中运算符和操作数的总数

产品度量

McCabe 复杂度度量

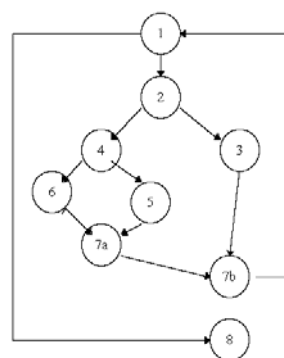
- ◆ 又称圈复杂度（Cyclomatic Complexity）
- ◆ 计算公式：
 - $M = e - n + 2p$
 - 结构化程度：# binary decisions + 1.
 - 控制图中IP(independent paths)的个数
 - 控制图中区域的个数
- ◆ 变量：
 - e 控制图的边数
 - n 控制图的节点数
 - p 控制图的不连通部分数

举例

```

1:  do while records remain
    read record;
2:    if record field 1 = 0
3:      then process record;
        store in buffer;
        increment counter;
4:    elsif record field 2 = 0
5:      then reset record;
6:      else process record;
        store in file;
7a:    endif
    endif
7b:  enddo
8:  end

```



圈复杂度=4

常用软件度量方法（进度）

- ◆ 度量项目的进度情况：
 - 测试的代码覆盖率
 - 开发的需求覆盖率
 - 里程碑成功达到的数目、比例和延误时间
 - 按期开始或结束的项目数、任务数和相应比例
 - 任务延误的时间

项目度量

常用软件度量方法（生产率）

- ◆ 度量产品规模（价值）与人力投入（费用）的比率：
 - 每人月平均非注解代码行数
 - 每人月平均功能点数
 - 每人月平均类个数
 - 每人月平均文档页数
 - 每人月平均测试的单元数

项目和过程度量

常用软件度量方法（可靠性）

- ◆ 可行性 - 指出可行时间百分比 (xx.xx%)、使用小时数、维护访问权、降级模式操作等。
- ◆ 平均故障间隔时间 (MTBF) - 通常表示为小时数，但也可表示为天数、月数或年数。
- ◆ 平均修复时间 (MTTR) - 系统在发生故障后可以暂停运行的时间。
- ◆ 精确度 - 指出系统输出要求具备的精确度（分辨率）和精确度（按照某一已知的标准）。
- ◆ 缺陷率 - 通常表示为每千行代码的错误数目或每个功能点的错误数目。

产品质量

Software Process and Management

25

沈备军

常用软件度量方法（缺陷）

- ◆ 缺陷密度和分布
- ◆ 缺陷排除有效性
- ◆ 缺陷发现效率

Software Process and Management

26

沈备军

缺陷密度和分布

- ◆ 缺陷数目、密度（千行缺陷数）
- ◆ 缺陷按严重度的分布(严重、一般、轻微)
- ◆ 缺陷按起因的分布（设计错误、代码错误、接口错误、硬件错误等）
- ◆ 缺陷在不同开发阶段的分布（需求、设计、编码、发布前、发布后等）
- ◆ 缺陷按模块的分布

Software Process and Management

27

沈备军

缺陷排除有效性(DRE)

- ◆ 缺陷排除有效性(DRE) 是对质量保证及控制活动的过滤能力的一个测量，这些活动贯穿于整个过程。
- ◆ 当把一个项目作为一个整体来考虑时，DRE按如下方式定义：

$$DRE = E / (E + D)$$

其中E=软件交付给最终用户之前所发现的错误数

D=软件交付之后所发现的缺陷数

Software Process and Management

28

沈备军

DRE值

- ◆ 最理想，DRE值是1，即软件中没有发现缺陷。
- ◆ 现实中，D会大于0，但随着E值的增加，DRE的值仍能接近1。
- ◆ 事实上，随着E的增加，D的最终值可能会降低(错误在变成缺陷之前已经被过滤了)。
- ◆ DRE鼓励软件项目组采用先进技术，以便在交付之前发现尽可能多的错误。

Software Process and Management

29

沈备军

缺陷发现效率

- ◆ 单位工作量或时间发现缺陷的数目
- ◆ 公式：
缺陷数/发现缺陷所化的工作量或时间

Software Process and Management

30

沈备军

常用软件度量方法（有效性）

- ◆ 有效性度量（即性能度量），包括时间和空间的有效性
- ◆ 度量方法
 - 从产品外部：
 - 对事务的响应时间（平均、最长）
 - 吞吐量（例如每秒处理的事务数）
 - 容量（例如系统可以容纳的客户或事务数）
 - 资源利用情况：内存、磁盘、通信等
 - 从产品内部：
 - 算法的有效性，即算法的复杂性

面向对象基本度量法

- ◆ LOC (lines of code)
 - 类的代码行数。
- ◆ NOA (number of attributes)
 - 类的属性个数。
- ◆ NCV (number of class variables)
 - 类的静态属性个数。
- ◆ NIV (number of instance variables)
 - 类的非静态属性个数。
- ◆ NOM (number of methods)
 - 类的方法个数。
- ◆ NCM (number of class methods)
 - 类的静态方法个数。
- ◆ NIM (number of instance methods)
 - 类的非静态方法个数。

CK度量法

- ◆ WMC (Weight method per class)
- ◆ DIT (Depth of the inheritance tree)
- ◆ NOC (Number of children)
- ◆ CBO (Coupling between object)
- ◆ RFC (Response for a class)
- ◆ LCOM (Lack of cohesion in methods)

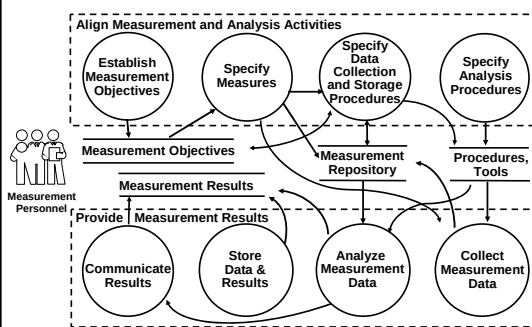
最基本的度量

S.R.Schach, 1992	U.S. Air Force, 93M-017报告	常用单位
Size 规模	Size 规模	LOC, KLOC
Effort 工作量	Effort 工作量	人-月
Duration 时间	Schedule 进度	月
Quality 质量	Quality 质量	错误数 / KLOC
Cost 成本	Rework 返工	元

本节内容

- ◆ 软件度量的概念
- ◆ 软件度量的模型
- ◆ 常用的软件度量
- ◆ 软件度量的步骤
- ◆ 软件度量的原则

Measurement & Analysis – CMMI PA



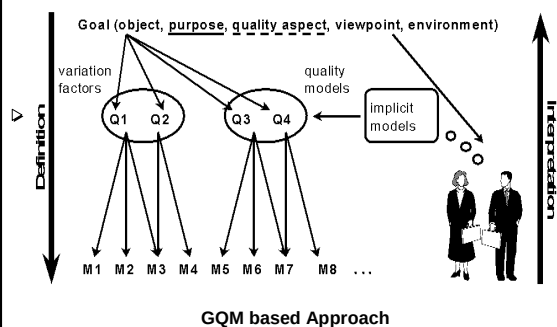
软件度量的步骤

1. Establishing and maintaining measurement objectives
2. Specifying measures
3. Specifying procedures for data collection, storage, and analysis
4. Collecting measurement data
5. Analyzing measurement data
6. Storing data and results
7. Reporting results to stakeholders

Step1 Establishing Measurement Objectives

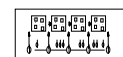
- ◆ Measurements only produce numbers
- ◆ To be useful, they must
 - relate to business objectives
 - be properly interpreted
 - lead to appropriate action
- ◆ If the business purposes for the measurements are not understood
 - the wrong data may be gathered
 - the data may not be properly used

Step 2 Specifying Measures



Step 3 Specifying Procedures

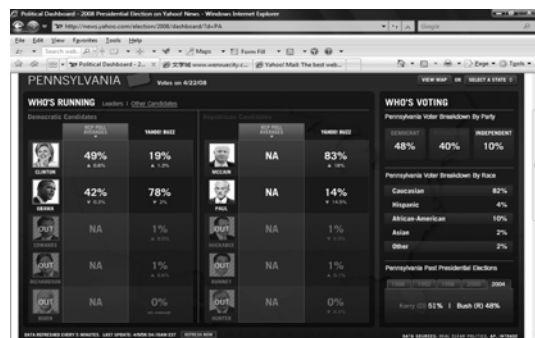
- ◆ Specify data collection procedures
 - Explicit specification of collection methods helps ensure that the right data are collected properly.
- ◆ Specify data storage procedures
 - Proper attention to storage and retrieval procedures helps ensure that data are available and accessible for future use.
- ◆ Specify data analysis procedures
 - Specifying the analysis procedures ensures that appropriate analyses will be conducted and reported to address the measurement objectives.



Step 4 Collecting Measurement Data

- ◆ Obtain the data for base measures
- ◆ Generate the data for derived measures
- ◆ Perform data integrity checks as close to the source of the data as possible.
 - Checks can include scans for missing data, out-of-bounds data values, and unusual patterns and correlation across measures

Which poll result you trust more?



Measurement Scales

- ♦ The collected data usually be calculated or analyzed statistically. So it is important to consider its scale type.

Type of Data	Possible Operations	Description of Data
Nominal	=, ≠	Categories
Ordinal	<, >	Rankings
Interval	+, -	Differences
Ratio	/	Absolute zero

Software Process and Management

43

沈备军

Measurement Scale Example

- ♦ Nominal
 - type of program type: OS, DB, Application...
- ♦ Ordinal
 - programmer experience level: low, *medium*, or high
- ♦ Interval
 - McCabe's complexity
 - There is no absolute zero. A program with complexity value of 6 is 4 units more complex than a program with complexity of 2, but it is probably not meaningful to say that the first program is three times as complex as the second.
- ♦ Ratio
 - LOC
 - A program of 2,000 lines can reasonably be interpreted as being twice as large as a program of 1,000 lines

Software Process and Management

44

沈备军

Data Collection Considerations

- ♦ Automation
- ♦ A personal process notebook
- ♦ Forms and templates
- ♦ Databases and spreadsheets
- ♦ Summary project report

Software Process and Management

45

沈备军

Step 5 Analyzing Measurement Data

- ♦ Conduct initial analyses, interpret the results, and draw preliminary conclusions
 - The results of data analyses are rarely self evident. Criteria for interpreting the results and drawing conclusions should be stated explicitly.
- ♦ Conduct additional measurement and analysis as necessary, and prepare results for presentation.
 - The results of planned analyses may suggest (or require) additional, unanticipated analyses.

Software Process and Management

46

沈备军

Step 6 Storing Data and Results

- ♦ Storing measurement-related information enables the timely and cost-effective future use of historical data and results.
- ♦ Information stored typically includes:
 - Measurement plans
 - Specifications of measures
 - Sets of data that have been collected
 - Analysis reports and presentations

Software Process and Management

47

沈备军

Step 7 Reporting Results to Stakeholders

- ♦ The results of the measurement and analysis process are communicated to relevant stakeholders in a timely and usable fashion to support decision making and assist in taking corrective action.
 - Keep relevant stakeholders apprised of measurement results on a timely basis.
 - Assist relevant stakeholders in understanding the results.

Software Process and Management

48

沈备军

本节内容

- ◆ 软件度量的概念
- ◆ 软件度量的模型
- ◆ 常用的软件度量
- ◆ 软件度量的步骤
- ◆ 软件度量的原则

如何使用软件度量

- ◆ 基于业务目标识别度量目标
- ◆ 根据GQM选择出合适的度量方法
- ◆ 建立历史数据基线，设立经验度量目标
- ◆ 在整个开发过程中，定期地(特别是在软件生命周期的基线和里程碑处)收集过程的执行数据，以准确地了解一个过程的实际结果(过程性能)以及该过程偏离期望结果(过程能力)的范围
- ◆ 整理、计算和评估度量值
- ◆ 将度量结果反馈给管理者和开发者，如项目阶段度量报告、项目度量报告、月度量报告、年度量报告
- ◆ 根据度量结果，调整计划和改进过程

软件度量原则

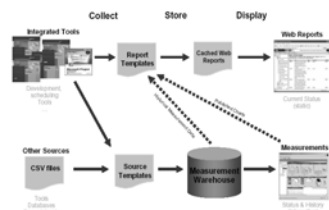
- ◆ 开始采用较少的度量，根据GQM从度量目标、问题导出度量
- ◆ 对收集测量和度量的个人及小组提供定期的反馈
- ◆ 不要使用度量去评价个人
- ◆ 与开发者和小组一起设定清晰的目标及达到这些目标的度量
- ◆ 不要用度量去威胁个人或小组
- ◆ 指出某个问题的度量数据不应该被看成是“否定的”含义。这些数据仅仅是过程改进的指标
- ◆ 不要被某个与其他重要度量不符合的度量迷惑

注意事项

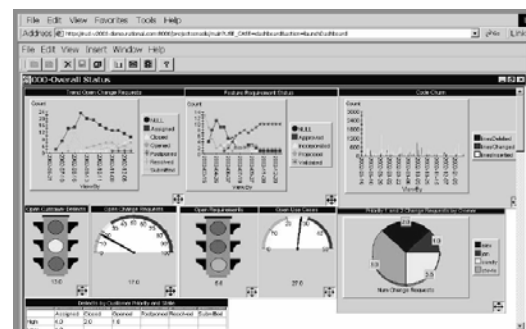
- ◆ 高级管理层的注意事项：
 - 度量目标必须明确，但永远不能使用度量值去奖励或惩罚进行度量的人；
- ◆ 项目管理员的注意事项：
 - 不要特别强调某一个度量，而不顾其他。不要度量个人；
- ◆ 项目团队成员的注意事项：
 - 提供精确、及时的数据，不要用软件测量值吹捧或贬低别人；

Software Measurement Tools

- ◆ Rational Project Console
- ◆ Man Machine System's JStyle
- ◆ LOC (freeware)
- ◆ PMD (Freeware)
- ◆



Rational Project Console



结束语

“没有度量就没有控制”

--Tom Demarco

讨论：软件度量

- ◆ 哪些度量应个人保留？哪些度量应是公开的？各举三个例子。
- ◆ 项目组A在最终的系统测试时发现了384个bug,项目组B则发现了184个。要决定哪个项目组改正bug更有效，应采用哪些度量？可以用到哪些历史数据？
- ◆ 尝试设计新的度量？