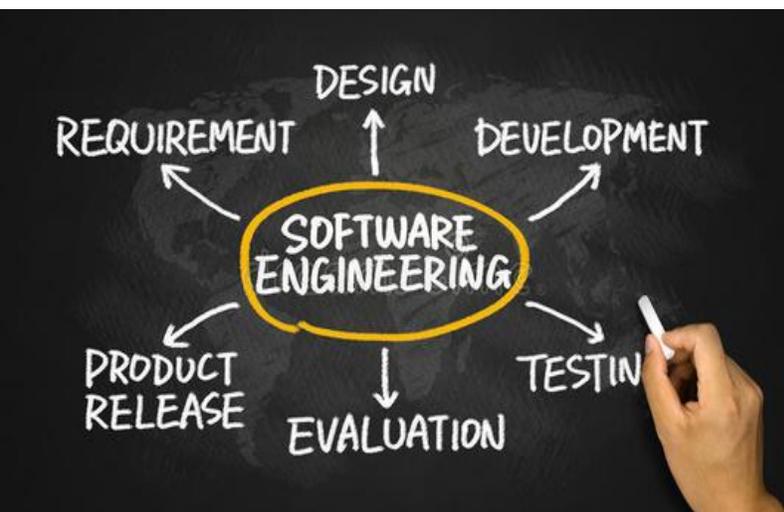




软件工程基础

—— 第23章 过程度量 and 项目度量



计算机学院 孟宇龙

23.1 过程领域和项目领域中的度量

23.2 软件测量

23.3 软件质量的度量

关键概念

- 缺陷排除效率
- 功能点
- 测量
- 度量

- 软件过程度量 and 项目度量是定量的测量
- 软件度量由软件管理者来分析和评估
- 测量数据通常由软件工程师来收集

- 术语“measure”（测量）、“measurement”（测度）和“metrics”（度量）经常被互换地使用，但注意到它们之间的细微差别是很重要的。因为“measure”（测量）和“Measurement”（测度）既可以作为名词也可以作为动词，所以它们的定义可能会混淆。
- 在软件工程领域中，“measure”（测量）对一个产品过程的某个属性的范围、数量、维度、容量或大小提供了一个定量的指示。“Measurement”（测度）则是确定一个测量的行为。
- IEEE的软件工程术语标准辞典(IEEE Standard Glossary of Software Engineering Terms)中定义“metric”（度量）为“对一个系统、构件或过程具有的某个给定属性的度的一个定量测量”。
- 当获取到单个的数据点(如在一个模块的复审中发现的错误数)时，就建立了一个测量。测度的发生是收集一个或多个数据点的结果(如调研若干个模块的复审，以收集每一次复审所发现的错误数的测量)。软件度量在某种程度上与单个的测量相关(如每一次复审所发现的错误的平均数，或复审中每人/小时所发现的错误的平均数)。
- 软件工程师收集测量结果并产生度量，这样就可以获得指标“indicator”。指标是一个度量或度量的组合，它对软件过程、软件项目或产品本身提供了更深入的了解。指标所提供的更深入的理解，使得项目管理者或软件工程师能够调整开发过程、项目或产品，这样使事情进行得更顺利，能被更好地完成。

- 例如，四个软件小组共同完成一个大型软件项目。每一个小组必须进行技术复审，但允许其自行选择所采用的复审类型。检查度量结果——每人/小时所发现的错误数，项目管理者注意到采用更加正式的复审方法的两个小组，每人/小时所发现的错误数比起另外两个小组高40%。假设所有其他参数都相同，这就给项目管理者提供了一个指标：正式的复审方法比起其他复审方法在时间投资上能得到更大的回报。他可能会决定建议所有小组都采用更加正式的方法。
- 度量给管理者提供了更深入的理解，而更深入的理解会产生更严谨、更正确的决策。

23.1 过程度量

- 评估正在进行中的项目的状态
- 跟踪潜在的风险
- 在问题造成不良影响之前**发现问题**
- **调整**工作流程或任务
- **评估**项目团队控制软件工作产品质量的能力

23.1 过程度量

- 测量数据由项目团队收集
- 转换成度量数据在项目期间使用

23.1 过程度量

- 软件过程的功效只能间接地测量，也就是说，根据从过程中获得的结果来导出的一组度量。结果包括
 - 在软件发布之前发现的错误数的测度
 - 提交给最终用户并由最终用户报告的缺陷的测度
 - 交付的工作产品（生产率）的测度
 - 花费的工作量的测度
 - 花费时间的测度
 - 与进度计划是否一致的测度
 - 其他测度
- 我们也可以通过测量特定软件工程任务的特性来导出过程度量。

过程度量规则

- 解释度量数据时使用常识，并考虑组织的敏感性。
- 向收集测量和度量的个人及团队定期提供反馈。
- 不要使用度量去评价个人。
- 与开发者和团队一起设定清晰的目标，并确定为达到这些目标需要使用的度量。
- 不要用度量去威胁个人或团队。
- 指出问题区域的度量数据不应该被“消极地”看待，这些数据仅仅是过程改进的指标。
- 不要在某一个别的度量上纠缠，而无暇顾及其他重要的度量。

- 质量相关性
 - 专注于工作产品和可交付产品的质量
- 生产率相关性
 - 与花费的工作量相关的工作产品的产量
- 统计SQA数据
 - 错误分类和分析
- 缺陷排除效率
 - 评估一个活动在错误传递到下一个活动之前发现错误的能力
- 重用数据
 - 已产生构件的数量和可重用性的程度

23.1.2 项目度量

- 利用度量能够对开发进度进行必要的调整，以避免延迟，并减少潜在的问题和风险，从而使得开发时间减到最少。
- 项目度量可用于在项目进行过程中评估产品质量，必要时可调整技术方法以提高质量。
- 每个项目必须测量：
 - 输入——工作中需要资源的测度（例如，人，工具）。
 - 输出——在软件工程过程中产生的可交付产品或工作产品的测度。
 - 结果——展示可交付产品功效的测度。

典型的项目度量

- 每个软件工程任务的工作量/时间
- 每小时评审发现的错误数
- 计划好的和实际的里程碑日期比较
- 变更(数量)和它们的特性
- 软件工程任务上工作量的分配

23.2 软件测量

- 软件过程的直接测量：花费的成本和工作量
- 产品的直接测量：产生的代码行（LOC）、运行速度、存储容量以及某段时间内报告的缺陷
- 产品的间接度量：功能、质量、复杂性、有效性、可靠性、可维护性，以及许多在第15章谈到的其他“产品特性”

23.2.1 面向规模的度量

- 每千行代码 (KLOC) 的错误数;
- 每千行代码 (KLOC) 的缺陷数;
- 每行代码 (LOC) 的成本;
- 每千行代码 (KLOC) 的文档页数;
- 每人月错误数;
- 每评审一小时错误数;
- 每人月千行代码数;
- 每页文档的成本。

23.2.2 面向功能的度量

- 每个功能点的错误数（上千行代码）
- 每个功能点的缺陷数
- 每个功能点的成本
- 每个功能点的文档页数
- 每人月功能点数

23.2.3 LOC和FP比较

编程语言	Programming	LOC per Function point			
	Language	平均 avg.	中等 median	最低 low	最高 high
	Ada	154	-	104	205
	Assembler	337	315	91	694
	C	162	109	33	704
	C++	66	53	29	178
	COBOL	77	77	14	400
	Java	63	53	77	-
	JavaScript	58	63	42	75
	Perl	60	-	-	-
	PL/1	78	67	22	263
	Powerbuilder	32	31	11	105
	SAS	40	41	33	49
	Smalltalk	26	19	10	55
	SQL	40	37	7	110
	Visual Basic	47	42	16	158

Representative values developed by QSM

为什么选择FP?

- 与编程语言无关
- 所使用的数据是在软件过程初期就确定的准备好的可数特征
- 对于其他更笨拙的版本来说，使用更少的代码行在实现过程中并没有什么不利
- 更容易衡量可重用的构件的影响

- 基于功能点的和基于LOC的度量都是对软件开发工作量和成本的比较精确的判定
- 不要将不同类型的小组进行比较度量
- 避免“苹果和橘子”问题

23.2.4 面向对象的度量

- 场景脚本的数量(用例)
- 关键类的数量
- 支持类的数量(实现系统所必需的但又不与问题域直接相关的类)
- 每个关键类的平均支持类数量 (分析类)
- 子系统的数量 (实现某个功能的类的集合, 该功能对系统最终用户可见)

WebApp项目度量

- 静态Web页的数量(最终用户不能控制页面显示的内容)
- 动态Web页 的数量(根据最终用户的操作或其他外部因素,页面上显示出相应的定制内容)
- 内部页面链接的数量(内部页面链接就是一个指针,它在Web应用中提供了到达其他某个Web页的超链接)
- 永久数据对象的数量
- 通过界面连接的外部系统的数量
- 静态内容对象的数量
- 动态内容对象的数量
- 可执行的功能的数量

- **正确性** ——程序操作符合规格说明的程度
每千行代码的缺陷数
- **可维护性** ——程序经受变化的程度
MTTC（平均变更时间）
- **完整性** ——程序不受外界攻击影响的程度
危险性（一个特定类型的攻击在给定的时间内发生的概）和安全性（一个特定类型的攻击将被击退的概率）
$$\text{完整性} = \sum [1 - (\text{危险性} \times (1 - \text{安全性}))]$$
- **可用性** ——程序容易被人使用的容易程度
性力图

23.3.2 缺陷排除效率

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

其中：

E 是软件交付给最终用户之前发现的错误数

D 是软件交付之后发现的缺陷数

小型组织的度量

- 从提出请求到评估完成所用的时间(小时或天) , t_{queue} .
- 进行评估所用的工作量(人-小时), W_{eval} .
- 从完成评估到把变更工单派发到员工所用的时间(小时或天), t_{eval} .
- 实现变更所需的工作量 (人-小时), W_{change} .
- 实现变更所需的时间 (小时或天) , t_{change} .
- 在实现变更过程中发现的错误数, E_{change} .
- 将变更发布给客户后发现的缺陷数, D_{change} .

建立度量大纲

- 明确你的业务目标。
- 弄清你要了解或学习的内容。
- 确定你的子目标。
- 确定与子目标相关的实体和属性。
- 确定你的测量目标。
- 识别可量化的问题和相关的指标，你将使用它们帮助你达到测量目标。
- 明确你要收集的数据元素，从这些数据元素中要得到帮助你回答问题的指标。
- 定义将要使用的测量，并使这些定义具有可操作性。
- 弄清楚实现测量需要做的操作。
- 准备一份实施测量的计划。