软件度量报告

1. 软件度量的应用范围

想要明白软件度量，首先要分清度量和测量的区别。度量具有前置性，它提供了一种定量研究软件问题的方法；测量具有实时性或后置性，主要集中在给度量提供数据或者处理数据的方法上。由于软件工程强烈的不确定性，使得软件工程的精确测量困难重重，但软件度量主要研究的是可能性的规律，通过概率和统计学的研究，寻找事物内在的规律。其并不具备1+1-2的特征，而是研究在多大可能性上这个结论是合理的，因为软件的主体是人，具有概率属性，设备和材料容易度量，但人很难度量。软件度量的主要作用是评估状况、跟踪进展情况、评价产品有效性和改进设计和过程的质量。定性分析可以提供迅速地判断能力，但定性分析终究需要定量分析的验证与支持，否则其结果很可能成为无目之本，出现错误。

软件度量的方法体系主要包括5个方面：

1. 项目度量，目的在于度量项目规模、成本、进度、顾客满意度等，辅助项目管理进行项目控制；
2. 规模度量，主要依靠经验和经验的模型，是决定项目成败的重要原因之一，是估算工作量、成本预算及策划项目进度的基础；
3. 成本度量；
4. 产品度量，实质上是软件质量的度量，软件的质量由一系列质量要素组成，每个质量要素又由一些衡量标准组成，主要度量方法是McCabe复杂性度量法；
5. 过程度量，对软件开发过程的个各方面进行度量，目的在于预测过程的未来属性，减少结果的偏差，主要包括成熟度度量（例如CMMI，GJB5000A、管理度量（主要包括里程碑管理、风险度量等项目管理度量，审查度量、质量保证度量等质量管理度量，变更控制、版本管理度量等配置管理度量、生命周期度量三个大的方面。

不同层次的人员对软件度量有不同的需求。高级管理人员，如CEO，coo关注点在上市时间、客户满意度、费用的节省等商业策略的组成部分上；中级管理层，如部门经理、总监等，则主要关注生产力、成本控制、效率等，他们更多的是着眼于总体的性能，交付情况以及产品的运行状态等，而不是项目每天完成的情况；项目管理层对度量的需求则是准确估计和控制软件产品，主要考虑通过每周的对比评测、研究进展，以确保项目开发方向的正确性，或者主动捕捉测量点，由度量分析师发展成一种模型，预测项目未来的结果。

1. 软件测量基础

软件测量是经典测量科学基础上的具体应用，为了使软件度量真正发挥作用，必须掌握测量的基础知识。软件度量不能用现成的公式进行计算，需要根据自己的实际情况建立模型，并通过历史数据来定义参数。首先，测量的表示理论。人们一般习惯从比较中获得对事物的理解，例如二元关系、三元关系中谁比谁大等，而这种关系可以映射到数学世界中，由此可以把测量定义为从经验世界到关系世界的一种映射，把度量定义为为了描述实体的某种属性，而为这个实体赋予的数字或者符号。比如为了描述人的年龄，将这个人从出生到现在的所经历的年数作为年龄属性赋予这个人。第二，测量和模型。模型是对显示的抽象，除去了实现的细枝末节，使我们可以我们从特定的角度看待实体和概念。测量可以分为直接测量和间接测量两种，而一个预测系统通常由一个数学模型和一组预测规程组成，其中，预测规程的作用是确定预测参数并对结果进行解释。第三，测量数据收集与分析。良好的数据应该具有正确性、准确性、一致性的特点，并要具有合适的精度，能与特定活动或持续时间相关联，且能够重复出现。就数据的收集而言，第一步要制定数据收集计划，然后决定测量项，根据需要选取合适的测试粒度，并确保产品处于配置控制之下，必须了解对产品的哪些版本进行测量。

1. 软件规模的估算与度量

软件规模的估算与度量部分，我主要想写一下我理解的功能点估算及用例点估算的主要流程及估算过程中需要注意的问题。在此之前先简单地描述一下传统的代码行度量，一般认为空白行和注释行不应该计算在代码行中，并把不带注释的行数缩写为NCLOC又称为有效代码行ELOC，并将注释行定义为CLOC，则总长度为LOC-NCLOC+ELOC，注释行的密度用CLOC/LOC表示。但由于所用的语言不同导致代码行不同等原因，代码行不适于作估算，更适合用作完成之后的测量。接下来是功能点估算。功能点估算提出的目的是使得不同国家，不同人对同样的需求估算得到的规模大小基本相同；其缺点是对需求描述的要求比较高。在这个方法中，功能点是一个度量单元，度量所得到的值和软件的代码量没有关系，也就不再依赖于选择的编程语言。至于功能点估算的过程，最简单地来说包括4个步骤：1、估算数据功能规模；2、估算事务处理规模；3、计算调整因子；4、根据公式计算功能点数。

1. 过程规划、预测与监控中的度量

这一节主要简单地说一下对项目评估预评审技术，PERT、原始的CoCoMo模型、诤值分析以及项目监控中数据分析的理解。由于理解不是特别深刻，所以只能简单的描述一下现有的了解。关于PERT，我认为最主要的就是对三个数据的评估，即乐观的OD，不考虑效率和中断，完成任务的最小时间量；悲观的PD，考虑各种培训、生病以及工作时间做与工作无关的事情等各种延误情况和最有可能的ED，不是OD和PD的中间值，而是根据经验估算认为的最可能的情况。根据这三个值得出项目的beta分布及其图像，使用beta分布而不是用正态分布的原因是人们的评估往往偏向于乐观，图中使得左右两侧面积近似相等的分割线所对应的时间即为最可能的工作量。这种估算方法需要策划小组人员分别进行估算得到结果后，再对结果按照一定的策略进行对比分析，得到最终的估计值。

1. 产品设计质量度量与控制

我认为非功能性需求是软件度量中最容易被忽略的，也最不容易被清晰掌握的部分。在需求分析中，常见的非功能性需求虽然都能设计感官需求、易使用性、安全性、可靠性、可维护性等方面，但对其要求的描述都只停留在定性的描述上，没有明确的度量标准。而需求是度量的重要标准，也是设计和测试的参考，需求描述的不明确很容易造成最后交付过程中出现分歧，这就需要在对产品的非功能性需求描述、设计和实现过程中引入可度量的体系。

**人员度量统计分析报告**

度量一：各阶段开发人员数

目的：生产率度量

度量单位：阶段开发人员数（按人次计）

|  |  |  |
| --- | --- | --- |
| 阶段 | 计划人数 | 实际人数 |
| 需求分析 | 3 | 3 |
| 概要设计 | 2 | 2 |
| 详细设计 | 3 | 3 |
| 编码 | 5 | 5 |
| 测试 | 5 | 5 |

度量二：项目人员总数统计（按实际人员发生数计）

|  |  |
| --- | --- |
| 管理人员总数 | 5 |
| 开发人员总数 | 5 |

**进度度量统计分析报告**

度量一：软件产品工程活动的进度

目的：软件产品工程活动的进度是否与计划相同

度量单位：开始及完成时间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务名称 | 计划开始时间 | 实际开始时间 | 计划结束时间 | 实际结束时间 | 周期 |
| 可行性分析 | 2020.5.11 | 2020.5.11 | 2020.5.14 | 2020.5.14 | 3天 |
| 项目计划 | 2020.5.15 | 2020.5.15 | 2020.5.16 | 2020.5.16 | 2天 |
| 需求分析 | 2020.5.17 | 2020.5.18 | 2020.5.19 | 2020.5.20 | 3天 |
| 概要设计 | 2020.5.20 | 2020.5.21 | 2020.5.24 | 2020.5.26 | 5天 |
| 详细设计 | 2020.5.25 | 2020.5.25 | 2020.6.2 | 2020.6.2 | 7天 |
| 系统开发 | 2020.6.3 | 2020.6.3 | 2020.6.25 | 2020.6.25 | 23天 |
| 软件测试 | 2020.6.26 | 2020.6.26 | 2020.6.27 | 2020.6.27 | 2天 |
| 软件修改 | 2020.6.27 | 2020.6.27 | 2020.6.30 | 2020.6.30 | 3天 |
| 用户配置 | 2020.7.1 | 2020.7.1 | 2020.7.1 | 2020.7.1 | 1天 |

**测试缺陷度量统计分析报告**

本轮缺陷分布统计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 第一轮 | 缺陷类别 | | | | |
| 缺陷状态 | 致命缺陷 | 严重缺陷 | 一般缺陷 | 微小缺陷 | 建议 |
| 打开 | 0 | 0 | 5 | 5 | 2 |
| 关闭 | 0 | 0 | 25 | 35 | 0 |

缺陷趋势统计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 缺陷类别 | | | | |
| 测试轮次 | 致命缺陷 | 严重缺陷 | 一般缺陷 | 微小缺陷 | 建议 |
| 1 | 0 | 0 | 30 | 40 | 2 |
| 2 | 0 | 0 | 10 | 20 | 4 |
| 3 | 0 | 0 | 5 | 5 | 2 |