

软件能力与过程质量定量控制的研究

马 慧 田新民 (首都经济贸易大学 信息学院 100026)

摘要:软件实施定量控制构成了软件质量管理发展进程的重要里程碑。该文章说明了控制图技术为判断过程稳定和预测过程的行为奠定了重要的理论基础;文章陈述了软件成熟能力度模型对定量分析的客观需求以及实践中具体的分组方式。

关键词:软件成熟能力 控制图 控制边界

1 引言

从 80 年代中期由美国国防部资助,卡内基·梅隆大学软件工程研究所(CMU/SEI)最先提出的“软件能力成熟度模型(SW-CMM—Software Capability Maturity Model)”理论及其应用,在 90 年代正式发表为研究成果。这一成果已经得到了众多国家软件产业界的认可,并且在北美、欧洲和日本等国家及地区得到了广泛应用,成为了事实上的软件过程改进的工业标准。现发展到 CMM I (Capability Maturity Model Integration) 即能力成熟度模型集成。在模型中提出再也不能在“糊涂”的状态下进行软件质量管理了,在模型 4 级中,明确提出软件定量管理的过程关键域,客观上需要有相适应的定量工具支持。对此,统计控制技术发挥了非常重要的作用。

2 软件成熟能力的定量控制技术

预想改进过程并生产出具有竞争力的软件产品,过程稳定性是关键。为了测试未来并做出预测,必须首先保证过程行为是稳定的。稳定是保证过程改进的基础,也就是说必须首先保证软件过程的相关指标在它固有的绩效边界之内运动。CMM 模型对过程的稳定能力有具体的、特殊的要求。

CMM(软件成熟能力度模型)的 4 级组织的过程能力是定量的质量改进。质量改进是等级 5 的关注焦点。已管理级(4 级)是建立在可重复级(2 级)和已定义级(3 级)之上的。在 4 级中,主要包括定量过程管理和软件质量管理两个关键过程域。借助于等级 3 的

过程资产,基于所收集的数据进行分析,做出客观的决策,使项目能定量理解并稳定的过程。当绩效超出限制时,识别绩效变化的特殊原因,采取合适的改正措施。

CMM 模型客观上需要控制技术作为辅助工具。控制图能形象地反映被控制对象的稳定水平。例如图 1 为一个工作时间周均值统计控制图的例子。随机变量较均匀地分布在可控的区间范围之内,并且分布的点没有违背统计控制的若干条判断标准,详见本文中的第三部分。

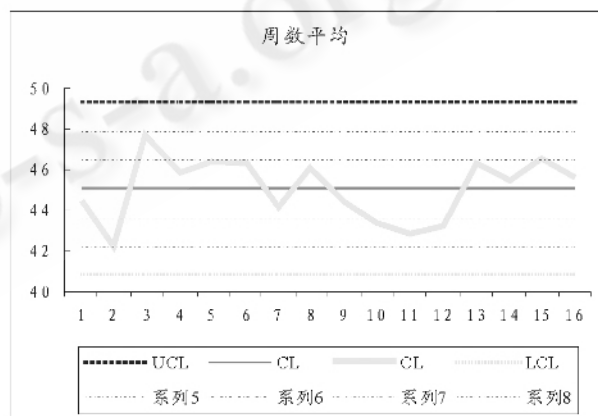


图 1 16 周均值的控制图

定量过程关注过程质量,软件质量管理关注产品质量。过程能力描述了软件过程期望结果的范围,这些过程数据被用来建立和修正过程性能目标,分析软件过程性能。定量过程管理的目的是定量控制软件项目的过程性能,软件过程性能决定了实施软件过程所

获得的实际结果。定量过程管理主要有以下目标:

(1) 有计划地进行定量过程管理活动。

(2) 能够对项目定义的软件过程的性能进行定量控制。

(3) 组织标准软件过程的过程能力是已知的、定量的。

软件质量管理侧重建立对项目软件产品质量的定量了解和实现特定的质量目标。软件质量管理包括:确定软件产品的质量目标;制定实现这些目标的计划;并监控及调整软件计划、软件工作产品、活动和质量目标,以满足客户和最终用户对高质量产品的需要和期望。软件质量管理的实践基于三个关键过程域(集成软件管理、软件产品工程定量、定量过程管理)。前两个是建立和实施项目定义软件过程的关键过程域,而后一个则是对项目定义软件过程实现所期望的结果能力建立定量了解。基于组织、客户和最终用户的需要来建立软件产品的质量目标,这样的目标才能实现。为实现质量目标,组织要制定战略和计划,项目则具体调整其已定义的软件过程。实施软件质量管理要达到以下三个目标:

① 项目的软件质量管理活动是有计划进行的;

② 软件产品质量的可测目标和这些目标的优先级是确定的;

③ 实现软件产品质量目标的实际进程能够被量化和管理的。

在第 4 级中,软件过程具有精确定义的、一致的评价方法,这些评价方法为评估项目的软件产品和质量奠定了一个量化的基础。量化控制将使软件开发真正变成一种工业生产活动。这时过程是可评价的,而且,执行过程的活动也是在可评价限度之内的,使得组织可以在定量限度范围内,预测过程和产品质量的发展趋势。因为过程是稳定的,可评价的,所以一有意外情况出现,就可以确定导致这些变化的“特定原因(special cause)”。一旦过程跨越了已知的限度,就会采取适当的措施来加以矫正。可想而知,软件产品是高质量的。

3 统计控制技术的基本方法

通过大量的实践可以发现,在某种质量水平下,随

机因素所引起的质量特征值波动的分布具有一定的规律。相对来讲,比较能反映规律的有正态分布。我们知道:分布概率的总和为 1。在 $\mu \pm \sigma$ 的范围内,曲线与横坐标之间包围的面积为 0.6827; 在 $\mu \pm 2\sigma$ 的范围内,曲线与横坐标之间包围的面积为 0.9545; 在 $\mu \pm 3\sigma$ 的范围内,曲线与横坐标之间包围的面积为 0.9973,过程落在 3σ 以外的概率仅为 0.27%,可以看成是一个小概率事件发生。 μ 代表分布中心的均值, σ 代表分散程度的标准偏差,它决定了分布曲线的形状。当取 3σ 为判断的标准时,如果一个小概率发生了(0.27%),则可以判断系统出现了异常波动——即软件开发过程处于非控制状态。

控制图的基本图形样式如图 2 所示。一般情况下,控制图是在原坐标系的基础上,描述随机变量的动态规律。在此图上至少包含三条统计线段即中心线、上控制线、下控制线。中心线: $CL = \mu$; 上控制限 $UCL = \mu + 3\sigma$; 下控制限 $LCL = \mu - 3\sigma$ 。此外,为了按统计控制标准判断,将中线与上下边界按 3σ 分别分成 3 等份。如果一个过程处于稳定的状态,那么,它不应该出现下列情况:

测试 1: 一个点落在 3σ 控制之外;

测试 2: 三个相继的点中至少两个点落在中心线的同一侧,并且距离中心线有两个以上的 σ 单位;

测试 3: 五个相继的中至少有四个点落在中心线的同一侧,并且距离中心线有一个以上的 σ 单位;

测试 4: 至少八个相继的点落在中心线的同一侧;

图 2 是一个反映 5 人工作量的统计控制图,在图中没有上述非稳定状态的现象发生。可初步判断在此阶段过程处于稳定的状态。

4 控制技术运用于软件能力质量管理

建立和维护稳定性是通过利用过去的经验,至少在某限度内预测现象在未来变化的走向。如果现象是可控的,即可以满足成本预算与进度承诺,提供所要求的产品功能与质量,制定可实现的计划,并且得到持续改进。总的变化 = [共同原因引起的变化] + [特殊原因引起的变化]。如果一个受控过程还不能满足用户的要求,那么这个过程就必须找出公共原因,实施改进

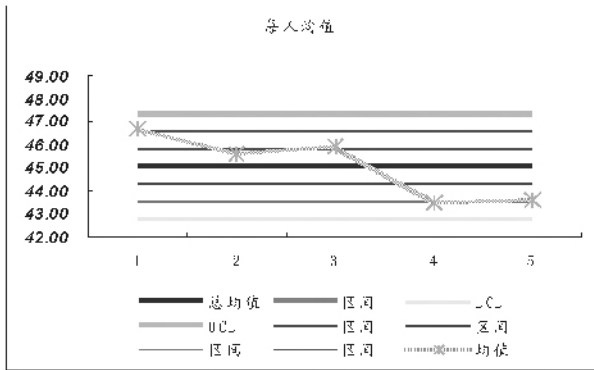


图 2 每人均值控制图

或重订指标。

在软件质量控制的过程中需要注意的是：

(1) 软件测定的支撑点通常选择时间或者是产品的里程碑；

(2) 合理的分组至关重要。同一组样本, 如果按不同的分类方式, 分组的结果却不尽相同。分组的主要原则之一是需要满足质量分析的具体目标。

例如: 有一组 80 个样本的数据。数据分别描述了 5 个人在 16 周的工作中完成工作时间的具体数据。在这种情况下, 至少有三种分组方式。其中包括按周数分组(16 组), 按每次活动分组(80 数据, 一组), 按人进行分组(5 组)。样本数据以及相关均值的控制图 1 图 3 以及表 1 所示。

综上所述, 软件定量控制构成了软件质量管理进程中重要里程碑。利用了统计推断的方法对软件生产过程进行监控具有其独特的优点。软件成熟能力度模型中对定量分析提出了客观要求。在定量控制的实践中, 合理的分组并按照组的不同规模选择不同的统计模型。如果分析的结果显示过程是不稳定的, 需要及时排除故障, 并采取相应的措施以控制此类故障不再发生; 如果分析的结果显示过程是在控制的范围内即稳定的, 可按照用户的改进需求, 通过软件工程等技术调整其均值、控制图上下限等指标。其中, 控制技术在软件定量管理中起到了重要的作用。

表 1 每人 16 周 80 个样本数据

	一	二	三	四	五
1	50.5	43.5	45.5	39.8	42.9
2	44.3	44.9	42.9	39.8	39.3
3	48.8	51	44.3	43	51.3
4	46.3	45.2	48.1	45.7	44.1
5	40.6	45.7	51.9	47.3	46.4
6	44.4	49	47.9	45.5	44.8
7	46	41.1	44.1	41.8	47.9
8	44.9	43.4	49	45.5	47.4
9	50	49	42.6	41.7	38.5
10	44.5	46.5	41.7	42.6	41.7
11	43.8	41.8	45.5	44.5	38.6
12	43.2	43.8	44.8	43.5	40.9
13	50	43.4	48.3	46.4	43.4
14	52.3	45.2	42.2	44.8	42.8
15	50	46.2	47.4	42.2	47
16	47.3	49.7	48	42	41

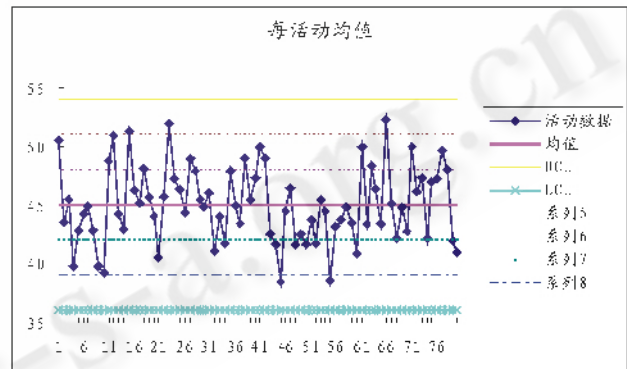


图 3 80 次活动的控制图

参考材料

- 1 杨一平,《现代软件工程技术 with CMM 的融合》,“人民邮电出版社”,2002 年 10 月。
- 2 William A. Florac and Anita D. Carleton , Measuring the Software Process - Statistical Process Control for Software Process Improvement , ,1999.
- 3 马慧,“Crystall Ball 仿真软件的探索及其在微机上的应用”,《计算机系统应用》2004 年 9 期。