对软件局部化的认识

麻锦涛 16281262 计科1602

软件工程的原则有抽象、信息隐藏、模块化和局部化等。推迟实现是软件方法学的基本指导思想。软件开发过程应该理性地 “推迟实现”，即把逻辑设计与物理设计清楚地划分开来，尽可能推迟软件的物理实现。对于大中型的软件项目，在软件开发过程中，如果过早而仓促地考虑程序的具体实现细节，可能会导致大量返工和损失。为了正确、高效地进行软件开发，必须在开发的前期安排好问题定义、需求分析和设计等环节，进行周密、细致的软件实现的前期工作，并明确规定这些环节都只考虑目标系统的逻辑模型，不涉及软件的物理实现。  
 逐步求精。逐步求精（也称逐步细化）是基于承认人类思维能力的局限性提出的，一般认为人类思维与理解问题的能力限制在7±2大小的范围内，求解一个复杂问题采用有条理的从抽象到具体的逐步分解与细化方法和过程进行。这是人类把复杂问题趋于简单化控制和管理的有效策略。逐步求精（细化）解决复杂问题的策略是软件工程方法学中的一项通用技术，与分解、抽象和信息隐蔽等概念紧密相关。  
 分解与抽象。分解是把复杂问题趋于简单化处理的有效策略。论证分解，即“分而治之”的有效性。若将一个复杂问题分解成若干容易解决的小问题，就能够减少解决问题所需要的总工作量。分解必须是科学而合理的，否则可能会增加解决问题的难度和工作量。  
 抽象是人类在认识或求解复杂问题的过程中，科学而合理地进行复杂系统分解的基本策略之一。抽象是把一些事物（状态或过程）中存在的相似的方面（忽略它们的差异）概括成“共性”的。抽象的主要思想是抽取事物的本质特性，而暂不考虑它们的细节。这是一种分层次的渐进过程。软件工程方法学中广泛采用分层次的从抽象到具体的逐步求精技术。建立模型（建模）是软件工程常用的方法和技术之一。软件工程中，整个软件开发过程需要建模，软件开发过程的各个阶段也需要建模。不同的软件开发方法的最主要的区别表现在它们的模型不同。所以，软件开发过程的一系列模型的建立标准、描述形式、应用规范等是软件开发方法最核心的研究内容。  
 **信息隐蔽。信息隐蔽是指使一些关系密切的软件元素尽可能彼此靠近，使信息最大限度地局部化。软件模块中使用局部数据元素就是局部化的一个例子。信息隐蔽的指导思想始终贯穿在软件工程的面向过程、面向功能和面向对象的软件开发方法的发展中。**  
 质量保证。质量保证是为保证产品和服务充分满足消费者要求的质量而进行的有计划、有组织的活动。质量保证是面向消费者的活动，是为了使产品实现用户要求的功能，站在用户立场上来掌握产品质量的。同样道理，这种观点也适用于软件开发。软件质量保证要求软件项目的实施活动符合产品开发中的对应的需求、过程描述、标准及规程。质量保证的基本思路是提倡“预防”而不是事后“补救”，强调“全过程控制”和“全员参与”。 软件质量是“软件与明确和隐含地定义的需求相一致的程度”。更具体地说，软件质量是软件与明确叙述的功能和性能需求、文档中明确描述的开发标准以及任何专业开发的软件产品具有的隐含特征相一致的程度。

基本原则

美国著名的软件工程专家 B. W. Boehm综合软件领域专家的意见，并总结了多家公司开发软件的经验，提出了软件工程的7条基本原则。Boehm 认为，这7条基本原则是确保软件产品质量和开发效率的原理的最小集合。它们是相互独立的，是缺一不可的最小集合；同时，它们又是相当完备的。  
分阶段的软件开发。根据这条基本原则，可以把软件生存周期划分为若干个阶段，并相应地提定出切实可行的计划，然后严格按照计划对软件开发与维护进行管理。在软件开发与维护的生存周期中，需要完成许多性质各异的工作。Boehm 认为，在整个软件生存周期中应指定并严格执行6类计划：项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划、运行与维护计划。  
 坚持进行阶段评审。统计结果显示：在软件生命周期各阶段中，编码阶段之前的错误约占63%，而编码错误仅占37%。并且，错误发现得越晚，更正它付出的代价就会越大，要差2～3个数量级甚至更高。因此，软件的质量保证工作不能等到编码结束以后再进行，必须坚持进行严格的阶段评审，以便尽早地发 现错误。  
 实行严格的产品控制。实践告诉我们，需求的改动往往是不可避免的。这就要求我们要采用科学的产品控制技术来尽可能达到这种要求。实行基准配置管理（又称为变动控制），即凡是修改软件的建议，尤其是涉及基本配置的修改建议，都必须按规定进行严格的评审，评审通过后才能实施。基准配置指的是经过阶段评审后的软件配置成分及各阶段产生的文档或程序代码等。当需求变动时，其他各个阶段的文档或代码都要随之相应变动，以保证软件的一致性。   
 采用先进的程序设计技术。先进的程序设计技术既可以提高软件开发与维护的效率，又可以提高软件的质量和减少维护的成本。  
明确责任。软件是一种看不见、摸不着的逻辑产品。软件开发小组的工作进展情况可见性差，难于评价和管理。为更好地进行管理，应根据软件开发的总目标及完成期限，尽量明确地规定开发小组的责任和产品验收标准，从而能够清楚地审查。  
 开发小组的人员应少而精。开发人员的素质和数量是影响软件质量和开发效率的重要因素，应该少而精。事实上，高素质开发人员的工作效率比低素质开发人员的工作效率要高几倍到几十倍，开发工作中犯的错误也要少得多。   
不断改进开发过程。软件过程不只是软件开发的活动序列，还是软件开发的最佳实践。在软件过程管理中，首先要定义过程，然后合理地描述过程，进而建立企业过程库，并成为企业可以重用的资源。对于过程，要不断地进行改进，以不断地改善和规范过程，帮助提高企业的生产效率。  
遵从上述七条基本原则，就能够较好地实现软件的工程化生产。但是，它们只是对现有的经验的总结和归纳，并不能保证赶上技术不断前进发展的步伐。因此，我们不仅要积极采纳新的软件开发技术，还要注意不断总结经验，收集进度和消耗等数据，进行出错类型和问题报告统计，评估新的软件技术的效果，指明必须着重注意的问题及应该采取的工具和技术。