软件重构技术研究

周隆亮，于 翔，唐学军
（西南电子设备研究所 四川 成都 610036）

【摘要】本文主要研究软件重构技术的相关问题，阐述了软件重构的定义和作用，分析了软件重构的时机，总结了软件重构的基本方法，并介绍了与软件重构密切相关的设计模式、极限编程的相关概念，为软件重构的深入研究奠定了基础。

【关键词】软件重构；极限编程；设计模式
【中图分类号】TP311.5 【文献标识码】A 【文章编号】1009-5624（2020）11-0214-02

Research on the technology of software Refactoring

Zhou Longliang, Yu Xiang, Tang Xuejun
Southwest Institute of electronic equipment, Chengdu, Sichuan 610036, China

【Abstract】This paper is mainly researching on the problems of software refactoring technology. The definition and effects of software refactoring are expatiated, the occasion situation for software refactoring are analyzed, and the basic ways for software refactoring are summarized. At finally, the design pattern and extreme programming, which are associating with software Refactoring nearly, are introduced, It lays a foundation for the further research of software refactoring.

【Key words】Software refactoring; Extreme programming; Design patterns

1 引言

软件设计开发是一个反复迭代的过程，随时面临着需求的变更，功能的增删，应用的调整，在软件设计开发的初始阶段很难建立可适应各种需求的软件架构，只能尽可能地将软件功能模块化、高内聚、低耦合，在面向新的需求及软件后期维护时，主要依靠软件重构实现，软件重构的一个重要原则是仅改变软件内部结构，而不改变其外部行为，通过软件重构可显著减少软件的维护成本，延长软件的使用寿命。

2 软件重构的定义与作用

对于重构的定义，可以追溯到面向对象语言Smalltalk的研究和使用，当软件设计人员在精炼代码，对软件框架进行整理时，逐步形成了软件重构的概念。为适应需求调整或者部署环境的变化，对函数名称的改变、利用函数来代替重复代码、或者改变函数的接口参数等都可认为进行了软件重构。

重构可以从动词和名词两个方面理解[1]。把重构理解为名词，是对软件的内部结构所作的一种改变，这种改变在可观察行为不变的条件下使软件更易理解，更易维护；把重构理解为动词，是应用一系列不改变软件可观察行为的重构操作对软件进行重新组织。这两种理解都强调重构是改进设计的过程，只改进软件的内部结构，而不改变代码的外部行为。它通过不断地改善软件的内部结构来减少软件的复杂性，使软件更易理解和维护。

软件重构通常可以起到以下作用：

2.1 优化软件结构

重构的一个重要目标就是要降低软件结构的复杂度。如果设计中试图为将来考虑过多的灵活性，则会产生不必要的复杂性和错误，重构强调优化软件结构，支持多种方式的扩展，而实际实现时仅面向任务提供最匹配的解决方案，只有在新需求到来时，再触发扩展实现。

2.2 增强软件的可维护性

随着系统的演进，针对软件的功能补丁不断增加，通常代码的结构会越来越差，甚至变得不可理解，代码的维护难度加大。程序中设置会出现万千重复的代码，重构的目标之一就是要消除重复，力求相同代码只出现一次，在不改变软件效率的前提下，让代码更容易维护。

2.3 增强代码的可理解性

重构实现功能单一的类、职责单一的函数、减少局部变量，减小系统耦合，设计有意义的命名规则，让不同的开发人员的代码可以一致理解。

2.4 简化测试

如果对代码进行了重构，就会对代码有一个更深层次的理解，建立在深层次的理解上，对于代码的测试将更加容易。重构强调清晰化代码结构，要求“小步骤进行”，并且对每一步都进行严格的测试，这些都将使得问题更加容易凸显，测试更加简单。

3 软件重构的时机

软件重构并不需要安排专门的时间，也不是一个周期性的重复工作，通常在软件生命周期内，需求的变更和软件的维护会自然触发重构。

3.1 软件集成时[2]

当我们开发一个新的软件时，通常都会借用一些已经成熟的代码，减少开发软件的费用。然而，可复用的软件通常需要大量的迭代设计，通过不断调整已有代码的结构，达到当前的软件要求，这就是对已有的代码与设计进行重构，抽取可复用单元。

3.2 功能调整时

需求不会是永远一成不变的，在不同的条件下使用，功能会有增删，甚至硬件升级也会导致软件调整，这就要求对原有的代码与设计进行调整，当原有的代码很难再加进新的功能或是在加进新的功能的时候，出现代码急剧变

坏的现象，此时需要对原有的代码与设计进行重构，以使其容易被扩展。

3.3 软件修复时

软件运行过程中经常会遇到一些意想不到的问题，这就需要我们根据问题现象对软件进行修复。在软件修复的过程中，不可避免地需要考虑软件重构的问题，如果是软件结构上的问题，就需要调整软件结构；如果是代码实现逻辑问题，就需要重写代码实现，直到解决软件出现的问题。

3.4 代码复查时

交付使用之前通常需要对软件进行测试，对代码进行复查。如果在复查过程中发现错误或者不合理的地方，这时基本都要对软件进行适当的重构。在开始进行重构之前，先阅读代码，对代码有一定程度的理解；然后做出一些建议，看能否使代码的设计更加简单；最后进行重构，就可以清楚地看到效果。重复几次后，将会对代码有更深的理解。

4 软件重构的方法

在软件重构领域，至今已提出软件度量、图变换、概念分析、程序分片等多种形式化重构技术，图变换提供了对软件重构操作的支持，程序分片用于处理函数或过程获取等特定类别的重构。但是，目前在源代码层次上的重构技术相对更加成熟，对于一个需要重构的软件，通过阅读其代码，总是能发现种种问题，比较常见的有软件结构不合理导致数据访问过程复杂；完成同一个功能不是通过函数调用而是将代码复制到相应位置；函数实现臃肿，有过多的未使用变量；设计一个过大的类，而不注重类的层次；一个函数的参数列表冗长；过多的注释，或者几乎没有注释等，通过修改代码把软件中存在的问题避免掉是软件重构的最直接的软件重构方法。而应对这些代码缺陷，在代码修改的时候，我们常常采取的措施有重新组织函数，将一些重复代码合并为函数，在需要的地方调用即可；在对象之间搬移特性，比如搬移函数、提炼类、将类内联化；重新组织数据，有继承关系的数据可以按层次重新组织；简化条件表达式，分解合并条件表达式、以多态取代条件表达式等。

软件源代码的重构应遵循小步骤的原则，即每一步总是做很少的工作，且每做少量修改便做测试，一次修改太多，往往会引入很多错误，不便于程序调试，且发现了修改错误也很难返回到原有状态。这些小步骤包括：（1）寻找需要重构的地方。（2）如果需要重构的代码还没有单元测试，则应首先编写单元测试。（3）进行单元测试，保证原先的代码是正确的。（4）认真考虑该做什么样的重构，或者找一本关于重构分类目录的书查找相似的情况，然后按照书中的指示一步一步地进行重构。（5）每一步完成时，都进行单元测试，保证可观察行为没有发生改变。（6）如果重构改变了接口，则需要修改某些测试。（7）全部做完后，进行全部的单元测试、功能测试，保证整个系统的可观察行为不受影响。

5 软件重构的相关技术

5.1 设计模式

设计模式是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结[3]。用面向对象的观点来看，设计模式可以被定义为描述特定场景下解决一般设计问题的类和对象框架。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性，因为对于设计模式的很多研究都集中于良好编程风格以及程序各部位之间有用的交互模式，而这些都可以影响到结构特征和重构手法。设计模式有助于对框架结构的理解，成熟的框架通常使用了多种设计模式，设计模式是软件工程的基石。重构有时会失去目标，而设计模式能够为重构提供明确的方向和目标，它不但可以用于引导重构并作为重构的目标，而且也为重构行为提供全局性的引导，对于一些典型实现，我们总是让我们的设计遵循已经成熟的设计模式。

5.2 极限编程

极限编程是敏捷开发思维的重要支持者，是目前较流行的一种轻量级的软件开发过程，同时它也是一个非常严谨和周密的方法。它是专门针对小到中型规模的软件开发团队在面对需求模糊与快速变化的情况下进行软件开发的一种轻量级软件开发方法。极限编程在需求开发阶段将用户也纳入开发人员范畴，将需求模块化，优先分析高风险模块；在设计阶段基于测试工作驱动开发，优化设计；单元测试的编写先于程序本身，每次模块整合都进行单元测试；其整个开发过程形成一个螺旋形增量开发过程。极限编程的一个主要原则是融入变化，在极限编程中，重构被作为一种核心设计方法，重构所具有的优点与极限编程所倡导的价值与规则相吻合，使得重构成为极限编程的一个重要的实践，成为极限编程不可分割的一部分。重构是极限编程中实现简单设计与增量变化的基础，所有的代码都必须经历持续不断地重构以保持设计的干净、简单和完整。

6 结语

重构是一种增强软件可理解性、可维护性、可复用性的软件方法，它已成为面向对象领域的研究热点和重要的实践之一。本文从软件的定义和作用、重构的时机、重构的方法，以及重构的相关技术这几个方面，对软件重构进行了分析。在软件工程化的要求下，对软件的重构将一直占据软件开发的重要位置。

【参考文献】

[1] 刘毅，安鲁陵．关于软件工程中的软件重构 [J]．科技广场，2004（9）：20-22.

[2] 华奇兵许文波费娜．面向对象软件重构 [J]．重庆邮电学院学报，2004，16（2）：96-100.

[3] 王宏伟，董丽丽．设计模式在软件重构中的应用与实现 [J]．中国科技信息，2010（6）：88-92.