

软件开发 与管理研究

闫俊伢 著

北京工业大学出版社

软件工程开发与管理研究

闫俊伢 著

北京工业大学出版社

图书在版编目(CIP)数据

软件工程开发与管理研究 / 闫俊伢著. — 北京 :
北京工业大学出版社, 2019.7
ISBN 978-7-5639-6969-2

I. ①软… II. ①闫… III. ①软件开发—研究②软件开发—项目管理—研究 IV. ①TP311.52

中国版本图书馆CIP数据核字(2019)第185334号

软件工程开发与管理研究

著 者: 闫俊伢

责任编辑: 刘连景

封面设计: 点墨轩阁

出版发行: 北京工业大学出版社

(北京市朝阳区平乐园100号 邮编: 100124)

010-67391722(传真) bgdcbs@sina.com

经销单位: 全国各地新华书店

承印单位: 定州启航印刷有限公司

开 本: 787毫米×1092毫米 1/16

印 张: 12.25

字 数: 245千字

版 次: 2019年7月第1版

印 次: 2019年7月第1次印刷

标准书号: ISBN 978-7-5639-6969-2

定 价: 45.00元

版权所有 翻印必究

(如发现印装质量问题, 请寄本社发行部调换 010-67391106)

作者简介

闫俊伢，女，山西大学商务学院信息学院副院长，教授，硕士生导师，中国高校计算机教育 MOOC 联盟山西省工作委员会委员，企业级开发工程师，高级软件工程师。

主要从事计算机学科的教学与科研工作，主编或参编出版教材 20 余部，著作 1 部。作为项目负责人承担山西省科技厅、山西省教育厅、教育部高等教育司产学研合作协同育人项目等 10 余项，参与省部级项目 10 余项。在国家核心期刊及重要学术刊物上发表论文 40 余篇，其中核心期刊及以上论文 10 余篇，获得国家实用新型专利授权 6 项，软件著作权 1 项。获第二届山西省教育科学研究优秀成果二等奖，省级教学成果二等奖，获山西省“百部（篇）工程”三等奖，获第三届全国商科教育实践教学大赛课程建设实践教学方案二等奖。多次指导学生参加山西省“挑战杯”竞赛、全国大学生电子设计竞赛、全国软件和信息技术专业人才大赛等各种专业赛事并获奖。其中“挑战杯”竞赛，获省级一等奖 5 次，二等奖 2 次；大学生电子设计竞赛，获省级二等奖 1 次，三等奖 6 次；全国软件和信息技术专业人才大赛，获省级一等奖 1 次，二等奖 2 次，三等奖 3 次。

前 言

自 20 世纪 40 年代中期现代计算机诞生以来,作为现代计算机“灵魂”的计算机软件经历了程序设计、软件设计和软件工程等发展阶段。尤其是自 1968 年提出“软件工程”概念以来,软件行业经历了风起云涌的飞速发展。软件工程是一个关于人的行业,其中包括了需要并使用软件的人、构建软件的人、测试软件的人、管理软件项目的人以及那些支持和维护软件的人。如今,软件工程专业已经有了成千上万的新编程语言、形形色色的新方式方法、多如牛毛的新软件工具,不计其数的软件应用不断如雨后春笋般涌现。迄今为止,也许没有哪个工程行业像软件工程领域一样如此欣欣向荣。这种蓬勃发展的强劲势头的确令人欢欣鼓舞。从表面上看,我们有充足的理由对软件工程的未来充满信心和期待。对于软件这样一个年近古稀的行业,我们理所当然地认为它应该是非常成熟、可靠的。

本书由山西大学商务学院闫俊伢教授撰写,全书共九个章节,第一章主要围绕软件工程进行了阐述;第二章对软件过程进行了分析,其中包括了典型软件过程模型以及软件开发方法等内容;第三章对软件项目立项进行了阐述,其中包括了软件项目立项的方法、软件可行性分析、软件项目规模成本估算、制定软件项目开发计划以及软件项目立项文档等内容;第四章对软件需求分析进行了阐述;第五章对软件架构与体系结构设计进行了分析;第六章对软件配置管理进行了分析;第七章介绍了软件质量管理;第八章对软件项目管理进行了分析;第九章对软件安全管理进行了分析。

软件工程是计算机科学的一个重要分支,所涉及的范围非常广泛,包括软件开发技术、软件工程环境、软件经济学、软件心理学和软件工程管理等诸多方面的知识。本书着重从实用角度讲述软件工程的基本概念、原理和方法,介绍如何规范地开发和维护软件;合理地安排软件开发和维护过程;规范地书写软件工程的文档资料,以提高软件开发的效率和质量。

为了确保研究内容的丰富性和多样性,笔者在写作过程中参考了大量理论与研究文献,成果基于山西省科技厅重点研发计划项目(编号:

201603D321112），并应用于课题的实际项目开发中，在撰写过程中也得到了项目组成员的大力支持以及清华大学出版社朱云飞编辑的审阅，在此向涉及的专家学者表示衷心的感谢。

最后，由于笔者水平有限，加之时间仓促，本书难免存在疏漏和不足之处，在此，恳请同行专家和读者朋友批评指正！

自“航天”升腾于天际之际，我国的空间探索开始由地面走向太空，从20世纪80年代以来，我国航天事业取得了举世瞩目的成就，在载人航天、月球探测、卫星应用、深空探测等方面取得了重大突破。载人航天作为人类探索太空的重要一步，我国在这一领域取得了举世瞩目的成就。从1992年启动载人航天工程，到2003年神舟五号首次载人飞行，再到2013年神舟十号与天宫一号交会对接，我国载人航天事业取得了举世瞩目的成就。在月球探测方面，我国也取得了重大突破。从1990年代开始，我国就开始了月球探测的准备工作，经过多年的努力，终于在2007年成功发射了嫦娥一号月球探测器，实现了我国首次月球探测。在卫星应用方面，我国也取得了重大突破。从1990年代开始，我国就开始了卫星应用的准备工作，经过多年的努力，终于在2003年成功发射了神舟五号载人飞船，实现了我国首次载人飞行。在深空探测方面，我国也取得了重大突破。从1990年代开始，我国就开始了深空探测的准备工作，经过多年的努力，终于在2007年成功发射了嫦娥一号月球探测器，实现了我国首次月球探测。

载人航天作为人类探索太空的重要一步，我国在这一领域取得了举世瞩目的成就。从1992年启动载人航天工程，到2003年神舟五号首次载人飞行，再到2013年神舟十号与天宫一号交会对接，我国载人航天事业取得了举世瞩目的成就。在月球探测方面，我国也取得了重大突破。从1990年代开始，我国就开始了月球探测的准备工作，经过多年的努力，终于在2007年成功发射了嫦娥一号月球探测器，实现了我国首次月球探测。在卫星应用方面，我国也取得了重大突破。从1990年代开始，我国就开始了卫星应用的准备工作，经过多年的努力，终于在2003年成功发射了神舟五号载人飞船，实现了我国首次载人飞行。在深空探测方面，我国也取得了重大突破。从1990年代开始，我国就开始了深空探测的准备工作，经过多年的努力，终于在2007年成功发射了嫦娥一号月球探测器，实现了我国首次月球探测。

载人航天作为人类探索太空的重要一步，我国在这一领域取得了举世瞩目的成就。从1992年启动载人航天工程，到2003年神舟五号首次载人飞行，再到2013年神舟十号与天宫一号交会对接，我国载人航天事业取得了举世瞩目的成就。在月球探测方面，我国也取得了重大突破。从1990年代开始，我国就开始了月球探测的准备工作，经过多年的努力，终于在2007年成功发射了嫦娥一号月球探测器，实现了我国首次月球探测。在卫星应用方面，我国也取得了重大突破。从1990年代开始，我国就开始了卫星应用的准备工作，经过多年的努力，终于在2003年成功发射了神舟五号载人飞船，实现了我国首次载人飞行。在深空探测方面，我国也取得了重大突破。从1990年代开始，我国就开始了深空探测的准备工作，经过多年的努力，终于在2007年成功发射了嫦娥一号月球探测器，实现了我国首次月球探测。

目 录

第一章 软件工程概述.....	1
第一节 软件与软件危机	1
第二节 软件工程	7
第三节 软件生命周期	14
第四节 软件开发工具	17
第二章 软件过程.....	19
第一节 软件过程概述	19
第二节 典型的软件过程模型	20
第三节 软件开发方法	34
第三章 软件项目立项.....	37
第一节 软件项目立项的方法	37
第二节 软件可行性分析	42
第三节 软件项目规模成本估算	44
第四节 制定软件项目开发计划	47
第五节 软件项目立项文档	53
第四章 软件需求分析.....	59
第一节 需求分析概述	59
第二节 需求获取	66
第三节 需求建模	69
第四节 需求规格说明与需求验证	81
第五章 软件架构与体系结构设计	87
第一节 软件设计概述	87
第二节 软件架构设计	90

第三节	软件结构设计的基本原理	94
第四节	软件结构化设计方法	99
第五节	软件体系结构设计常用技术	102
第六章	软件配置管理	109
第一节	基线技术	109
第二节	软件配置管理的任务	110
第三节	软件配置管理需求	112
第四节	实施软件配置管理	114
第五节	软件研制过程的配置控制	122
第七章	软件质量管理	125
第一节	软件质量概述	125
第二节	质量度量模型	127
第三节	软件复杂性	130
第四节	软件可靠性	132
第五节	软件评审	138
第六节	软件容错技术	146
第八章	软件项目管理	149
第一节	项目管理概述	149
第二节	软件项目管理概述	155
第三节	软件项目技术与进度安排	160
第九章	软件安全管理	169
第一节	软件系统的安全性工作	169
第二节	软件安全性设计准则	179
参考文献		187

第一章 软件工程概述

随着现代信息技术的发展，计算机的使用范围越来越广，软件的需求量也不断增多，软件系统的规模越来越大。为提高软件开发与管理的质量，工程化被应用于软件开发与管理的过程中。本章主要阐述软件与软件危机、软件工程、软件生命周期以及软件开发工具。

第一节 软件与软件危机

一、软件的概念

软件的定义是随着计算机技术的发展逐渐完善的。20 世纪 50 年代，人们认为软件等于程序；20 世纪 60 年代，人们认为软件等于程序和文档；20 世纪 70 年代，人们认为软件等于程序、文档和数据。软件是计算机中与硬件相互依存的部分，具体包含如下内容。

- ①在运行中能够提供相应功能的程序。
- ②描述软件设计过程的文档。
- ③让程序正确运行的数据。

二、软件的特点

软件与硬件都属于产品，但二者也存在一定差别。了解并理解这种差别，对理解软件工程是非常重要的。软件的特点包括以下几点。

（一）双重性

软件的双重性，一方面表现在软件是一个产品，能利用软件来表现硬件的计算潜能；另一方面表现在软件是产品交付使用的载体，能实现计算机之间的通信。

（二）开发性

软件是被开发或设计出来的，而不是传统意义上所说的被制造出来的。一般意义上的产品，包括硬件产品，都要经过分析、设计、制造、测试等过程，也就是说要经过一个从无形的设想到一个有形的产品的过程。软件不是有形的系统元件，只是逻辑上的产品，是被开发或设计出来的，而不是被制造出来的。软件的开发成本大于生产成本，一旦被开发或设计出来，能进行大量的复制，这也意味着软件的开发不能像制造其他产品那样进行管理。

（三）退化性

软件不会“磨损”，但会退化。一般情况下，有形的硬件产品在使用过程中总会有磨损。在使用初期，硬件磨损往往比较严重（这实际上是磨合），而经过了一段不长时间的磨合后，将进入相对的稳定期。随着时间的流逝，硬件的磨损再次趋于严重，这也意味着硬件的寿命快要到了。

软件并不是一种有形的产品，因此也就不存在所谓的“磨损”问题。在软件的运行初期，未知的错误使得程序有较高的故障率，当修正了这些错误而未引入新的错误时，软件将进入较为理想的平稳运行期。这说明软件是不会“磨损”的。但在实际情况中，软件尽管不会“磨损”，但会退化。这是因为软件在其生命周期中会经历多次修改，每次修改都会引入新的错误，而对这些错误又要进行新的修改，使得软件的故障曲线呈现一种齿形，导致最后的故障率慢慢升高，即软件产生了退化，而这种退化缘于修改。

（四）定制性

绝大多数软件都是定制的且是手工编写的。在硬件制造业，构件的复用是非常自然的。理想情况下，软件构件应该能够被复用于不同的程序。但由于软件的定制性，软件构件的复用才刚刚起步，不能像硬件那样拿来即用，需要进行必要的软件定制，如构件之间的组合、接口的设计、功能的修改与扩充等，而且软件开发中构件的使用比例有限。因此，软件设计基本上依赖于手工编写。

（五）复杂性

随着现代软件结构的发展，软件规模越来越大，使得软件的开发过程越来越复杂。所以已经不是靠一个人单枪匹马开发一套软件的时代了，而是需要一个组织严密的团队才能完成的了。

三、软件的分类

(一) 按功能分类

①系统软件：与计算机硬件接口并为其他程序服务的软件，如操作系统、驱动程序等。

②支撑软件：用于开发软件的工具性软件，如开发平台、数据库管理系统、各种应用工具软件等。

③应用软件：为解决某一领域的应用而开发的软件，如商业软件、嵌入式软件、个人计算机软件、Web 软件、人工智能等。

(二) 按版权分类

①商业软件：版权受法律保护、经授权方可使用且必须购买的软件。

②共享软件：与商业软件类似，但可以“先尝后买”，其获取途径主要是通过互联网。

③免费软件：无须支付许可证费用便可得到和使用的软件，发行渠道类似于共享软件。

(三) 按工作方式分类

①实时软件：在事件或数据产生时需要立即处理并及时反馈信号，以控制需要监测的部分和控制过程的软件。

②分时软件：多个联机用户同时使用计算机的软件。

③交互式软件：能够实现人机通信的软件。

④批处理软件：把一批数据以批处理的方式依次运行，再根据顺序逐个处理的软件。

(四) 按软件规模分类

①微型软件：一个人在几天内完成的少于 500 行的程序。它通常没有严格设计和测试的文档。

②小型软件：不超过 2 个人在半年内完成的 1000 ~ 2000 行的程序。它通常没有与其他程序的接口。

③中型软件：不超过 5 个人在 1 ~ 2 年内完成的 5000 ~ 50000 行的程序。这类软件通常需要有严格的文档和设计规范。

④大型软件：由 5 ~ 20 个人在 2 ~ 3 年内完成的 50000 ~ 100000 行的程序。这类软件需要按照软件工程方法进行管理。

⑤超大型软件：由 100 ~ 1000 个人在 4 ~ 5 年内完成的具有 100 万行程序的软件项目。这类软件必须按照软件工程方法开发，有严格的质量管理措施。

⑥巨型软件：由 2000 ~ 5000 个人在 5 ~ 10 年内完成的具有 100 万 ~ 1000 万行程序的软件项目。这类软件必须按照软件工程方法开发，有严格的质量管理措施。

（五）按销售方式分类

①定制软件：受某个特定客户委托，在合同的约束下开发的软件。

②产品软件：由软件开发机构开发，可以为众多用户服务并直接提供给市场的软件。

四、软件的发展

（一）程序设计

时间：20 世纪 50 ~ 60 年代。

软件所指对象：程序。

主要程序设计语言：机器及汇编语言。

软件工作范围：程序编写。

需求者：程序设计者本人。

开发软件的组织：个人。

软件规模：小型。

决定质量的因素：个人编程技术。

开发技术和手段：子程序、程序库。

维护者：程序设计者。

硬件特征：存储容量小，价格高，工作可靠性差。

软件特征：完全不受重视。

（二）程序系统

时间：20 世纪 60 ~ 70 年代。

软件所指对象：程序及说明书。

主要程序设计语言：高级语言。

软件工作范围：包括设计和测试。

需求者：少数用户。

开发软件的组织：开发小组。

软件规模：中小型。

决定质量的因素：小组设计水平。

开发技术和手段：结构化程序设计。

维护者：开发小组。

硬件特征：价格降低，速度、容量及工作可靠性有明显提高。

软件特征：软件技术的发展不能满足需要，出现软件危机。

（三）软件工程

时间：20 世纪 70 年代以后。

软件所指对象：程序、文档和数据。

主要程序设计语言：软件语言，具体包含需求定义语言、程序设计语言、软件设计语言、软件功能语言等。

软件工作范围：软件生存期。

需求者：市场用户。

开发软件的组织：开发小组及中大型软件开发机构。

软件规模：大中小型。

决定质量的因素：管理水平。

开发技术和手段：开发环境、开发工具、数据库、标准和规范、面向对象技术、网络及分布式开发、工程化开发方法。

维护者：专职维护人员。

硬件特征：向超高速、大容量、微型化、网络化方向发展。

软件特征：软件技术有一定的发展，价格高，未能摆脱软件危机。

五、软件危机的概念

软件危机是计算机软件开发和维护过程中遇到的一系列问题的集中体现。这些问题不仅包括软件不能正常工作，还包括如何满足软件需求、如何开发软件、如何运行软件、如何维护软件等。

六、软件危机的表现

（一）不符合用户需求

软件开发人员对用户的需求缺乏全面、准确和深入的理解，往往急于编程，导致最后实现的软件系统与用户的实际需求相距甚远。

（二）软件供不应求

软件开发生产率提高的速度远远低于计算机硬件的发展速度，软件应用需求的增长得不到满足，出现了供不应求的局面，从而使人类不能充分利用计算机硬件提供的巨大潜力。

（三）软件预估不准确

软件的实际开发成本比估算成本高出几倍，实际进度比预期进度推迟几个月甚至几年。软件开发者为了追赶进度、降低成本和减少损失，采取一些策略，致使软件质量降低。这种现象损害了开发者的信誉，同时又引起了用户的不满。

（四）软件质量不可靠

软件质量保证技术没有严密有效地贯穿到软件开发的全过程，导致交付给用户的软件质量差，在运行过程中频繁产生问题，甚至带来极其严重的后果。

（五）软件可维护性差

程序中出现的很多错误非常难改正，要使软件适应新的硬件环境又几乎不可能，也不能根据用户提出的新需求在原有程序上添加一些新的功能，造成软件维护困难和不可重用，使得开发人员重复开发基本类似的软件。

（六）文档资料不完整

计算机软件不仅包含程序，还包括软件开发过程中各个阶段的文档资料，如各阶段的说明书、软件测试用例等。这些文档是软件开发人员交流信息的工具，对于维护人员来说更是不可缺少。缺少必要的文档资料或文档资料不严密、不正确，必然为软件开发和维护工作带来许多困难。

七、软件危机产生的原因

（一）软件自身特点

软件是计算机系统逻辑部件，难以提前预测。在计算机上运行之前，难以对软件的质量进行评价，因此给软件开发和维护带来了困难。另外，软件的规模越来越庞大，软件程序越来越复杂，为了在规定时间内开发规模庞大的软件，必须由多人合作完成开发，在这个合作开发的过程中不仅涉及技术问题，还必须有严格、科学的管理。

（二）开发与维护不善

软件自身的特点给软件开发与维护带来一定困难，但如果在开发过程中使用已被证明是正确的方法和成功的经验，许多困难是可以被克服的。但是许多软件开发人员仍然对软件开发与维护工作有错误的认识，并且使用落后、错误的技术。错误的认识和方法主要表现为缺乏正确的理论指导、忽视软件需求分析的重要性和轻视软件维护等。

八、软件危机的解决方法

在分析软件危机产生的原因后，人们开始探索使用软件工程的原理、概念、方法、技术等，进行软件的设计、开发、运行、维护和更新，于是计算机科学领域产生了软件工程这一概念。为解决软件危机，首先应对软件有正确的认识，消除计算机发展初期“软件是程序”的错误认识。软件由完整的程序、文档以及数据的配置组成。程序是能够完成特定功能的序列指令；文档是开发、运行及维护软件所需的图文资料；数据是让软件处理信息的数据结构。其次，应充分认识到软件开发是需要各种软件开发人员密切配合，合作完成的工程项目，而不是某种个体劳动的神秘技巧。软件开发应吸收和借鉴人们在各种工程项目中积累的概念、原理、方法和技术，尤其是要吸收人们从事计算机硬件开发的经验和教训。最后，要吸收和借鉴软件开发的成功经验，并进行推广使用，探索更好的软件开发方法，减少软件危机。如果说机械工具能“放大”人的体力，那么软件工具则能“放大”人的智力。软件开发过程复杂，每个阶段都有许多繁杂的工作，借助软件工具辅助，软件开发人员可以提高开发效率和质量。软件开发应构建软件工程支撑环境，即将软件开发过程中使用的软件工具有机结合成一个整体，支持软件开发的全过程。

第二节 软件工程

一、软件工程的观念

通俗地说，软件工程即借用传统工程设计的基本思想，运用工程化的原理、概念、方法、技术来开发和维护软件，将经过时间证明正确的管理技术和软件开发的科学方法有机结合，从而缩短开发周期，降低开发成本，提高软件的开发效率和质量。经过多年的发展，软件的工程化生产已成为软件产业。软件已

成为产品，它涉及产值、市场、版权、法律保护等方面的问题。

软件工程是一门交叉学科，需要运用管理学的方法和原理来进行软件开发管理，运用工程学的方法设定开发方案、估算运费和实施方案，运用数学来分析各种算法，建立可靠性模型。

二、软件工程的要素

软件工程以关注软件质量为目标，由方法、工具和过程三个要素构成。其关系如图 1-1 所示。

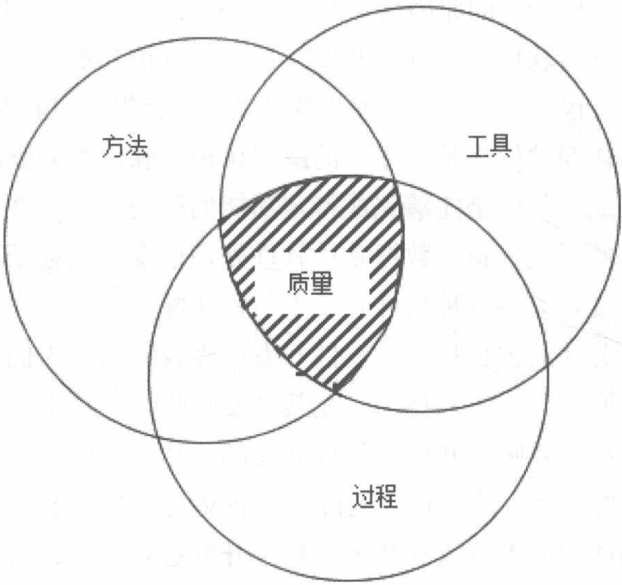


图 1-1 软件工程三要素

软件工程方法：提供如何进行软件开发的技术，涉及软件工程的多个方面。例如，软件系统需求分析、软件项目计划设计与估算、软件系统总体结构设计、软件系统数据结构设计等。

软件工程工具：提供自动或半自动的软件支撑环境。目前，已经推出了许多软件工程工具，这些软件工程工具集成起来，建立起软件开发支撑系统。计算机辅助软件工程将各种软件开发工具、软件工程工具以及软件工程数据库有机结合，形成软件支持环境。

软件工程过程：软件工程过程定义了软件工程方法的使用顺序、要求交付的文档资料、保证软件质量的管理、软件开发各阶段的里程碑。软件工程过程将方法与工具结合起来，进行软件开发。

三、软件工程的目标

（一）有效性

软件系统能有效利用计算机的空间和时间资源。时空开销是衡量软件质量的重要技术指标之一。有时空间资源和时间资源的有效性会产生一定矛盾，因此不得不做出一些取舍，牺牲空间效率换取时间有效性，或者牺牲时间效率换取空间有效性。经验丰富的软件开发人员会恰当使用折中概念，巧妙地调节时间和空间的有效性，实现软件设计并满足用户需求。

（二）可靠性

软件的可靠性是指软件具有能够防止因设计、原理以及结构等方面的不足导致的软件系统失效的能力。尤其对于实时嵌入式计算机系统来说，可靠性更是一个重要的目标。这是因为软件要实时控制一个物理过程，如核电站的运行、宇宙飞船的导航等。如果不能保证可靠性，后果不堪设想。因此在软件开发过程中，要注重可靠性。

（三）可修改性

可修改性是指允许对软件系统进行修改，支持软件的维护和调试，是一个难以达到且难以度量的目标。

（四）可理解性

可理解性是指软件系统具有清晰的结构，能直接反映问题需求，有利于控制软件系统的复杂性。

（五）可维护性

在软件产品交付用户使用后，软件开发人员依然能够对软件进行修改，改正已发现或潜在的错误，改进性能和功能，让软件更能适应用户的使用环境。因为软件属于逻辑产品，只要用户存在需求，就能长期应用下去，所以要进行软件维护。在软件工程中，可维护性是软件工程中一个非常重要的目标。在软件开发费用中，软件维护费用占有很大比重。

（六）可重用性

软件中概念或功能相对独立的模块可以定义为软部件。所谓可重用性即软部件在各种场合的应用程度。可重用的软部件能不进行修改直接使用，有些则需要进行修改后使用。可重用软部件要具有正确的编码、清晰的结构与注解以