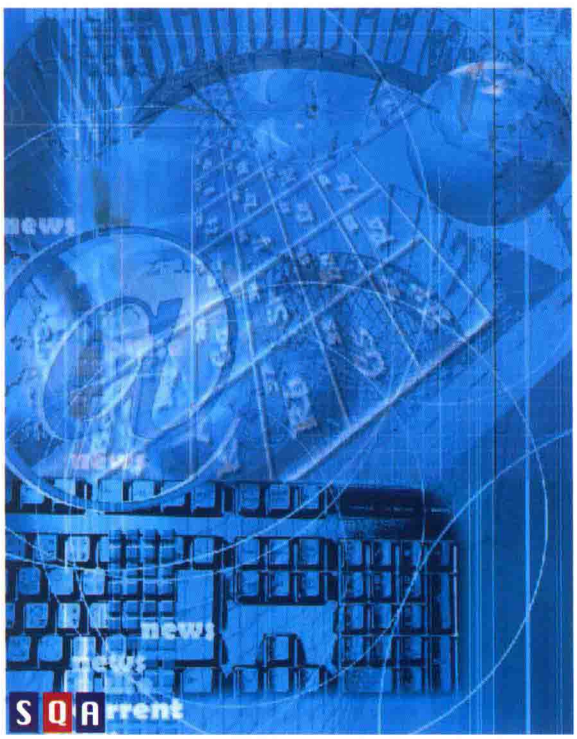
高等学校计算机应用规划教材

软件测试技术与实践

- 软件与软件危机
- 软件测试基础
- 软件测试分类与分级
- 软件缺陷管理
- 基于生命周期的软件测试方法
- 软件测试过程及测试过程管理
- 软件静态测试
- 软件动态测试
- 软件单元测试
- 软件集成测试和确认测试
- 软件系统测试
- 面向对象软件测试



蔡建平　叶东升　康　妍
菲　周百顺　李大奎　编著

清华大学出版社

高等学校计算机应用规划教材

软件测试技术与实践

蔡建平 叶东升 康 妍
王爱菲 周百顺 李大奎 编著

清华大学出版社

北 京

内 容 简 介

本书共分为软件测试基础、软件测试管理、软件测试方法与技术三部分，覆盖了软件测评的各个环节和知识点，内容包括软件及软件测试的基本概念、软件测试分类与分级、软件缺陷管理、软件全生命周期测试、软件测试及其过程管理、软件静态测试与动态测试，以及面向对象软件测试的方法等。对于其中的一些重要环节，设计了基于案例驱动，利用典型开源工具进行软件测试实践的教学内容，如缺陷管理、测试管理、静态测试、单元测试、集成测试、系统测试(包括功能测试及性能测试)等。

本书可作为高等院校计算机相关专业的教材和参考书籍，还可作为软件测试应用型人才的培训教材，也可供软件测试、软件质量保证及软件开发和软件项目管理从业人员参考。

图书在版编目(CIP)数据

软件测试技术与实践 / 蔡建平 等编著. —北京：清华大学出版社，2018
(高等学校计算机应用规划教材)
ISBN 978-7-302-48688-6

Ⅰ. ①软… Ⅱ. ①蔡… Ⅲ. ①软件－测试－高等学校－教材 Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2017)第 271394 号

责任编辑：王　军　李维杰
装帧设计：孔祥峰
责任校对：牛艳敏
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：29　　字　　数：706 千字
版　　次：2018 年 1 月第 1 版　　印　　次：2018 年 1 月第 1 次印刷
印　　数：1~3000
定　　价：79.80 元

产品编号：070863-01

序

当前及今后，“软件定义一切”已充分说明了软件的重要性。软件作为产品及产品中的核心，其质量仍然可以认为是产品的生命。

在国家大力推进信息化与工业化融合之际，信息软件及工业软件的质量保证决定了“两化融合”的成败。软件质量保证的最重要手段之一就是软件测试。《软件测试技术与实践》作为航天中认推出支持两化软件测试的系列教材(《软件测试技术与实践》、《信息软件系统测试与实践》及《嵌入式软件测试与实践》)的基础，就显得非常重要。软件测试集技术、工程及实践于一身，如果没有好的技术基础和工程意识，眼高手低，那么在开展信息软件测试或工业软件测试时就会力不从心，多走弯路，无法把好质量关。

蔡建平教授长年从事软件工程、软件测试以及软件质量保证的研究、实践和教学，并为编写此书做了较长时间的准备，也出版了多本这方面的教材。特别是《软件测试方法与技术》被评为“全国工程硕士专业学位教育指导委员会推荐教材”和“软件工程专业核心课程系列教材”。本教材是在《软件测试方法与技术》和《软件测试实践教程》基础上改编而成的，具有如下主要特点：

(1) 该教材重要知识点的组织和讲述满足国内企业，特别是国内各种评测机构或组织对现代软件测试人才培养的要求；

(2) 该教材在传统软件测试技术和方法的基础上，特别强调软件测试是质量把控的重要手段之一，必须要与软件质量度量和评价相结合，要满足软件工程全生命周期软件测试的要求，要充分重视软件开发方法和应用方式对软件测试的影响，要注意软件测试工具对软件测试支持的重要作用等；

(3) 该教材还为高等院校和企业培训提供了软件测试课程实践教学方案、平台和案例，满足应用型软件测试人才培养的需要，满足软件测试工作人员的自学需要。

总之，该教材在软件测试技术的学习、普及、推广和软件测试应用型人才培养以及软件测试理论教学知识体系及实践教学体系的建立等方面进行了有益的探索和尝试。该教材内容全面、详实，实用性强。该教材的改版，将有益于国内软件测试人员和计算机相关专业的本科生及研究生的学习与能力的培养，有益于推动现代软件测试技术和方法的研究、教学和实践的进一步发展，同时对我国软件测试业的发展和软件测试紧缺人才的培养起到积极的促进作用。

航天中认软件测评科技(北京)有限责任公司总经理

2017 年 8 月 30 日于北京

前　言

当前，软件测试已从传统的软件工程瀑布模型中测试阶段的软件测试变化为覆盖包括需求分析、系统设计、详细设计、程序编码、内部测试、系统测试、系统安装、确认验收以及系统维护整个软件工程生命周期的软件测试；从过去单纯的测试概念发展到包括静态分析、质量度量与评价在内的评测结合的软件评测思想；从传统的测试内容分类到基于质量特性、子特性的测试内容分类；从传统的结构化程序测试方法到面向对象的软件测试方法；从早期的单机或桌面测试到网络应用测试及嵌入式应用测试；从以手工测试为主发展到离不开测试工具支持的测试及管理。事实上，软件测试也成为耗费人力、财力和时间的一项复杂的工作，对测试人员提出了高素质、专业化的要求。对软件测试人员不但要求精通各种软件测试技术和方法，有一定的软件测试工程实践经验；还要求他们熟悉软件开发技术和软件开发流程，具有快速学习专业知识或领域知识、掌握新技术和应用新工具的能力；另外，软件测试人员要有团队合作意识，善于和人沟通与交流，并能承受被人误解和指责的心理素质。

随着计算机技术的快速发展，软件越来越普遍地应用到各个领域和各个方面，且应用规模越来越大，应用形式越来越复杂，软件质量要求越来越高，软件测试越来越重要。对于高等院校而言，人才的培养是其核心的工作，鉴于高素质的软件测试专业人才越来越奇缺，航天中认积极开展校企合作，与大连理工大学软件学院共建软件测试课程和实验室，大力开展教学实践、教学改革的探索和实践，编写专业教材，凝练软件测试人才培养的成果。

本书是在《软件测试方法与技术》和《软件测试实践教程》基础上改编而成，将两本书的内容压缩、整合和完善，并试图将本书重点突出在两方面：软件测试技术与软件测试实践。本教材既包括成熟的理论基础，也包括软件测试工具的使用，同时还可以通过书中案例实践的学习巩固学习成果。本书是作者多年从事软件测试技术研究、软件测试课程教学及软件项目测试成果和经验的总结。全书共分 12 章，分为 3 部分：第 I 部分(第 1 至第 3 章)是软件测试基础，涉及软件测试的一些基本概念和基础知识，如软件与软件危机、软件测试基本概念、软件测试分类与分级；第 II 部分(第 4 至第 6 章)是本书的重点内容之一，详细讲述生命周期的软件测试方法与技术，包括软件缺陷与缺陷管理、软件测试及其过程管理、软件测试管理工具的使用及案例实践；第III部分(第 7 至第 12 章)也是本书的重点内容之一，详细讲述软件测试的方法与技术，包括软件静态测试及软件动态测试。其中，基于 McCabe 设计与集成复杂性的集成测试方法解决了集成覆盖测试的理论问题。

本书系统全面地从软件测试基础理论到软件测试全生命周期实践角度，系统地为读者解决从事软件测试工作的问题。另外，本书几乎在各章对支撑该章软件测试方法和技术应用的开源软件测试工具详细地进行了应用介绍，并配有具体测试案例。这些工具及案例的

应用对于支持高校软件测试课程实践是有意义的。最后，本书取材新颖、内容翔实、通俗易懂、技术实用、覆盖面广、指导性强，既可作为软件测试相关课程的研究生(特别是工程硕士专业学位研究生)与本科生的教材，同时还可供软件测试培训和软件测试人员自学的书籍。

致谢，再一次感谢《软件测试方法与技术》和《软件测试实践教程》中提到的人员，感谢取材互联网上的有关原创作者，感谢清华大学出版社的大力支持和帮助。

2017 年 8 月 28 日于北京

作者简介

蔡建平，在军队从事教学与全军军用共性软件、软件工程、软件质量保证等项目的论证及研究工作20多年，获军队科技进步一等奖一项、二等奖两项、三等奖两项，编著《Ada程序设计语言高级教程》，发表各类学术文章20多篇。

在企业工作期间，主持开发了嵌入式软件工程和软件测试工具，这些工具已成功地用于航空、航天等国防项目的测试和软件工程化，极大地保证了这些项目的质量。

在北京工业大学工作期间，在软件学院的学科、专业、实验室、“211工程”、教育部和北京市特色专业、科技创新平台以及学科交叉等建设方面做了大量的工作，取得了突出成果。获国家教育教学成果二等奖。“软件测试”及“高级软件编程技术”分别评为学校精品课程和研究生重点建设课程，《软件测试大学教程》、《软件测试实验指导教程》、《嵌入式软件测试实用技术》、《软件综合开发案例教程》4部教材和专著已在清华大学出版社出版发行。其中《软件测试大学教程》于2013年被评为全国工程硕士专业学位教育指导委员会推荐教材。

科研上，发表各类论文20多篇，申请专利、软著多项，指导的学生科技活动成果获第十二届“挑战杯”全国大学生课外学术科技作品竞赛三等奖，指导的两篇硕士论文被评为校优秀论文。

作为惠普国际软件人才及产业基地的学术总监，负责全国各高校共建专业合作论证及顶层策划与设计，培养方案、课程体系、实训方案的设计与制定，与实训课程配套教材的研发组织及各门课程的研发组织，基地师资队伍及课程团队建设，以及1000多名学生实习/实训的组织与实施。

目前，在航天中认负责公司的咨询、研究及对内和对外的技术培训等业务，负责和参与了交通部、体育总局、中海油、大连理工大学、浪潮、长虹、美的、小天鹅、格力、轨道交通、国家电网、中国质量认证中心、汽车电子、医疗电子、家用电器等信息化建设项目及嵌入式系统项目的软件工程化、软件质量保证、软件测试以及配套实验室建设的咨询与培训。

蔡建平教授还是国家科学技术奖励、国家专利奖励、山东省科学技术奖励、北京市科学技术奖励、海淀区科学技术奖励、北京市文化创意产业、海淀区文化创意产业等专家库成员。

目　录

第 I 部分　软件测试基础篇

1947 年，计算机还是由机械式继电器和真空管驱动的、有房间那么大的庞然大物，由哈佛大学制造的 MARK-II 则是体现当时技术水平的计算机。在一次整机运行中，它突然停止了工作。技术人员爬到计算机上找原因，发现是一只飞蛾受光和热的吸引飞到了计算机内部一组继电器的触点之间，然后被高电压击死。由此，计算机的缺陷产生了，虽然最后该缺陷被技术人员解决了，但是我们从此认识到了它就是缺陷。

如今软件已经渗透到我们的日常生活中，从办公设备到家用电器，从通信工具到航空航天事业，软件无处不在，然而却又很难完美无缺。

1994 年的秋天，迪士尼公司发布了第一个面向儿童的多媒体光盘——《狮子王动画故事书》(*The Lion King Animated Storybook*)。对此，迪士尼公司做了大量的宣传。因此，销售额非常可观。然而圣诞节过后，公司接到了大量的投诉电话，称游戏不能运行。经证实，造成这种后果的原因是迪士尼公司未对市面上使用的许多不同类型的 PC 机型进行测试，软件只能在少数系统中运行。

同样是 1994 年，英特尔奔腾浮点除法缺陷事件，不仅使英特尔公司的形象受到严重影响，并且为自己处理软件缺陷的行为付出了 4 亿多美元的代价。

类似的还有美国航天局火星极地登陆者号探测器事件、爱国者导弹防御系统事件、千年虫问题等，这些事件的后果有的是带来不便，例如游戏玩不成，有的可能是灾难性的后果——导致机毁人亡。它们的发生都是由于在软件中隐藏着错误。

软件为什么会频繁出问题，如何杜绝或将它们减至最少，将影响降至最低呢？在论述软件测试概念之前先介绍一下软件、软件危机及软件工程等概念，然后再讲解软件测试的相关知识。

第1章　软件与软件危机

我们都知道软件的重要意义：软件是信息化的核心，现代国民经济、国防建设、社会发展及人民生活都离不开软件。软件产业是增长最快的朝阳产业，是高投入、高产出、无污染、低能耗的绿色产业。软件产业关系到国家经济和文化安全，体现了国家综合实力，是决定未来国际竞争地位的战略性产业。

但软件到底什么？它具有什么样的特性？它能够干什么？

1.1.1　软件特性

软件是人通过智力劳动产生的，软件产品是人的思维结果，是一个逻辑部件，而不是一个物理部件。因此，软件生产水平最终在相当程度上取决于软件人员的教育、训练和经验的积累。所以，软件具有与硬件不同的一些特点。

1. 软件与硬件的不同

软件与硬件的不同或差别主要反映在以下几个方面。

1) 表现形式不同

硬件有形，有色，有味，看得见，摸得着，闻得到。而软件无形，无色，无味，看不见，摸不着，闻不到。软件大多存在于人们的脑海里或纸面上，它的正确与否，是好是坏，一直要到程序在机器上运行才能知道。这就给设计、生产和管理带来许多困难。

2) 生产方式不同

软件开发是智力的高度发挥，而不是传统意义上的硬件制造。尽管软件开发与硬件制造之间有许多共同点，但这两种活动是根本不同的。在两种活动中，通过好的设计能够得到好的质量，但硬件制造阶段可能引入的质量问题在软件开发中却不会出现，反之亦然。这两种活动都依靠人，但人的作用和工作专长之间的关系是完全不同的。因为软件是逻辑产品，如几个人共同完成一个软件项目时，人与人之间就有一个思想交流的问题，称之为通信关系。通信是要付出代价的，不仅要花费时间，现实中由于通信中的疏忽常常会使错误增加。人虽然是最聪明的，但人也是最容易犯错误的。

3) 要求不同

硬件产品允许有误差，如加工一根轴，其外径精度要求为Φ500±0.1。生产时，只要达到规定的精度要求就算合格。而软件产品却不允许有误差，要 1 就是 1。如美国金星探测器水手 1 号，导航程序的一条语句的语法正确，但语义错了，结果飞行偏离航线，最终导致试验的失败。又如，阿波罗宇宙飞船飞行控制软件，由于粗心，把一个逗号写成一个句号，又没能检查出来，几乎造成悲剧性的后果。这就给软件开发和维护，以及它的质量保证体系提出了很高的要求。

4) 维护不同

硬件是会用旧、用坏的这是因为硬件在使用过程中，由于受到环境的影响，如灰尘、温湿度变化、空气污染、振动等因素而使产品产生腐蚀或磨损，使硬件故障率增加，甚至损坏，以致不能使用。解决的办法，换上一个相同的备件就是了。而软件不受那些引起硬件损坏的环境因素的影响。因此，在理论上，软件不会用旧、用坏。但实际上，软件也会变旧、变坏。因为在软件的整个生命周期中，一直处于改变(维护)状态。而随着某些缺陷的改变，很可能引入一些新的缺陷，因而使软件的故障率增加，品质变坏。硬件某一部分变坏，可以使用备件，而软件则不存在这种备件，因为软件中任何缺陷都会在机器上导致同样的错误。所以，软件维护要比硬件维护复杂得多。

从上我们可以总结出软件具有同传统的工业产品相比的一些特性。

2. 软件的特性

1) 软件是一种逻辑实体，具有抽象性

这个特点使它与其他工程对象有着明显的差异。人们可以把它记录在纸上、内存中和磁盘、光盘上，但无法看到软件本身的形态，必须通过观察、分析、思考、判断，才能了解它的功能、性能等特性。

2) 软件没有明显的制造过程

一旦研制开发成功，就可以大量拷贝同一内容的副本。所以对软件的质量控制，必须着重在软件开发方面下工夫。

3) 软件在使用过程中，没有磨损、老化的问题，但有退化问题

软件在生命周期后期不会因为磨损而老化，但会为了适应硬件、环境以及需求的变化而进行修改，而这些修改又不可避免地会引入错误，导致软件失效率升高，从而使得软件退化。当修改的成本变得难以接受时，软件就被抛弃。

4) 软件对硬件和环境有着不同程度的依赖性

这导致软件移植的问题。

5) 软件的开发至今尚未完全摆脱手工作坊式的开发方式，生产效率低

6) 软件是复杂的，而且以后会更加复杂

软件是人类有史以来生产的复杂度最高的工业产品。软件涉及人类社会的各行各业、方方面面，软件开发常常涉及其他领域的专门知识，这对软件工程师提出了很高的要求。

7) 软件的成本相当昂贵

软件开发需要投入大量、高强度的脑力劳动，成本非常高，风险也大。现在软件的开销已大大超过了硬件的开销。

8) 软件工作牵涉很多社会因素

许多软件的开发和运行涉及机构、体制和管理方式等问题，还会涉及人们的观念和心理。这些人的因素，常常成为软件开发的困难所在，直接影响到项目的成败。

1.1.2　软件种类

软件已经渗透到我们的日常生活，用于各行各业。具体地说，软件按如下形式分类：

(1) 系统软件(如操作系统、数据库管理系统、设备驱动程序、通信处理程序等)；

(2) 应用软件(如事务软件、实时软件、工程和科学软件、嵌入式软件、娱乐软件、个人计算机软件、人工智能软件等)；

(3) 工具软件(如文本编辑软件、文件格式化软件、磁盘向磁带传输数据的软件、程序库系统以及支持需求分析、设计、实现、测试和管理的软件等)；

(4) 可重用软件。

1.2　软件危机

前面提到的美国于 20 世纪 60 年代初飞向金星的第一个空间探测器(水手 I 号)，因其飞行舱内计算机导航程序中一条语句的语义出错，致使偏离航线无法取得成功。这条语句的错误并不是语法错误，而是具有完全不同于程序员所期望的意思——语义错误。可以称得上是世界上最精心设计，并花费了巨额投资的美国阿波罗登月飞行计划的软件，仍然没有避免出错；另外，阿波罗 8 号太空飞船的一个计算机软件错误，造成存储器的一部分信息丢失；还有，阿波罗 14 号在飞行的 10 天里，出现了 18 个软件错误。当时，软件系统的可靠性得不到保证，几乎没有不存在错误的软件系统。

那时，计算机已有近 20 年的历史，一方面硬件成本每隔两到三年降低一半，内存和外存则成本每年降低 40%左右，硬件性能价格比每十年提高一个数量级，但所需的软件很少能在成本、时间进度、功能规模、维护能力等方面达到要求，特别是可靠性难以符合人们的需要。正如 E.E.David 指出的那样，大型系统的软件生产已经成为管理人员担惊受怕的项目，于是，人们在 20 世纪 60 年代后期惊呼发生了软件危机(software crisis)。

1.2.1　软件危机的分析

产生软件危机的原因是多方面的，但不管怎样，软件危机有它内在的或本质上的原因。下面就软件危机产生的诸多原因进行分析，揭示软件危机内在或本质上的原因。

1. 早期编程的特点

从 20 世纪 40 年代开始，人们从在 MARK-I 和 ENIAC 计算机上编制程序，到软件危机发生时为止的 20 多年时间里，对软件开发的理解就是编程，且编程是在一种无序的、崇尚个人技巧的状态中完成的。

和今天的软件开发相比，那时的编程具有一些特点。

1) 软件规模相对较小

原因有二：

① 人们对软件可能达到的功能认识有限，那时最为关心的是计算机硬件的发展。作为一名计算机专业人员，他不太关心软件问题(只有为数不多的专业人员才去关心软件)，

但他必须懂得计算机的结构。作为一个机构，其大量资金也被用于计算机硬件开销，软件只是作为展现其软件性能的一种手段而投入少量资金。

② 硬件性能从某种意义上左右着人们对软件的需求，人们总是不自觉地在心中盘算着硬件支持的可能性，这种现象从根本上阻碍了软件的广泛应用，从而限制了软件规模。

2) 编程作为一门技艺

大部分软件技术人员不太关心他人的工作，他们往往陶醉于自己的编程技巧，并且那时也无编程规范与标准，这也是由软件规模决定的。决定软件质量的唯一因素就是编程人员的素质，时间进度亦是如此。根据 H.Sackman 等人的调查，素质好的人与素质差的人，在软件生产效率上的比例是 10∶1，在程序处理速度和存储容量上的比例是 5∶1。

3) 缺少有效方法与软件工具的支持

在当时几乎谈不上有效的编程方法，使用最多的亦只是简单的控制流图。软件工具只有子程序库、装入程序、编辑程序、排错程序以及汇编和编译程序(后来才有链接程序)，以至于许多程序员沉溺于编程技艺的掌握和使用上。

4) 不重视软件开发的管理

由于人们重视个人技能，再加上软件开发过程能见度低，许多管理人员甚至根本不知道软件技术人员在干什么，究竟做得如何，从而造成管理活动几乎不存在。直到今天，这种观念在一些机构中仍有残留，为此，我们对软件开发的管理问题必须给予更多的重视。

5) 软件开发后的维护工作很难进行

由于人们重视个人技能，一旦需要做某些修改，就要原编程人员进行修改。如果他已离开本机构，就需要别人去读懂他的程序，再进行修改和补充。此项工作极其辛苦，说不定修改了一处，反而出现上百处漏洞，变得得不偿失。

上述编程特点导致出现人们常说的软件危机现象。

2. 大型软件开发问题

进入 20 世纪 60 年代，应客观需求需要制作一些大型软件，这样就出现了像第一个空间探测器所描述的例子。国外在开发一些大型软件系统时遇到了许多困难，有些系统最终彻底失败了；有些系统虽然完成了，但比原定计划推迟了好几年，而且费用大大超出了预算；有些系统未能达到用户当初的期望；有些系统则无法进行修改、维护。IBM 公司 OSS/360 系统和美国空军某后勤系统都花费了几千人/年努力，历尽艰辛，但结果令人失望。

随着软件开发应用范围的增广，软件开发规模越来越大。大型软件开发项目需要组织一定的人力共同完成，而多数管理人员缺乏开发大型软件开发系统的经验，而多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确，有时还会产生误解。软件开发项目开发人员不能有效地、独立自主地处理大型软件开发的全部关系和各个分支，因此容易产生疏漏和错误。这也是导致软件危机产生的一个原因。

3. 软件生产的知识密集和人力密集的特点

由于计算机技术和应用发展迅速，知识更新周期加快，软件开发人员经常处在变化之中，不仅需要适应硬件更新的变化，而且还要涉及日益扩大的应用领域问题研究；软件开

发人员所进行的每一项软件开发几乎都必须调整自身的知识结构以适应新的问题求解的需要，而这种调整是人所固有的学习行为，难以用工具代替。

软件生产的这种知识密集和人力密集的特点是造成软件危机的根源所在。从软件危机的种种表现和软件作为逻辑产品的特殊性，可以发现软件危机的具体原因。

4. 用户需求难以明确

对于大型软件往往需要许多人合作开发，甚至要求软件开发人员在软件开发过程中深入应用领域对相关问题进行研究，了解用户需求。这样就需要在用户与软件开发人员之间以及软件开发人员之间相互通信。在此过程中难免发生理解上的差异，导致用户需求不明确的问题产生。

用户需求不明确问题主要体现在四个方面：

① 在软件开发出来之前，用户自己也不清楚软件开发的具体需求；

② 用户对软件开发需求的描述不精确，可能有遗漏，有二义性，甚至有错误；

③ 在软件开发过程中，用户还提出修改软件开发功能、界面、支撑环境等方面的要求；

④ 软件开发人员对用户需求的理解与用户本来的愿望有差异。这些需求问题将导致后续软件错误的设计或实现，而要消除这些需求上的误解和错误往往需要付出巨大的代价。这同样是产生软件危机的一个原因。

5. 缺乏正确的理论指导，缺乏有力的方法学和工具方面的支持

由于软件开发不同于大多数其他工业产品，其开发过程是复杂的逻辑思维过程，其产品极大程度地依赖于开发人员高度的智力投入。由于过分地依靠程序设计人员在软件开发过程中的技巧和创造性，因此加剧了软件开发产品的个性化，这也是发生软件开发危机的一个重要原因。

6. 软件开发复杂度越来越高

软件开发不仅仅是在规模上快速地发展扩大，而且其复杂性也急剧地增加。软件开发产品的特殊性和人类智力的局限性，导致人们无力处理复杂问题。所谓复杂问题的概念是相对的，一旦人们采用先进的组织形式、开发方法和工具提高了软件开发效率和能力，新的、更大的、更复杂的问题又摆在人们的面前。

7. 软件危机的本质原因

从前面软件危机的原因分析来看，软件危机产生的本质原因主要有两点：

① 与软件本身的特点有关；

② 与软件的开发人员有关。

从上我们可以看出，软件危机是指在计算机的开发和维护过程中所遇到的一系列严重问题。这些问题绝不仅仅是不能正常运行的软件才具有的，实际上，几乎所有软件都不同程度地存在这些问题。