

王英龙 张伟 杨美红 主编

软件测试技术

教育部“软件工程课程体系研究”规划教材



清华大学出版社

教育部“软件工程课程体系研究”规划教材

软件测试技术

王英龙 张 伟 杨美红 主编

王英龙 张 伟 杨美红
王振操 韩明军 张海燕 编著
郭 莹 韩庆良 王 鲁

清华大学出版社
北京

内 容 简 介

本书首先从一个宏观的角度对软件测试做了准确定位,然后对软件测试做了总体论述并描绘了软件测试的一个全貌;随后,以软件测试生命周期这样一个基本的软件测试过程为线索,逐层深入地向读者解密软件测试的内容和技术,在本书的后面还论述了测试过程的组织与管理、测试的度量和过程改进及相关模型。理论与实践的紧密结合是本书的最大特点,本书精心准备了一个独立软件项目,自始至终保持将各类测试内容附着于该项目中,做到了测试的连续性和完整性,实景回放式的叙述方式使读者如临其境,带来有如真实的工作体验。

本书可以用作各类高等学校的软件测试专业及相关专业的教学用书,也可以作为认证考试的参考用书,供测试职业鉴定类的高级检验员、检验师,国家软件资格水平考试软件评测师、国际软件测试工程师使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件测试技术/王英龙等主编. —北京:清华大学出版社, 2009. 9

(教育部“软件工程课程体系研究”规划教材)

ISBN 978-7-302-20878-5

I. 软… II. 王… III. 软件—测试—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2009)第 159254 号

责任编辑:张瑞庆 顾 冰

责任校对:焦丽丽

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260

印 张:26

字 数:601 千字

版 次:2009 年 9 月第 1 版

印 次:2009 年 9 月第 1 次印刷

印 数:1~3000

定 价:37.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:033563-01

P R E F A C E

软
件
工
程
系
列
教
材

前 言

在几年前的一次业内研讨会的间隙,我与几位老朋友在一起聊天,请他们谈一谈对软件测试的一些看法。这几位多是一些名企的资深项目经理或是技术骨干,他们对软件测试的评价却颇高,归结一下的话,可以用12个字概括:**不可或缺、卓有成效、大有可为**。伴随信息产业的发展,现代软件应用提出了更高需求,应用开发的规模和复杂度、系统集成的复杂度都超越以往,信息化风险陡增,客户对软件质量更加关注、要求更高。软件测试作为重要的质量保障手段,为软件项目的成功做出了自己的贡献,并为项目过程改进提供了真实的一线数据,甚至有人戏言患上了“测试依赖症”,这是其“不可或缺”的一面;随着行业对软件测试的持续关注和投入,软件测试发展迅速,测试手段和方法获得了长足进步,软件测试用事实说话,获得了业界广泛认可,软件测试“卓有成效”;但是另一方面,作为一个年轻的领域,有关软件测试的基础理论研究和实践仍然有待探索和积累,投身于这个领域,将获得更多的机遇和挑战,未来“大有可为”。

山东省计算中心多年来精耕于信息技术领域,承担了众多省内外科研项目,为软件测试的发展做出了重要贡献。同时,在对外提供软件测试服务过程中积累了丰富的测试经验,并借此成为业内重要的一极。“十年树木,百年树人”,软件测试的发展最终需要更多人才的培养,应业内同行的邀请,我们组织力量精心编撰了这本著作。在编写本书的过程中,我们力争详尽准确,对每一个知识点都极尽考究。但是限于时间和精力,书中一定还有不少瑕疵和纰漏,恳请读者朋友批评指正,我们将不胜感激。

为此,我们也希望能够有机会在未来继续对这本书持续完善,使它成为一本与众不同的、饱含生命力的“活书”,能够持续地从行业中汲取营养,并将知识远播。

全书共分5篇,系统地讲述了软件测试的一般过程和方法,从软件工程的高度对软件测试重新定位,从软件质量的视角对软件测试的价值重新诠释,注重基础理论讲解的同时也重视实践经验的总结,使全书既有知识面的宽度也有认识理解上的深度。本书编写过程中参阅研讨了大量测试著作以及一些培训和认证的资料,力争使本书知识覆盖面更加广泛,使之可以满足高等院校的日常教学要求,同时对于有进一步深造要求的企事业在职人员也大有裨益。

理论与实践的紧密结合是本书的最大特点,为了编写本书,我们精心准备了一个独立软件项目作为本书介绍方法的实践案例,使读者在本书层层推进、逐步深入的介绍过程中,完成了案例项目的整个测试,不知不觉地在学到理论知识的同时也获得了宝贵的实际项目体验。全书内容结构紧凑、前后呼应、一气呵成。

本书内容组织以行业内的国家标准为依据,是介绍相关标准最多、最全面的测试类书籍,使全书保持了一定的业界权威性,同时又不失于实践性。

另外,衷心感谢山东省信息系统测评工程技术研究中心团队成员张伟、王振操、韩明军、郭莹、张海燕、韩庆良等为本书最终成稿所做出的工作,在本书的编撰过程中他们付出很多,正是他们的辛勤劳动和汗水,最终凝聚成今天的这本著作。

最后,还应该感谢清华大学出版社的诸位朋友和同仁,感谢他们所提供的大力支持和协助,使得本书最终得以面世。

编 者
2009年7月

CONTENTS

软
件
工
程
系
列
教
材

目 录

第一篇 软件工程概述

第 1 章 软件工程的起源概述	3
1.1 软件的发展及特点	3
1.1.1 计算机硬件的发展	3
1.1.2 计算机软件的发展	4
1.1.3 计算机软件的特点	5
1.2 软件危机	6
1.2.1 软件危机的表现	6
1.2.2 软件危机的形成原因	8
1.2.3 软件工程的提出	10
本章小结	11
第 2 章 软件工程概览	12
2.1 软件工程的观念	12
2.2 软件工程的要素	13
2.2.1 软件工程方法	13
2.2.2 软件工程过程	16
2.2.3 软件工程工具	18
2.2.4 CASE 简介	18
2.3 软件工程的基本原则	19
2.4 软件工程的原理	19
2.5 软件开发过程模型	21
2.5.1 瀑布模型	21
2.5.2 原型模型	22

2.5.3	增量模型	22
2.5.4	喷泉模型	23
2.5.5	螺旋模型	23
2.6	软件工程标准	24
2.6.1	标准概述	24
2.6.2	标准分类	24
2.6.3	软件工程相关标准介绍	27
	本章小结	37
第3章	软件过程能力评估与 CMM/CMMI	38
3.1	CMM/CMMI 综述	38
3.2	CMM/CMMI 基本框架	40
3.2.1	CMM	40
3.2.2	CMMI	40
3.3	CMM/CMMI 与软件测试	42
	本章小结	44

第二篇 软件测试概述

第4章	软件质量	47
4.1	什么是软件质量	47
4.2	软件质量管理	48
4.2.1	质量管理基础	48
4.2.2	软件质量管理的手段和方法	48
4.3	软件质量与软件开发、测试	49
	本章小结	50
第5章	软件测试基础	52
5.1	软件测试的历史及演变	52
5.2	什么是软件测试	53
5.3	软件测试的原则	54
5.4	软件测试的分类	55
5.5	软件测试基本方法	57
5.5.1	黑盒测试	57
5.5.2	白盒测试	58
5.5.3	黑盒测试与白盒测试的关系	59

本章小结	60
第 6 章 软件测试过程模型	61
6.1 什么是软件测试过程模型	61
6.2 常见的软件测试过程模型	61
6.2.1 V 模型	61
6.2.2 W 模型	62
6.2.3 X 模型	62
6.2.4 前置测试模型	63
6.2.5 H 模型	64
6.2.6 软件测试模型比较	65
本章小结	65
第 7 章 软件测试生命周期	67
7.1 测试计划	67
7.2 测试分析	68
7.3 测试设计	68
7.4 测试执行	69
7.5 测试评估	70
本章小结	70

第三篇 软件测试一般过程与方法

第 8 章 测试计划	73
8.1 项目启动场景	73
8.2 测试计划	78
8.3 测试计划的编制过程及要素	79
8.3.1 测试的质量需求	80
8.3.2 风险分析	82
8.3.3 测试范围的识别	88
8.3.4 制定测试策略	90
8.3.5 测试资源评估	95
8.3.6 计划任务	97
8.3.7 其他特殊要求	101
8.4 测试计划的编写格式	102
8.5 测试计划实例及点评	104

8.6	测试计划的最佳实践	113
	本章小结	114
第 9 章	测试分析	115
9.1	什么是软件测试需求	115
9.2	测试需求分析过程	116
9.2.1	需求采集	116
9.2.2	测试需求分析	117
9.2.3	测试需求评审	124
	本章小结	125
第 10 章	测试方法与测试设计	126
10.1	静态测试	126
10.1.1	文档检查/审查	126
10.1.2	代码检查/审查	130
10.2	动态测试	134
10.2.1	测试用例概述	134
10.2.2	白盒测试用例设计方法	140
10.2.3	黑盒测试用例设计方法	148
10.3	应用实例讲解	164
10.3.1	单元测试	165
10.3.2	集成测试	189
10.3.3	系统测试	197
10.3.4	验收测试	207
	本章小结	211
第 11 章	测试实施	212
11.1	测试准备	212
11.1.1	测试设备检查	213
11.1.2	测试环境搭建	213
11.1.3	测试环境检查	214
11.2	测试执行	215
11.2.1	测试执行流程	216
11.2.2	监控执行过程	216
11.2.3	测试执行记录	220
11.3	缺陷管理	228
11.3.1	什么是缺陷	228

11.3.2	缺陷分类	229
11.3.3	缺陷报告编写	230
11.3.4	缺陷处理流程及状态跟踪	231
11.4	回归测试	233
11.4.1	回归测试方法	233
11.4.2	回归测试过程	234
11.5	HRMIS 的测试执行过程	234
11.5.1	测试环境搭建	235
11.5.2	单元测试执行情况	237
11.5.3	集成测试执行情况	252
11.5.4	系统测试执行情况	254
	本章小结	280
第 12 章	测试评估	281
12.1	测试评估工作模型	281
12.2	测试评估内容	282
12.2.1	测试计划中的评估	282
12.2.2	测试需求分析中的评估	284
12.2.3	测试方法与设计中的评估	284
12.2.4	测试执行中的评估	285
12.3	测试报告	289
12.3.1	测试报告的一般性要求	289
12.3.2	测试报告要素及实例	289
12.3.3	测试报告的管理	300
	本章小结	300
第四篇 测试管理与过程改进		
第 13 章	软件测试过程组织与管理	303
13.1	软件测试组织	303
13.1.1	人员与团队	303
13.1.2	测试过程组织	306
13.2	软件测试管理	306
13.2.1	测试过程管理	307
13.2.2	配置管理	311
13.2.3	风险管理	313
	本章小结	315

第 14 章 测试度量与过程改进	316
14.1 测试度量	316
14.1.1 什么是度量	316
14.1.2 测试度量内容	317
14.1.3 测试度量分类	317
14.1.4 测试度量过程	318
14.2 测试过程改进	319
14.2.1 测试过程改进内容	319
14.2.2 测试过程改进过程	323
14.2.3 测试过程改进注意事项	323
14.3 测试过程改进模型	324
14.3.1 IDEAL 模型	324
14.3.2 6-Sigma 模型	325
14.3.3 PDCA 模型	327
14.3.4 TMM 模型	329
14.3.5 TPI 模型	330
本章小结	334

第五篇 软件测试工具及其应用

第 15 章 软件测试工具及其分类	339
15.1 软件测试工具分类	339
15.1.1 按照原理分类	339
15.1.2 按照用途分类	341
15.2 软件测试工具的实现原理	344
15.3 软件测试工具的选择原则	345
本章小结	346
第 16 章 功能测试工具	348
16.1 WinRunner	348
16.1.1 概述	348
16.1.2 WinRunner 的应用	349
16.1.3 常见问题解答	355
16.2 Quick Test Professional	356
16.2.1 概述	356

16.2.2 Quick Test Professional 的应用	357
16.2.3 常见问题解答	363
16.3 WinRunner 和 QTP 的区别	364
本章小结	365
第 17 章 性能测试工具	366
17.1 性能测试概述	366
17.1.1 常见的软件性能指标	366
17.1.2 性能测试的步骤	367
17.2 HP LoadRunner	369
17.2.1 概述	369
17.2.2 LoadRunner 的应用	370
17.2.3 常见问题解答	378
17.3 OpenSTA	381
17.3.1 概述	381
17.3.2 OpenSTA 的应用	382
本章小结	387
第 18 章 测试管理工具	388
18.1 TestDirector	388
18.1.1 概述	388
18.1.2 TestDirector 的应用	390
18.2 Bugzilla	395
18.2.1 工具介绍	395
18.2.2 Bugzilla 功能特点	396
本章小结	396
附录 A 案例项目业务及技术背景介绍	397
附录 B 软件工程国家标准目录	399
参考文献	401

PART 1

第

一

篇

软件工程概述

本篇主要对软件工程的起源、软件工程相关基本概念以及软件过程能力评估模型 CMM/CMMI 进行简要介绍,目的是使读者对软件工程及其发展现状有一个概括性了解,从而有助于了解软件测试在软件工程中所处的地位及其发挥的重要作用。

本篇名词解释

软件(software): 是计算机系统中与硬件(hardware)相互依存的另一部分,它包括程序(program)、相关数据(data)及其说明文档(document)。

软件危机(software crisis): 落后的软件生产方式无法满足迅速增长的计算机软件需求,从而导致软件开发与维护过程中出现一系列严重问题的现象。

软件工程(software engineering): 软件工程是研究和应用如何以系统性的、规范化的、可量化的过程化方法去开发和维护软件,以及如何把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。

软件开发过程模型(software development process model): 软件开发过程模型是软件开发全过程、软件开发活动以及它们之间关系的结构框架,它为软件项目的管理提供里程碑和进度表,为软件开发提供原则和方法。

过程(process): “针对一个给定目的的一系列操作步骤”(IEEE-STD-610),或为实现一个给定目标而进行的一系列运作步骤。过程具有这样一些的性质:时间性、并发性、

嵌套性和度量性等。

软件过程 (software process): 开发和维护软件以及相关产品涉及的一系列活动。其中软件相关产品包括项目计划、设计文档、源代码、测试用例和用户手册等。过程是活动的集合,活动是任务的集合,任务是把输入转换为输出的操作。

软件过程能力 (software process capability): 描述了在遵循一个软件过程后,所期待结果的界限范围。

软件过程成熟度 (software process maturity): 指一个特定的软件过程被明确地定义、管理、度量、控制,并且是有效的程度。成熟度可以用于指示企业加强其软件过程能力的潜力。当一个企业达到了一定的软件过程成熟级别后,它将通过制定策略、建立标准和确立机构结构使它的软件过程制度化,而制度化又促使企业通过建立基础设施和公司文化来支持相关的方法、实践和过程,从而使之可以持续并维持一个良性循环。

软件过程管理 (software process management): 对软件产品及其开发、维护和支持所涉及的工作过程进行管理。

软件过程改进 (software process improvement): 为了更有效的达到优化软件过程的目的而实施的改善或改变其软件过程的系列活动。

验证 (verification): “验证”的目的在于按照需求(包括顾客需求、产品需求和产品构件需求)对产品和中间产品进行验证。“验证”过程强调验证准备、验证执行和确定纠正措施。“验证”是一种渐进的过程,因为它要在产品和工作产品整个开发过程中执行,即从对需求进行验证开始,然后是对推进中的工作产品进行验证,最后是对完成的产品进行验证。

确认 (validation): “确认”的目的在于证明,产品或产品构件被置于其预定使用环境中时,适合于其预定用途。“验证”过程与“确认”过程看起来类似,但是它们处理的问题不同。“确认”是要证明所提供的(或将要提供的)产品适合其预计的用途,而“验证”则是要查明工作产品是否恰当地反映了规定的要求。换句话说,验证保证“做得正确”,而确认则保证“做正确的东西”。

内部质量 (internal quality): 产品属性的总和,决定了产品在特定条件下使用时,满足明确和隐含要求的能力。

外部质量 (external quality): 产品在特定条件下使用时,满足明确或隐含要求的程度。

使用质量 (quality in use): 特定用户使用的产品满足其要求,以在特定的使用环境下达到有效性、生产率、安全性和满意度等特定目标的程度。

软件工程的起源概述

软件工程诞生于 1968 年,是北大西洋公约组织(North Atlantic Treaty Organization, NATO)在讨论应对“软件危机”的对策的过程中提出的一个概念,意图通过工程化的生产方式使软件走上工业化道路,从而解决“软件危机”所带来的“危害”。本章从软件的发展阶段及其特点入手,解释了“软件危机”出现的原因,以及软件工程的出现及其发展。

1.1 软件的发展及特点

软件(software)是计算机系统中与硬件相互依存的另一部分,它包括程序、相关数据及其说明文档。用一个简单的公式来表示,即“软件=程序+文档+数据”。

从一定意义上来讲,软件是伴随着世界上第一台计算机的诞生而同时出现的,并且随着计算机硬件的发展而不断寻求突破。而随着软件的复杂程度、规模、应用领域、开发方法等的不断完善,又进而对计算机的运算速度提出了更高的要求。因此,计算机软件与计算机硬件的发展是相互影响、相互制约的。

1.1.1 计算机硬件的发展

计算机硬件作为软件的载体,经历了以下几个发展阶段。

简单的计算工具及机械。人类出于农业生产和产品交换的目的,需要借助一些机械工具进行数字记录和简单的计算,如中国人在公元 13 世纪发明的算盘、苏格兰数学家在 17 世纪早期发明的一种用于计算乘法的“骨质拼条”,以及 17 世纪中期由法国科学家发明的齿轮式加减法等。到 19 世纪 40 年代为止,所有的机械式计算机都没有硬件和软件之分,算法被固化在机械中,如果需要修改算法,只能重新制造计算机。

第一台电子计算机。1946 年 2 月 15 日,世界上第一台通用电子数字计算机 ENIAC(埃尼阿克)宣告研制成功。ENIAC 共使用了 18 000 个电子管,另加 1500 个继电器以及其他器件,其总体积约 90m²,重达 30T,占地 170m²。ENIAC 的成功,是计算机发展史上的一座纪念碑,是人类在发展计算技术的历程中,到达的一个新的起点。这台耗电量为 140kW 的计算机,运算速度为每秒 5000 次加法,或者 400 次乘法,比机械式的继电器计算

机快 1000 倍。ENIAC 最初是为了进行弹道计算而设计的专用计算机。但后来通过改变插入控制板里接线方式来解决各种不同的问题,而成为一台可以进行多种科学计算和军事计算的通用计算机。ENIAC 程序采用外部插入式,每当进行软件中一项新的计算时,都要重新连接线路。有时几分钟或几十分钟的计算,要花几小时或 1~2 天的时间进行线路连接准备。这种外加式的计算机程序和极小的存储容量,使其尚未完全具备现代计算机的主要特征。

电子管计算机时代(1946—1959)。这一阶段的计算机主要用于科学计算,主存储器是决定计算机技术面貌的主要因素。当时,主存储器有水银延迟线存储器、阴极射线示波管静电存储器、磁鼓和磁心存储器等类型,通常按此对计算机进行分类。

晶体管计算机时代(1959—1964)。主存储器均采用磁心存储器,磁鼓和磁盘开始用作主要的辅助存储器。不仅科学计算用计算机继续发展,而且中、小型计算机,特别是廉价的小型数据处理用计算机开始大量生产。

1964 年,在集成电路计算机发展的同时,计算机也进入了产品系列化的发展时期。半导体存储器逐步取代了磁心存储器的主存储器地位,磁盘成了不可缺少的辅助存储器,并且开始普遍采用虚拟存储技术。随着各种半导体只读存储器和可改写的只读存储器的迅速发展,以及微程序技术的发展和运用,计算机系统中开始出现固件子系统。

20 世纪 70 年代以后,计算机用集成电路的集成度迅速从中小规模发展到大规模、超大规模的水平,微处理器和微型计算机应运而生,各类计算机的性能迅速提高。随着字长 4 位、8 位、16 位、32 位和 64 位的微型计算机相继问世和广泛应用,对小型计算机、通用计算机和专用计算机的需求量也相应增长了。

新一代计算机是把信息采集存储处理、通信和人工智能结合在一起的智能计算机系统。它不仅能进行一般信息处理,而且能面向知识处理,具有形式化推理、联想、学习和解释的能力,将能帮助人类开拓未知的领域和获得新的知识。

1.1.2 计算机软件的发展

正如计算机硬件在 60 年内发生了极大变化,同样对于计算机软件来说,从观念到目标等方面都发生了很大变化。

20 世纪 40—50 年代:这一时期,计算机主要用于科学或军事计算。计算机没有装入任何软件,只能识别二进制代码,计算程序被固化到计算机硬件中,完全由计算机硬件实现。软件开发可被称为程序编写。由于每个计算机系统的指令系统都是单独设计,没有规律,0 与 1 组成序列的含义难于辨认和记忆,因此编写、阅读、调试程序都非常困难。此时,软件尚未从计算机硬件中脱离出来。

20 世纪 60 年代:计算机应用不再局限于计算问题,应用范围扩大。计算机大规模进入商业、银行等领域,开始出现操作系统、汇编语言、高级编程语言等系统软件。这一时期编程的目的不再是关心计算机硬件的动作,而是要确定程序人员定义的动作序列。这种动作不是代表计算机的一条指令,而是一系列计算机指令的执行。计算机程序与计算机的指令系统和内部结构无关,它们更直接地接近人们要处理的问题,因此解决问题的规模与复杂性大为增加。

20 世纪 70 年代：这一时期，计算机应用主要是大量的各种类型的非数值计算为特征的商业事务应用，并设计到大量智能性很强的领域。这一时期也开启了计算机网络互联技术，操作系统、数据库系统都得到大力发展。

20 世纪 80 年代：这一时期最为突出的表现是大量的数据库应用系统。特别是关系数据库的发展，使各种商业事务及社会生活，如银行、保险、民航等各方面的问题都转化为文字、数字，以数据形式进入计算机及数据库，形成各种类型的计算机信息系统。计算机应用得到极大的发展。计算机系统不仅用于科学计算、增强军事及社会企业集团的功能，而且个人计算机的字处理程序等软件已广泛应用，提高了个人生产率。

20 世纪 90 年代，在经济全球化背景的推动下，计算机使用方式发生根本变化，网络化成为整个计算机业的长远发展趋势。计算机应用不仅要提高个人的生产率，而且是通过支持跨地区、跨部门、跨时间的群组共享信息协同工作来提高群组、集团的整体生产率。

综上所述，计算机软件的复杂程度从简单到复杂，计算机软件类型从以系统软件为主到以应用软件为主，计算机软件的用户从科学与军事机构到各类企事业单位，经历了极大的变化。现在，计算机软件已经是人们在生产、生活中必不可少的工具。

结合计算机硬件的发展历程可以看出，计算机硬件的发展已在一定程度上超越了计算机软件的发展，随着计算机网络、通信技术的飞速发展，在分布式计算平台上架设各类软件系统已经具备了良好的硬件基础。软件要实现更好的发展前景，已基本不会受到硬件条件的限制。

1.1.3 计算机软件的特点

计算机软件是一种与传统的工业产品截然不同的产品。传统的工业产品通常在既定的生产线下进行物质原材料的组装而成，具有有形的外观，并且成本影响因素较稳定，具有一定的磨损和老化期。

软件同传统的工业产品相比，具有如下特性：

① 软件是一种逻辑产品，与物质产品有很大区别。软件产品是看不到摸不着的，因而具有无形性，是脑力劳动的结晶，是以程序和文档的形式出现的。软件保存在计算机存储器和光盘介质上，通过计算机的执行才能体现其功能和作用。

② 软件的主要生产过程是研制，其成本主要体现在软件的开发和研制上。软件一旦研发成功后，通过复制即可产生大量的软件产品。

③ 软件的生产线，没有原材料组装过程，其生产过程无法通过生产线、机器、流程等进行严格的工艺控制。

④ 软件在使用过程中，不存在磨损、老化的问题。但是为了适应硬件、外界环境、需求等的变化，需要对软件进行修改。这些修改不可避免的引入错误，导致软件失效率升高，从而使软件退化。当修改的成本难以接受时，软件就被抛弃。

⑤ 软件的成本昂贵。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。

⑥ 软件工作牵涉到很多社会因素。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等因素。这些人的因素，常常成为软件开发的困难