# 软件测试主流技术研究

黄霞 著

RUANJIAN CESHI ZHULIU JISHU YANJIU

中国水利水电出版社
www.waterpub.com.cn

# 软件测试主流技术研究

黄 霞 著

中国水利水电出版社
www.waterpub.com.cn
·北京·

## 内 容 提 要

软件测试是一门新兴的学科，同时，又是一门越来越重要的学科。本书主要论述了实用、先进和成熟的测试技术及工具，重点对软件测试计划与策略、软件测试的核心技术：黑盒测试、软件测试的核心技术：白盒测试、软件生命周期中测试的实施等内容进行了分析。

本书结构合理，条理清晰，内容丰富新颖，可供从事软件测试工作的相关技术人员参考使用。

**图书在版编目(CIP)数据**

软件测试主流技术研究/黄霞著. —北京：中国水利水电出版社，2019.1

ISBN 978-7-5170-7401-4

Ⅰ.①软… Ⅱ.①黄… Ⅲ.①软件—测试—研究 Ⅳ.①TP311.55

中国版本图书馆 CIP 数据核字(2019)第 025377 号

| | |
|---|---|
| 书　　名 | 软件测试主流技术研究<br>RUANJIAN CESHI ZHULIU JISHU YANJIU |
| 作　　者 | 黄　霞　著 |
| 出版发行 | 中国水利水电出版社<br>(北京市海淀区玉渊潭南路 1 号 D 座 100038)<br>网址：www. waterpub. com. cn<br>E-mail：sales@waterpub. com. cn<br>电话：(010)68367658(营销中心) |
| 经　　售 | 北京科水图书销售中心(零售)<br>电话：(010)88383994、63202643、68545874<br>全国各地新华书店和相关出版物销售网点 |
| 排　　版 | 北京亚吉飞数码科技有限公司 |
| 印　　刷 | 三河市华晨印务有限公司 |
| 规　　格 | 170mm×240mm　16 开本　17 印张　220 千字 |
| 版　　次 | 2019 年 4 月第 1 版　2019 年 4 月第 1 次印刷 |
| 印　　数 | 0001—2000 册 |
| 定　　价 | 84.00 元 |

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

# 前　言

近年来我国软件行业的迅猛发展，带动了软件测试行业的快速发展，同时提升了我国软件产品的质量，使软件产品得到了更加广泛的应用，为我国软件产业提供了极大的安全保证与信誉保障，并且为我国软件事业未来的发展奠定了基础。

软件测试是开发一个软件必要的步骤，也是一种减少程序错误、证实软件质量的方法。然而，在软件测试中，如何选择测试用例，按什么样的顺序设计测试路径一直是人们研究的重要问题。虽然我国软件测试的规范性正在不断提高，但能够真正担当软件测试工作的人却很少，存在以下问题：

(1)从事软件测试的人员基本功不够牢固，缺乏系统学习和培训，缺乏测试理论知识，只懂得一些表面上的测试技术，不能作更进一步的研究。

(2)专业知识不够扎实，优秀的软件测试工程师既需要专业的软件测试技能、具备软件编程能力，还需要掌握网络、操作系统、数据库、中间件等计算机基础知识，并且熟悉行业领域知识。

(3)没有建立相对完整的测试体系，忽视理论知识，大部分人对软件测试的基本定义和目的不清晰，对自己的工作职责理解不到位。

(4)理论与实践脱节，学习的软件测试技术比较肤浅并且零杂，没有深入理解测试的基本道理，不能进行实际的应用。

由于软件测试新技术、新需求、新观念的发展和变化，因此，在学习软件测试技术的过程中，不仅要掌握其理论原则和方法，更重要的是技术的应用。软件测试人员需要利用大量的时间去

思考、理解软件测试的思想和理念，并运用测试技术和技巧去解决问题。

本书共8章，主要内容为软件测试概述、软件测试计划与策略、黑盒测试技术、白盒测试技术、软件生命周期中测试的实施、面向对象软件测试、主流信息应用系统测试、测试工具。

本书在撰写过程中加入了自己的研究积累以及国内外专家学者的研究成果和论述。由于软件体系结构方面的参考资料众多，所涉及的文献难免会有疏漏，在此表示歉意，同时还要向相关内容的原作者表示诚挚的敬意和谢意。

由于作者水平有限，加之时间仓促，遗漏之处在所难免，恳请读者批评指正。

作　者

2018年5月

# 目　录

# 第 1 章　软件测试概述

近 20 年来，随着计算机和软件技术的飞速发展，软件测试技术研究也取得了很大的突破。软件测试的重要性主要体现在两个方面：软件系统的层次性越来越复杂，上层系统越来越依赖于底层模块的稳健性；软件测试贯穿于整个软件生命周期，无处不在。因此，软件测试是软件质量保证的一个重要手段，只要有软件生产和运行就必然有软件测试。但目前，软件研发人员过剩，软件测试人才不足，需求旺盛。

## 1.1　软件测试的产生背景及发展

### 1.1.1　软件测试的背景

随着软件产业的发展，软件系统的规模和复杂性与日俱增，软件的生产成本和软件因自身缺陷故障造成的损失都大大增加，甚至会带来灾难性的后果。比如美国迪士尼公司的狮子王游戏软件 bug、火星登陆事故、跨世纪“千年虫”问题、爱国者导弹防御系统问题以及 Intel 奔腾浮点除法问题等。软件产品不同于其他科技和生产领域的产品，它是人脑高度智力化的体现，由于这一特殊性，软件存在缺陷几乎在所难免。在开发大型软件系统的漫长过程中，面对纷繁复杂的各种现实情况，人的主观认识和客观现实之间往往存在着差距，开发过程中各类人员之间的交流和配

合也往往不是尽善尽美的。

软件发生错误时会对人类生活造成各种各样的影响。软件测试可以降低各种影响，在一定程度上解放了程序员，使他们能够更专心于解决程序的算法效率，同时也减轻了售后服务人员的压力，因为交到他们手里的程序再也不是那些“一触即死机”的定时炸弹，而是经过严格检验的完整产品。软件测试的发展还为程序的外形、结构、输入和输出的规约和标准化提供了参考，并推动了软件工程的发展。

虽然软件测试技术的发展很快，但是其发展速度仍落后于软件开发技术的发展速度，使得软件测试在今天仍然面临着很大的挑战。软件规模越来越大，功能越来越复杂，在信息化领域所起的作用越来越重要，但如何进行充分且有效的测试成为难题，对分布式系统和实时系统尚缺乏有效的测试方法，测试的安全性也没有统一的标准。

### 1.1.2 软件测试的发展历程

在计算机诞生的初期，并没有系统意义上的软件测试，只是采用调试的方法对系统进行测试，证明系统可以正常运行即可，没有测试方法和测试计划，测试用例的选择主要依靠测试人员的经验。

20 世纪五六十年代，各种高级语言的出现促使软件测试系统的诞生。当时的软件程序相比之前复杂性大大增加，软件测试仍然处于不受重视的地位，软件正确与否很大程度上依赖于编程人员的技术水平。在这一阶段，软件测试的方法和理论进步不大。

20 世纪 70 年代以后，计算机硬件快速发展，为软件测试的发展提供了硬件基础，软件测试在整个软件开发周期中占据的分量越来越大；软件开发技术已经趋向成熟和完善，规模和复杂度日益增加，这给整个软件测试工作带来巨大的挑战，出现很多软件测试方法和理论，随之产生软件测试的理论体系，出现一大批软

件测试人才。

目前软件已经形成产业化，只对软件进行质量控制已经无法满足人们对软件测试的要求，需要对软件质量、成本和进度都进行严格测试。在程序代码活动的基础上，软件测试贯穿整个软件开发过程的各个阶段。

## 1.2　软件缺陷

### 1.2.1　软件缺陷的定义和类型

所谓“缺陷(Bug)”，即计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。

软件缺陷的主要类型有：

1)软件没有实现产品要求的功能。

2)软件出现了不该出现的错误。

3)软件实现了说明没提到的功能。

4)软件没实现规格说明中未明确提及但应实现的目标。

5)软件难理解，不易使用。

### 1.2.2　缺陷严重等级

由于采用的缺陷管理工具不同，缺陷严重等级的级别也会有差异。

1. Blocker(阻碍的)

阻碍开发或软件测试工作，冒烟测试没有通过，不能进行正常的软件测试工作。

2. Critical(紧急的)

1)系统无法测试,或者系统无法继续操作,应用系统异常中止。

2)对操作系统造成严重影响,系统死机,被测程序挂起,不响应等情况。

3)造成重大安全隐患情况,如机密性数据的泄密。

4)功能没有实现,无法进行某一功能操作,影响系统使用。

3. Major(重大的)

1)功能基本上能实现,但在特定情况下导致功能失败。

2)导致输出的数据错误,如数据内容出错、格式错误、无法打开。

3)导致其他功能模块无法正常执行。

4)功能不完整或者功能实现不正确。

5) 导致数据最终操作结果错误。

4. Normal(普通的)

功能部分失败,对整体功能的实现基本不造成影响。

5. Minor(较小的)

链接错误、系统出错提示或没有捕获系统出错信息、数据的重要操作(增删查改)没有提示,出现频率极低,对功能实现造成非致命性的影响。

6. Trivial(外观的)

产品外观上的问题或一些不影响使用的小毛病,如菜单或对话框中的文字拼写或字体问题等。

7. Enhancement(改进的)

对系统产品的建议或意见。除了严重性之外,还必须关注软

件缺陷处于一种什么样的状态，以便跟踪和管理某个产品的缺陷，三种基本的缺陷状态包括：

1）激活状态（Active 或 Open）。问题尚未解决，测试人员新报告的缺陷，或验证后缺陷仍然存在。

2）已修正状态（Fixed 或 Resolved）。开发人员针对缺陷，修改程序，认为已解决问题，或者通过单元测试。

3）关闭或非激活状态（Close 或 Inactive）。测试人员验证已修正的缺陷后，确认缺陷不存在后的状态。

### 1.2.3　缺陷管理流程

根据 SEI TSP 国际标准，缺陷管理流程可以定义如下：

研发计算机必须分为开发机、测试机和发布机。开发工作在开发机上进行，软件测试工作（系统测试）在测试机上运行，最后产品验收和运行在发布机上运行，发布机可能在客户处。

1）每轮测试开始，开发部门提出本次测试重点，开发机上的版本同步到软件测试机上（或通过配置管理工具实现同步）。

2）软件测试工程师进行冒烟软件测试，如果冒烟测试没有通过，则退回给开发部门，等待开发部门重新提交软件测试任务，返回 1）。

3）冒烟测试通过，测试工程师继续执行测试活动，包括传统正规测试和基于经验的测试，如探索式软件测试等。发现缺陷，记录在缺陷管理工具中。

4）开发工程师修改被确认的缺陷（状态为 Assigned）。

5）当软件测试工程师认为软件测试结束，大部分缺陷都发现完毕，开发机上的版本再一次同步到软件测试机上。

6）软件测试工程师对缺陷进行复测，如果问题仍旧存在，则标记为 Reopen，否则标记为 Closed。此时还要对以前测试过的功能进行回归测试。

7）开发工程师对于 Reopen 的缺陷进行修改。

8)当一轮软件测试达到出口标准,软件测试机上的版本同步到发布机上,软件测试任务完成,否则返回第5)步。

## 1.3 软件测试的定义及原则

### 1.3.1 软件测试的定义

软件在交付使用之前需要进行严格测试,软件测试的概念起源于20世纪70年代中期。1972年在美国的北卡罗来纳大学组织了历史上第一次正式的关于软件测试的会议。1973年首先给出软件测试的定义:测试就是建立一种信心,确信程序能够按期望的设想进行。1983年修改为:评价一个程序和系统的特性或能力,并确定它是否达到期望的结果,软件测试就是以此为目的的任何行为。

软件测试不能认为是单纯的程序测试,并非只有在程序编程结束之后才能开始,而是贯穿在整个软件开发过程中。测试的重要性在于,它必须保证所开发的软件达到设计时的需求,免除由于软件自身的"缺陷"带来的"漏洞",最大限度地降低软件开发的成本。

软件测试过程的终极目标是将软件的所有功能在所规定的环境中全部运行并通过,并确认这些功能的适合性和正确性。这是一种使自己确信产品能够工作的正向思维方法。

20世纪80年代早期,软件行业开始逐渐关注软件产品质量,并在公司建立软件质量保证部门QA或SQA。此时的软件测试涵盖了验证(Verification)和确认(Validation)两个概念。

1)验证,即检验软件是否已正确地实现了产品规格书所定义的系统功能和特性。

2)确认,即通过检查和提供客观证据来证实特定目的的功能或应用是否已经实现,一般是由客户或代表客户的人执行,主要

通过各种软件评审活动来实现。

因此，软件测试更为普遍的定义是：

1）软件测试是使用人工或者自动的手段检测（包括验证和确认）一个被测系统或部件的过程，其目的是检查系统的实际结果与预期结果是否保持一致，或者是否与用户的真正使用要求（需求）保持一致。

2）软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计的一批测试用例，并利用这些测试用例运行程序以及发现错误的过程，即执行测试步骤，它是软件质量保证的关键步骤。

### 1.3.2　软件测试的基本原则

在软件测试工作中应当遵守的经验与原则如下：

1）所有测试的标准都是建立在用户需求之上的，测试的目的在于发现系统是否满足规定的需求。

2）应当把“尽早地和不断地测试”作为软件开发者的座右铭，越早进行测试，缺陷的修复成本就会越低。

3）程序员应避免检查自己的程序，由第三方进行测试会更客观、更有效。

4）充分注意测试中的群集现象。一段程序中已发现的错误数越多，其中存在的错误概率也就越大，因此对发现错误较多的程序段，应进行更深入的测试。

5）设计测试用例时，应包括合理的输入和不合理的输入，以及各种边界条件，特殊情况下要制造极端状态和意外状态。

6）穷举测试是不可能的。

7）注意回归测试的关联性，往往修改一个错误会引起更多错误。

8）测试应从“小规模”开始，逐步转向“大规模”。

9）测试用例是设计出来的，不是写出来的，应根据测试的目

的，采用相应的方法去设计测试用例，从而提高测试的效率，更多地发现错误，提高程序的可靠性。

10）重视并妥善保存一切测试过程文档（测试计划、测试用例、测试报告等）。

11）对测试错误结果一定要有一个确认过程。

## 1.4 软件测试模型

### 1.4.1 V模型

图 1-1 所示为 V 模型测试。

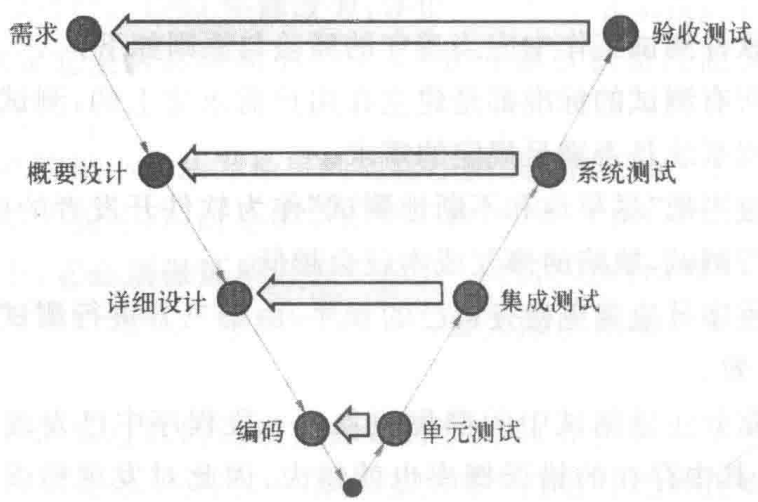


图 1-1 V模型测试

步骤 1：单元测试。单元测试主要由开发工程师在编码时执行。

步骤 2：集成测试。集成测试是相对于详细设计阶段而言的，主要采用由上到下、由下到上或混合方式将模块逐步集成，测试的内容主要是模块与模块、类与类之间的关联性。

步骤 3：系统测试。系统测试是相对于概要设计阶段而言的，主要由软件测试工程师从整体出发，对系统进行全面测试。

步骤 4：验收测试。验收测试是用户对产品进行的测试，一般分为 Alpha 测试和 Beta 测试。验收测试往往由系统维护人员或者用户来完成，需要完全站在用户的立场上进行测试，测试环境也要尽可能与用户的实际环境保持一致，大多数时候，需要到用户现场去进行验收测试工作。

### 1.4.2　W 模型

图 1-2 所示为 W 模型测试。W 模型其实是 V 模型的变种，它提倡的主要思想是软件前置测试理念（即软件测试需要贯穿软件研发的始终）。所以，W 模型又称双 V 模型或前置模型，在需求、设计和编码阶段对产生的工件进行文档评审，一个目的是提出自己的建议和意见，另外一个目的是尽可能理解产品的需求和实现方式。使用前置软件测试法，Bug 在软件前期就可以发现，从而降低软件开发的成本。

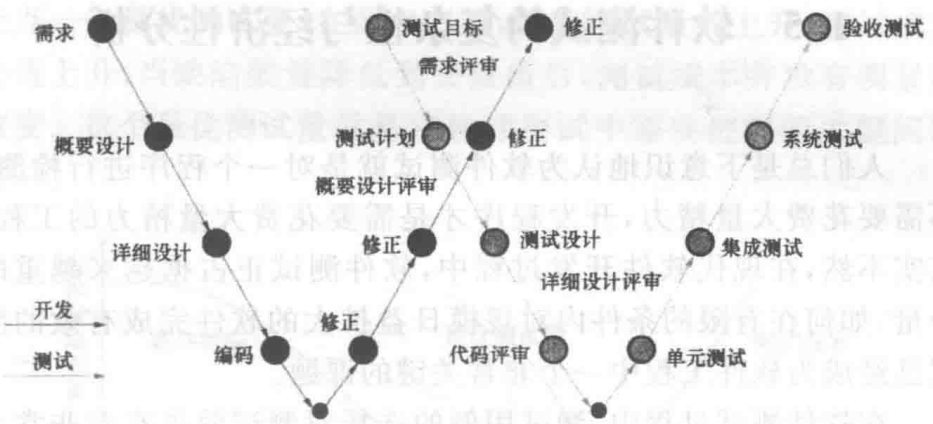


图 1-2　W 模型测试

### 1.4.3　X 模型

图 1-3 所示为 X 模型测试。X 模型将软件系统分为若干模块，对每个模块进行单元测试、集成测试以及系统测试，然后统一对模块进行集成测试。事实上，这里已经提出了“探索式软件测

试”的概念。

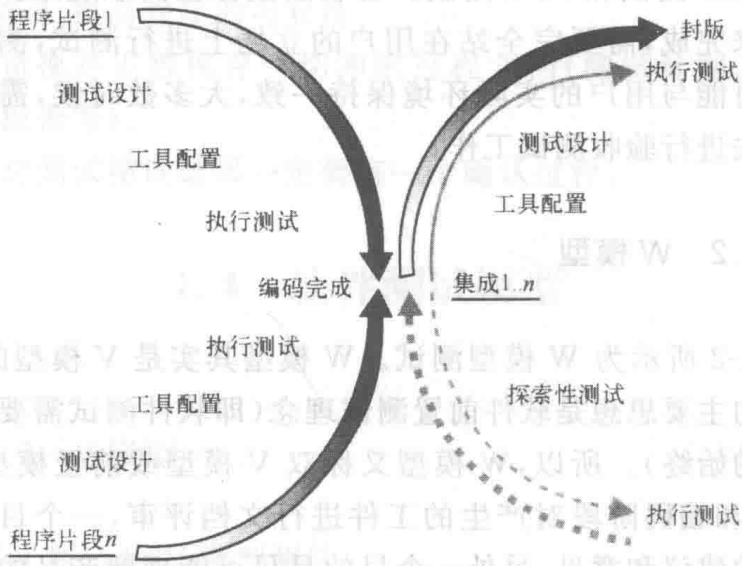


图 1-3　X 模型测试

## 1.5　软件测试的复杂性与经济性分析

人们总是下意识地认为软件测试就是对一个程序进行检测，不需要花费大量精力，开发程序才是需要花费大量精力的工程。其实不然，在现代软件开发过程中，软件测试正占据越来越重的分量，如何在有限的条件内对规模日益扩大的软件完成有效的测试已经成为软件工程中一个非常关键的课题。

在软件测试过程中，测试用例的选择对测试效果有着非常大的影响。设计测试用例时必须非常细致，并且应具备非常高深的技巧，如若不然，就很有可能发生疏漏。这是由以下四个方面的因素决定的。

### 1.5.1　完全测试的不现实性

一般认为，测试工作应将所有可能的输入情况都执行一遍，