测试员培训速成教材

版本号<1.1>

中国测试员论坛 http://www.renju2002.com/cgi.asp

2003-3-12

1. 修订历史

创建者姓名：郭荣力
创建时间：2003-3-12 15:21
版本号：<1.0 >

修改者姓名：郭荣力
修改时间：2003-3-14
版本号：<1.1>
修改内容：将测试原则内容重新规划，添加省略部分。

2. 目录

测试员培训速成教材 1

1. 修订历史 2
2. 目录 3
3. 引言 4
   - 3.1. 本文编写目的 4
   - 3.2. 读者范围 4
   - 3.3. 专业术语说明 4
   - 3.4. 参考资料 5
4. 说在前面的话 5
5. 软件缺陷的正式定义 5
6. 软件测试员的工作内容 6
7. 优秀软件测试员的素质要求 6
8. 软件测试的基本原则 6
9. 测试的基本定律 6
   - 9.1. 完全测试是不可能的 6
   - 9.2. 软件测试是有风险的 7
   - 9.3. 软件缺陷的集群现象 7
   - 9.4. 软件缺陷的抗药性 7
   - 9.5. 测试无法证明软件没有缺陷 7
   - 9.6. 不是所有缺陷都能修复 7
   - 9.7. 测试人员在开发小组中不受欢迎 8
   - 9.8. 测试工作是一项有前途的职业 8
10. 如何对产品说明书进行测试 8
    - 10.1. 高级审查 8
    - 10.2. 低级审查 9
11. 如何进行没有产品说明书的测试 10
12. 动态黑盒测试技术 10
    - 12.1. 通过测试和失败测试 10
    - 12.2. 等价划分 10
    - 12.3. 如何选择测试的数据 10
    - 12.4. 如何进行状态测试 11
    - 12.5. 其他黑盒测试技术 11
13. 报告缺陷的要点 11
    - 13.1. 有效的软件缺陷描述要点 11
    - 13.2. 分离和再现软件缺陷 11
    - 13.3. 软件严重性和等级 12
    - 13.4. 软件跟踪系统 12

3. 引言

3.1. 本文编写目的

这是为培训临时抽调的非专业测试人员参加测试工作，
而编写的包含测试基础知识的速成培训教材。

3.2. 读者范围

临时参加测试的非专业人员。

3.3. 专业术语说明

3.3.1. 软件缺陷

软件中含有符合下面 5 条规则之一的问题称为软件缺陷：

- 软件未达到产品说明书标明的功能。
- 软件出现产品说明书指明不会出现的错误。
- 软件功能超出产品说明书指明的范围。
- 软件未达到产品说明书未指出但应达到的目标。
- 软件测试人员或用户认为软件难以理解，不易使用，运行速度缓慢等问题。

3.3.2. 黑盒测试

指测试人员通过各种输入和观察软件的各种输出结果来发现软件的缺陷，而不关心程序具体如何实现的一种测试方法。

3.3.3. 静态测试

指测试不运行的部分，例如测试产品说明书，对此进行检查和审阅。

3.3.4. 动态测试

通过运行和使用软件进行测试。

3.3.5. 探索测试

通常用于没有产品说明书的测试，这需要把软件当作产品说明书来看待，分步骤逐项探索软件特性，记录软件执行情况，详细描述功能，综合利用静态和动态技术来进行测试。

3.3.6. 等价区间

指测试相同目标或者暴露相同软件缺陷的一组测试用例。

3.4. 参考资料

《Software Testing》（美）Ron Patton 著 Copyright© 2001 by Sams Publishing

4. 说在前面的话

你也许还不怎么了解软件测试为何物，但是希望你已经清楚，无论人们如何努力，缺陷仍然会出现。你需要了解如何计划测试，从那里找到缺陷，如何报告他们，那些软件缺陷要修复，那些要推迟。

最后，你应该明白，软件测试是一项复杂而艰巨的任务。要想获得成功，需要组织、训练和实践。软件缺陷给人们带来很大的危害，测试员需要在他们传播之前，高效而专业地找出这些缺陷。

5. 软件缺陷的正式定义

它的具体定义参见专业术语说明部分，在对于其中不易理解的地方重点给予说明：

其中第三条规则：
如果软件含有产品说明中根本没有存在的功能，这是缺陷。

第四条原则：
产品说明书虽然没有提到，但是按照常理应该达到的功能。

关于第五条原则：
每一个使用过一些软件的人都会对如何改进软件有一些看法，软件无法满足所有人的需求，所以，作为测试人员，需要全面客观地作出合情合理地评价。

6. 软件测试员的工作内容

软件测试员的目标是发现软件缺陷。需要注意，有的测试员只是为了证明软件可以运行，而不是找到缺陷，这是非常错误的做法。
软件测试人员是第一个客户，需要代表客户说话，反映客户的感受，力求完美。
同时应当早一些找到缺陷。

7. 优秀软件测试员的素质要求

- 探索精神：不怕接触新的事务，进入陌生的环境，乐于拿到新软件，并运行和观察它。
- 发现缺陷的能力：善于发现问题，喜欢猜测。
- 不懈努力：不停尝试，从不心存侥幸，而是尽一切能力去寻找。
- 创造性：不满足于显而易见的问题，想出富有创意甚至超常的手段来寻找缺陷。
- 追求完美：力求完美，但是知道某些无法企及时，不去苛求。
- 判断准确：软件测试人员需要具有决定测试内容，测试时间，以及判断所看到的现象是否算真正缺陷的能力。
- 老练稳重：不怕坏消息，知道如何告诉程序员他的程序有缺陷，同时尊重程序员的劳动成果，做到态度诚恳而冷静。
- 说服能力：测试人员需要具有说服程序员为什么软件缺陷必须修复的能力。

8. 软件测试的基本原则

测试的原则是 Good enough。当软件缺陷下降到一定水平的时候，再花很大的力气也将收效甚微，软件测试开发和测试工作有其工作成本，不可能无限测试修改下去，所以应该知道什么时候是足够好了，测试之前应该确立一个测试标准，过度测试和测试不足同样是不可取的。

9. 测试的基本定律

9.1. 完全测试是不可能的

作为非专业测试人员，可能会认为拿到软件就可以进行完全测试，找出所有软件缺陷，遗憾的是，这是不可能的，即使最简单的程序，也无法做到。主要原因有一下四点：

- 输入的工作量太大
- 输出的结果太多
- 软件运行时可能的分支太多
- 软件说明书缺少客观标准

9.2. 软件测试是有风险的

因为无法使用全部的测试用例，所以软件测试具有漏掉软件缺陷的风险。增大测试用例数量，会增大测试成本，反之会漏掉大量缺陷，所以，需要使用一些方法来合理选择测试用例，优化测试过程，以达到在有限的时间和有限数量的测试用例中发现大多数缺陷的目的。

9.3. 软件缺陷的集群现象

发现一个缺陷，通常会在他的周围发现很多缺陷。原因有三点：

- 程序员的怠倦。程序员是人，有疲劳的时候，这段时期编的代码可能会有较多问题。
- 程序员会犯同样的错误。一个程序员喜欢在类似的地方犯自己容易犯的错误。
- 缺陷的关联性。一系列缺陷可能与某个比较深层的关键缺陷有关。

9.4. 软件缺陷的抗药性

软件测试越多，发现的缺陷会越少，越难发现曾经发现的缺陷，这需要不断采用新的方法来测试，避免这种现象。

9.5. 测试无法证明软件没有缺陷

只要发现缺陷即可证明软件不完美，但是没有发现缺陷却并不能证明软件完美。

9.6. 不是所有缺陷都能修复

测试员必须接受的现实是，并非所有缺陷都能修复，原因有以下几点：

- 没有足够的时间
- 不算真正的缺陷
- 修复的风险太大
- 不值得修复
- 是否需要修复

修复软件缺陷的决策过程，通常由软件测试员，项目管理员和程序员共同参与，他们各自站在自己的角度，对软件缺陷是否修复都有自己的看法和观点。

9.7. 测试人员在开发小组中不受欢迎

检查和批评同事的工作，挑毛病，公布发现的问题，是会受人欢迎的。
下面是一些建议：

- 早点找出软件的缺陷。
- 控制情绪。
- 不要总报告坏消息。

9.8. 测试工作是一项有前途的职业

软件业刚刚开始发展的时候，软件测试一般是事后考虑的，那时候，软件产品很小，也不复杂，使用计算机的人也不多，项目小组几乎没有人会去调试其他人的程序。软件即使出问题，也相当容易修复。
现在，软件招聘广告上很容易可以找到招聘软件测试员的栏目，软件业已经发展到强制使用专业测试人员了。毕竟，生产低劣软件的代价太高了。
我们仍然可以看到一些软件公司使用很原始的开发方法进行开发，边写边改，不重视软件质量控制，当时，大多数软件都是采用正规的开发方法完成的，软件测试员是核心小组的成员，有经验的测试人员是被尊重的，同时对测试人员的本身素质和技术能力的要求也越来越高。这对于测试人员来说，无疑是个好消息。测试已经成为一种非常有前途的职业了。

10. 如何对产品说明书进行测试

10.1. 高级审查

10.1.1. 设身处地的为用户着想

测试员要接触到产品说明书时最简单的做法就是把自己当作客户。如果遇到不懂的地方不要放过，因为早晚需要依此来测试它，晚做不如早做，如果能早一点发现问题最好。

10.1.2. 研究现有的标准和规范

在 Windows 和 Apple 出来之前，几乎每种软件都有不同的界面。现在硬件和软件都已经标准化了，并且对用户的使用计算机的方式作了大量的研究，现在的产品看起来确实符合人类工程学的设计。应该考虑的是产品中应该采用何种标准。下面给出一些参考：

- 公司惯用语和约定。
- 行业要求。
- 国家标准。
- 图形用户界面（GUI）。

- 硬件和网络标准。

测试人员的任务不是定义软件需要符合何种规范，而是检验说明书是否套用正确的标准，有没有遗漏。

10.1.3. 审查和测试同类产品

了解竞争对手的产品或者小组开发的其他类似产品，同类软件产品有助于制定测试条件和测试方法。审查竞争产品时要注意的问题：

- 规模。大软件和小软件的测试方法是不同的。
- 复杂性。软件是否复杂，是否会影响测试。
- 测试性。是否有足够的资源、时间和经验来测试软件。
- 质量/可测试性。软件是否完全依据质量标准计划编写的？是否可靠？
- 多动手。对同类软件要尽可能的使用，追根问底。为审查产品说明书积累经验。

10.2. 低级审查

10.2.1. 检查说明书的属性

需要检查下面的属性：

- 完整性。单独使用是否包含足够信息。
- 准确。目标是否明确，是否有错误。
- 清晰。描述是否合理。
- 一致。产品功能描述是否自相矛盾。
- 贴切。是否有多余信息，功能是否符合用户要求。
- 合理。特定的资源限制下，目前是否可以实现目标。
- 代码无关。定义产品，而不是定义其所依赖的软件设计，架构和代码。
- 可测试。是否为测试员进行验证提供了足够的信息。

10.2.2. 检查说明书的用语

- 总是，每一种，所有，没有，从不。这些是可以相对应设计测试用例的。
- 明显，显然，必然都是诱使人接受假定情况的用语，需要注意。
- 某些，有时，通常，几乎等等是无法测试的。
- 诸如此类、依此类推等等是无法测试的。
- 良好，确定，高效等等是不确定的说法，同样无法测试。
- 如果－那么没有否则的句式，需要考虑否则的情况。即“ 如果 ”没有发生会如何。

11. 如何进行没有产品说明书的测试

如果产品没有项目说明书，对于测试员而讲这是一个难堪的境地。测试人员的任务是根据产品说明书尽早找出软件的缺陷。没有产品说明书或者说明书本身没有经过测试，有很多问题，无法参照，那么，测试将是一个无法完成的任务。
但是，尽管产品说明书没有写，或者不完善，总有人知道产品应该做成什么模样。这个人可能是开发人员，项目管理员或者销售人员甚至是用户，和他们交流，记录收集到的信息并且反复斟酌就可以得到更详细的资料，可以得到可用的产品说明书的替代物。
如果得到的信息仍然不够完善就需要使用探索测试方案。

12. 动态黑盒测试技术

不深入代码细节的软件测试方法称为动态黑盒子测试，经常也被称为行为测试。有效的动态测试需要关于软件行为的一些定义，需求文档或者产品说明书，不必了解内部情况。下面介绍一下经常使用的技术：

12.1. 通过测试和失败测试

通过测试是指确认一个软件能够做什么，并不考验其能力，只使用最简单最直观的测试案例。在设计和执行测试案例时总是首先进行通过测试，在进行破坏测试之前，看看软件基本功能是否实现是很重要的。
纯粹为了破坏软件而设计和执行的测试案例成为失败测试或者迫使出错测试。虽然他的过程与通过测试的过程相同，但是它是为蓄意攻击软件的薄弱环节准备的。

12.2. 等价划分

选择测试用例的重要方法，他是指分步骤地使用等价区间把过多地测试用例缩小到同样有效地小范围的过程。

12.3. 如何选择测试的数据

使用等价划分的边界条件数据。提出边界条件时，必须测试临近边界的合法数据，即测试最后一个可能合法的数据，以及刚超过边界的数据。使用默认、空白、空值、零值或者无来测试。使用非法、错误和垃圾数据进行测试。

12.4. 如何进行状态测试

设计各种状态转换图，测试软件的逻辑流程。

12.5. 其他黑盒测试技术

系统使用上面的测试方法已经可以发现大多数软件缺陷，下面给出一些常见的补充手段。

- 像个愚笨的用户那样去做。
- 在已经找到软件缺陷的地方再去找找。
- 凭借经验、直觉和预感记录那些可行和不可行的技术，如果认为可疑就仔细探究。

13. 报告缺陷的要点

发现了软件缺陷，需要记录下来，不但要记录结果，同时需要详细描述发现的步骤，以备程序员重现问题，并解决它。要求报告写的清楚明了和准确。有时利用截屏技术把当时的情况保存成图片，可以达到一图胜千言的效果。

13.1. 有效的软件缺陷描述要点

- 短小。只解释事实和演示、描述软件缺陷必需的细节。
- 单一。每一个报告只针对一个软件缺陷。
- 明显和通用。使用可以看的懂的通用步骤来描述。
- 再现。软件缺陷报告必须展示再现缺陷的能力。
- 报告缺陷时不做评价。缺陷报告应该只针对产品而不是具体的人。
- 补充完善软件缺陷报告。从发现软件缺陷时起，程序员的责任就是保证它被正常地报告，并且得到修复。

13.2. 分离和再现软件缺陷

分离缺陷的一些技巧：

- 不要想当然接受任何假设。
- 查找时间依赖和竞争条件的问题。
- 与压迫和负荷相关的边界条件、软件缺陷、内存泄漏和数据溢出也许需要使用动态白盒技术查找。
- 状态缺陷仅在特定软件状态中显露出现。

- 考虑资源依赖性和内存、网络、硬件共享的相互作用。
- 不要忽视硬件。

有时候测试人员无法分离和重现缺陷，但是程序员根据提供的线索可以找出问题所在。所以无法分离的缺陷也需要记录下来。

13.3. 软件严重性和等级

软件的严重性表示软件缺陷的恶劣程度，优先级表示修复缺陷的重要程度和应该何时修复。

严重性：

- 系统崩溃
- 操作性错误
- 小问题
- 建议

优先级：

- 立即修复
- 产品发布之前必须修复
- 如果时间允许应该修复
- 可能修复，但也能发布

软件缺陷的优先级在项目期间会发生变化。原来标记为优先级 2 的软件缺陷随着开发时间即将用尽，可能变为优先级 4。发现该缺陷的测试员需要继续监视缺陷状态，确保自己能够同意对其所作的变动。并提供进一步测试数据或说服他人使其得以修复。

13.4. 软件跟踪系统

大多数项目小组采用规则约束由谁来改变软件缺陷的状态。例如只有项目管理员可以决定推迟软件缺陷修复，只有测试员允许关闭软件缺陷等。

重要的是一旦登记了软件缺陷，就要跟踪其生命周期，不可以跟丢了，并且提供必要的信息驱使其得到修复。

缺陷的生命周期：

简单周期：

测试员找到并登记软件缺陷，软件缺陷移交到程序员=>程序员修复软件缺陷，软件缺陷移交到测试员=>测试员确定软件缺陷被修复，测试员关闭软件缺陷。

复杂周期：

发现缺陷（测试员发现并登记缺陷，软件缺陷转到程序员）=>软件缺陷移交到项目管理员=>（以不修复形式解决）项目管理员认为软件缺陷不重要，软件缺陷移交到测试员=>重新激活缺陷（测试员不同意，找出通用失败案例，软件缺陷移交到项目管理员）=>项目管理员同意缺陷需要修复，缺陷转给程序员=>以修复形式解决（测试员确认软件缺陷得以修复，测试员关闭软件缺陷）=>缺陷关闭。