



SOFTWARE TESTING

苏临之

sulinzhi029@nwu.edu.cn



Exercise

- 写出以下英文术语的中文名称。

1. functional testing

2. shared testing

3. alpha-testing

4. dynamic testing

5. specification

6. assertions

7. business logic layer

8. bottom-up testing

9. test-to-pass

10. code-based testing

11. threat modeling

12. compatibility testing

13. condition coverage

14. documentation testing

15. stub module

16. incremental testing

17. test-to-fail

18. top-down testing



Exercise

- 19. equivalence partitioning
- 20. software failure
- 21. system testing
- 22. load testing
- 23. stress testing
- 24. statement coverage
- 25. usability testing
- 26. configuration testing
- 27. driver module
- 28. cause-effect diagram
- 29. software defect
- 30. deductive debugging
- 31. test of independence
- 32. localization
- 33. static testing
- 34. black-box testing
- 35. inclusive constraint
- 36. multiple condition coverage
- 37. backward compatibility
- 38. spiral model
- 39. decision table
- 40. cyclomatic complexity
- 41. basic path set
- 42. acceptance testing



Shared Testing

- ❖ 测试共享：在一定时间内互换测试任务，可理解为“你执行我的测试，我执行你的测试”。
- ❖ 这样可以让他他人想出其他的测试用例来测试，如在动态黑盒测试里至少可以帮助审查等价类划分和测试用例，为测试提供新思路。
- ❖ 请求协助寻找软件缺陷的最佳伙伴是产品支持或客户服务小组。这些人对软件缺陷非常敏感（特别是易用性缺陷）。



Bug Bash

- ❖ 缺陷轰炸指的是在一段时间（一般为若干个小时）内整个测试小组停下指定的常规测试任务，选择软件中某一区域，所有测试员集中测试某个区域或者某组特性，仿佛对这个区域进行“轰炸”。
- ❖ 选择轰炸的区域可能是软件缺陷聚集之处，看是否还有更多的潜伏问题；也可能是怀疑不存在软件缺陷的区域。



α -Testing (Alpha-Testing)

- ❖ α 测试是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试，不能由程序员或测试员完成。
- ❖ α 测试发现的错误，可以在测试现场立刻反馈给开发人员，由开发人员及时分析和处理。其目的是评价软件产品的功能、可使用性、可靠性、性能和支持。尤其注重产品的界面和特色。 α 测试可以从软件产品编码结束之后开始，或在模块测试完成后开始，也可以在确认测试过程中产品达到一定的稳定和可靠程度之后再开始。有关的手册（草稿）等应该在 α 测试前准备好。



β -Testing (Beta-Testing)

- ❖ β 测试是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场，不能由程序员或测试员完成。所以， β 测试是在开发者无法控制的环境下进行的软件现场应用。
- ❖ 在 β 测试中，由用户记下遇到的所有问题，包括真实的以及主管认定的，定期向开发者报告，开发者在综合用户的报告后，做出修改，最后将软件产品交付给全体用户使用。 β 测试着重于产品的支持性，包括文档、客户培训和支持产品的生产能力。只有当 α 测试达到一定的可靠程度后，才能开始 β 测试。由于 β 测试的主要目标是测试可支持性，所以 β 测试应该尽可能由主持产品发行的人员来管理。



Differences

- ❖ 两者主要区别是测试的场所、环境和时序不同：
 - α 测试可以是指把用户请到开发方的场所来测试， β 测试是指在一个或多个用户的场所进行的测试。
 - α 测试可以的环境是受开发方控制的，用户的数量相对比较少，时间比较集中。而 β 测试的环境是不受开发方控制的，且用户数量相对比较多，时间不集中。
 - 一般 α 测试可以先于 β 测试执行。



Outsourcing Testing

- ❖ Outsourcing Testing即外包测试。许多公司采用的一种常用做法是向擅长各方面软件测试的其他公司外包或提交部分测试工作。这虽然比有产品小组的测试员来完成更麻烦、更昂贵，但是一种更有效的共享测试途径。
- ❖ 配置和兼容性测试通常是外包测试的理想选择。
- ❖ 本地化测试是另一个通常被外包测试的例子。



Why Outsourcing Testing?

- ❖ 客户在自己招募人员的过程中，不可避免地会遇到许多棘手的问题，诸如：
 - 招募过程中无法评估该人员的测试水准和能力；
 - 招募后没有为测试人员良好的测试技术培训体系；
 - 无法正确快速地组建测试团队以融合到研发过程中；
 - 无法系统地使测试人员最快限度地掌握最新测试技术用以研发需求；
 - 如果各个项目的进度不是连续性的，测试人员的工作有可能会阶段性停滞而增大了成本；
- ❖ 上述这些问题的解决会大大增加客户的额外成本。如果把这些成本与人员工资成本合并起来，有可能远高于外包成本。



To Be an Outsourcing Tester...

❖ 优点:

- 暂不符合甲方公司的人员可以通过外包公司进入甲方公司工作，从而跟随甲方公司一起工作和学习；
- 甲方公司将更多的非核心项目进行外包处理，就业机会就能够有所增加，同时甲方的工作负担也能够减轻。

❖ 缺点:

- 外包测试员因为必须在甲方公司劳动而缺乏归属感，在地位和待遇上也不如甲方公司正式职员；
- 甲方公司很大程度决定了测试员的去和留，尤其对于软件测试员这样可能不受欢迎的职位来说更是如此。



Outsourcing Testing

- ❖ 外包以优势互补的出发点达到甲乙双方共赢的局面，企业将获得以下收益：降低成本、降低风险、提高质量、提高响应速度和更好的用户使用性。
- ❖ 外包测试有三种形式：现场测试、公司内部测试、设立联合研发中心。



The Development Team Should...

- ❖ 为使外包测试更加有效，开发小组的测试人员应做出以下考虑：
 - 测试公司究竟要执行那些测试任务？
 - 他们遵守哪个进度？
 - 应为测试公司提供哪些内容？如软件说明书、阶段性软件更新以及测试用例。
 - 测试公司提供哪些内容？如至少要提供发现的缺陷。
 - 如何与测试公司联系？
 - 怎样知道测试公司是否满足期望？
- ❖ 总之，不要把软件“扔过去”让测试公司进行测试。



Defect Management

- ❖ 软件缺陷管理是在软件生命周期中识别、管理、沟通任何缺陷的过程。该过程从缺陷的识别开始，直到到缺陷的解决关闭为止，确保缺陷被跟踪管理而不丢失。一般的，需要跟踪管理工具来帮助进行缺陷全流程管理。
- ❖ 每一个软件组织需要明确必须妥善处理软件中的缺陷，因为这关系到软件组织机构生存、发展的质量根本。但是并非所有的软件组织都明确应当如何有效地管理自己软件中的缺陷。



How to Describe a Defect

- 缺陷ID：每一个缺陷需要和一个ID号一一对应，根据某一ID号就可以追踪到对应缺陷的解决状态。
- 缺陷标题：用一句简单的话描述该缺陷，形成缺陷标题。
- 缺陷详细描述：对缺陷的详细描述，缺陷如何复现的步骤等等，之所以把这项单独列出来，是因为对缺陷描述的详细程度直接影响开发人员对缺陷的修改，描述应该尽可能详细。
- 缺陷提交人和提交时间：提交缺陷的人员名和对应提交时间。



How to Describe a Defect

- 缺陷状态：指的是某缺陷在缺陷管理系统中处于解决的进度情况，常见缺陷状态有七种。
- 缺陷严重程度：常见的缺陷严重程度可分为“致命”“严重”“一般”和“细微”四种。
- 缺陷紧急程度：缺陷的优先处理等级即缺陷紧急程度，一般分为1-4级，其中1级最紧迫，4级最不紧迫。
- 缺陷等级：有时候结合严重程度和紧急程度，将缺陷统一划分为若干个等级。



How to Describe a Defect

- 缺陷所属项目/模块：缺陷所属的项目和模块，尽可能更精确地定位至模块。
- 缺陷指定解决人和解决时间：解决缺陷指定的人员名单和最终截止日期。
- 缺陷复核情况：包括复核人、复核时间、复核结果通过与否等内容。
- 测试环境：对处理缺陷所用的测试环境描述。
- 必需附件：对于文字描述不足的地方，需要使用其他媒体设备 (音频、图形或视频等)。



Seven Defect Statuses

- New (发现错误 • 新建)
- Open (打开)
- Fixed (已修复)
- Reopen (再次打开)
- Closed (关闭)
- Rejected (拒绝)
- Delayed (延迟)



Seven Defect Statuses

状态	状态说明	操作人员
发现错误	首次录入	测试人员
打开	已经确认是缺陷，并等待修改	开发经理/项目经理
已修复	修改完成后待回归测试验证	开发经理/开发人员
再次打开	回归测试验证不通过，再次等待修改	测试人员
关闭	回归测试验证通过	测试人员
拒绝	讨论后认为不是缺陷或拒绝修改	开发经理/开发人员
延迟	延迟修改	开发经理/开发人员/缺陷 评审委员会



Process of Defect Management

- 缺陷管理的流程可以概括为：测试人员提交新的错误入库，缺陷状态为“发现错误·新建”；项目经理将缺陷分配给相应的开发人员，缺陷状态为“打开”；开发人员查重现缺陷，做如下处理：如果不是缺陷，则置状态为“拒绝”；如果是缺陷则修复并置状态为“已修复”；如果不能解决的错误，要留下文字说明并保持错误为“拒绝”状态。测试人员查询状态为“已修复”的错误，验证错误是否已解决，做如下处理：如问题解决了置错误的状态为“关闭”，如果问题得不到立即解决则需要置状态为“延迟”。

Five Defect Levels

等级	说明	举例
低	可在发布后再商量是否改进	A. 某些测试的建议
中	不影响功能的正常使用,可以在时间和资源允许的情况下再解决	A. 辅助性说明描述不清楚 B. 显示格式不规范 C. 长时间操作未给用户进度提示 D. 提示窗口文字未采用行业术语 E. 可输入区域和只读区域没有明显的区分标志 F. 系统处理未优化
高	事件是重要的,但是由于解决问题需要花费一定的时间,所以可以用较长的时间解决,如果时间紧可以留下个版本解决	A. 界面文字等错误 B. 打印内容、格式错误 C. 简单的输入限制未放在前台进行控制 D. 删除操作未给出提示



Five Defect Levels

等级	说明	举例
很高	事件是重要的，并且应该在紧急的事件处理之后尽快得到解决，必须在发布前解决	A. 功能不符 B. 缺少功能，与需求不符 C. 数据流错误 D. 程序接口错误 E. 轻微的数值计算错误
紧急	事件非常重要，测试工作无法继续进行，需要马上给予关注解决。	A. 由于程序所引起的死机，非法退出，运行中断，应用程序崩溃 B. 死循环 C. 导致数据库发生死锁 D. 数据通讯错误 E. 严重的数值计算错误



Homework

1. 用java语言写一个判断一个数是否为质数的代码，将方法命名为isPrime；然后编写测试代码，对所编写的的isPrime进行测试。
2. 用java实现获取数组中的最大值和最小值，并将获取最大值和获取最小值分别写在两个方法中（可以命名为findmax和findmin）。编写测试类，对这两个方法进行测试。
3. 以上两题请在2023年12月31日23:59前发送至邮箱sulinzhi029@nwu.edu.cn，邮件标题格式“软件测试-JUnit大作业-姓名-学号”。



THANK YOU!