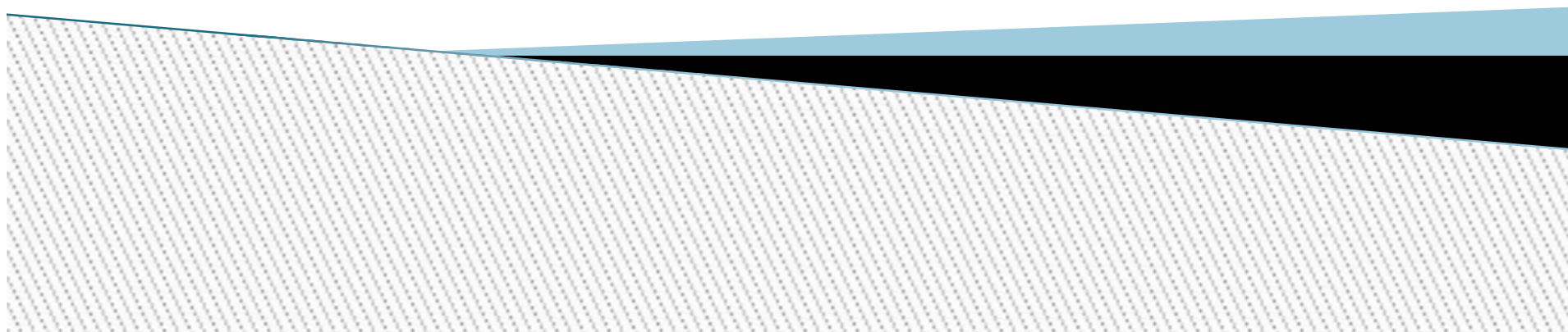
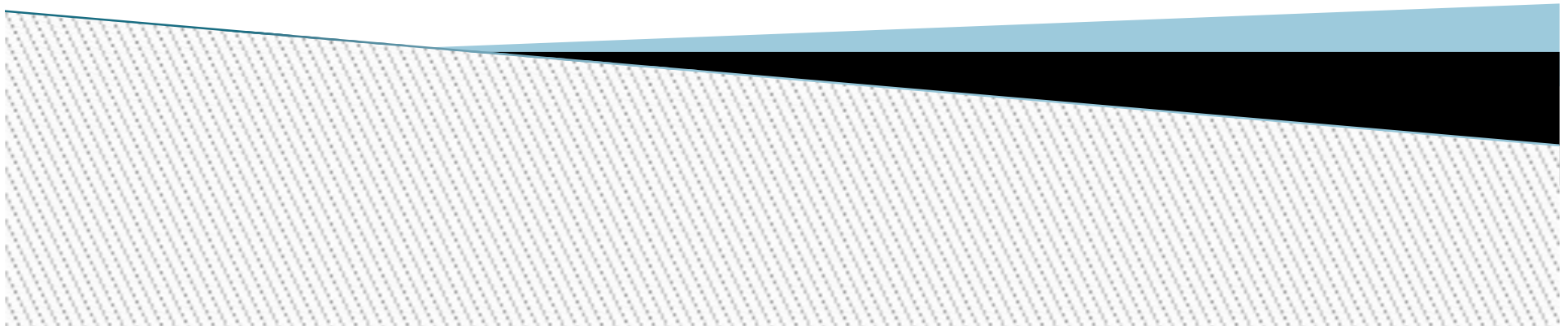


软件测试流程及规范

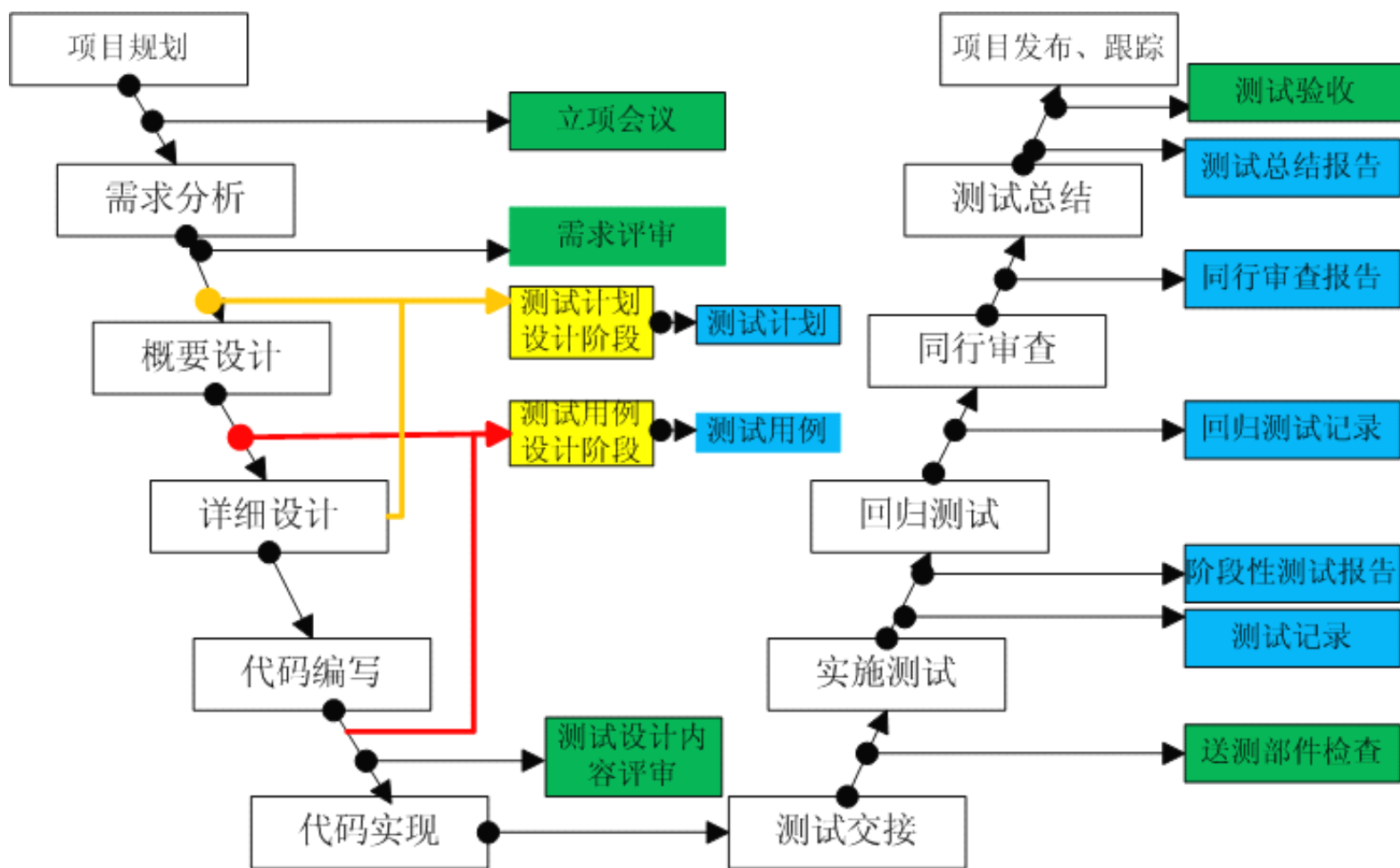


目录

- **1.1 测试流程图**
 - 1.1.1 完整开发流程
 - 1.1.2 测试流程
 - 1.1.2.1 计划与设计阶段
 - 1.1.2.2 实施测试阶段
 - 1.1.2.3 测试总结阶段
- 1.2 计划与设计阶段**
 - 1.2.1 立项会议
 - 1.2.2 需求评审
 - 1.2.3 测试工作启动
 - 1.2.4 测试设计阶段
 - 1.2.4.1 设计测试计划
 - 1.2.4.2 设计测试用例
 - 1.2.5 设计内容评审
- 1.3 实施测试阶段**
 - 1.3.1 测试交接
 - 1.3.2 实施测试
 - 1.3.2.1 实施测试
 - 1.3.2.1 提交阶段性报
 - 1.3.3 回归测试
 - 1.3.4 同行审查
- 1.4 总结阶段**
 - 1.4.1 测试总结报告
 - 1.4.2 测试验收
 - 1.4.3 测试归档
 - 1.4.4 测试工作总结

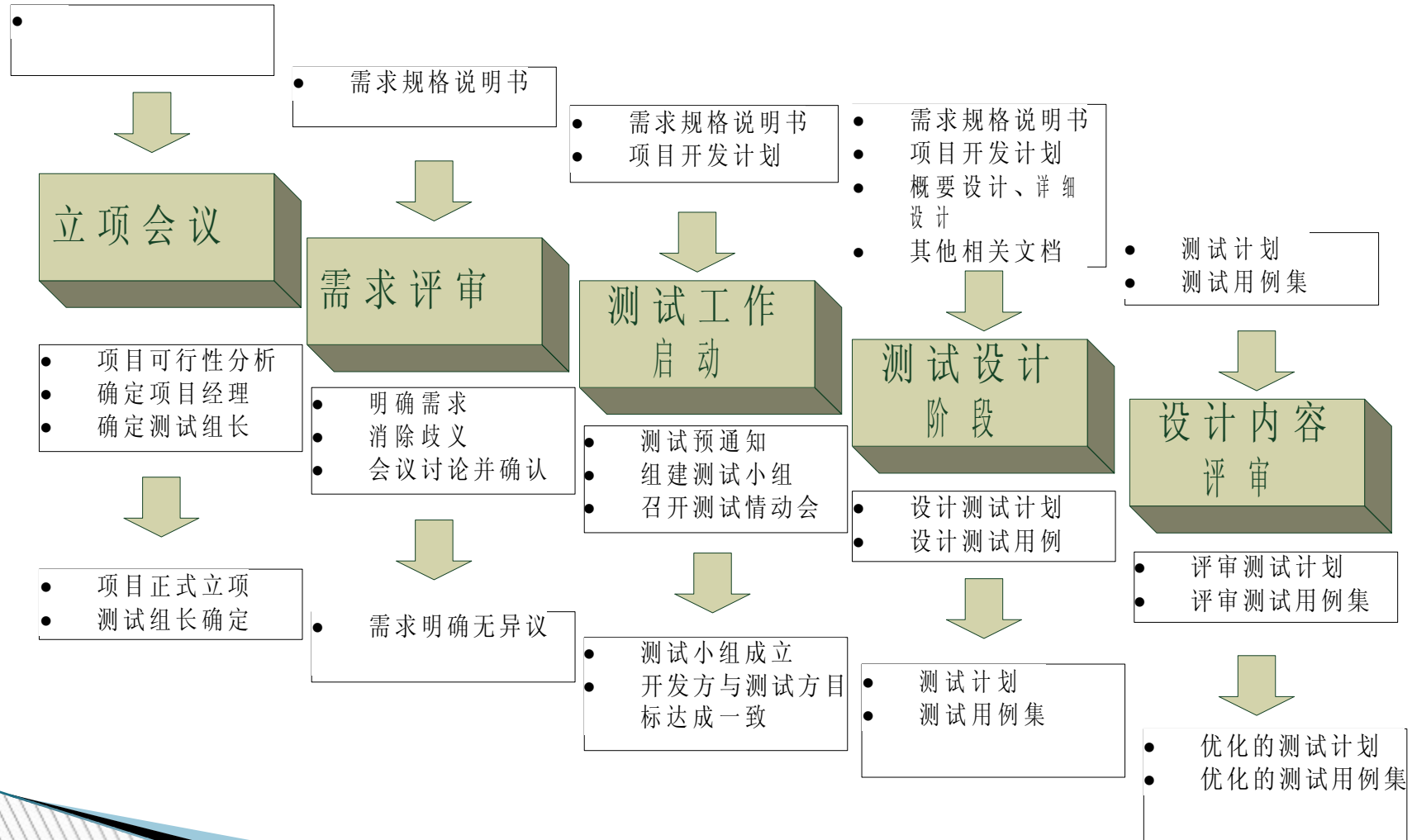


1.1.1 完整开发流程



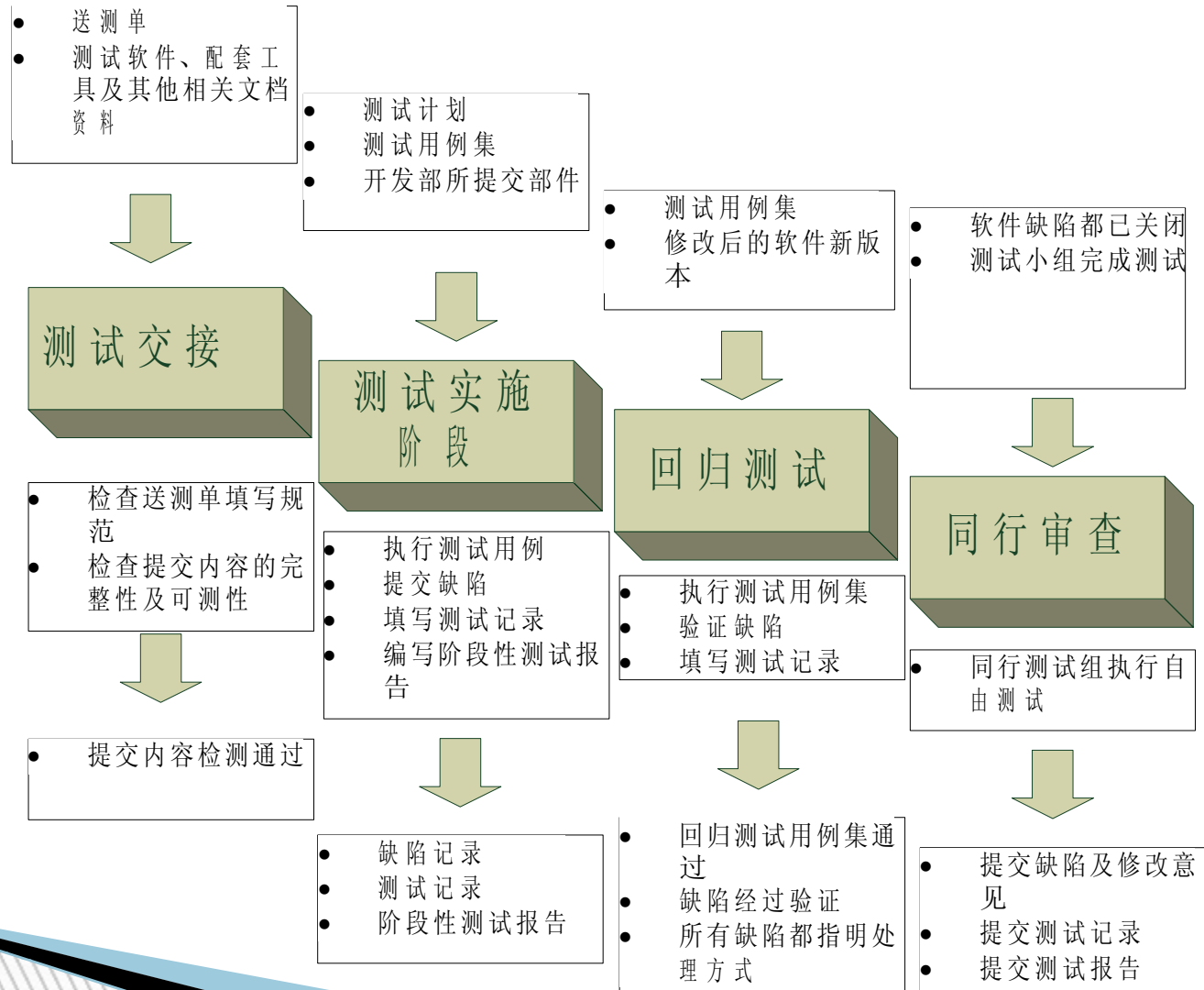
1.1.2 测试流程

1.1.2.1 计划与设计阶段

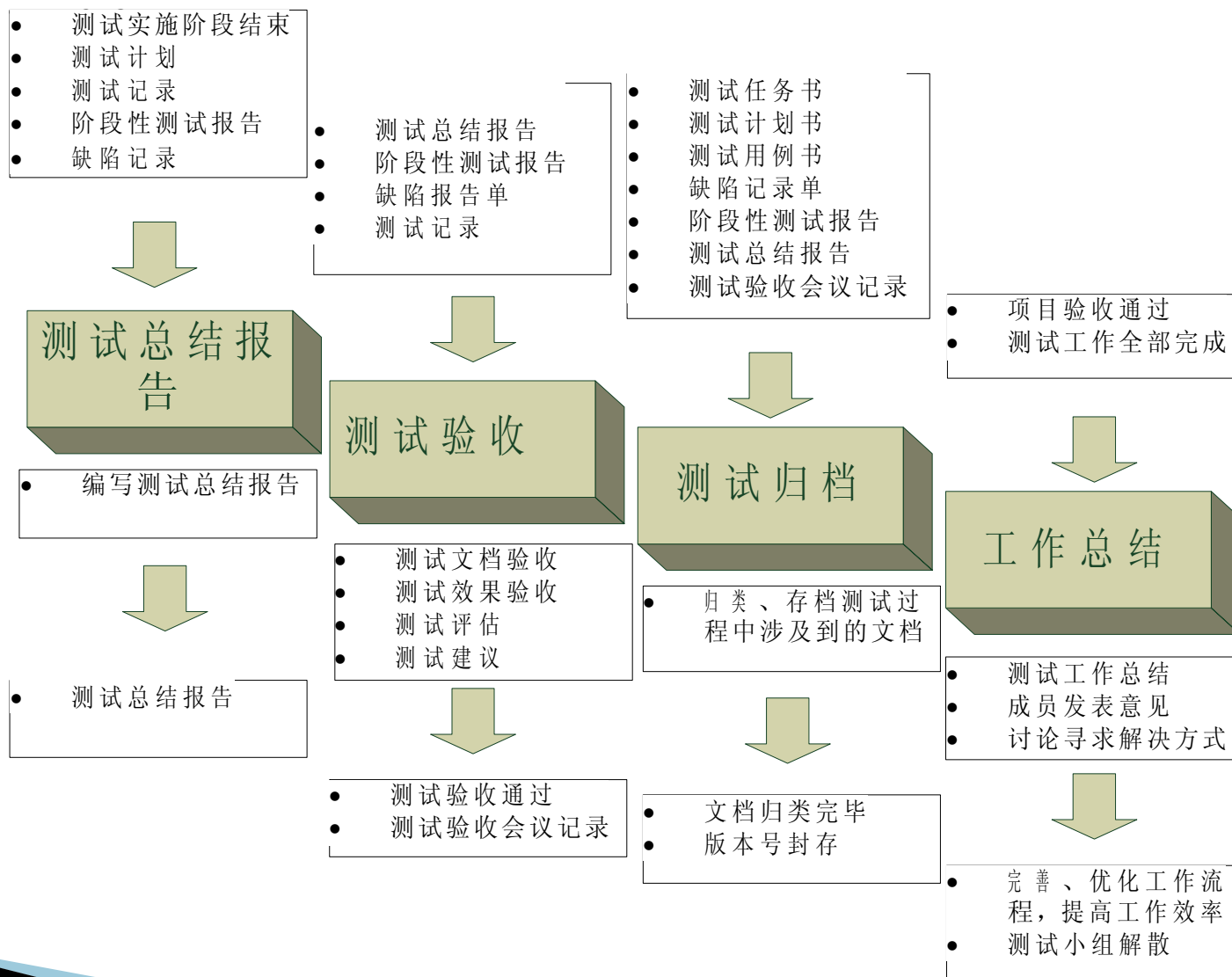


1.1.2 测试流程

1.1.2.2 实施测试阶段



1.1.2 测试流程

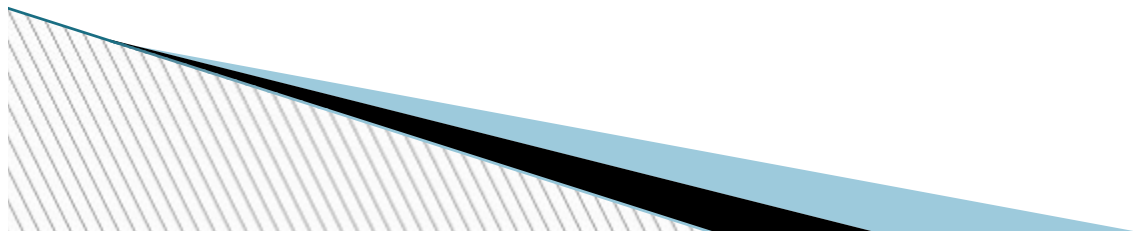


1.2 计划与设计阶段

1.2.1 立项会议

由工程技术委员会召开立项会议，会议主要对项目的可行性进行分析，并且确定项目经理及项目测试组长。

过程要点	详细说明
输入条件	立项会议
工作内容	<ul style="list-style-type: none">● 项目（产品）可行性分析。● 项目经理的确定。● 根据项目信息，测试经理确定测试组长。
退出标准	测试组长确定。
责任人	测试经理（确定测试组长）



1.2 计划与设计阶段

1.2.2 需求评审

注：

1. 需求定义基本完成，此时应在评审会议召开之前发给测试团队，预留时间给测试相关人员熟悉、理解。
2. 测试部参与人员由测试部经理指定，主要由测试组长、测试设计等人员组成（还应包括配置管理人员、质量保证人员）。

1.2计划与设计阶段

1.2.3 测试工作启动

过程要点

详细说明

输入条件

项目（产品）开发计划完成

工作内容

1. 项目/产品经理邮件通知测试组长正式测试交接时间，测试规模预估等，同时提交相关最新项目资料：
 - 项目需求及软件规格定义文档
 - 项目开发计划
 - 开发设计过程中提供概要设计、详细设计文档。
 - 其他相关资料
2. 组建测试小组，确定小组成员。并指定测试设计工程师及测试实施工程师。
3. 召开测试启动会议，开发团队提供需求规格说明书和开发计划，确认开发组与测试组对需要交接的测试内容、测试目标达成一致，统一项目组的目标和测试的工作重点。

退出标准

测试小组成立，双方对测试目标及内容达成一致。

责任人

产品（项目）经理，测试组长

在正式测试任务下达前，开发团队应在项目（产品）开发计划完成后及时向测试团队下达预通知，告之较为确切的测试日期，提供当前最新的相关资料。部门经理和测试组长组建测试小组，并视具体情况决定是否需要调整人力、时间安排、测试环境等其它资源。测试小组成员可预先熟悉必要的项目（产品）资料。

1.2 计划与设计阶段

1.2.4 测试设计阶段

1.2.4.1 设计测试计划

过程要点

详细说明

输入条件

项目需求文档建立，项目开发计划完成

工作内容

根据项目的需求文档、设计文档，按照测试计划文档模板编写测试计划。测试计划中应该至少包括以下关键内容：

- 依据项目背景及要求，确定测试环境。
- 测试需求——需要测试组测试的范围，估算出测试所花费的人力资源 and 各个测试需求的测试优先级
- 测试策略——确定项目的测试计划内容，整体测试的测试方法和每个测试需求的测试方法，同时做好测试进度安排及人员调整。
- 测试资源——本次测试所需要用到的人力、硬件、软件、技术的资源
- 测试组角色——明确测试组内各个成员的角色和相关责任
- 可交付工件——在测试组的工作中必须向项目组提交的产物，包括测试计划、测试报告等等
- 风险管理——列举出测试工作所可能出现的风险

测试计划编写完毕后，必须提交给项目组全体成员，并由项目组中各个角色组联合评审。

退出标准

- 测试计划由项目组评审并通过。
- 在项目开发过程中，要适时的对测试计划进行跟踪，以及评估此计划的完整性、可行性，在项目结束时还要最后评估一下测试计划的质量

责任人

测试设计工程师

针对需求分析文档和项目开发计划文档测试完成后，测试组需要编写测试计划文档、制定测试策略及预估测试过程中的风险，并设计出合理的规避风险的策略，为后续的测试工作提供直接的指导。

1.2 计划与设计阶段

1.2.4 测试设计阶段

1.2.4.1 设计测试用例

过程要点

详细说明

输入条件

测试需求明确，测试计划明确

工作内容

根据测试计划设计测试用例，设计参考原则：

- 等价类划分
- 边界值分析
- 错误推测等
- 业务知识及相关流程

退出标准

- 测试用例需要覆盖所有的测试需求
- 测试用例集需进行评审并通过
- 项目进行过程中，适时的根据需求变更来对测试用例进行维护

责任人

测试组成员

在需求分析文档确立基线以后，测试组需要针对项目的测试需求编写测试用例，在实际的测试中，测试用例将是唯一实施标准。在用例的编写过程中，具体的任务和责任人如下：

1.2计划与设计阶段

1.2.5设计内容评审

过程要点	详细说明
输入条件	测试计划、测试用例集完成
工作内容	评审测试计划内容的正确性及合理性： <ul style="list-style-type: none">● 测试环境、测试资源；● 测试需求范围，各个测试需求的优先级；● 测试策略及风险管理等； 评审测试用例集： <ul style="list-style-type: none">● 测试用例优先级● 测试用例集基于需求的覆盖程度
退出标准	测试计划及测试用例集评审通过
责任人	同行测试组，项目经理，

测试计划及测试用例的设计工作完成后，需通知项目组相关成员召开评审会议。在这之前需要将待评审的内容发给相关人员熟悉和理解。

1.3 实施测试阶段

1.3.1 测试交接

过程要点	详细说明
输入条件	测试设计内容评审完毕，开发团队编码工作完成，并已完成内部测试；
工作内容	<ol style="list-style-type: none">1. 开发组根据测试启动会上所规定的内容，填写送测单，向测试组提交测试内容。2. 测试小组检查提交部件的完整性和可测性：<ul style="list-style-type: none">● 检查接收的测试内容（按照测试启动会上所规定的交接内容）；● 检查程序是否有病毒；● 能否正确安装/卸载；● 检查送测的软件是否完整，能否进行测试；
退出标准	提交部件经测试组检验通过
责任人	产品（项目）经理，测试组长

1.3 实施测试阶段

1.3.2 实施测试

1.3.2.1 实施测试

过程要点	详细描述
输入条件	测试组长于前一工作日定出当日的测试计划，确定可用的测试用例。
工作内容	<ul style="list-style-type: none">● 测试实施工程师根据测试计划中分配给自己的测试任务和提供的测试用例，实施相应的测试用例。● 记录实施用例的结果，提交当日测试纪录。● 提交缺陷。
退出标准	测试用例中的所有任务被执行，结果被记录。
责任人	测试组成员

实施测试用例将花费测试组大部分时间，这些工作都是建立在前期很多计划工作的基础上。

1.3 实施测试阶段

1.3.2 实施测试

1.3.2.1 提交阶段性报告

过程要点	详细描述
输入条件	测试组完成了预定周期的测试任务
工作内容	<p>测试组长根据此轮测试的结果，编写阶段性测试报告（参考测试阶段性报告模板），主要应包含以下内容：</p> <ul style="list-style-type: none">● 测试报告的版本● 测试的人员和时间● 测试所覆盖的缺陷——测试组在这轮测试中所有处理的缺陷，报告测试组长处理的缺陷和实施工程师验证的缺陷。不仅要写出覆盖缺陷的总数，还要写明这些缺陷的去向● 测试新发现的缺陷数量● 上一版本活动缺陷的数量● 经过此轮测试，所有活动缺陷的数量及其状态分类● 测试评估——写明在这一版本中，那些功能被实现了，那些还没有实现，这里只需写明和上一版本不同之处即可● 急待解决的问题——写明当前项目组中面临的最优先的问题，可以重复提出
退出标准	在每轮测试结束之后应尽快将符合标准的测试报告发给全项目组

责任人

测试组长

在约定的测试周期完成之后，测试组长需要总结此次测试的结果，编写阶段性测试报告。

1.3 实施测试阶段

1.3.3 回归测试

过程要点	详细描述
输入条件	在每轮测试中，按照现有的测试用例没有新的缺陷被发现，测试报告中全部的活动缺陷都被解决。
工作内容	<ul style="list-style-type: none">● 测试组将按照测试计划中对于回归测试的策略对产品进行回归测试，回归测试的用例属于测试用例的一部分或者是全部测试用例，但不能超出原先预定的测试用例的范围。● 记录用例实施结果，提交回归测试记录。
退出标准	<ul style="list-style-type: none">● 回归测试所运行的用例全部通过● 缺陷经过验证● 所有缺陷都被指明处理方式
责任人	测试实施工程师

在每轮测试结束之后，由测试组重新拷贝修改后的最新版本，进行回归测试。

1.3 实施测试阶段

1.3.4 同行审查

过程要点

详细描述

输入条件

回归测试结束，所有缺陷都被关闭。

工作内容

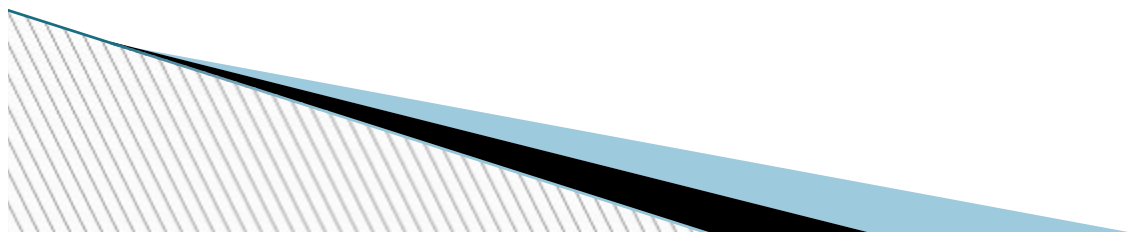
1. 进行对测试组所测试项目或产品的测试审查工作. 基本原则:
 - 不依据所设计测试用例, 进行自由测试.
 - 测试时间保持在3个正常工作日以内.
 - 如发现严重缺陷, 则一轮测试结束后, 更新版本, 执行回归测试.
2. 提交当日测试纪录.
3. 编写同行审查总结报告(报告以简单为好).

退出标准

同行审查没有新的缺陷或没有严重缺陷产生.

责任人

同行测试组



1.4 总结阶段

1.4.1 测试总结报告

过程要点

详细描述

输入条件

测试组完成了所有的测试实施工作, 同行审查结束.

工作内容

测试组长根据测试的结果, 按照测试总结报告的文档模板编写测试报告 (参考测试总结报告模板), 测试报告必须包含以下重要内容:

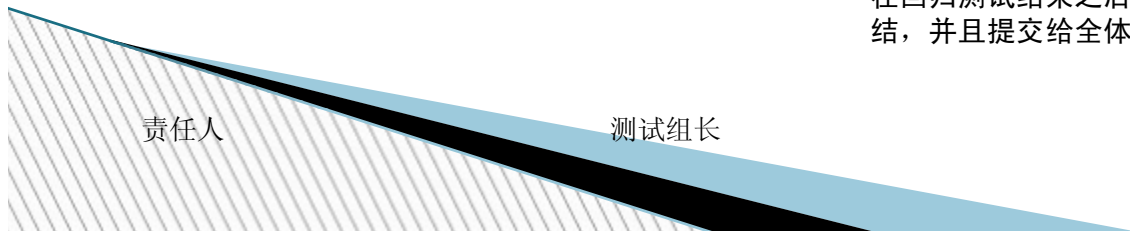
- 测试资源概述——多少人、多长时间。
- 测试结果摘要——分别描述各个测试需求的测试结果, 产品实现了哪些功能点, 哪些还没有实现
- 缺陷分析——按照缺陷的属性分类进行分析
- 测试需求覆盖率——原先列举的测试需求的测试覆盖率, 可能一部分测试需求因为资源和优先级的因素没有进行测试, 那么在这里要进行说明
- 测试评估——从总体对项目质量进行评估
- 测试组建议——从测试组的角度为项目组提出工作建议

退出标准

测试组长完成了符合标准的测试报告, 发送给全项目组。
在回归测试结束之后, 测试组长将要编写测试总结报告, 对测试进行总结, 并且提交给全体项目组, 为产品的后续工作提供重要的信息支持。

责任人

测试组长

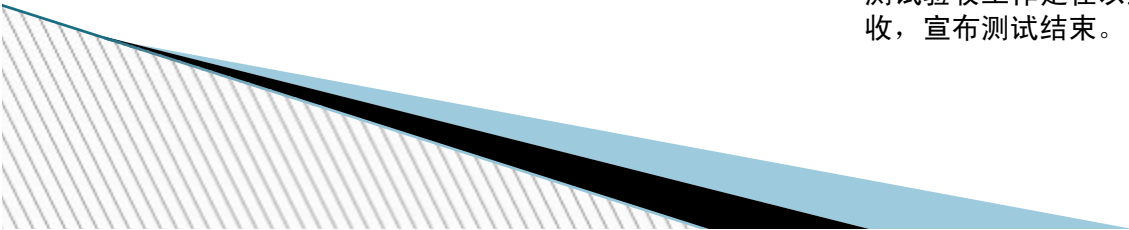


1.4 总结阶段

1.4.2 测试验收

过程要点	详细描述
输入条件	测试组完成了所有的测试实施工作，测试组长完成符合标准的测试总结文档
工作内容	<p>由测启会上约定的验收组成员，对本次测试收进行验收，验收内容包括：</p> <ul style="list-style-type: none">● 测试效果验收——测试是否达到预期目的● 测试文档验收——测试过程文档是否齐全，可信，符合标准● 测试评估——从总体对测试的质量进行评估● 测试建议——对本次测试工作指出不足，需要在以后工作中改进的地方● 宣布测试结束——测试验收组成员签字宣布本次测试结束
退出标准	测试验收通过，测试验收会议记录整理完毕
参与人员	验收组人员，测试经理，测试组长，产品（项目）经理

测试验收工作是在以上工作全部结束后，对测试的过程，效果进行验收，宣布测试结束。

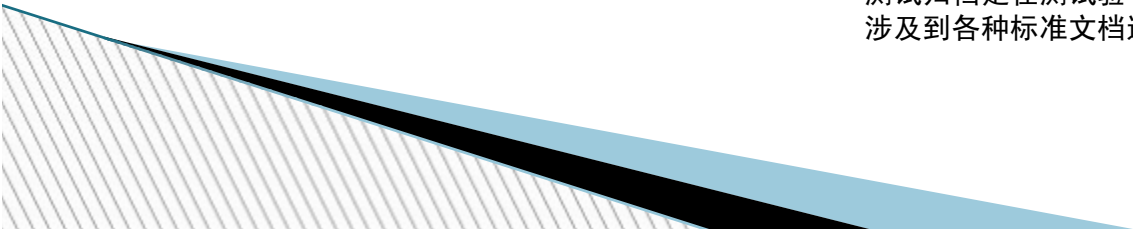


1.4 总结阶段

1.4.3 测试归档

过程要点	详细描述
输入条件	测试验收通过
工作内容	归类、存档测试过程涉及到的文档，主要包括以下文档（必须） <ul style="list-style-type: none">● 测试任务书● 测试计划书● 测试用例书● 阶段性测试报告● 测试总结报告● 测试验收会议记录
退出标准	全部文档归类完毕，版本号封存
责任人	测试组长

测试归档是在测试验收结束宣布测试有效，结束测试后，对测试过程中涉及到各种标准文档进行归类，存档。



1.4 总结阶段

1.4.4 测试工作总结

过程要点

详细描述

输入条件

项目验收工作完成。

工作内容

由质控部经理，测试组长召开项目测试工作总结会议，会议内容主要为：

- 测试组长对项目期间的整个测试组的工作情况进行总结，指出测试工作中存在的问题，同时也对工作中表现好的地方给与肯定。（具体包括整个测试情况、流程实施、人员安排、测试方法等）
- 参与本次项目测试工作的所有成员个人体会和建议。
- 讨论测试工作中出现的问题，寻求更好的解决办法。
- 宣布解散测试小组。

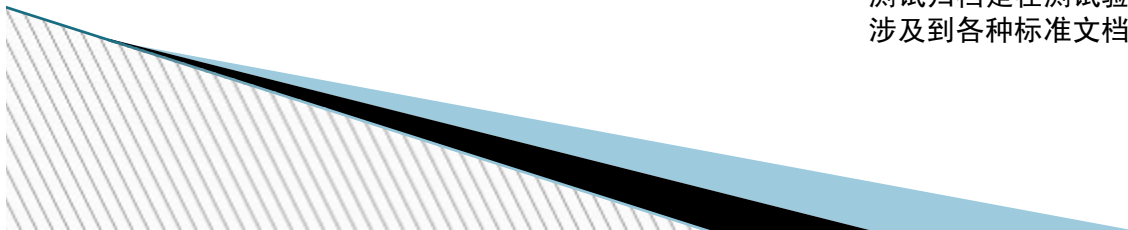
退出标准

所提问题寻求到较好解决方式，测试小组解散

参与人员

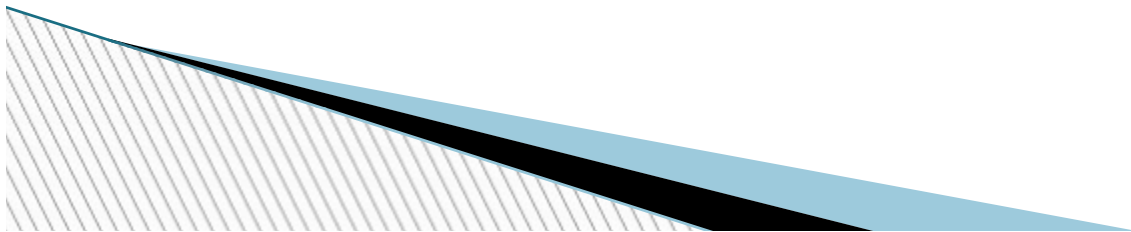
测试部所有成员

测试归档是在测试验收结束宣布测试有效，结束测试后，对测试过程中涉及到各种标准文档进行归类，存档。



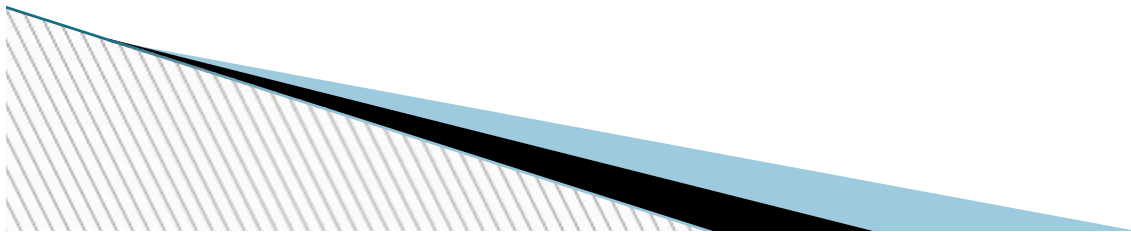
测试计划

- ▶ 产品基本情况
- ▶ 测试需求说明
- ▶ 测试策略和记录
- ▶ 测试资源配置
- ▶ 计划表
- ▶ 问题跟踪报告
- ▶ 测试计划的评审和结果



测试策略

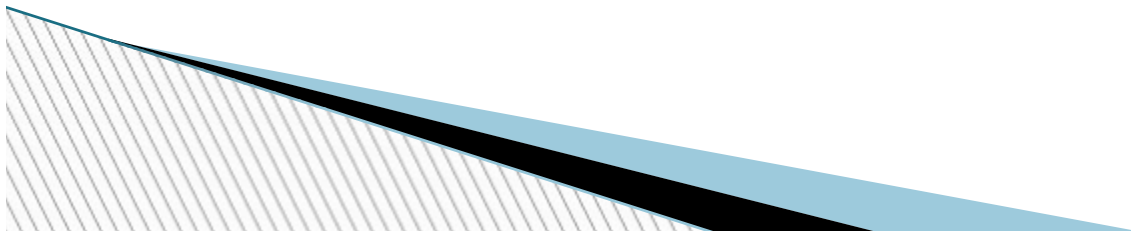
- ▶ 测试策略是制定测试计划的重要参考依据
- ▶ 目的：如何以最少的人力、物力和时间等资源投入来达到最佳测试效果的综合方法。
- ▶ 影响因素：
 - 测试完成的标准
 - 资源状况
- ▶ 针对需求定义测试类型、方法及工具等



测试用例设计考虑因素及基本原则

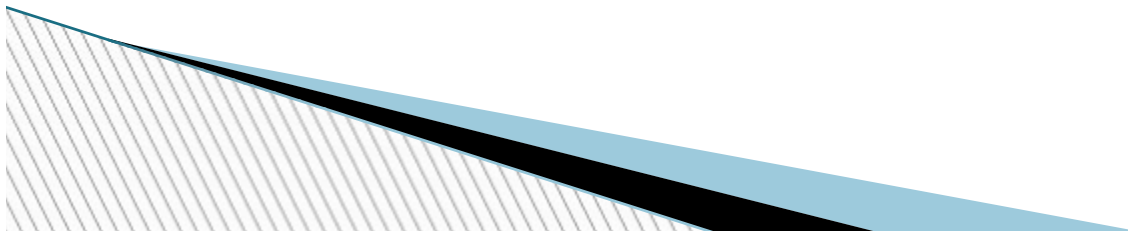
- ▶ 代表性、典型性
- ▶ 正确和错误的或者异常的输入
- ▶ 多考虑用户实际使用场景

- ▶ 避免含糊的测试用例（三种状态）
- ▶ 尽量将具有相类似功能的测试用例抽象并归类
- ▶ 尽量避免冗长和复杂的测试用例



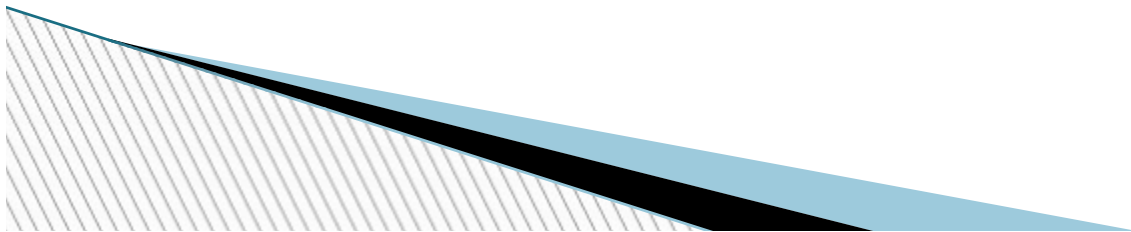
测试用例书写标准

- ▶ 标识符 identification
- ▶ 测试项 test item
- ▶ 测试环境要求 test environment
- ▶ 输入标准 input criteria
- ▶ 输出标准 output criteria
- ▶ 测试用例之间的关联



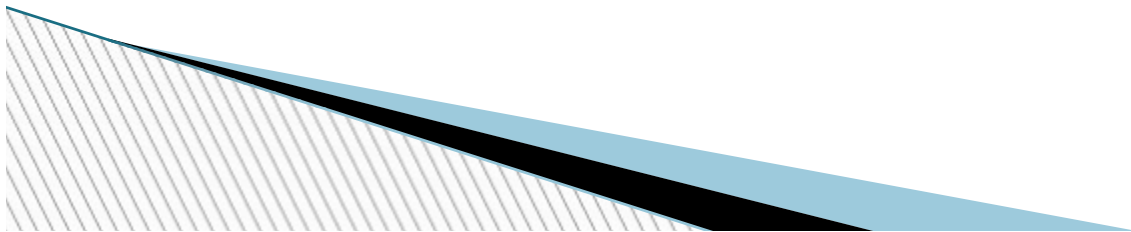
跟踪测试用例

- ▶ 例如，执行一轮测试中，需要跟踪总共执行了多少测试用例，每个人员平均每天使用多少测试用例，测试用例中通过、未通过以及未使用的占多少，未使用的原因是多少
- ▶ 测试用例覆盖率的跟踪
- ▶ 测试跟踪表



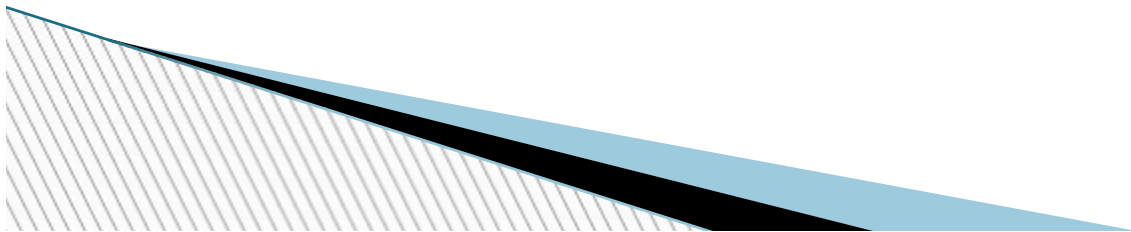
维护测试用例

- ▶ 先前的测试用例设计不全面或不准确
- ▶ 部分严重的软件错误未在测试用例中覆盖
- ▶ 新的版本有新功能的需求或改动
- ▶ 编写的测试用例不规范或者语句错误
- ▶ 旧的测试用例不再适用



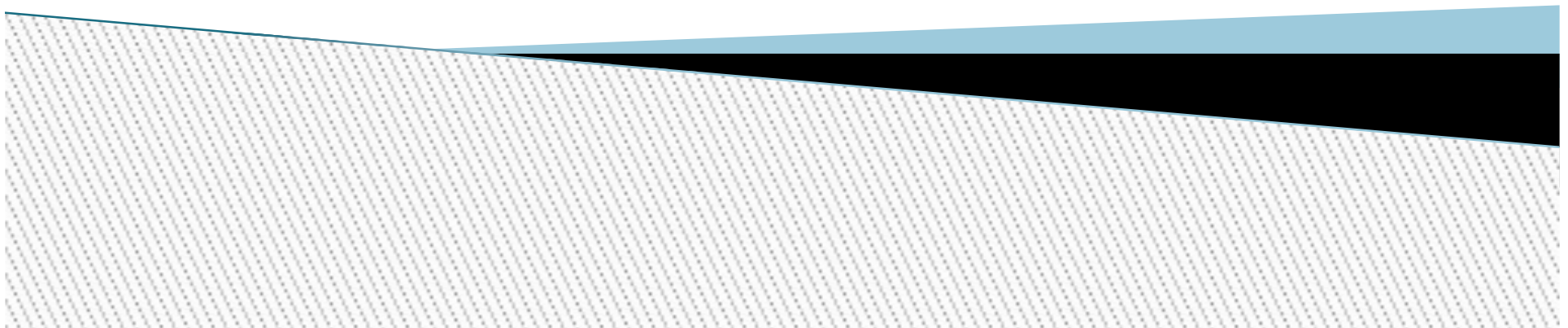
测试阶段分类

- ▶ **单元测试**
 - 程序系统中的最小单元——模块
 - UT的测试用例针对的是被测单元的具体功能。
- ▶ **集成测试:**
 - IT的测试用例关注的是模块间的接口，接口间的数据传递关系，单元组合后是否实现预计的功能。
- ▶ **系统测试:**
 - 验证系统各部件是否都能正常工作并完成所赋予的任务。
 - 压力测试、容量测试、性能测试、安全测试、容错测试
- ▶ **验收测试**
 - 验证系统是否达到了用户需求规格说明书（项目和产品验收准则）中的要求，希望尽可能地发现软件中存留的缺陷，保证系统或软件产品最终被用户接受。

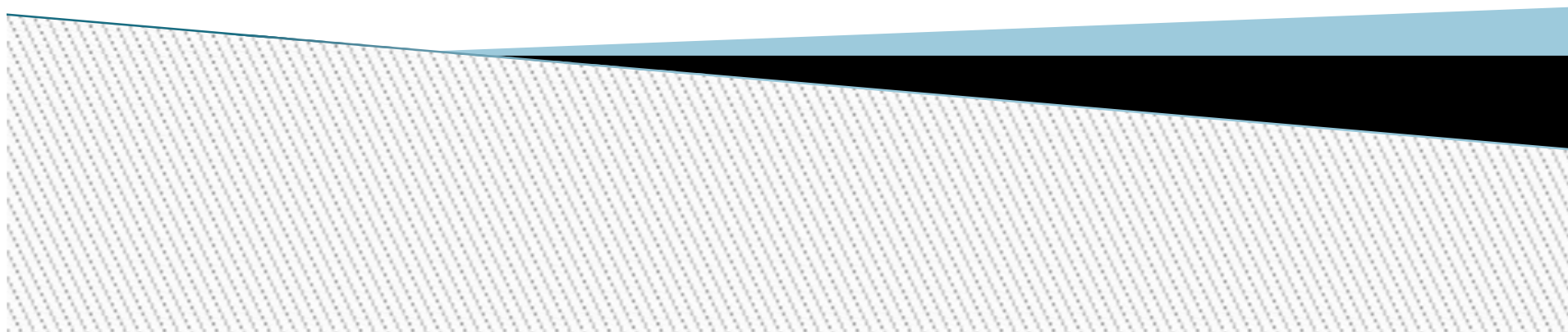


---The end---

Sweety

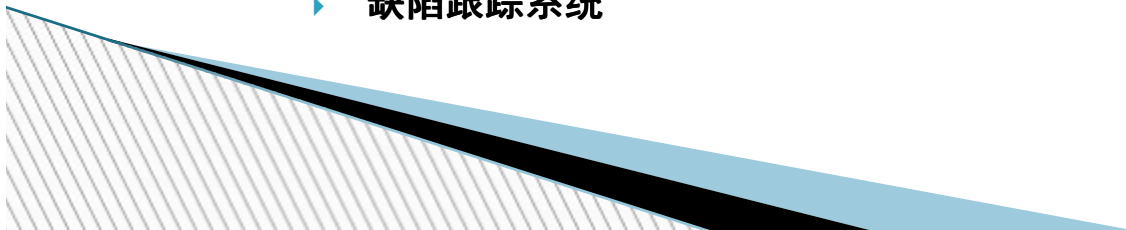


缺陷报告及跟踪



• 目录

- ▶ 一个简单的缺陷报告
- ▶ 缺陷报告的描述
 - 缺陷的严重性和优先级
 - 缺陷的类型和来源
 - 缺陷分布
 - 完整的缺陷信息列表
- ▶ 如何有效的报告缺陷
 - 有效的缺陷带来的益处
 - 有效报告缺陷
- ▶ 软件缺陷的跟踪和处理
 - 软件缺陷的生命周期
 - 缺陷的跟踪处理
 - 缺陷状态
- ▶ 缺陷跟踪系统



• 一个简单的缺陷报告



缺陷报告示例

标题：直接打开一个新邮件时，出现 404 错误

测试环境：Windows XP 和 FireFox 3.0 或 IE 6.0/7.0

前提：事先发送邮件到 Gmail 信箱

步骤：1) 打开浏览器，进入 <http://www.google.com/ig?hl=zh-CN>

2) 点击一封新邮件

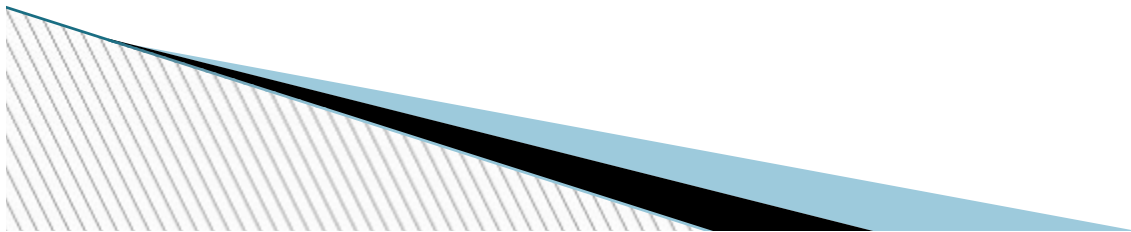
期望结果：在新打开的浏览器窗口显示邮件内容

实际结果：出错，显示“Not Found” 404 错误信息。见附件 Gmail_error.gif

频率：30%

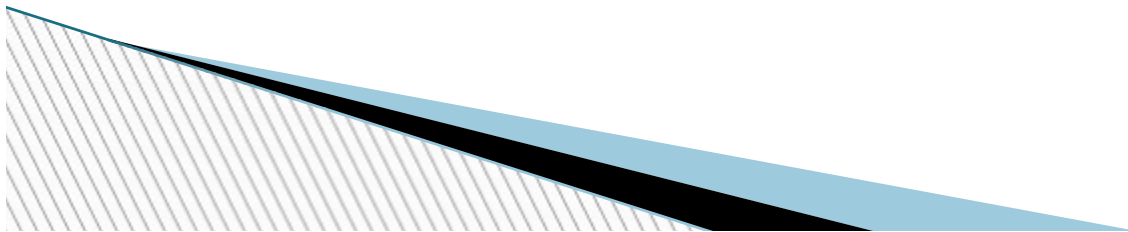
• 缺陷的严重性和优先级

- ▶ **严重性**： 武二线对软件产品使用的影响程度
- ▶ **优先级**： 缺陷必须被修复的紧急程度
- ▶ 缺陷越严重，越要优先得到修正，缺陷严重等级和缺陷优先级相关性很强
- ▶ 有例外？



•缺陷的严重性和优先级

- ▶ 有，如有些缺陷比较严重蛋由于级数的限制或第3方产品的限制，暂时没办法修正，其优先级就会低

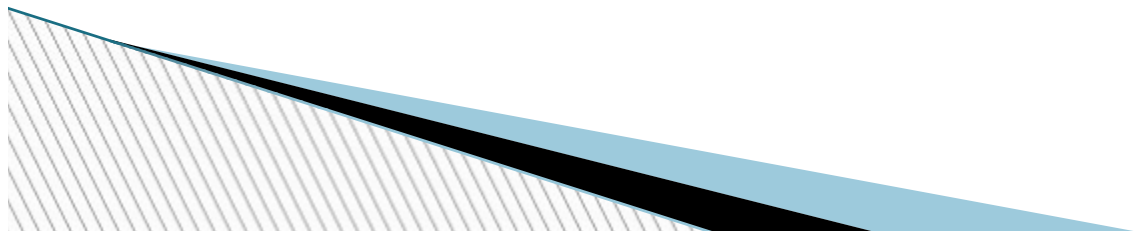


具体说明

缺陷严重等级	描述
0级：致命 (Fatal)	最严重等级，缺陷导致系统任何一个主要功能完全丧失、用户数据受到破坏、系统崩溃、悬挂、死机等。
1级：严重 (Critical)	系统的主要功能部分丧失、数据不能完整保存，系统的次要功能完全丧失，系统所提供的功能或服务受到明显的影响。
2级：一般 (Minor)	系统的次要功能没有完全实现，但不影响用户的正常使用。例如提示信息不太准确；或用户界面差、操作时间稍长等问题。
3级：较小 (Minor)	操作者不方便或遇到麻烦，但不影响功能的操作和执行，如字体不美观、按钮大小不是很合适、文字排列不对齐等一些小问题。

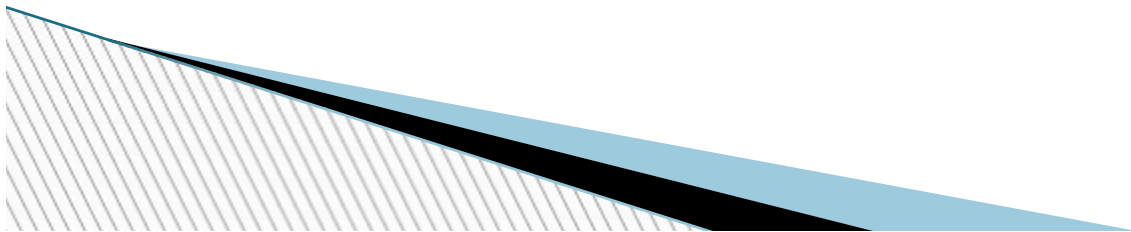
表 9-3 软件缺陷的优先级

缺陷优先级	描述
立即解决 (P1级)	缺陷导致系统几乎不能运行、使用，或严重妨碍测试的执行，需立即修正、尽快修正。
高优先级 (P2级)	缺陷严重，影响测试，需要优先考虑修正。例如不超过 24 小时修正。
正常排队 (P3级)	缺陷需要修正，但可以正常排队等待修正。
低优先级 (P4级)	缺陷可以在开发人员有时间的时候被修正，如果没有时间，可以不修正。



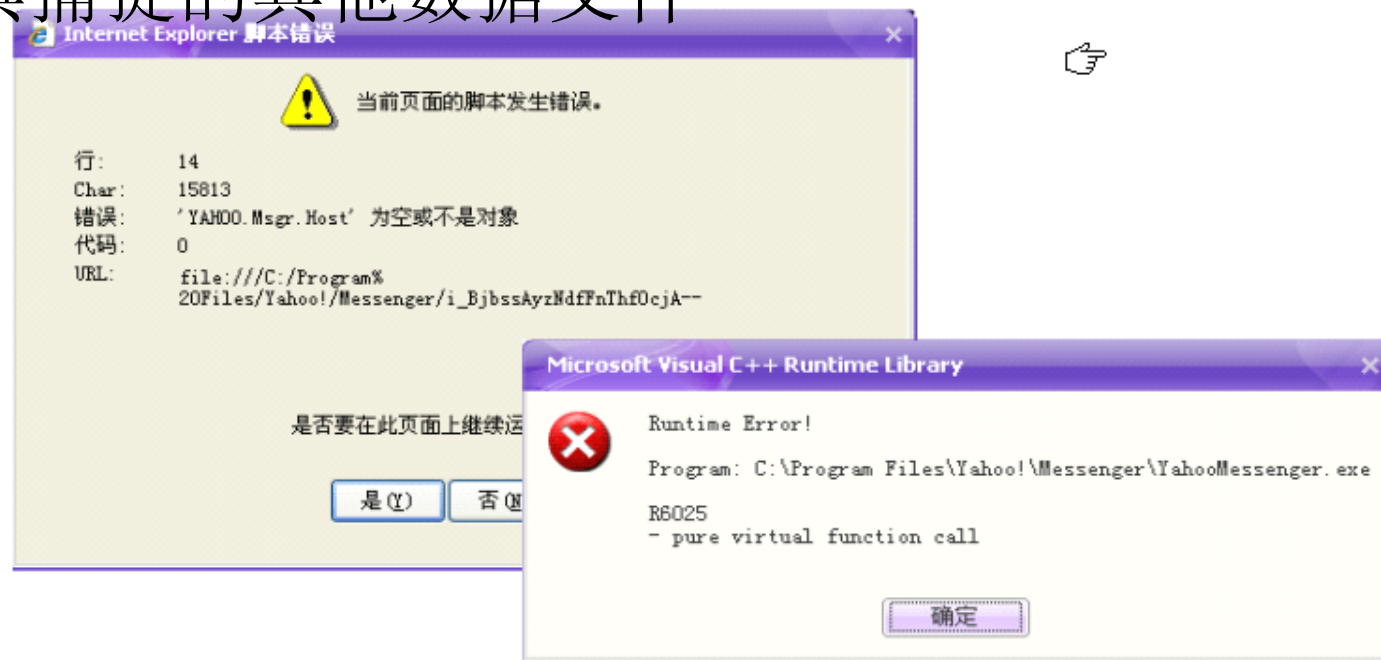
缺陷的类型和来源

- ▶ 弄清楚缺陷的来源，有助于分清责任、权力，有利于缺陷的修正
 - 缺陷类型可以分为业务逻辑、数据处理、接口、UI、性能、安全性、兼容性、配置、文档等
 - 缺陷来源，如需求说明书、涉及规格说明书、代码、用户手册
 - 缺陷关联的模块名、缺陷来自于产品的特定的模块的名称
 - 缺陷发生的阶段，例如需求、系统架构设计、详细设计、编码等



缺陷分布

- ▶ 一张图能胜过千言万语
- ▶ Log file
- ▶ 工具捕捉的其他数据文件



• 完整的缺陷信息列表

❖ ID

❖ 标题

❖ 前提

❖ 环境

❖ 操作步骤

❖ 期望结果

❖ 实际结果

❖ 频率

❖ 严重程度

❖ 优先级

❖ 类型

❖ 缺陷提交人

❖ 缺陷指定解决人

❖ 来源

❖ 产生原因

❖ 构建包跟踪

❖ 版本跟踪

❖ 提交时间

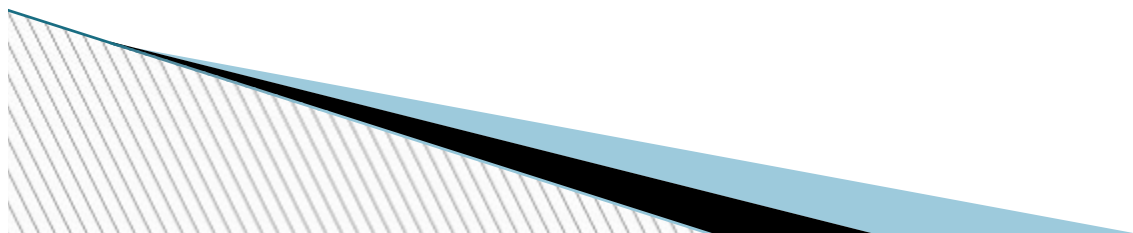
❖ 修正时间

❖ 验证时间

❖ 所属项目/模块

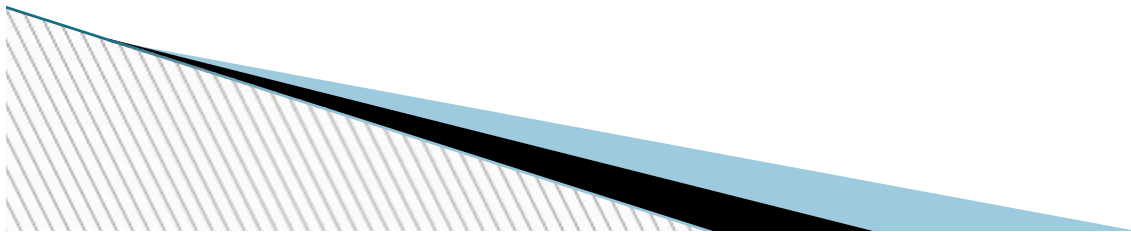
❖ 产品信息

❖ 状态



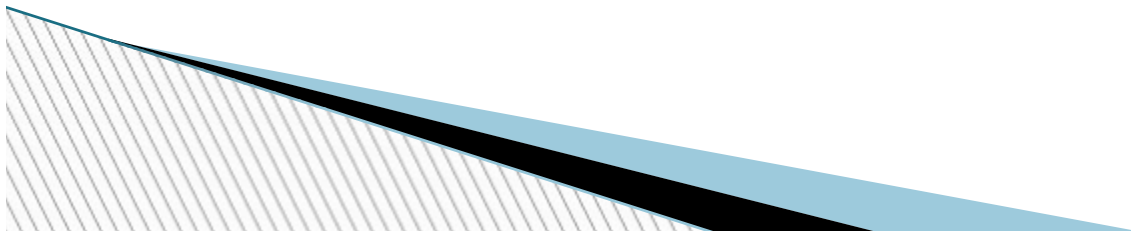
•有效的缺陷描述所带来的益处

- ▶ 容易再现所报告的问题，加快缺陷的修正
- ▶ 提高工作效率
- ▶ 提高测试人员的信任度，有利于开发团队和测试团队之间的沟通和合作
- ▶ 客观、准确的产品质量评估
- ▶ 预防缺陷

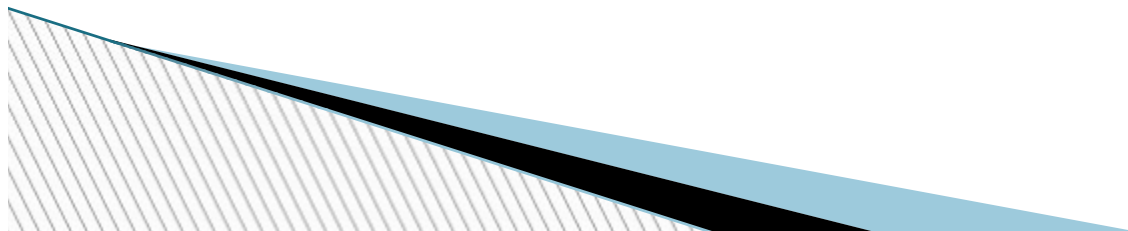
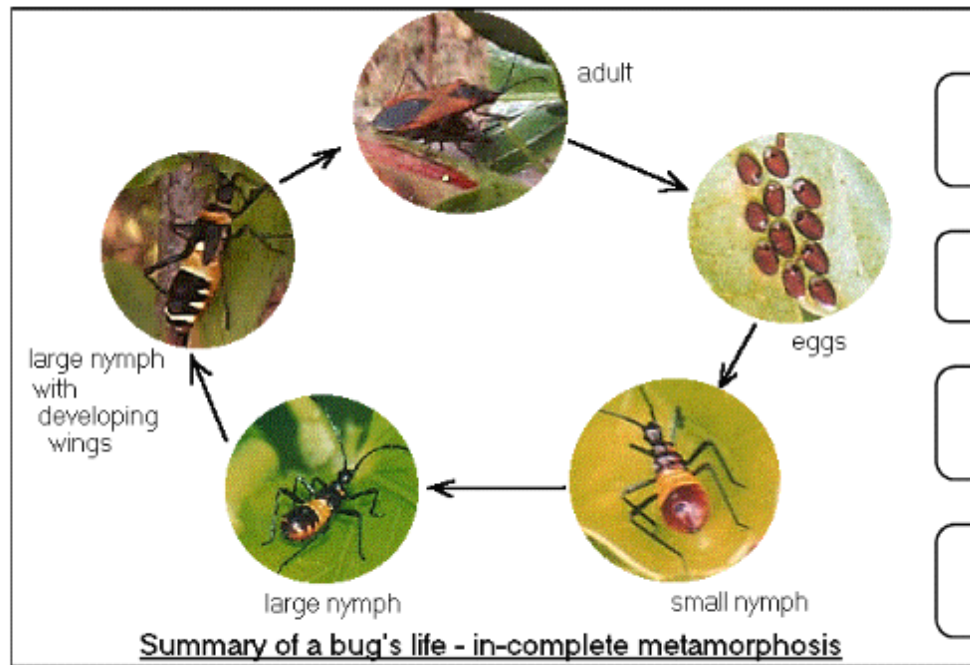


有效报告缺陷

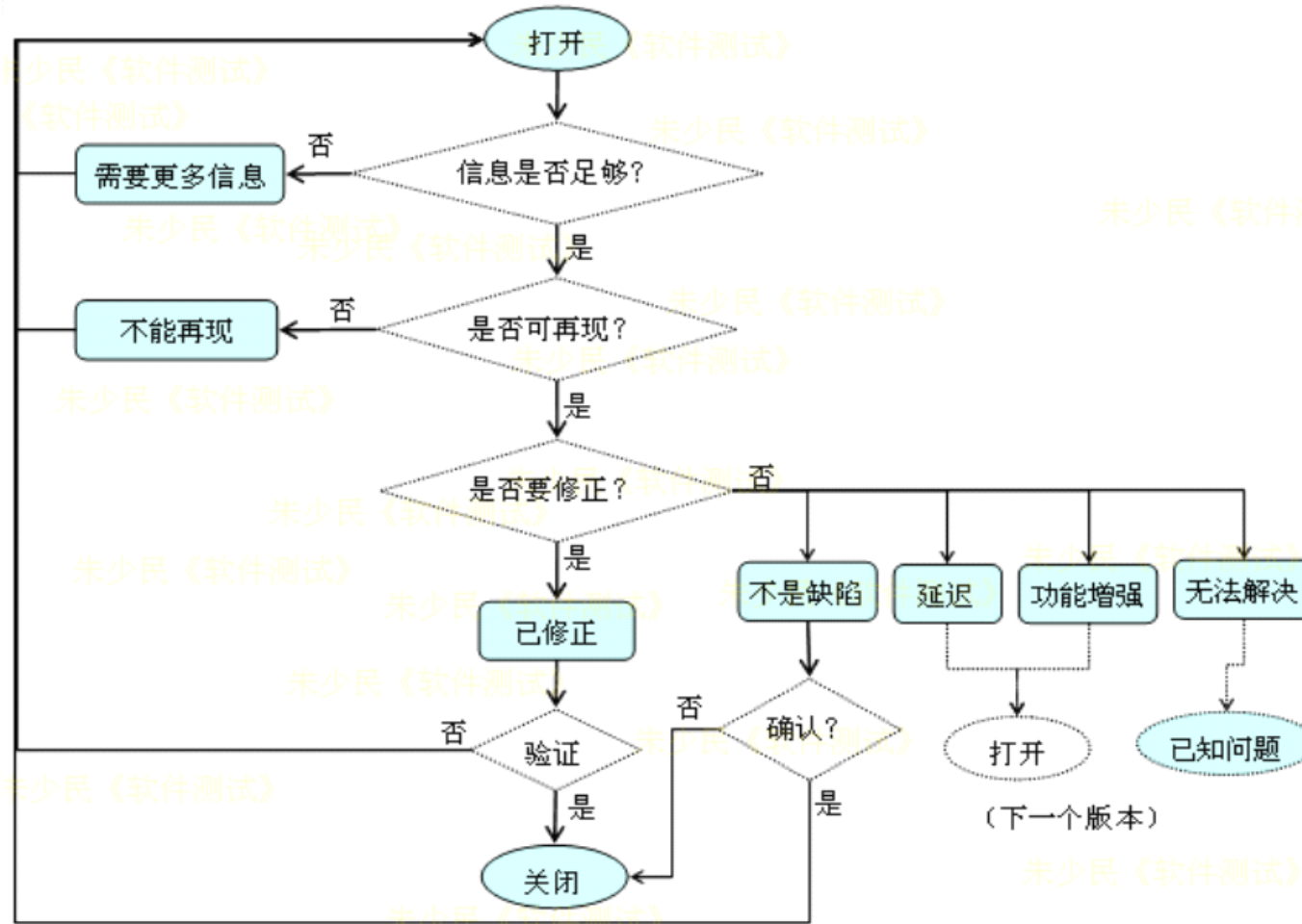
- ▶ 单一准确 每个报告只针对一个软件缺陷
- ▶ 可以再现 不要忽视或省略任何一项操作步骤，特别是关键性的操作一定要描述清楚，确保开发人员照所述的步骤可以再现缺陷
- ▶ 完整统一 提供完整的缺陷描述信息
- ▶ 短小精炼 如使用业务关键词
- ▶ 特定条件 必须注明缺陷发生的特定条件
- ▶ 不做评价 客观描述



•软件缺陷生命周期

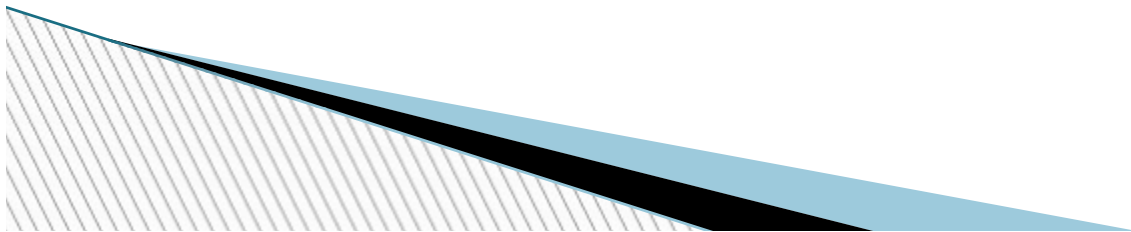


软件缺陷的生命周期



缺陷的跟踪处理

- ▶ 密切跟踪缺陷状态的变化，及时处理缺陷，使项目按预定的计划进行
- ▶ 动态报表，及时更新数据
- ▶ 自动邮件机制

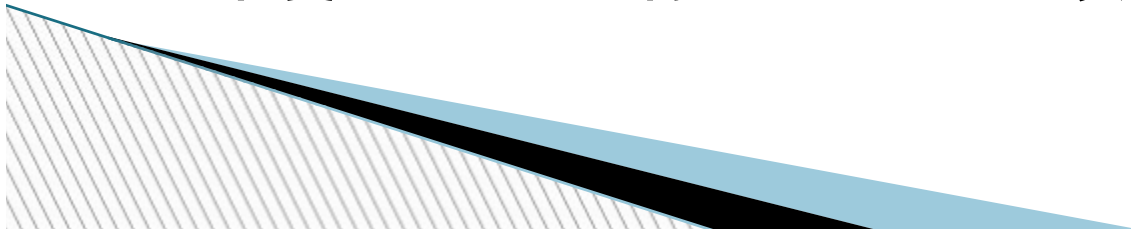


缺陷状态

缺陷状态	描述
打开/激活的 (Open /Active)	缺陷的起始状态，或重新被打开的状态。问题存在或依旧没有解决，等待修正，如新报的缺陷、补充完信息后再打开等。
已修正 (Fixed /Resolved)	已被开发人员检查、修复过的缺陷，通过单元测试，认为已解决但还有待测试人员验证。
关闭/非激活 (Close /Inactive)	测试人员验证后，确认缺陷不存在之后所处的状态。
延迟 (deferred)	这个缺陷不严重被推迟修正，可以在下一个版本中解决。
无法解决 (Onhold)	由于技术原因或第三者软件的缺陷，开发人员目前不能修复的缺陷。
功能增强 (Feature Enhancement)	该问题符合当前的设计规格说明书，但是一个有待改进的问题。
不是缺陷 (Not a Bug)	开发人员认为这不是问题，是测试人员误报的缺陷。
不能再现 (cannot duplicated)	开发不能复现这个软件缺陷，需要测试人员检查缺陷复现的步骤。
需要更多信息 (NeedMoreInfo)	开发能复现这个软件缺陷，但开发人员需要一些信息，例如：缺陷的日志文件，图片等。

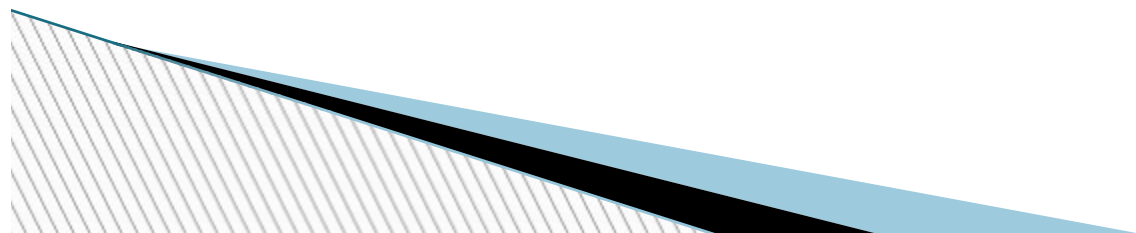
缺陷数据库所带来的益处

- ▶ 不仅可以统一数据格式、完成数据校验，而且确保每一个缺陷不会被忽视，使开发人员的注意力保持在那些必须尽快修复的高优先级的缺陷上
- ▶ 可以随时简历符合各种需求的查询条件，而且有利于建立各种动态的数据报表，用于项目状态报告和缺陷数据统计分析
- ▶ 可以随时得到最新的缺陷状况大家活得一致又准确的信息，掌握相同的实际情况，消除沟通上的障碍
- ▶ 可以将缺陷和测试用例、需求等关联起来，完成更深度的分析，有利于产品的质量改进等。



开源缺陷跟踪系统

- Mantis [•http://mantisbt.sourceforge.net/](http://mantisbt.sourceforge.net/)
- Bugzilla [•http://www.mozilla.org/project/bugzilla/](http://www.mozilla.org/project/bugzilla/)
- Bugzero [•http://bugzero.findmysoft.com/](http://bugzero.findmysoft.com/)
- Scarab [•http://scarab.tigris.org/](http://scarab.tigris.org/)
- TrackIT [•http://trackit.sourceforge.net/](http://trackit.sourceforge.net/)
- Itracker [•http://www.itracker.org/](http://www.itracker.org/)



---The end---

