

# 系统测试设计

## System Test Design

---

# 目录

## CONTENTS

- 测试设计要点
- **需求识别**
- 可测需求的特征
- 测试目标识别
- **建立测试设计流程模型**
- **建立测试结果模型**
- 测试设计准备度量
- 测试用例设计有效性

# 目录

## CONTENTS

- **测试设计要点**
- 需求识别
- 可测需求的特征
- 测试目标识别
- 建立测试设计流程模型
- 建立测试结果模型
- 测试设计准备度量
- 测试用例设计有效性

## 11.1 测试设计要点

系统测试设计的**核心工作**是识别系统的输入和预期的输出，判断系统是否处理了正确的**特征（一组相关联的需求）**

# 11.1 测试设计要点

系统测试设计的评价要素：

**(1)覆盖度量**

**(2)有效性**

**(3)生产效率**

**(4)正确性**

**(5)维护**

**(6)用户技能**



# 11.1 测试设计要点

## 覆盖度量

**覆盖指标**关注设计为满足某种标准的一个测试套件**检测DUT(被测设备)的程度**。覆盖指标允许我们**度量**系统测试的进展



# 11.1 测试设计要点

| 测试用例标识符  | 需求标识符    |          |          |          |
|----------|----------|----------|----------|----------|
|          | $N_1$    | $N_2$    | ...      | $N_p$    |
| $T_1$    | $A_{11}$ | $A_{12}$ | ...      | $A_{1p}$ |
| $T_2$    | $A_{21}$ | $A_{22}$ | ...      | $A_{2p}$ |
| $T_3$    | $A_{31}$ | $A_{32}$ | ...      | $A_{3p}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $T_q$    | $A_{q1}$ | $A_{q2}$ | ...      | $A_{qp}$ |

覆盖矩阵 $[A_{ij}]$ （一种系统测试覆盖度量标准）



# 11.1 测试设计要点

## 有效性和正确性



而为了开发**有效的**、能够发现缺陷的测试用例，**需求**的**正确性**起着非常关键的作用

系统测试设计的**重点**应该放在需求的识别和分析上，**从需求得到被测目标，基于被测目标创建测试用例**



# 11.1 测试设计要点

→ 在测试用例设计时：

要确保测试用例的**可靠性**：

要确保测试用例具有全面的**错误触发过程**和精确的**通过或失败标准**

# 11.1 测试设计要点

## 生产效率

必须尽可能地使用**结构化测试用例开发技术**来生产一个测试套件。**结构化测试用例开发技术**要将维护的成本降到最低，并且能够改善**工作效率**



# 11.1 测试设计要点

## 维护

必须在测试套件开发的前期就仔细设计测试用例，以保证测试套件的**可维护性**

# 11.1 测试设计要点

## 用户技能

必须考虑测试套件的潜在**用户**。测试套件必须很容易在另一个环境中部署、执行，并且这些过程需要适当的文档化



# 目录

## CONTENTS

- 测试设计要点
- **需求识别**
- 可测需求的特征
- 测试目标识别
- 建立测试设计流程模型
- 建立测试结果模型
- 测试设计准备度量
- 测试用例设计有效性

## 11.2 需求识别

**需求**是用户的需要和期望的描述，这些需要和期望是一个系统预期实现的

统计数据表明，**接近50%**的系统缺陷可以归因于**需求**的“**误解**”

## 11.2 需求识别

需求的  
定义过  
程的难  
题：

1、确保需求获取的正确性，其本质是满足用户的期望（需求定义的“**形式**”问题）

2、保证需求对开发人员和测试人员的**沟通**不存在模糊不清的情况（对产品质量存在不同理解），以便所交付的系统没有意外问题

## 11.2 需求识别的规范模型

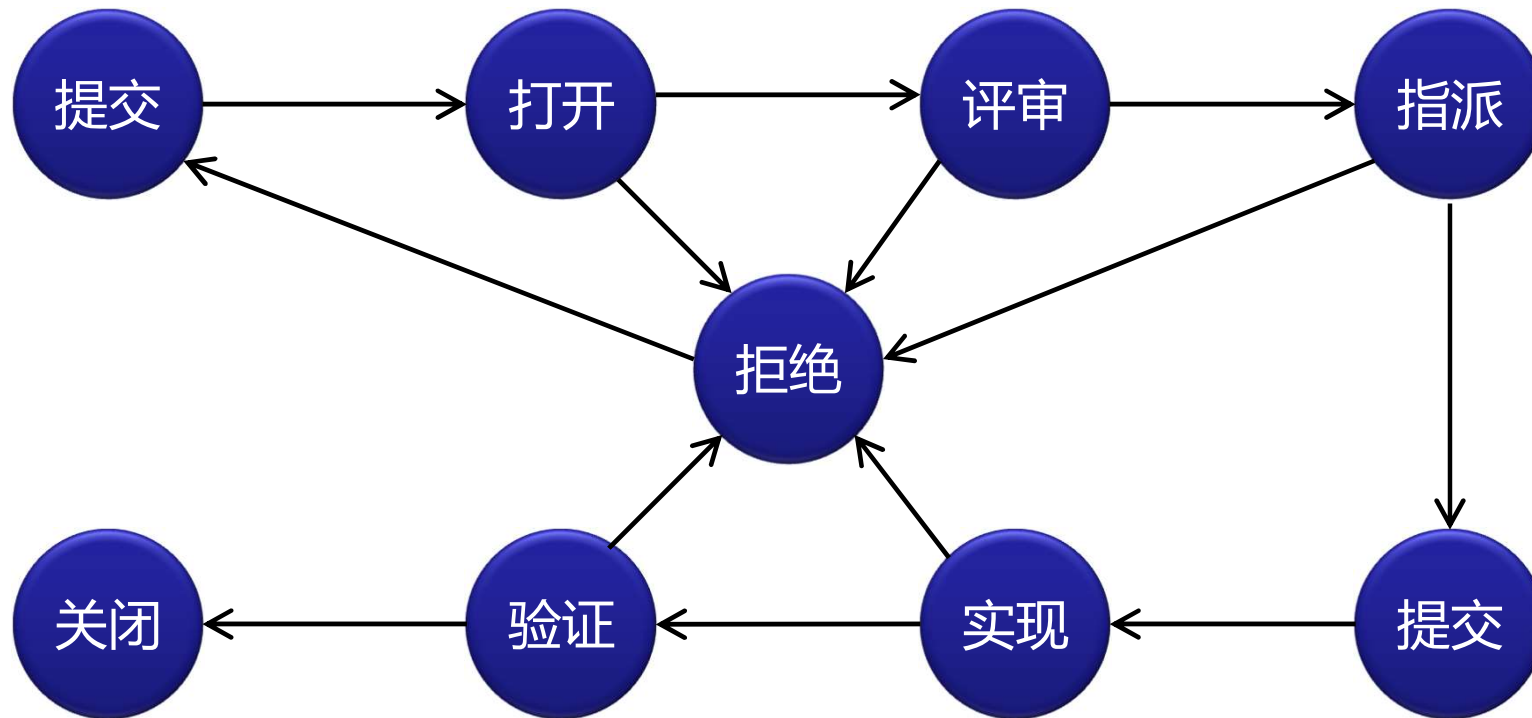
需求从开始到结束的全过程：

|           |                |
|-----------|----------------|
| <b>提交</b> | (submit)       |
| <b>打开</b> | (open)         |
| <b>评审</b> | (review)       |
| <b>指派</b> | (assign)       |
| <b>提交</b> | (commit)       |
| <b>实现</b> | (implement)    |
| <b>验证</b> | (verification) |
| <b>关闭</b> | (closed)       |
| <b>拒绝</b> | (decline)      |





## 11.2 需求识别



需求识别模型的状态转移图



## 11.2 需求识别

**需求模式字段汇总表** (Pg221 表11.2) : 在需求识别过程中每个步骤需要填写的信息字段 (可用于自动化过程)

## 11.2 需求识别

需求识别规范模型中的角色：**客户、营销经理、项目经理、软件工程主管、测试经理、测试工程师**

## 11.2 需求识别

---

### 提交状态 (submit)

---

将一个新需求置为**提交**状态，目的是使其他人都可以使用它。这个状态的**负责人是需求提交者**。一个新需求有不同的来源：**客户、营销经理和项目经理**

## 11.2 需求识别

**提交  
状态**  
需填  
写需  
求模  
式字  
段

**需求号：**一个唯一的与需求相关的标识符

**优先级：**需求的优先级别：高或普通（需求实现的顺序）

**标题：**需求的标题

**描述：**一个需求的简短描述

**记录：**关于该需求的一些记录（如果有）

**产品：**需求所要求的产品名称

**客户：**请求这个需求的客户名称



## 11.2 需求识别

### 打开状态 (open)

在这个状态，**营销经理**负责维护需求并协调相应活动。包括确认需求无重复，否则将重复的需求从**打开**状态变成**拒绝**状态；或者因为某种原因将需求从**打开**状态变成**拒绝**状态；或者指派需求给**软件工程主管**，将需求从**打开**状态转移到**评审**状态

## 11.2 需求识别

**打开  
状态**  
需填  
写需  
求模  
式字  
段

**优先级：**重新评价需求的优先级：高或者普通

**严重性：**给需求指派一个严重性：普通或关键

**拒绝记录：**如果拒绝了，给出一个需求的解释

**软件发布：**为需求建议一个更适合的软件版本

**营销理由：**为需求提供一个营销理由

**描述：**如果有任何不明确的，应描述需求

**记录：**如果有必要，做一些有用的注释



## 11.2 需求识别

### 评审状态 (review)

在**评审**状态，**软件工程主管**是需求的负责人，其通过评审需求来理解和评估软件实现所需的时间，并为需求准备一个初级功能规范版本。**软件工程主管**通过把所有者改为**营销经理**，将需求从**评审**状态转移到**指派**状态；如果需求不可能实现，则**拒绝**该需求



## 11.2 需求识别

**评审  
状态  
需填  
写需  
求模  
式字  
段**

**工程注释：**在评审过程中产生的注释在这个区域中记录。在功能规范开发、生成代码和执行单元测试时，这些注释是十分有用的

**实现时间：**这个字段含有为实现需求的评估时间，以人-周为单位

**附件：**一个分析文档，包括将来的功能规范开发中十分有用的图和描述

**功能规范标题：**将为这个需求编写的功能规范的名称

**功能规范名称：**功能规范的文件名

**功能规范版本：**功能规范的最新版本号

**工程指派：**由主管指派的评审需求的工程师名字



## 11.2 需求识别

### 指派状态 (assign)

在**指派**状态，**营销经理**是需求的负责人。其负责把需求指定到一个特定软件发布版本，把需求变为**提交**状态（转移给**项目经理**）；或可能拒绝需求和终止需求的开发过程，把需求转移到**拒绝**状态



## 11.2 需求识别

### 提交状态 (commit)

在**提交**状态，**项目经理**是需求的负责人

**项目经理**评审所有需求，通过与**营销经理**、**软件工程主管**和**客户**进行商议，可能重新将一个特定的需求指派到一个不同的软件发布版本。将一个特定需求提交到软件发布后，**项目经理**可以把需求转移到一个**实现**状态，同时**冻结**功能规范，退出**提交**状态



## 11.2 需求识别

### 实现状态 (implement)

在**实现**状态，**软件工程主管**是需求的负责人

此时表示软件工程部目前正在编码该需求并进行单元测试，在实现完成和软件发布给系统测试后，**软件工程主管**可能把这个需求从**实现**状态转移到**验证**状态；如果相关技术不能实现这个需求，**主管**也可拒绝该需求，并把这个需求转移到**拒绝**状态

## 11.2 需求识别

**实现状态**  
需填写需求模式字段

**拒绝记录：**如果需求转移到拒绝状态，给出一个拒绝需求的解释

**EC号：**一个EC（工程改变）文档号

**附件：**EC文档（表11.3：工程变更文档信息）



## 11.2 需求识别

### 验证状态 (verification)

在**验证状态**，**测试工程师**是需求的负责人。测试工程师验证需求。**测试经理**可能解释实现和需求不一致，则转移到**拒绝**状态，或验证后通过，则转移到**关闭**状态

## 11.2 需求识别

|                                       |   |
|---------------------------------------|---|
| 验证<br>状态<br>需填<br>写需<br>求模<br>式字<br>段 | <b>拒绝记录：</b> 拒绝这个需求的理由                                  |
|                                       | <b>EC（工程改变）号：</b> 一个EC文档号                               |
|                                       | <b>附件：</b> EC文档   |
|                                       | <b>验证方法：</b> 可以从集合{测试，分析，演示，审查}的4个值中取一个                 |
|                                       | <b>验证状态：</b> 可以从集合{通过，失败，不完全}的3个值中取一个，从而标识需求最终的状态       |
|                                       | <b>依从：</b> 可以从集合{全部依从，部分依从，完全不依从}的3个值中取一个，标识软件镜像依从需求的程度 |



## 11.2 需求识别规范模型

### 验证状态

需填写需求模式字段

**TC号：**覆盖需求测试用例的标识符

**TC结果：**可从{**未测试，通过，失败，阻塞，无效**}的5个值中取一个

**缺陷号：**一个缺陷的标识符

**测试节点：**可能含有测试工程师提供给最终用户的一个关于分析、审查和示范的解释





## 11.2 需求识别

### 关闭状态 (closed)

在完成了验证之后，**测试经理**把需求从**验证**状态转移到**关闭**状态



## 11.2 需求识别

### 拒绝状态 (decline)

在这个状态，**营销经理**是需求的负责人。一个需求到达这种状态是因为以下几个原因：

- 1、**营销经理**拒绝这个需求
- 2、在技术上不可能实现这个需求，而且可能有一个相关的EC号
- 3、**测试经理**用EC号拒绝实现



## 11.2 需求识别-需求数据库

在需求识别过程中，通过填写**需求模式  
字段汇总表**（Pg221 表11.2）的全部内容，  
逐步构成**需求数据库**

## 11.2 需求识别-追溯矩阵

客户可从**需求数据库**系统产生一个**追溯矩阵**，追溯矩阵可以允许我们发现需求 and 测试用例之间的2路双向映射：

- 从**需求**到一个**功能规范**到执行它们的特定的**测试用例**
- 从每个**测试用例**返回到**需求**和**功能规范**

## 11.2 需求识别-追溯矩阵

**追溯矩阵**的两种应用：

- I. 识别和追溯测试的**功能覆盖**
- II. 当一个**系统改进**时识别哪一个测试用例必须执行或更新

# 目录

## CONTENTS

- 测试设计要点
- 需求识别
- **可测需求的特征**
- 测试目标识别
- 建立测试设计流程模型
- 建立测试结果模型
- 测试设计准备度量
- 测试用例设计有效性

## 11.3 可测需求的特征

基于要被验证的**需求**来设计**系统测试**。测试工程师分析**需求**、相关的**功能规范**和**标准**来确定**需求的可测性**。可测性分析的方法通过评定需求的静态行为特征来揭示测试目标

## 11.3 可测需求的特征

确定需求描述是否可测的一种方法

- 1、获得下列需求描述：系统必须执行**X**
- 2、概括需求描述来创建测试目标：验证系统执行**X**的正确性
- 3、通过提问来评审测试目标：是否是可行的？换句话说，就是弄清假定系统和测试环境是可用的，是否可能执行**它**
- 4、如果对上述问题的回答是肯定的，也就是说需求的描述对测试目标是清楚和详细的。否则，需要做更多的工作来**修订和补充**需求描述





## 11.3 可测需求的特征

**例：**

**初始需求描述：**

软件的镜像必须很容易随着网络的发展而升级和降级

**重新确定的需求描述：**

软件镜像能够在100点的网络元素上进行升级和降级

## 11.3 可测需求的特征

除了需求的可测性之外，在评审时，系统测试工程师也必须分析其它因素：如**生命安全、信息安全、完全性、正确性、一致性、清楚性、相关性、可行性、可验证性、可跟踪性**等（Pg225）

还要对**功能规范**进行评估，标准参见表11.4

# 目录

## CONTENTS

- 测试设计要点
- 需求识别
- 可测需求的特征
- **测试目标识别**
- 建立测试设计流程模型
- 建立测试结果模型
- 测试设计准备度量
- 测试用例设计有效性

## 11.4 测试目标识别

**第一步：** 识别测试目标的第一步就是阅读、理解和分析功能规范

**例子：** 软件镜像必须很容易的在100点的网络元素上升级或降级 (Pg227)

## 11.4 测试目标识别

**第二步：**识别测试目标的第二步是通过质疑需求来获取这些需求中的**推断需求**（期望系统能够支持但是没有被明确定义的需求）

**例子：**系统必须按照一定的顺序来排序一个元素列表（Pg228）

## 11.4 测试目标识别

- 1.当给出一个已排序的列表作为输入时，需要验证这个系统产生的排序列表。
- 2.当给出可变长度的列表作为输入时，需要验证这个系统产生的排序列表。
- 3.验证输出项的数量等于输入项的数量。
- 4.验证已排序的输出记录的内容和输入记录的内容相同。



## 11.4 测试目标识别

- 5. 当一个空的列表作为输入时，验证系统会产生一个空的系统列表。
- 6. 通过给出包含一个或多个空记录的输入列表，检验系统的行为和输出列表。
- 7. 验证这个系统可以排序包含大量未分类整理的项的列表。



## 11.4 测试目标识别

确定测试目标后，需要形成**测试用例组**  
**或子组**，最终形成**测试套件**



## 11.4 测试目标识别

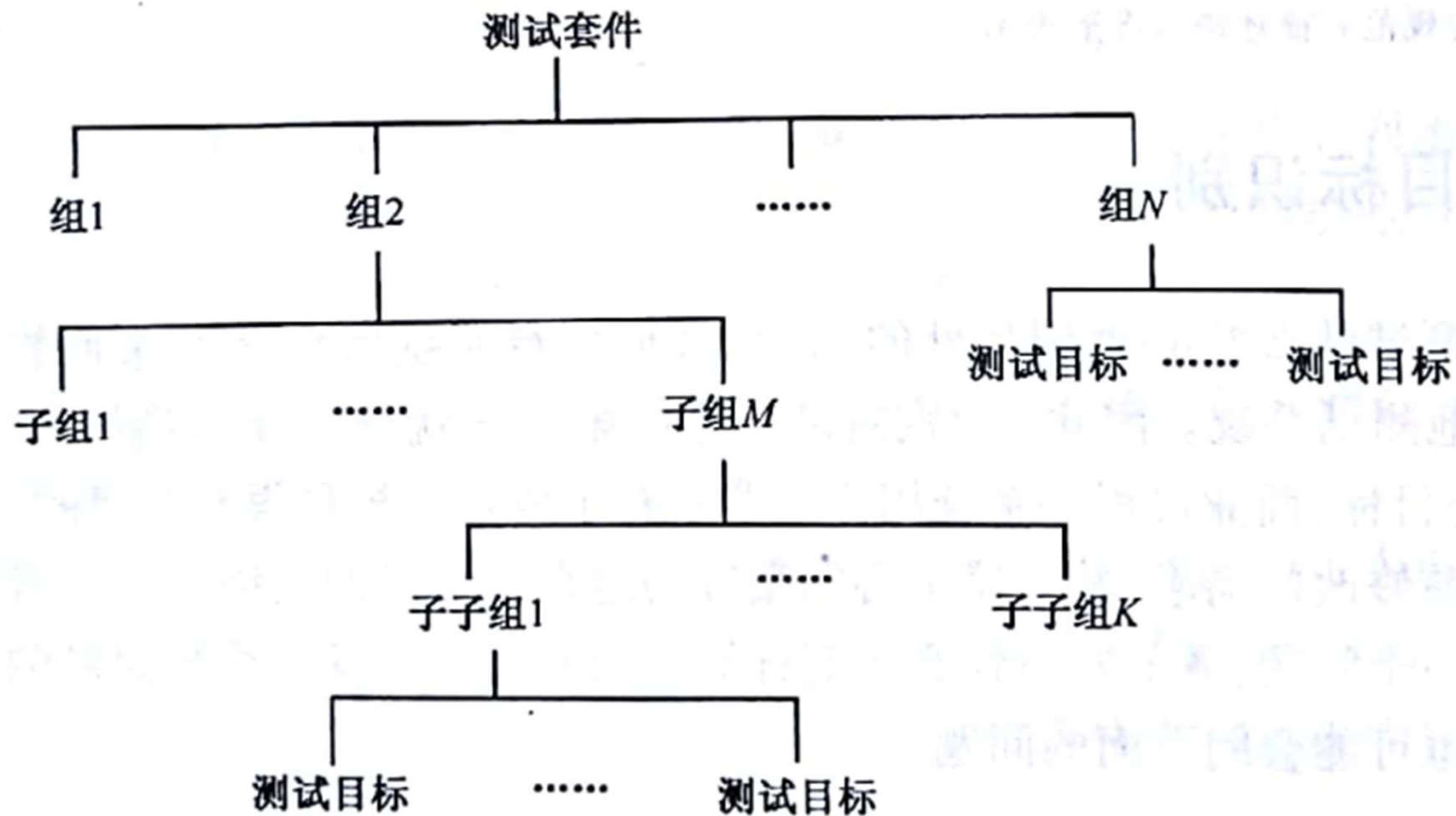


图 11.2 测试架构

# 目录

## CONTENTS

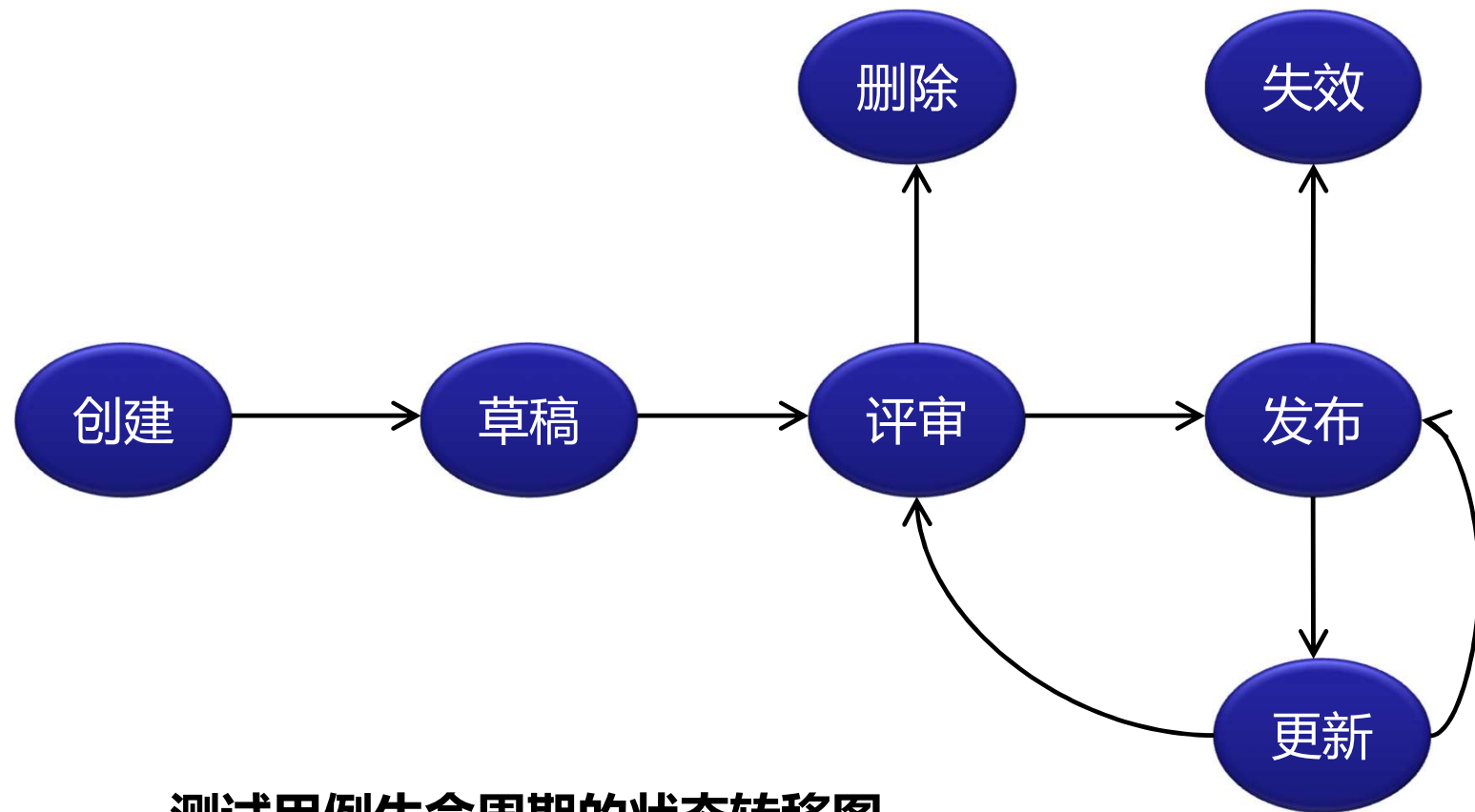
- 测试设计要点
- 需求识别
- 可测需求的特征
- 测试目标识别
- **建立测试设计流程模型**
- 建立测试结果模型
- 测试设计准备度量
- 测试用例设计有效性

## 11.6 建立测试设计流程模型

测试目标是从需求规范中定义的，而且为每一个测试目标创建一个**测试用例**。每一个测试用例都设计为模块化组件的组合，称为测试步骤

在一个**测试用例**的**生命周期**中，从开始到完成经过以下几个状态：**创建、草稿、评审、删除、发布、更新和失效**

## 11.6 建立测试设计流程模型



测试用例生命周期的状态转移图

## 11.6 建立测试设计流程模型

- **创建状态** 创建者将一个测试用例设置为最初状态，创建者也称所有者，由他初始化测试用例的设计
- **草稿状态** 这个状态的所有者是**测试组**，即系统测试组（管理准备）
- **评审和删除状态** 评审状态的所有者是**测试用例的创建者**。他邀请**测试工程师**和**开发人员**来评审和确认测试用例的有效性。若通过评审则进入**发布**状态，否则进入**删除**状态

## 11.6 建立测试设计流程模型

- **发布和更新状态**     **发布**状态的用例是可以执行的，而且它成为测试套件的一部分。**更新**状态意味着测试用例正在被修改（增强、调优或修复）。更新状态可直接进入**发布**状态，或进入**评审**状态
- **失效状态**     一个过时的测试用例可以转入**失效**状态。过时的原因有：1、测试用例未随系统更新进行维护；2、有新测试用例代替；3、测试用例无法重用；4、测试用例的最初设计目标已消失

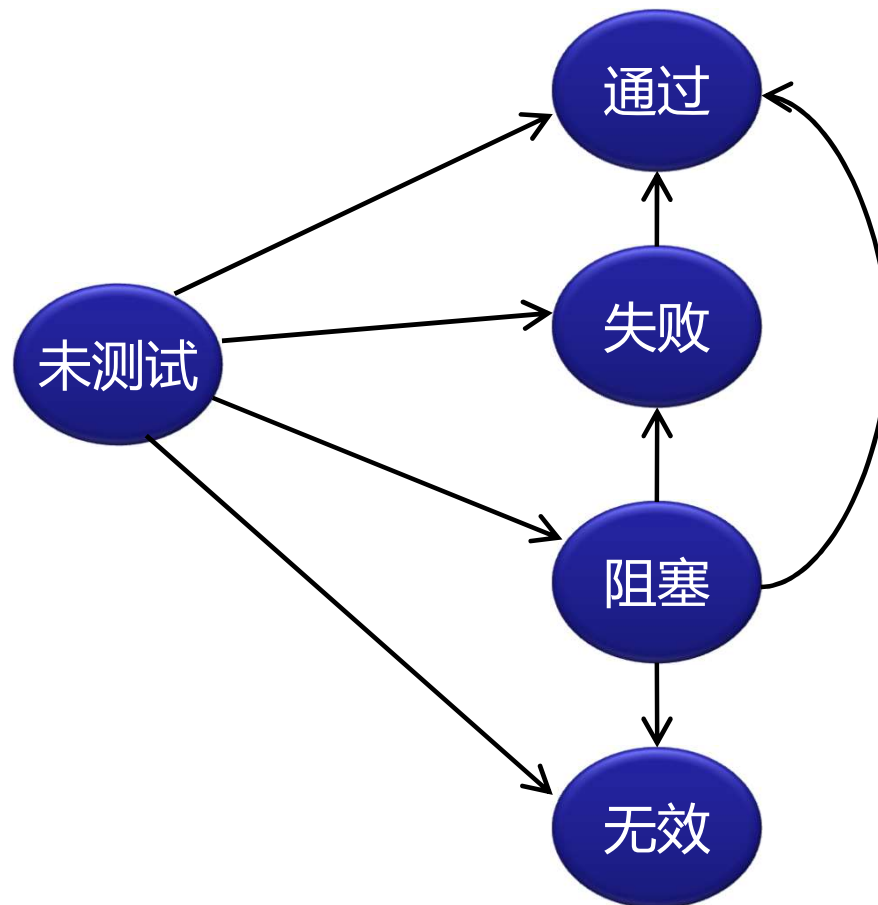


# 目录

## CONTENTS

- 测试设计要点
- 需求识别
- 可测需求的特征
- 测试目标识别
- 建立测试设计流程模型
- **建立测试结果模型**
- 测试设计准备度量
- 测试用例设计有效性

## 11.7 建立测试结果模型



测试用例结果状态转移图



## 11.7 建立测试结果模型

设计并选择一个测试用例之后，测试用例的执行状态被置为**未测试**的初始状态。如果这个测试用例对当前的软件发布版本是无效的，测试结果就会移动到**无效**的状态。当测试用例执行开始后，测试用例的状态可能会变为如下几种状态中的一种：**通过**、**失败**、**无效**或**阻塞（无法完全执行测试用例）**。如果测试用例执行完且测试结果满足通过条件，测试工程师就可以将测试用例结果从**未测试**状态转移到**通过**状态

# 目录

## CONTENTS

- 测试设计要点
- 需求识别
- 可测需求的特征
- 测试目标识别
- 建立测试设计流程模型
- 建立测试结果模型
- **测试设计准备度量**
- 测试用例设计有效性

## 11.8 测试设计准备度量

- **测试用例的准备情况 (PST) :** 在正式发布成有效的测试用例前, 它会经过一些阶段或状态, 如草稿和评审阶段。对测试用例处于不同测试设计的状态进行计量以监测测试设计的进展
- **测试用例设计花费的平均时间(ATS):** 计量一个测试用例从初始构想 (即创建状态) 到认为它是可用的 (即发布状态) 所需的时间
- **可用测试用例的数量 (NAT) :** 当前项目中处于发布状态的测试用例数量

## 11.8 测试设计准备度量

- **计划测试用例的数量 (NPT) :** 在系统测试开始时, 那些在测试套件中处于准备好执行的测试用例数量
- **测试套件的覆盖率 (CTS) :** CTS是需要选择的测试用例数量的度量, 或者需要设计为对系统需求有一个比较好的覆盖率的测试用例数量的度量

# 目录

## CONTENTS

- 测试设计要点
- 需求识别
- 可测需求的特征
- 测试目标识别
- 建立测试设计流程模型
- 建立测试结果模型
- 测试设计准备度量
- **测试用例设计有效性**

# 11.9 测试用例设计有效性

**测试用例设计有效性**度量的目标是 (i) 测量测试套件的“缺陷发现能力”和 (ii) 使用这种度量来优化测试设计的流程

## 11.9 测试用例设计有效性

**测试用例设计产出率 (TCDY) ;** 其中逃逸测试用例 (**TCE**) 为针对未在计划的测试过程中发现的缺陷设计的测试用例

$$\text{TCDY} = \frac{\text{NPT}}{\text{NPT} + \text{TCE}} \times 100\%$$

# 系统测试设计

**End**

---