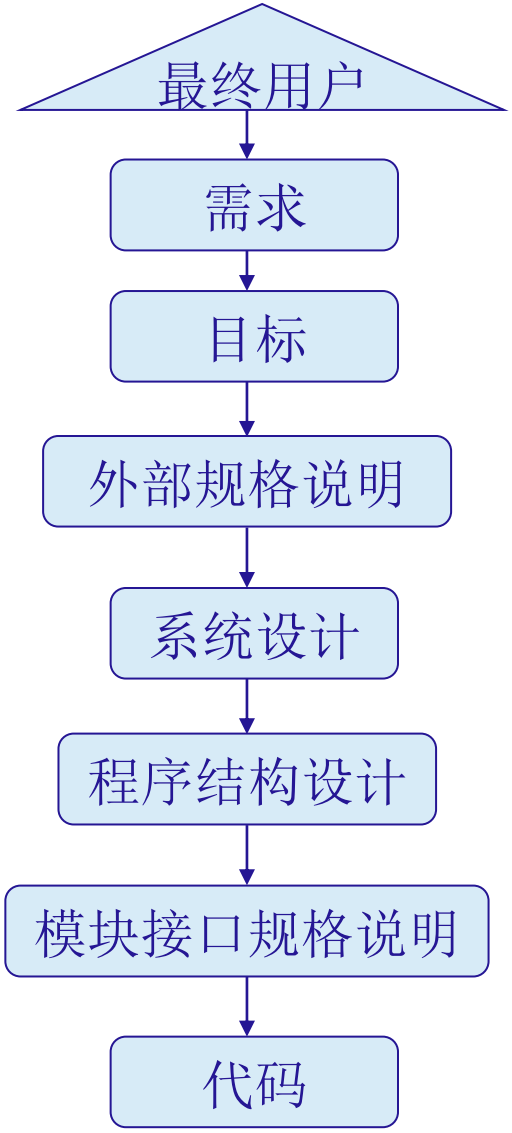
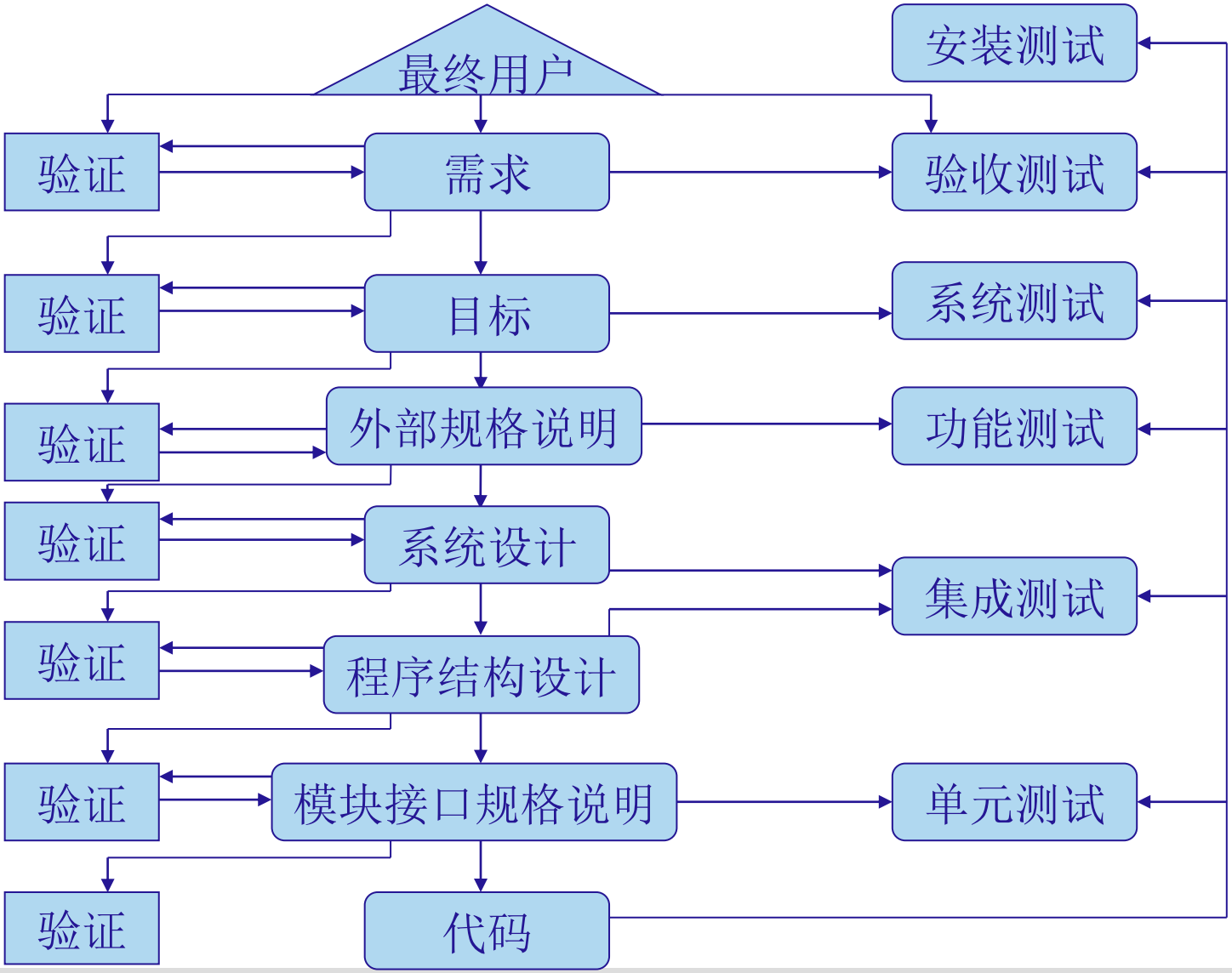
SOFTWARE TESTING

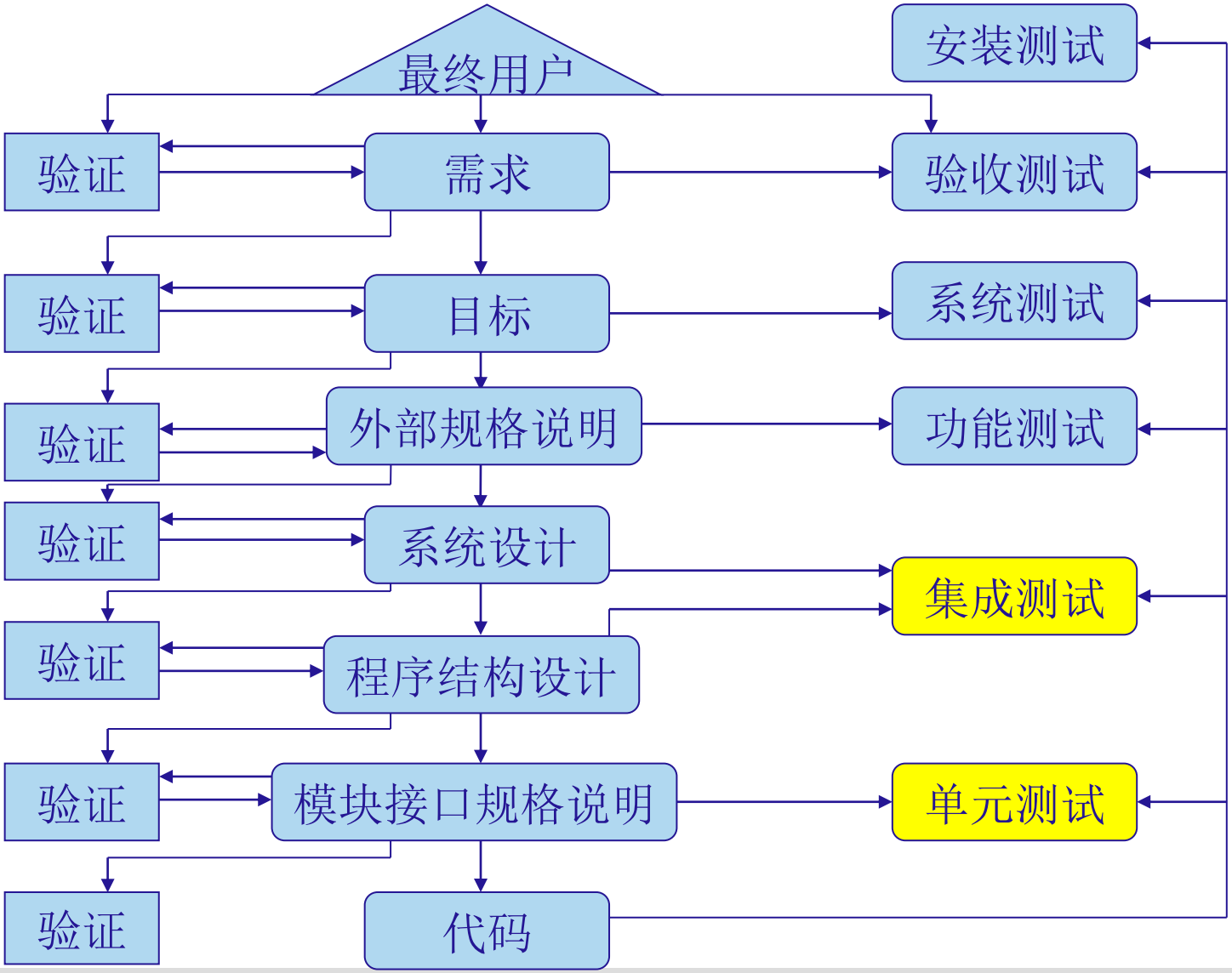
苏临之

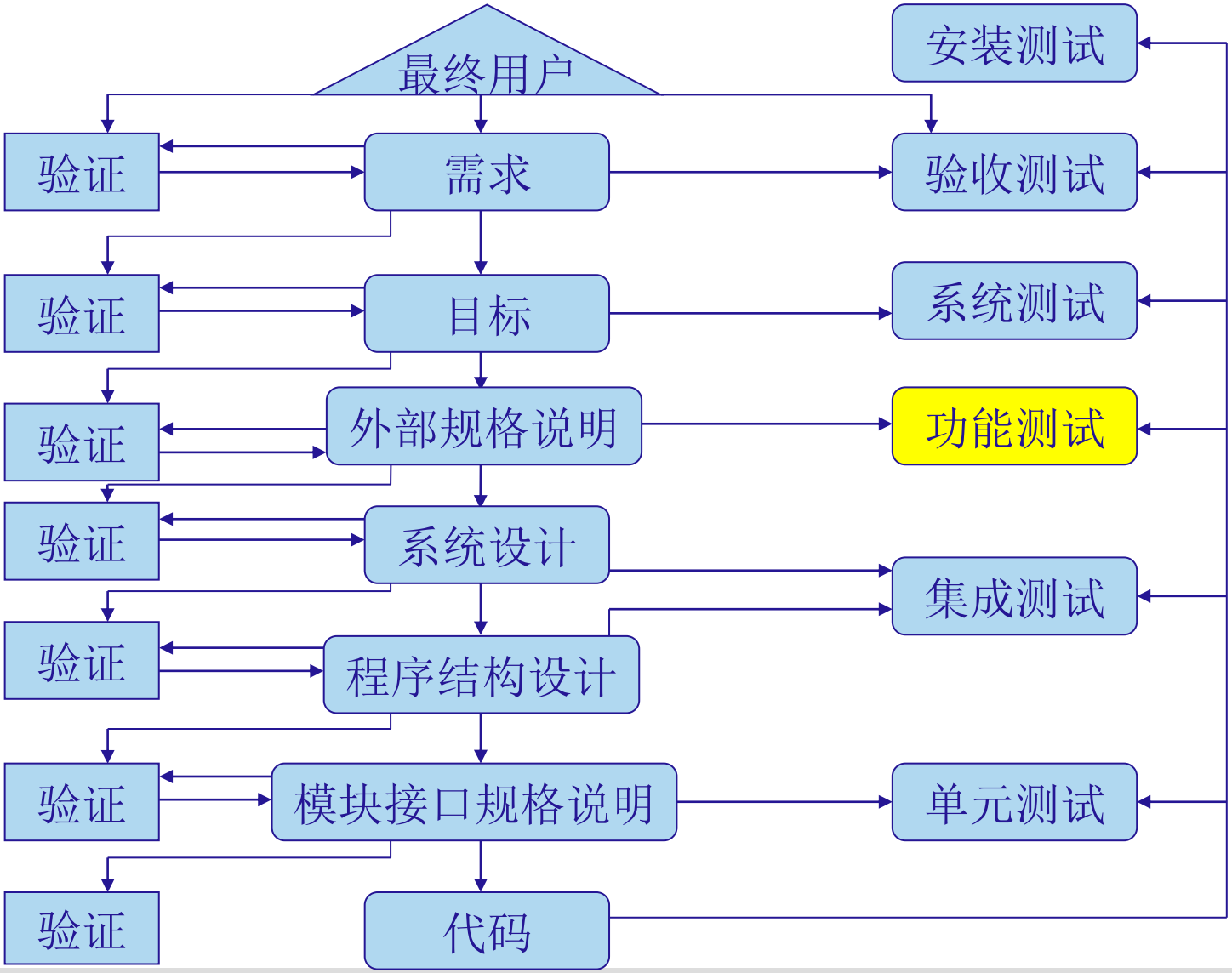
sulinzhi029@nwu.edu.cn

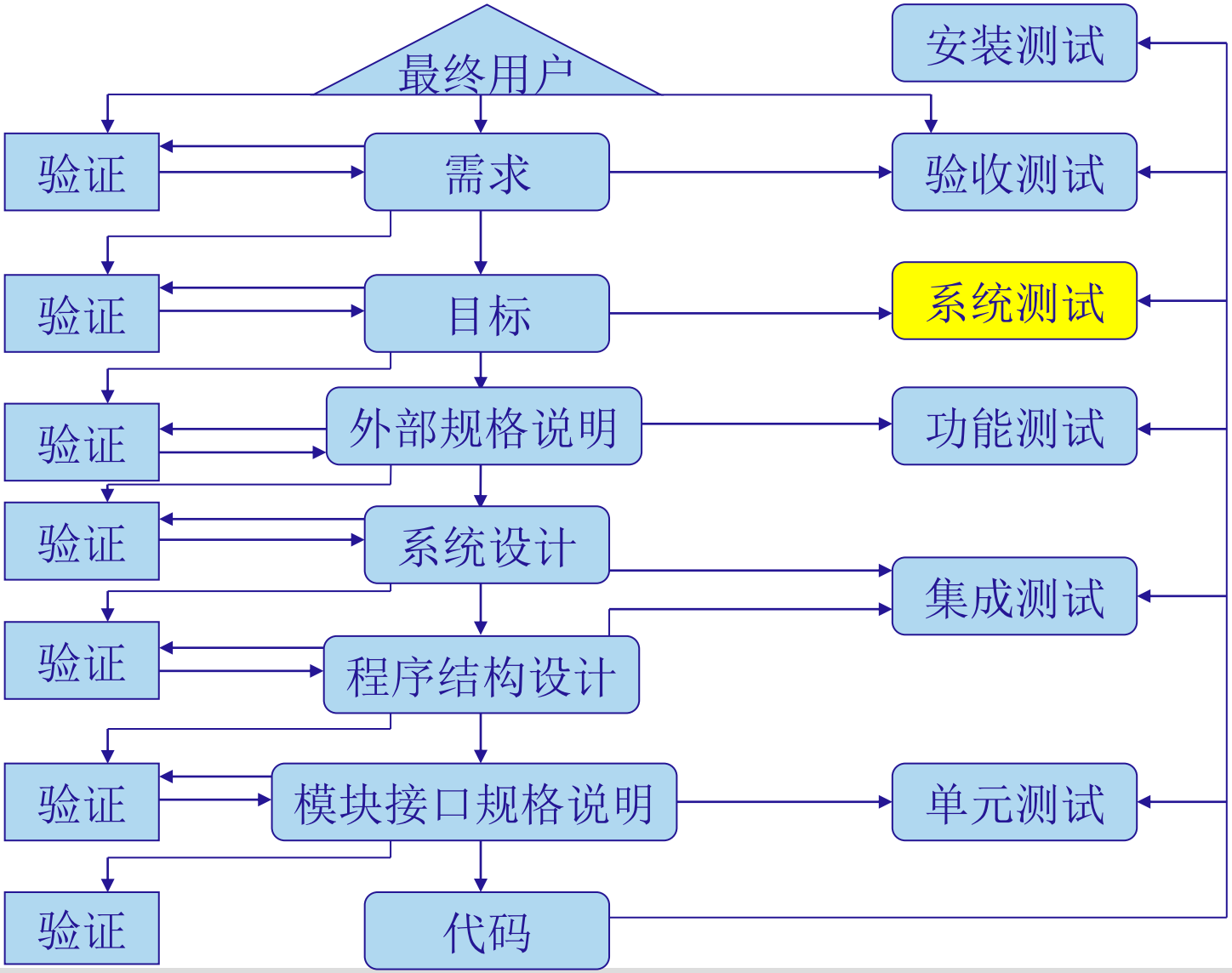
Development of Software

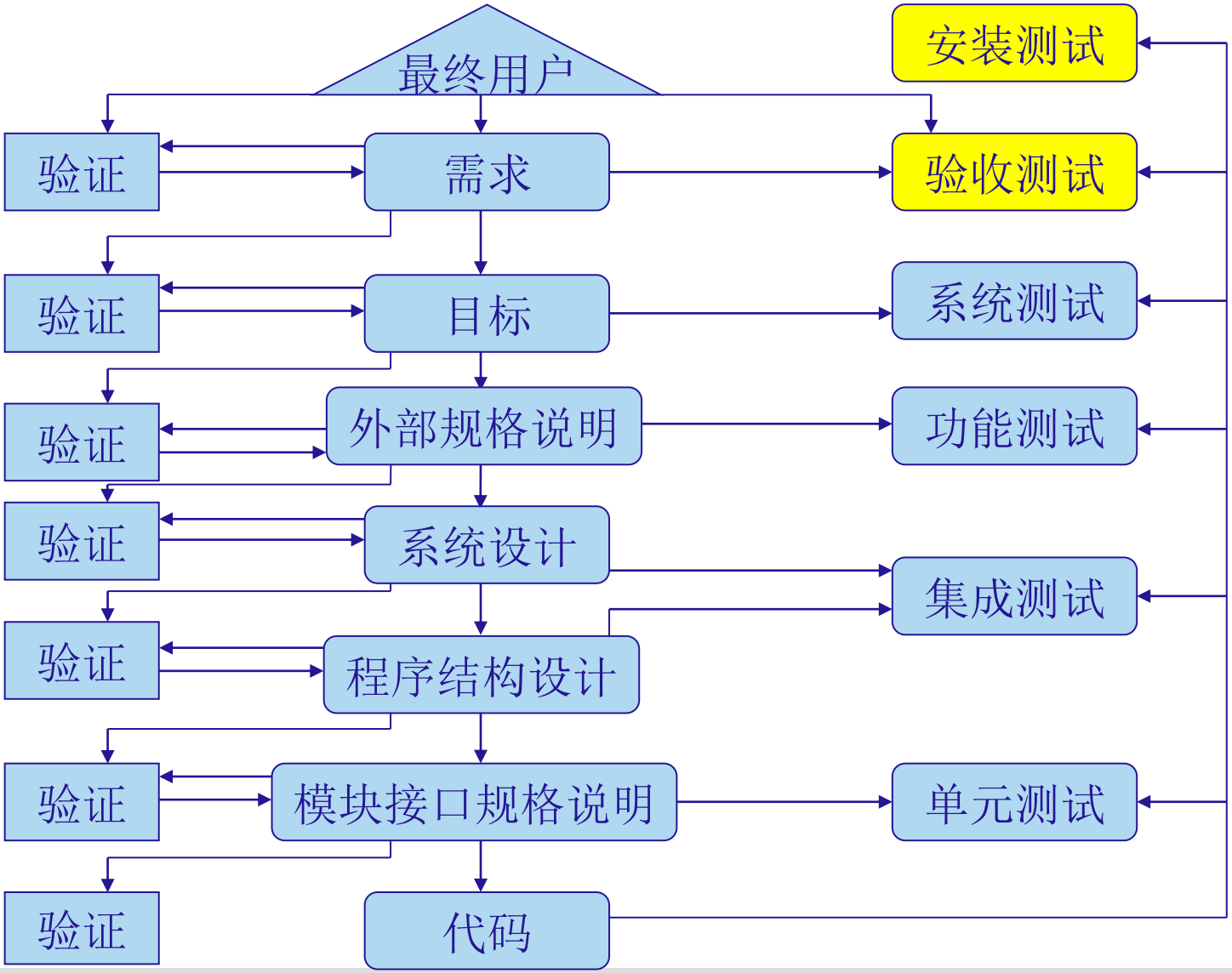



最终用户
安装测试
验证
需求
验收测试
验证
目标
系统测试
验证
外部规格说明
功能测试
验证
系统设计
集成测试
验证
程序结构设计
验证
模块接口规格说明
单元测试
验证
代码


最终用户
安装测试
验证
需求
验收测试
验证
目标
系统测试
验证
外部规格说明
功能测试
验证
系统设计
集成测试
验证
程序结构设计
验证
模块接口规格说明
单元测试
验证
代码


最终用户
安装测试
验证
需求
验收测试
验证
目标
系统测试
验证
外部规格说明
功能测试
验证
系统设计
集成测试
验证
程序结构设计
验证
模块接口规格说明
单元测试
验证
代码


最终用户
安装测试
验证
需求
验收测试
验证
目标
系统测试
验证
外部规格说明
功能测试
验证
系统设计
集成测试
验证
程序结构设计
验证
模块接口规格说明
单元测试
验证
代码


最终用户
安装测试
验证
需求
验收测试
验证
目标
系统测试
验证
外部规格说明
功能测试
验证
系统设计
集成测试
验证
程序结构设计
验证
模块接口规格说明
单元测试
验证
代码

System Testing

- System testing, whose aim is to compare the entire system with the initial goals, is a process to find out whether the system is not capable of meeting the goals. So without these goals, one can never executing the system testing.
- It is worth noting that the system testing is not to test the function of the system because it is not based on the specification. Actually the system testing can be viewed as a non-functional testing.
- It is crucial to determine who will be in charge of system testing because the testers here must know customers' attitude and the circumstance for using. Those who develop the software can not act as the system tester.

Common Issues in System Testing

- Ability Testing
- Load Testing
- Stress Testing
- Performance Testing
- Memory Testing

Load Testing & Stress Testing

- The load testing makes the system experience large amount of data, aiming to check whether the software can cope with the load as demonstrated. It will occupy much source. For example, the popular artificial neural networks need load testing when they are run in a computer.
- The stress testing aims to test whether the system can handle high stress, which means that in a short time the data or operations reach their maximum. For example, the actual power source supply will consider the maximum power transfer, so the stress testing is indispensable.

Specific Issues in System Testing

- Configuration Testing
- Compatibility Testing
- Foreign Language Testing
- Usability Testing
- Documentation Testing
- Safety Testing

Specific Issues in System Testing

- **Configuration Testing**
- Compatibility Testing
- Foreign Language Testing
- Usability Testing
- Documentation Testing
- Safety Testing

Configuration Testing

- A process for various hardware to run a software.
- 配置测试是必不可少的，原因是硬件的生产厂商并没有执行严格的标准，有时候仅执行松散的规范，这样导致软件使用某种硬件配置无法正常工作。
- 如何判断缺陷是由配置问题产生而不仅仅是一个普通缺陷？最可靠的方法是：在另外一台由完全不同硬件配置的机器上执行导致问题的相同操作。
- 配置测试的工作量可能是巨大的，因此可以使用等价类划分的方法来减少工作量。

Specific Issues in System Testing

- Configuration Testing
- **Compatibility Testing**
- Foreign Language Testing
- Usability Testing
- Documentation Testing
- Safety Testing

Compatibility Testing

- Forward Compatibility: Being able to use the versions in the **future**.
- Backward Compatibility: Being able to use the versions in the **past**.
- 绝大多数软件都需要进行某种程度的兼容性测试，但并不是所有的软件都要进行。
- 兼容性是一种产品特性，可以有不同程度的符合标准。软件测试员通过确定兼容性检查的工作量大小，为兼容性测试的确立提供相应的依据。

Specific Issues in System Testing

- Configuration Testing
- Compatibility Testing
- **Foreign Language Testing**
- Usability Testing
- Documentation Testing
- Safety Testing

Various Languages

- The existing languages are various in many aspects.
  - Pronunciation system
  - Vocabulary system
  - Grammar system
  - Users and their regions
  - ……

Localization & Internationalization

- 逐字直译单词是容易的，但要想使整个操作提示意思明确、实用，就需要投入更多的时间和精力。好的翻译要是外文翻译得读起来和原文一样。
- 软件适应特定地域特征，照顾到语言、方言、地区习俗和文化的过程称为本地化（Localization）或国际化（Internationalization）

Translation Issues

- 文本扩展
- ASCII等编码问题
- 热键和快捷键
- 扩展字符
- 字符计算
- 阅读方向
- 图形中的文字
- 文本和代码脱离

Specific Issues in System Testing

- Configuration Testing
- Compatibility Testing
- Foreign Language Testing
- **Usability Testing**
- Documentation Testing
- Safety Testing

Usability Testing

- The software is developed for using, but sometimes the developers were busy developing and thus too much emphasis has been put on coding, whereas the users are overlooked.
- The lack of usability will lead to one of the software defects: the software is difficult to understand, hard to use, slow, or in the tester's eyes will be viewed not good by the end user.

Necessities for Usability Testing

- 并非每一个软件开发小组都会很科学地设计UI。由于技术的问题或者是时间限制。
- 可能是由于软件没有正确本地化。
- 软件使用者的构成越来越复杂，对软件的易用性要求越来越高、越来越不宽容，即要求越来越苛刻。

User Interface

- 用户界面（UI）用于与软件交互的方式，用来与软件交互时提供输入和接受输出。所有的软件都有UI。
  - 早期计算机有触发开关和发光二极管。
  - 上世纪60、70年代，纸带、穿孔卡和打字机是最流行的用户界面
  - MS－DOS使用视频监视器和简单的行编辑器作为UI
  - 现在我们使用的个人计算机都有复杂的图形用户界面（GUI）

What Makes a Good UI?

- There are seven basic element to design a good UI.
  - Following standards and guidelines (符合要求规范)
  - Intuitive (直观)
  - Consistent (一致)
  - Flexible (灵活)
  - Comfortable (舒适)
  - Correct (正确)
  - Useful (实用)

Following Standards and Guidelines

- 最重要的用户界面要素是软件符合现行的标准和规范。如软件要在Windows等现有的平台上运行，标准是已经确立的。比如说它们描述了软件应该具有什么样的外观和感觉，从何时使用复选框而不是单选按钮，到何时使用提示信息、警告信息或是出错信息等。
- 标准和规范：由软件易用性专家开发。经过大量正规测试而设计出的方便用户的规则。
- 标准和规范非常重要，重要到要把它们当作是软件产品说明书的一部分进行测试。

Intuitive

- 衡量软件UI直观程度的标准
  - 用户界面是否洁净、不唐突、不拥挤？UI不应该为用户使用造成障碍。所需功能或者期待的响应应该明显、并出现在预期出现的地方。
  - UI的组织和布局合理吗？是否允许用户轻松地从一个功能转到另一个功能？下一步做什么明显吗？任何时刻都可以决定放弃、退回、退出吗？输入得到确认了吗？菜单或者窗口是否太深了？
  - 有多余功能吗？是否有太多特性把工作复杂化了？是否感到信息太庞杂？
  - 如果其他所有努力失败，帮助系统真能帮忙吗？

Consistent

- 被测试软件本身以及与其他软件的一致是一个关键属性。用户希望对一个程序的操作方式能够带到另一个程序中。
- 不一致会使用户从一个程序转向另一个程序时感受到挫折。当然同一个程序不一致更糟糕。
- 审查软件一致性要注意：
  - 快速键和菜单选项，如F1总是用于得到帮助信息。
  - 术语和名称：软件使用同样的术语和命名方式吗？如Find是否一直叫Find，而不是叫Search。
  - 听众：软件是否一直面向统一级别的听众？
  - 诸如OK和Cancel按钮的位置。

Flexible

- 用户为数不多的选择足以允许他们选择想要做的和怎样做。例如Windows计算器程序一般有三种视图：标准型、科学型和程序员型。
- 灵活性带来了复杂性，意味着需要经历更加严密和复杂的测试。
- 灵活性对于测试的主要影响是状态和数据：
  - 状态跳转
  - 状态的终止和跳过
  - 数据输入和输出

Comfortable

- 软件用起来舒服，而不应该为用户工作制造障碍和困难。软件的舒适是相当讲究感觉的。
- 鉴别软件舒适性好坏的方法：
  - 恰当：软件外观和感觉应该与所作的工作和使用者相符。不要太夸张也不要太朴素。
  - 错误处理：程序应该在执行关键操作之前提出警告，并且允许用户恢复由于错误操作而丢失的数据（撤销和重复）。
  - 性能：有些操作不要太快，如软件卡死时的对话框提示。

Correct

- 舒适性被公认为是模糊的，而正确性则是明确的。
- 测试正确性就是测试UI是否做了该做的事。
- 常见正确性问题：
  - UI与产品说明书不符
  - 市场定位偏差：是否与宣传材料不符？
  - 语言和拼写
  - 不良媒体，即UI中包含的所有图标、图像、声音和视频
  - 所见即所得（WYSIWYG）：保证UI显示的就是实际得到的。

Useful

- 是否实用是优秀UI的最后一个要素。
- 审查UI是否对软件具有实际的价值。它们有助于用户执行软件设计的功能吗？如果认为它们没必要，就要研究一下找出他们存在于软件的原因（有两种可能）。多余的特性，不论是在什么软件中（重要与否）对用户都是不利的，同时还意味着需要更多的测试工作。

Testing for the Disabled

- 以下几种残疾对使用计算机和软件会造成极大的困难：视力损伤、听力损伤、运动损伤、认知和语言障碍。
- Windows、Linux这样的都在一定程度上支持辅助选项，如Windows提供了以下能力：
  - 粘滞键
  - 筛选键
  - 切换键
  - 声音卫士
  - 声音显示
  - 高对比度
  - 鼠标键
  - 串行键

Specific Issues in System Testing

- Configuration Testing
- Compatibility Testing
- Foreign Language Testing
- Usability Testing
- **Documentation Testing**
- Safety Testing

Documentation Testing

- The documentation here means the text that is not or has something little to do with the code.
- If your software's documentation consisted of nothing but a simple readme file, testing it wouldn't be a big deal. But, the days of documentation consisting of just a readme file are gone.
- Today, software documentation can make up a huge portion of the overall product. Sometimes, it can seem as if the product is nothing but documentation with a little bit of software thrown in.

Types of Documentation

- 包装文字和图形
- 市场宣传材料、广告以及其他插页
- 授权/注册登记表
- 最终用户许可协议（EULA）
- 标签和不干胶条
- 安装和设置指导
- 用户手册
- 联机帮助
- 指南、向导和计算机基础训练（CBT）
- 样例、示例和模板
- 错误提示信息

Significance

- 软件用户把文档这样的非软件部分当作整个软件的一部分，而在使用时不会区分这些东西是由程序员、作家还是艺术家创作的，他们关心的是整个产品的质量。
- 比如说，安装指导有误，或者不正确的错误提示把用户引入歧途，他们就会认为存在软件缺陷。
- 好的软件文档以下述3种方式确保产品质量：
  - 提高易用性
  - 提高可靠性
  - 降低支持费用

What to Look for?

- Testing the documentation can occur on two different levels. If it's non-code, such as a printed user's manual or the packaging, testing is a static process. If the documentation and code are more closely tied, such as with a hyperlinked online manual or with a helpful paper clip guy, it becomes a dynamic test effort. In this situation, you really are testing software.
- Whether or not the documentation is code-linked, a very effective approach to testing it is to treat it just like a user would.
- Finally, if the documentation is software driven, test it as you would the rest of the software.

Realities

- 文档开发和测试有别于软件开发和测试的原因：
  - 文档常常得不到足够的重视、预算和援助
  - 编写文档的人可能对软件做什么不甚了解。
  - 印刷文档制作要花不少时间，可能是几周，也可能是几个月。

Specific Issues in System Testing

- Configuration Testing
- Compatibility Testing
- Foreign Language Testing
- Usability Testing
- Documentation Testing
- **Safety Testing**

Safety Testing

- 机器的安全问题似乎每天都在出现。黑客、病毒、蠕虫、间谍软件、流氓软件、木马都成为了常见术语。
- 遭到攻击的后果就是丢失数据，浪费时间。
- 软件的安全性，确切的说软件缺少安全性，已经成为巨大的问题。
- 安全性漏洞是一种软件缺陷，要求软件测试员要像黑客一样预先攻击被测软件，这对软件测试员要求很高。

Concepts for Safety

- 安全产品：指产品在系统的所有者或管理员的控制下，保护用户信息的保密性、完整性、可获得性，以及处理资源的完整性和可获得性。
- 安全漏洞：指产品不可行的缺陷—即使是正确的使用产品时——来防止攻击者窃取系统的用户权限、调节操作、破坏数据、或建立未授权的信任。
- 黑客：精通编程和使用的人或电脑玩家。使用变成技能来获得对计算机网络或文件的非法访问的人。真正“从良”的黑客可以用于测试安全性漏洞。

Motivations for Hackers

- 出于好奇
- 挑战/成名
- 恶意破坏，如报复
- 偷窃
- 心理、精神上的疾病
- ……

Threat Modeling Analysis

- 执行威胁模型并不是软件测试软的责任，而是整个项目小组执行的正式过程
- 步骤：构建威胁模型分析小组 → 确认价值 → 创建一个体系结构总体图 → 分解应用程序 → 确认威胁 → 记录威胁 → 威胁等级评定
- 等级评定的要点：1、潜在的损害；2、可反复性；3、可利用性；4、受影响的用户；5、可发现性。

System Testing

- Ability Testing
- Load Testing
- Stress Testing
- Performance Testing
- Memory Testing
- Configuration Testing
- Compatibility Testing
- Foreign Language Testing
- Usability Testing
- Documentation Testing
- Safety Testing

Test of Independence

- Test of independence corresponds to an independent group of people who participate the testing process. They do not develop the software.
- 优势：1) 测试人员从中立的角度看待每个缺陷；2)测试人员完全没有偏见，即不带主观人至偏颇；3)测试人员对质量没有任何假设。
- 劣势：1) 与开发团队的隔离有时会导致过时的文档或版本引用；2) 独立的测试执行通常是最后一个阶段，在此过程中任何延迟都会影响受到版本或产品的发布；3) 开发人员可能对质量不负责任，因为他们可能认为独立测试团队的测试系统中有问题；4) 独立测试一定会受到通信阻碍。

Manual & Automatic Testing

人工测试	自动测试
消耗时间并单调：由于测试用例是由人力资源执行，所以非常缓慢并乏味。	时间消耗少：快速自动化运行测试用例时明显比人力资快。
人力资源上投资巨大：由于测试用例需要人工执行，所以在人工测试上需要更多的试验员。	人力资源投资较少：测试用例由自动工具执行，所以在自动测试中需要较少的试验员。

Manual & Automatic Testing

人工测试	自动测试
可信度较低：人工测试可信度较低是可能由于人工错误导致测试运行时不够精确。	可信度更高：自动化测试每次运行时精确地执行相同的操作。
非程式化：编写复杂并可以获取隐藏的信息的测试的话，这样的程序无法编写。	程式化：试验员可以编写复杂的测试来显示隐藏信息。

THANK YOU!