中山大學
SUN YAT-SEN UNIVERSITY

Chapter 3
Software Static Testing

Software Testing: approaches & Technologies

School of Data & Computer Science, Sun Yat-sen University

Outline

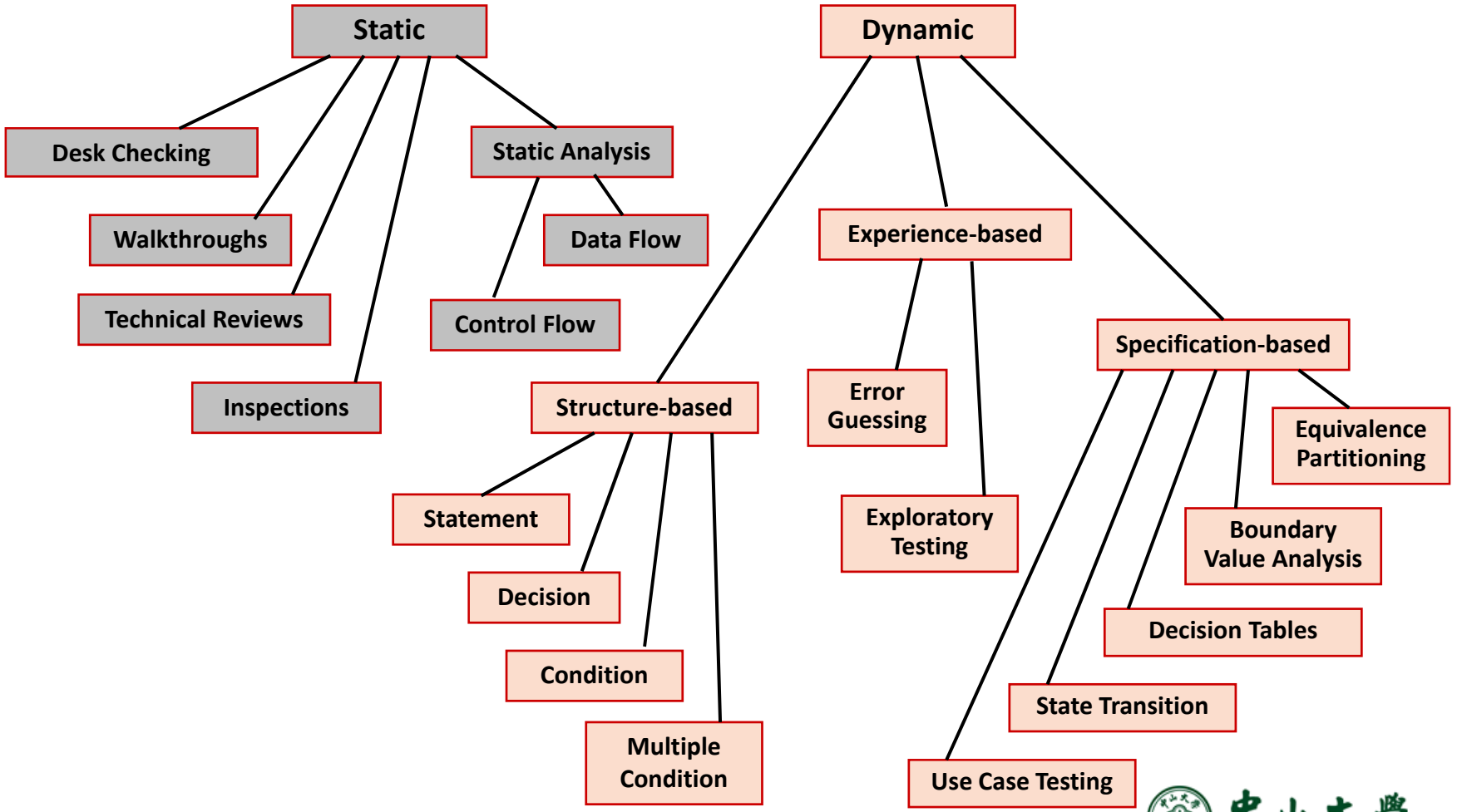
- 3.1 软件静态测试概述
- 3.2 阶段评审
  - 同行评审
  - 非正式审查和正式审查
  - 需求规格说明书的测试
- 3.3 代码检查
- 3.4 软件复杂性分析
- 3.5 软件质量控制
- 3.6 软件静态分析工具

3.1 软件静态测试概述

- **Test Design Techniques**
  - Definition
    - By design we mean to create a plan for how to implement an idea, and technique is a method or way for performing a task.
    - **Test Design** is creating a set of inputs for given software that will provide a set of expected outputs. The idea is to ensure that the system is working good enough and it can be released with as few problems as possible for the average user.
  - Two main categories
    - Broadly speaking there are two main categories of **Test Design Techniques**. They are:
      - Static Techniques
      - Dynamic Techniques

3.1 软件静态测试概述

- **Test Design Techniques**
  - Tree structure of the testing techniques



3.1 软件静态测试概述

- **Static Testing**
  - What is Static Testing
    - Static testing is the testing of the software work products manually, or with a set of tools, but the products are **not executed**.
    - Static testing starts early in the Life Cycle and so it is done during the verification process.
    - Static testing **does not need computer**
      - as the testing of program is done without executing the program
      - For example: reviewing, walk through, inspection, etc.
    - Most static testing techniques can be used to 'test' any form of document including source code, design documents and models, functional specifications and requirement specifications.

3.1 软件静态测试概述

- **Static Testing**
  - What is Static Testing for
    - ✧ Static testing techniques provide a great way to improve the quality and productivity of software development. It includes the reviews and provides the overview of how they are conducted. The primary objective of static testing is to improve the quality of software products by assisting engineers to recognize and fix their own defects early in the software development process.
    - ✧ Static tests contribute to the increased awareness of quality issues.

3.1 软件静态测试概述

- **Static Testing**
  - The Uses of Static Testing
    - ✧ Since static testing can start early in the life cycle so early feedback on quality issues can be established.
    - ✧ As the defects are getting detected at an early stage so the rework cost most often relatively low.
    - ✧ Development productivity is likely to increase because of the less rework effort.
  - Types of the defects that are easier to find during the static testing
    - ✧ deviation from standards
    - ✧ missing requirements
    - ✧ design defects
    - ✧ non-maintainable code
    - ✧ inconsistent interface specifications

3.1 软件静态测试概述

- **软件静态测试的概念**
  - 软件静态测试通常指不执行被测程序代码，而在软件文档或程序代码中查找错误或进行评估的过程。
    - 测试对象是各种与软件相关的产物，如文档、源代码等。
    - 通过扫描程序正文对程序的数据流和控制流等进行分析。
      - 找出软件缺陷，生成测试报告。
      - 不必进行测试用例的设计和结果分析等工作 。
    - 静态测试可以手工/人工进行，也可以借助软件工具自动进行。
      - 人工进行静态测试，能够充分发挥人的思维优势。
      - 静态测试对软件错误的发现和定位同时进行。
      - 静态测试不需要特别的运行环境支持，因此容易展开。
    - 静态测试对测试人员要求较高。
      - 测试人员起码应该具有编程经验。

3.1 软件静态测试概述

- **软件静态测试的必要性**
  - 软件静态测试对软件代码标准以及质量进行监控，以此提高代码的可靠性，使系统的设计符合模块化、结构化、面向对象的要求。
    - ✧ 一个软件产品即使实现了需求，但如果它的内部结构复杂、混乱，代码的编写不遵循一定的规范，这时软件中往往会隐藏一些不易被察觉的错误。这些错误在特定的条件下会造成重大的影响。
    - ✧ 一个软件产品即使满足了用户目前的需求，也有可能日后的维护升级工作相当困难。

3.1 软件静态测试概述

- **软件静态测试的必要性**
  - 软件静态测试的作用
    - ✧ 使系统的设计符合模块化、结构化、面向对象的要求
    - ✧ 使开发人员编写的代码符合规定的编码规范
    - ✧ 通过对代码标准及质量的监控提高代码的可靠性
    - ✧ 通过对源代码的检查尽可能早地发现缺陷
    - ✧ 通过组织代码审核，定位容易产生错误的模块
    - ✧ 为日后的维护工作节约大量的人力、物力
    - ✧ 提供有效的质量保证手段

3.1 软件静态测试概述

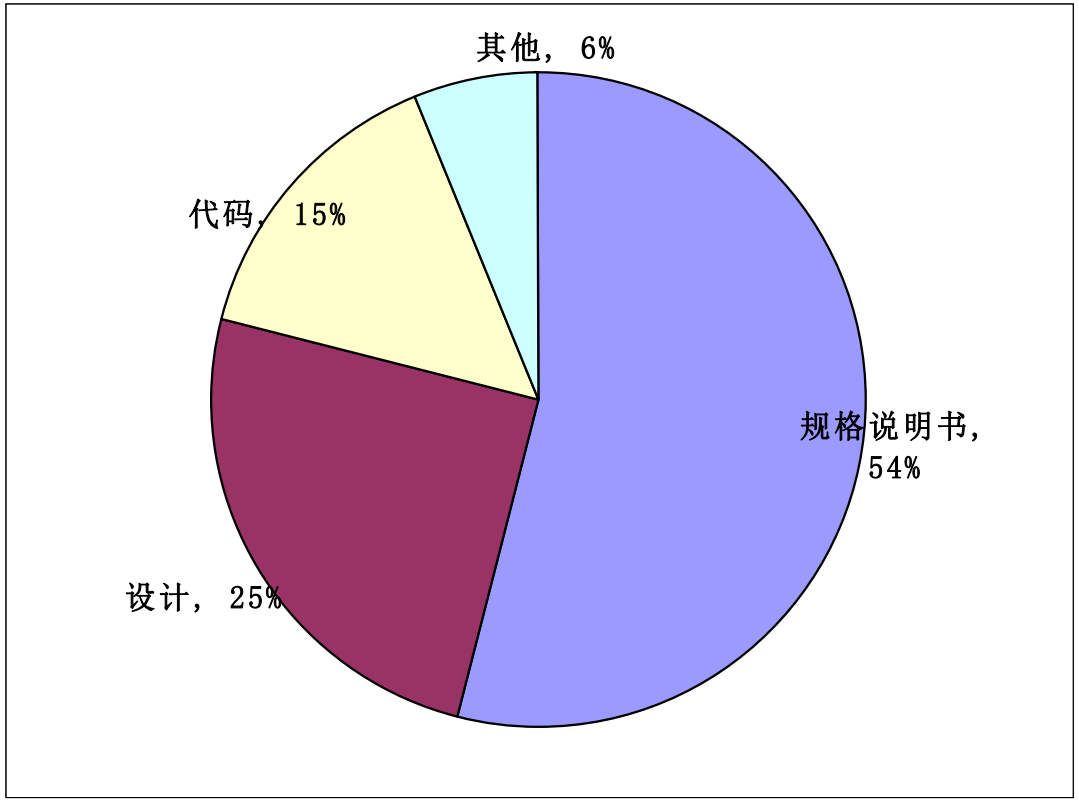
- **软件静态测试内容**
  - 软件静态测试主要包括各阶段的评审、代码检查、程序分析、软件质量度量等。
    - ✧ 阶段评审通常由人来执行。
    - ✧ 代码检查、程序分析和软件质量度量等既可由人工完成，也可使用工具完成。工具通常具有更好的作用和效果。
    - ✧ 经验表明，通过包括代码检查、桌面检查、走查、小组评审和技术审查在内的静态分析能够有效地发现 30%-70% 的逻辑设计和编码错误，而且这种方法一次能够解释一批错误，同时还能对错误进行定位。

3.2 软件的阶段评审

- **评审的概念**
  - 评审是对软件元素或项目状态进行评估的活动，以确定当前结果与预期结果的偏差，给出相应的改进意见。
    - ✧ 通常人工执行，包括非正式的 Review 和正式的 Inspection
    - ✧ 需求阶段的规格说明书是评审的重要内容
    - ✧ 评审的形式一般包括
      - ◌ 培训评审、预备评审、同行评审等
  - 除了期望在项目早期发现缺陷和降低项目失败风险外，项目需要进行评审的其它原因包括：
    - ✧ 分享知识
    - ✧ 培训团队成员
    - ✧ 为管理层决策提供依据
    - ✧ 为过程改进提供信息
    - ✧ 评估项目所处状态

3.2 软件的阶段评审

- **评审的概念**
  - 软件缺陷的构成



3.2 软件的阶段评审

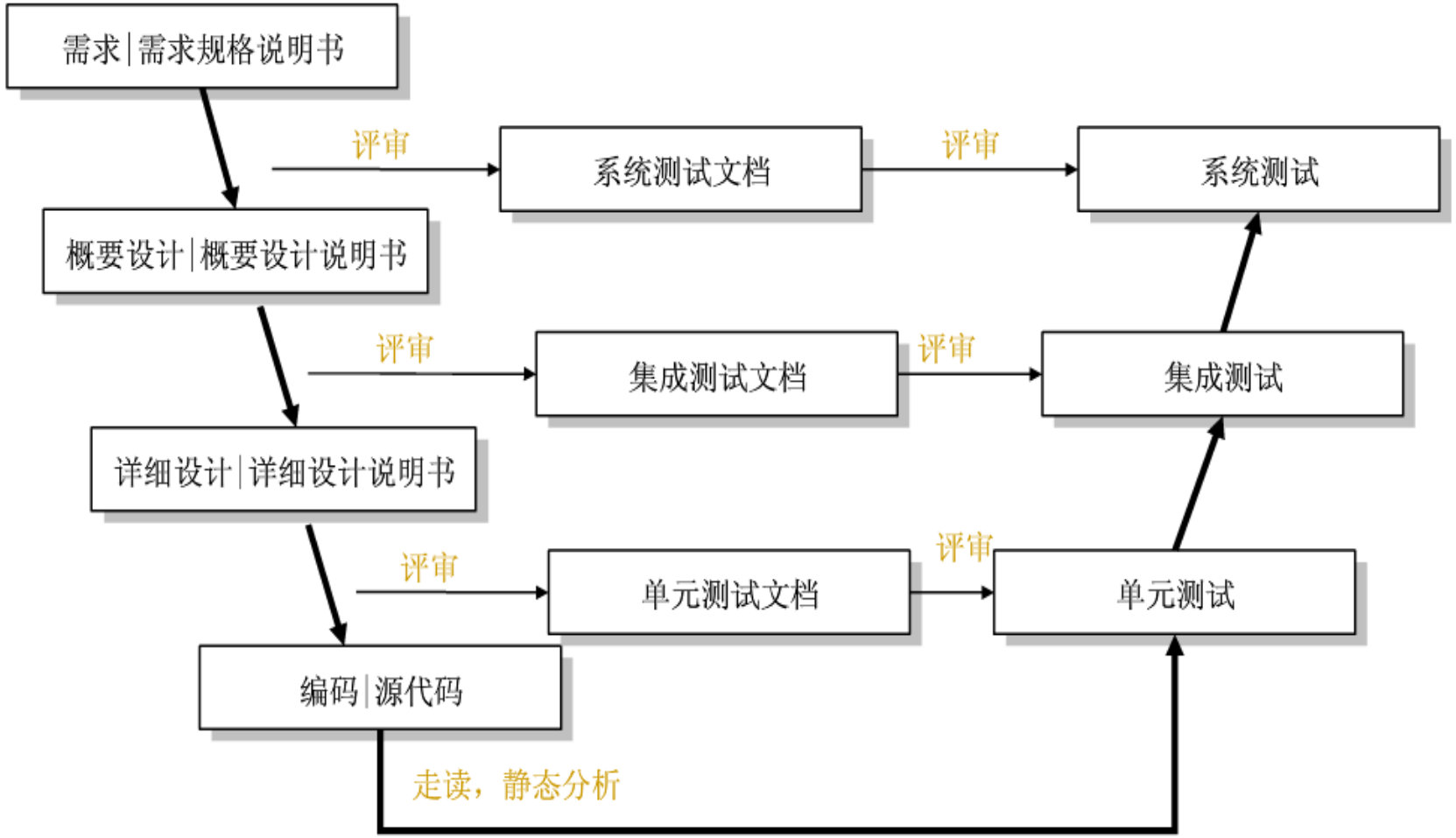
- 同行评审
  - 概述
    - ✧ 同行评审由软件产品开发者以外的其他人员检查产品，以期发现缺陷并寻找改进的机会。
      - ◌ 评审时间点
        - 一般设在工作产品到达了一个完成的里程碑节点，即将进入下一个开发阶段时。
      - ◌ 评审方法
        - 评审参与者逐行仔细阅读被评审的程序对象，尽量发现被测对象中的缺陷。

3.2 软件的阶段评审

- 同行评审
  - 概述
    - ✧ 同行评审形式一般包括技术审查、小组评审、走查、桌面评审、临时评审五种类型，它们的区别在于正式程度：
      - ◌ 技术审查最正式，然后是小组评审、走查、桌面评审 (桌查)，临时评审最随意。
      - ◌ 对于重要的或者高风险的评审对象，应该采用更为正式的评审方式。
      - ◌ 同行评审越正式，可能发现的缺陷就越多，但花费的成本也将升高。

3.2 软件的阶段评审

- 同行评审



3.2 软件的阶段评审

- 同行评审
  - 技术审查 Inspection
    - ✧ 技术审查是一种参加人员包括非软件作者等专家在内的针对特定对象进行检查，以期发现缺陷的过程。
    - ✧ 审查对象
      - ◌ 需求规格书、设计文档和源代码等
    - ✧ 审查流程
      - ◌ 计划、事前会议、准备、会议、返工、跟踪、因果分析
    - ✧ 每个阶段需要确定的内容
      - ◌ 审查小组成员：作者、评审组长、审查专家、读者、记录员
      - ◌ 相应的输入、输出

3.2 软件的阶段评审

- 同行评审
  - 技术审查 Inspection (续)
    - 技术审查是一种有结构、有规则的评审方法。审查规则包括：
      - 审查小组一般由3-7人构成。
      - 软件作者不能担当评审组长、读者或记录员等角色，要保持开放的思想，接受别人的意见，避免争论。
      - 评审组长不能同时担任记录员。
  - 小组评审 Technical Review
    - 小组评审是一种 “轻量级审查”，可参考采用技术审查的指导方针和流程。
      - 小组评审没有技术审查那么正式和严格，会议期间读者的角色可以由评审组长代替。
      - 小组评审方法发现的缺陷数量可能是技术审查的 2/3。

3.2 软件的阶段评审

- 同行评审
  - 走查 Walkthrough
    - 走查由软件产品的作者向一组同事进行产品说明，希望获得他们的意见。
      - 走查是一种非正式的评审，其过程由作者主持，没有标准的流程可循。
      - 走查发现的缺陷数量可能只是技术审查的 ½。
  - 同级桌面审查 Desk Checking
    - 同级桌面审查 (同级桌查) 是一对一评审。
      - 除作者以外只有一位评审专家对其工作产品进行检查。

3.2 软件的阶段评审

- 同行评审
  - 临时评审
    - ✧ 临时邀请团队的同事，在短时间内解决一些问题。
  - 软件评审指导书
    - ✧ 用于将评审过程和规则以文字形式固定下来，其内容包括:
      - ◌ 目的、范围、评审角色及职责、过程准则、目标、进入标准、活动、退出标准、度量、相关资料、过程监控。

3.2 软件的阶段评审

- **Informal reviews**
  - Informal reviews are applied many times during the early stages of the life cycle of the document.
    - ✧ A two person team can conduct an informal review.
    - ✧ In later stages these reviews often involve more people and a meeting.
    - ✧ The goal is to improve the quality of the document.
    - ✧ The most important thing to keep in mind about the informal reviews is that they are **not documented.**

3.2 软件的阶段评审

- **Formal reviews**
  - Formal reviews follow a formal process. It is well structured and regulated.
  - A formal review process consists of six main steps:
    - ✧ Planning
    - ✧ Kick-off
    - ✧ Preparation
    - ✧ Review meeting
    - ✧ Rework
    - ✧ Follow-up

3.2 软件的阶段评审

- **Formal reviews**
  - 1. Planning
    - ✧ The process begins with a request for review by the author to the moderator (or inspection leader). A moderator has to take care of the scheduling like date, time, place and invitation of the review. For the formal reviews the moderator performs the entry check and also defines the formal exit criteria. The entry check is done to ensure that the reviewer's time is not wasted on a document that is not ready for review. After doing the entry check if the document is found to have very little defects then it's ready to go for the reviews. So, the entry criteria are to check that whether the document is ready to enter the formal review process or not.

3.2 软件的阶段评审

- **Formal reviews**
  - 1. Planning
    - The entry criteria for any document to go for the reviews are:
      - The documents should not reveal a large number of major defects.
      - The documents to be reviewed should be with line numbers.
      - The documents should be cleaned up by running any automated checks that apply.
      - The author should feel confident about the quality of the document so that he can join the review team with that document.

3.2 软件的阶段评审

- **Formal reviews**
  - 1. Planning
    - ✧ Once the document clears the entry check, the moderator and author decide that which part of the document is to be reviewed. Since the human mind can understand only a limited set of pages at one time so in a review the maximum size is between 10 and 20 pages. Hence checking the documents improves the moderator ability to lead the meeting because it ensures the better understanding.

3.2 软件的阶段评审

- **Formal reviews**
  - 2. Kick-off
    - ✧ This kick-off meeting (启动会议) is an optional step in a review procedure. The goal of this step is to give a short introduction on the objectives of the review and the documents to everyone in the meeting. The relationships between the document under review and the other documents are also explained, especially if the numbers of related documents are high.

3.2 软件的阶段评审

- **Formal reviews**
  - 3. Preparation
    - ✧ In this step the reviewers review the document individually using the related documents, procedures, rules and checklists provided. Each participant while reviewing individually identifies the defects, questions and comments according to their understanding of the document and role. After that all issues are recorded using a logging form. The success factor for a thorough preparation is the number of pages checked per hour. This is called the checking rate. Usually the checking rate is in the range of 5 to 10 pages per hour.

3.2 软件的阶段评审

- **Formal reviews**
  - 4. Review meeting
    - The review meeting consists of three phases:
    - **Logging phase:** In this phase the issues and the defects that have been identified during the preparation step are logged page by page. The logging is basically done by the author or by a scribe. Scribe is a separate person to do the logging and is especially useful for the formal review types such as an inspection. Every defects and it's severity should be logged in any of the three severity classes given below:
      - Critical: The defects will cause downstream damage.
      - Major: The defects could cause a downstream damage.
      - Minor: The defects are highly unlikely to cause the downstream damage.

3.2 软件的阶段评审

- **Formal reviews**
  - 4. Review meeting
    - ✧ During the logging phase the moderator focuses on logging as many defects as possible within a certain time frame and tries to keep a good logging rate (number of defects logged per minute). In formal review meeting the good logging rate should be between one and two defects logged per minute.
    - ✧ **Discussion phase:** If any issue needs discussion then the item is logged and then handled in the discussion phase. As chairman of the discussion meeting, the moderator takes care of the people issues and prevents discussion from getting too personal and calls for a break to cool down the heated discussion. The outcome of the discussions is documented for the future reference.

3.2 软件的阶段评审

- **Formal reviews**
  - 4. Review meeting
    - ✧ **Decision phase:** At the end of the meeting a decision on the document under review has to be made by the participants, sometimes based on formal exit criteria. Exit criteria are the average number of critical and/or major defects found per page (for example no more than three critical/major defects per page). If the number of defects found per page is more than a certain level then the document must be reviewed again, after it has been reworked.

3.2 软件的阶段评审

- **Formal reviews**
  - 5. Rework
    - ✧ In this step if the number of defects found per page exceeds the certain level then the document has to be reworked. Not every defect that is found leads to rework. It is the author's responsibility to judge whether the defect has to be fixed. If nothing can be done about an issue then at least it should be indicated that the author has considered the issue.

3.2 软件的阶段评审

- **Formal reviews**
  - 6. Follow-up
    - ✧ In this step the moderator check to make sure that the author has taken action on all known defects. If it is decided that all participants will check the updated documents then the moderator takes care of the distribution and collects the feedback. It is the responsibility of the moderator to ensure that the information is correct and stored for future analysis.

3.2 软件的阶段评审

- **需求规格说明书的评审**
  - 软件需求规格说明书评审也称需求规格说明书测试。
    - ✧ 需求规格说明书评审一般采用逐行阅读的方式进行。
    - ✧ 需求规格说明书评审应该在需求阶段当规格说明书整体或者部分完成后立即开展。由于规格说明书的重要性，很多软件项目评审将规格说明书作为其重要的评审内容之一。
      - ◌ 目的：
        - 尽早发现缺陷，使规格说明书具有更好的可测试性，软件测试人员可以更加熟悉系统应用。
      - ◌ 方法：
        - 静态测试
      - ◌ 技术：
        - 概要评审和详细评审

3.2 软件的阶段评审

- **需求规格说明书的评审**
  - 规格说明书的概要评审
    - 目的：
      - 发现特定的缺陷，比如大的原理性问题、遗漏的描述或过度复杂的描述等。
    - 评审过程
      - 测试人员站在用户的角度，确保作为第一质量要素的用户要求得到满足；
      - 研究现有的标准和基线；
      - 借鉴类似软件系统的评审。

3.2 软件的阶段评审

- **需求规格说明书的评审**
  - 规格说明书的详细评审
    - ✧ 规格说明书的详细评审从检查规格说明书的属性开始。一个好的规格说明书，包括其中的文字和图片，应具有如下属性：
      - ○ 完整性
      - ○ 精确性
      - ○ 准确性
      - ○ 一致性
      - ○ 无二义性
      - ○ 相关性
      - ○ 可行性
      - ○ 代码无关性
      - ○ 可测试性等

3.2 软件的阶段评审

- **需求规格说明书的评审**
  - 问题词语列表
    - ✧ 测试规格说明书的时候应密切关注下面的一些词汇以及这些词汇的上下文含义是否清晰，这些词汇常常会带来需求缺陷：
      - ◌ 总是、每个、所有、没有一个、从来不。。。
        - 这些词表示肯定和确定的含义，必须确认应该用这些词语，或找出不应该使用的理由。
      - ◌ 当然、所以、明显地、无疑、显然。。。
        - 这些词有劝说人接受的意思，规格书中应尽量避免。
      - ◌ 一些、有时、经常、通常、大部分、主要的、等等、类似、好、快、便宜、高效、小和稳定。。。
        - 这些词可测试性差，必须进一步定义以给出确切的含义描述。

3.2 软件的阶段评审

- **需求规格说明书的评审**
  - 问题词语列表 (续)
    - ✧ 测试规格说明书的时候应密切关注下面的一些词汇以及这些词汇的上下文含义是否清晰，这些词汇常常会带来需求缺陷：
      - ◌ 有把握的、处理过的、拒绝的、跳过的、去掉的 。。。
        - 这些词可能隐藏一些本该详细说明的功能性需求。
      - ◌ 如果 ... 那么 ... 。。。
        - 避免使用这些依赖于其他因素的描述。

3.2 软件的阶段评审

- **设计检查**
  - 设计检查的时机
    - ✧ 在编码开始前进行
  - 检查功能设计说明，消除歧义
    - ✧ 功能的用意、总体位置
    - ✧ 输入、输出
    - ✧ 可能的错误/例外
    - ✧ 接口定义
    - ✧ 交互细节
    - ✧ 实施建议

Thank you!