软件质量保障

1. 软件质量保障基础

- 软件质量保障背景：分析、设计和实现软件过程中的错误和缺陷，以及软件测试的重要性。
- 软件质量保障定义：质量保证（QA）和质量控制（QC）的定义及其在软件开发中的作用。

2. 软件质量保障方法

- 质量定义：符合规约和达到目的两种定义的区别。
- 软件的确认和验证：验证（Verification）和确认（Validation）的区别及Boehm的解释。
- 软件失败的术语：包括缺点、谬误、故障、失效等的定义和区分。

3. 显式和隐式的需求和规约

- 需求和规约：显式需求和规约的识别，以及它们在软件开发中的作用。

4. 区分缺陷的严重性和优先级

- 严重性与优先级：衡量缺陷对用户或组织的影响和后果，以及缺陷在开发组织中的重要性。

5. 软件测试

- 软件测试定义：测试的目的、活动和测试标准。
- 测试的目的：发现软件缺陷和对质量进行总体评估。
- 传统的测试技术：白盒测试和黑盒测试的技术与方法。
- 测试策略：单元测试、集成测试、确认测试和系统测试的定义和区别。
- 测试完成标准：决定何时停止测试的标准。

6. 最新的测试技术

- 模糊测试、差分测试和蜕变测试：这些技术的基本原理和应用场景。

7. 静态分析

软件测试

- 软件测试定义：对程序行为的动态验证，使用有限的测试用例集。
- 错误观点：测试不能发现所有错误；测试不是证明程序正确，而是发现错误。
- 测试目的：发现错误、证明软件功能和性能符合需求、为可靠性分析提供数据、不能证明无错误。

2. 软件测试原则

- 尽早和不断测试：贯穿软件开发全程。
- 测试用例组成：测试输入数据和预期输出结果。
- 独立测试队伍：开发与测试分离。
- 合理与不合理输入：包括正常和异常输入条件。
- 群集现象：错误倾向于在程序的某些部分集中。
- 严格执行测试计划：避免随意性。
- 全面检查测试结果：确保没有遗漏错误。
- 回归测试：修改后重新运行测试用例。
- 保存测试资料：方便维护。

3. 白盒测试与黑盒测试

- 测试用例设计：设计最少的测试用例发现最多的错误。
- 白盒测试（结构测试）：基于程序内部逻辑结构设计测试用例。
- 黑盒测试（行为测试）：基于需求规约，不考虑内部逻辑。

4. 测试策略

- V模型：软件开发各阶段与测试策略的对应关系。
- 测试目标：量化产品需求、显式陈述测试目标等。

5. 单元测试

软件支持维护演化devops

1. 软件支持与维护概述

- 软件工程流程：问题定义、需求分析、概要设计、详细设计、编码、测试、维护。
- 售后服务类比：比较软件产品与电器产品、生活用品的售后服务。
- 新问题应对：面对新用户要求、新硬件设备，考虑任务执行和解决方案。

2. 软件支持

- 软件支持定义：产品发布后的支持和维护周期长，对用户体验至关重要。
- 支持类型：产品缺陷性支持和非缺陷性支持和服务。
- 支持成本和方式：商用软件和定制软件的支持成本，包括按呼叫收费和客户咨询服务。
- 开源软件支持：开源软件的支持方式，如社区支持和商业支持提供商。

3. 软件维护

- 软件维护定义：软件开发完成后的补充、修改和增加工作。
- 维护类别：矫正性维护、完善性维护、适应性维护、预防性维护。
- 影响因素：系统规模、年龄、结构合理性、应用类型、易维护性等。
- 维护步骤：检查待维护系统、确定维护位置和性质、研究可行性、实施维护、确认维护。
- 维护原则：不损害程序质量、保持风格一致性、有利于未来改变、对用户无不利影响。

4. 软件演化

- 软件演化定义：软件系统经过分析、测试/验证后，面对需求改变、环境变化等问题的持续维护和演化。

- 静态分析方法：基于语法和浅层次语义分析预测程序行为的方法。
- 静态分析的用途：检查中间文档和源代码，以及常见工具的介绍。

8. 静态分析的问题

- 误报和漏报：静态分析工具的常见问题及其对软件工程师的影响。

9. 形式化方法

- 形式化方法定义：用数学技术证明程序正确性的方法。
- 形式化方法的应用：需求规格说明、证明设计符合规约等。
- 形式化方法的缺点：包括掌握难度、适用范围限制等。

10. 检查和审查

- 检查和审查的定义：由团队对代码或中间文档进行审查的过程。
- 桌面检查和代码检查：检查项目、过程和注意事项。

11. 走查

- 走查的定义和目的：知识传播、头脑风暴、评估设计方案、找问题。

- 定义：针对模块或构件的测试。
- 内容：模块接口、局部数据结构、边界条件等。
- 过程：结合编码工作，使用驱动程序和桩模块。

6. 集成测试

- 定义：将独立模块组合后进行的测试。
- 方式：非增量式和增量式（自顶向下和自底向上）。

7. 确认测试

- 标准：基于软件需求规约，检查功能实现和文档完整性。

8. 系统测试

- 定义：对基于计算机的系统进行的测试。
- 种类：恢复测试、安全保密性测试、压力测试、性能测试。

9. 测试完成标准

- 统计标准：基于统计模型决定测试停止时间。
- 方法：使用指定测试用例、植入缺陷观察发现率、错误数目曲线观察。

- 变更驱动的演化：完善性、适应性、预防性变更。
- 变更控制：发起、批准、执行、跟踪和关闭变更请求的管理流程。

5. 程序理解

- 程序理解定义：以源代码为主要参考，理解软件系统行为的活动。
- 程序理解场景：遗产软件、重构、复用、维护。
- 程序理解方法：基于分析的方法（静态和动态）、基于类比的方法、基于学习的方法、基于大模型的方法。

6. DevOps

- DevOps定义：一套实践、工具和文化理念，用于自动化和整合软件开发与IT团队之间的流程。
- DevOps价值：提高软件生产运营效率、改善软件质量、改善开发与运维团队的协作和沟通。
- CI/CD：持续集成/持续部署，作为DevOps的实施途径之一。