Chapter 19. Architecture, Implementation, and Testing

【理解：实现与架构的一致性】

【掌握：将架构嵌入代码。框架方法。代码模版方法。防止架构侵蚀】

* 四种技术有助于保持代码和体系结构的一致性:
* 将设计嵌入到代码中（Embedding the design in the code）：
  + 架构作为实现的蓝图。这意味着

1. 实现者知道他们正在实现什么架构结构。
2. 他们可以在代码中以注释的形式记录体系结构结构。然后，任何熟悉代码的人都将知道一些约束。
3. 在所有级别上使用注释。
4. 项目应该有关于如何做到这一点的约定。然后，工具可以自动地将代码和体系结构联系起来。

* 框架（Frameworks）：
  + 框架是围绕特定主题组织的一组可重用类。
  + 程序员使用框架提供的服务。
  + 例子：
    - Ruby on Rails是基于MVC的，是为web应用程序设计的。它提供了从web服务器收集信息、与数据库通信或查询数据库以及呈现模板的功能。
    - AUTOSAR是为汽车内的计算机设计的。
  + 基类/接口
  + 相互作用机理
  + 控制器
  + 配置
* 代码模板（Code templates）：
  + 代码模板是程序员提供应用程序特定部分的代码的集合。
  + 下面的例子实现了一个故障转移协议:

如果在关键应用程序组件中检测到故障，则会发生如下切换:

1. 在另一个处理器上并行执行的辅助副本被提升到新的主副本。
2. 新的主节点与客户端一起重构
3. 启动一个新的辅助服务器作为新的主服务器的备份。
4. 新开办的中学对新开办的小学宣布成立。

如果在辅助服务器中检测到故障，则在其他处理器上启动一个新的辅助服务器。

* 过程:
* 对于必须具有热备份的每个关键组件，使用代码模板。
* 将特定于应用程序的代码放置在模板中的固定位置。
* 保持代码和体系结构的一致性(即避免“体系结构侵蚀”) ：

Keeping code and architecture consistent (i.e. avoiding “architecture erosion”)

* + 实现将偏离文档化的体系结构。
  + 实现者可能做出彼此或与体系结构不一致的决策。
  + 架构可能没有预见到出现的所有事件。
* 代码模板的优点（Advantages of Code Templates）
  + 具有相似属性的组件的行为方式相似。
  + 模板只需要调试一次。
  + 复杂的部分可以由技术人员完成，也可以交给技术较差的人员。
* 防止结构侵蚀（Preventing Architecture Erosion）
  + 使用工具执行架构约束。
    - 可以添加在构建或检入期间强制执行的体系结构规则。
  + 当侵蚀发生时，将文件标记为过期。会给剩余的部分更多的信任。
  + 安排文档/代码同步时间。

【了解：架构师在测试中的角色】

* 架构和测试（Architecture and Testing）：
  + 单元测试（Unit test）
    - 体系结构定义了要测试的单元。它们是组件或模块。
    - 体系结构定义了单元的职责和交互。
    - 测试线束将驱动要测试的元素。测试线束可以测试:
      * 功能正确性的责任
      * 综合载荷性能
      * 通过故障注入的可用性。
    - 还可以通过向测试团队分配更改来测试可修改性需求。
  + 集成测试（Integration test）
    - 与单元测试一样，集成测试可以测试功能、性能、可用性和安全性。
    - 可以通过让测试工具执行各种攻击场景来测试安全性。
    - 如果没有释放资源或配置指定错误，系统在长时间运行后可能会降级。
  + 网络效应（Network effects）
    - 假设一个错误导致2%的性能下降。
      * 这是在正常的变化，如果使用一个服务器
      * 如果将系统部署到数千台服务器上，可能会导致严重的退化。
    - 配置错误在安装过程中很常见，可能会导致网络效应。
    - 网络效应最好通过自感知系统来发现，即系统监控自身并使值对外可用。
  + 测试活动（Test activities）
    - 架构师应该积极参与
      * 测试计划，因为架构师知道系统的敏感区域。
      * 测试开发。测试驱动开发是一种技术，在这种技术中，系统的下一个增量被开发来满足预定的测试。
      * 测试解释。架构师知道各种被监视的值应该是什么，并且最适合解释测试结果。
      * 测试工具的创造。测试工具必须与系统紧密地交互，这需要体系结构知识。

【本章总结】

* 实现
  + - * 实现活动可以将体系结构知识嵌入到代码中
      * 模板可以用于重现的关键部分
      * 可以通过使用工具和管理流程来防止体系结构的侵蚀
* 测试
  + - * 单元和集成测试依赖于架构知识和测试工具。
      * 当将系统部署到1000台服务器时，很难发现网络影响。
      * 架构师应该参与各种各样的测试活动。