

Object Oriented Analysis & Design

面向对象分析与设计

Lecture_11 成为好的架构师

主讲: 姜宁康 博士



■ 3、成为一个好的架构师...

- 技术专家&领域专家
- 团队技术权威、领导
- 推销人员

3.1 架构师作为技术专家&领域专家

- 软件工程知识经验很重要（但不够）
- 问题领域存在重要的细微差别，如
 - 成熟度 Maturity
 - 稳定性 Stability
 - 系统用户的特殊性
- 影响架构方案的设计、选择
 - 分布性 Distribution
 - 扩展性 Scalability
 - 演化性 Evolvability

3.1 架构师作为技术专家&领域专家

■ 软件工程师与软件架构师

- 一方有丰富的软件开发经验，一方有丰富的领域经验 (roughly 50/50)
- 一方保持最新的编程技能，一方跟踪最新的技术趋势和标准

■ 有时架构师会钻研系统某一高风险模块的内部技术细节

- 明白底层细节，有助于验证初始的设计方案
- 做一些原型 Prototyping

■ 直觉 **intuition**

- 只有当 ‘深层神经元’ 记住了足够多的相关资料之后，才会有效用
- “熟读唐诗三百首，不会做诗也会吟”
- “水到渠成”

3.1 架构师作为技术专家&领域专家

- 理解软件开发中的**细微差别**
 - 原则 Principles
 - 方法与技术 Methods & techniques 、方法学 Methodologies
 - 工具 Tools
- 不必是“世界级”的程序员，但必须熟悉程序员使用的方法
 - 有时可以 审核代码 Code review
- 应当理解架构选择中的多种可能
 - 不能自以为是 Do not live in ivory tower
 - 有时架构选择约束了实现的选择
 - 有时实现级的技术/工具约束了架构选择
- 理解不同的软件开发方法
 - 如 object-oriented (OO) & component-based
- 在以下方面展示自己的专家级知识 **Demonstrated expertise in**
 - 系统建模、各种架构方案的折衷分析 Architectural trade-off analysis
 - 架构方案与系统需求之间的对应

3.2 架构师作为领导：是技术权威

■ 技术权威

- 靠的是知识丰富&成就 Based on knowledge & achievement
- 命令是通过想法、专家意见、词语表达和行动得到尊重
- 不能依赖组织赋予的职位
 - 即使没有行政职务，也能实现自己的权威

■ 保证遵循了设计原则

■ 为了改进生产效率&产品质量

- 引入新的想法、解决方案和技术
- 成为新手、青年员工的导师

■ 做出决策并且帮助软件工程师去实现

- 在这过程中寻取其他人的支持、尊重

3.2 架构师作为领导

- 架构师对工作要非常投入，有适当的权限在技术上做出决定
- 决策依据
 - 领域知识 Domain knowledge
 - 工作计划 Schedule
 - 不需要全部的信息、也不需要团队一致同意
 - 新技术的影响 Impact of new technology
 - 将责任转移到领域专家或开发负责人
- 决策的艺术在于“及时”
 - 设计方案需要灵活性、适应新的变化因素
 - 及时做出决策，后面再修改，常常优于迟迟不做决策以至于延误开发进程

3.2 架构师作为领导

■ 保持乐观 As Cheerleaders

- 面对周期长、规模大、复杂的项目更是如此
- 开发小组在项目的子集上工作
 - 组长缺乏对整个项目目标的远见 Managers lose sight of overall project goals
 - 客户在长久的等待中失去了耐心 Customers get impatient from long wait

■ 架构师必须

- 了解高层次的前景目标，以及在某些方面适当的细节
- 让不同的涉众确信架构的
 - 美、实用、可适应性、科技的影响、财务的影响等
- 保持“部队”士气高昂

3.2 架构师作为领导

■ 架构师是战略家 Architects As Strategists

■ 开发出优美的架构，还不够！

- 技术是公司发展的一部分，架构对组织的发展必须是最好的

■ 架构满足组织的

- 商业战略 Business strategy
- 背后的基本原理 Rationale behind it
- 商业实践 Business practices
- 决策流程 Decision making processes

■ 必须了解竞争对手 Must also be aware of competitors'

- 产品 Products
- 战略 Strategies
- 过程 Processes

3.2 架构师作为领导

■ 架构师还是政治家 As Politicians

- 必须让组织的核心领导致力于架构 Must get key organization players committed to architecture
- 必须做许多有影响的事 Must do a lot of influencing themselves
 - 找出组织内谁是核心领导 Find out who the key players are
 - 发现他们想要什么 Find out what they want
 - 发现组织背后的组织（问题的根源）

■ 架构师要不断地

- 倾听、打造人脉圈子、善于表达、推销前景设想、多角度思考问题

3.2 架构师作为领导

■ 架构师是成本预估师 As Cost Estimators

- 需要了解选择不同的架构方案带来的不同财务影响 Must understand financial ramifications (分枝) of architectural choices
 - 新开发还是重用旧的 new or reuse
 - 采用“商用货架产品”的成本 Cost of COTS adoption
 - 重用的开发成本 Cost of development for reuse
 - 组织内部开发 还是外包开发 In-house or out-sourcing
 - 组织的财务稳定性 以及市场的位置 Company's financial stability & position in marketplace
 - “技术优先”的方案不总是最适合的 Technologically superior solution is not always most appropriate one
 - 对成本、进度的影响 Impact on cost & schedule
- 快速、近似的成本估算，一般足够了
 - 一旦解决方案基本定型之后，再采用详细的成本估算技术

3.3 架构师作为推销员

- 架构师作为推销员 Architects as Salespeople
- 基于前面说的种种理由，架构师要推销
 - 总体的项目前景 Overall vision
 - 技术方案 Technological solutions
 - 架构方案关键的特性 Key properties of architecture
 - 架构方案将直接导致的最终系统的关键属性 Key properties of eventual system that architecture will directly enable
 - 成本/计划的说明 Cost/schedule profile
 - 坚持架构方案的重要性 Importance of sticking to architecture
 - 背离架构方案将带来的损失 Penalties of deviating from it

架构师标签：技术专家、领域专家、技术权威、决策者、乐观、战略家、政治家、成本预估师、推销员





■ **本讲结束**