**软件架构实践（第四版）\_读书笔记\_1**

第一部分：简介

第一章·什么是软件架构

1. 软件架构的基本原则归纳：
2. 软件架构的设计和实现必须与组织的业务对接，因为软件系统都是为了满足组织的业务目标而构建的。
3. 软件架构是业务目标与最终系统之间的桥梁，它将抽象的业务需求转化为具体的技术实现。
4. 软件架构的设计、分析和记录可以利用已知的技术来实现，这些技术经过验证，能够帮助实现业务目标，换言之，它使用的是已知且可操作的技术。
5. 软件架构要具备足够的灵活性和可扩展性，以适应未来业务目标的变化。
6. 软件架构应该能够被设计、分析和记录。

总而言之，软件架构的核心目的在于形成一套方法，使得软件工程变得更加具体、可实现以及后续的扩展。

1. 软件架构是什么/不是什么 组件和连接结构

一组软件结构的集合 模块结构

分配结构

软件架构是 一种抽象：架构只关心接口所示西安的共有部分

一种设计：架构是设计，但并非所有设计都是架构

1. 每个软件系统都有软件架构，这是因为每个系统都有元素和关系，这也反映出架构文档的重要性。
2. 并非所有架构都是好的，同时不可以将试错作为选择架构的方式，所以架构文档非常重要。
3. 架构包含行为，如系统和企业架构；系统架构和企业架构
4. 架构结构和视图：架构结构可以分为三大类：

①组件和连接器（C&C）结构（Component-and-connector (C&C) structures），它侧重于元素在运行时相互交互以执行系统功能的方式，描述了如何将系统构建为一组具有运行时行为（组件）和交互（连接器）的元素。

②模块结构（Module structures），它将系统划分为实现单元，显示了系统构建或获得的一组代码或数据单元。

③分配结构（Allocation structures），它建立了从软件结构到系统非软件结构的映射。

1. 什么造就了好的架构？本书通过观察结构，分为了两类：流程建议和产品（或结构）建议。

关于流程建议：

①软件架构应该是一个架构师或者一小部分架构师的产物，并且具有确定的技术领导者。

②架构师（或架构团队）应在明确指定的质量属性优先级列表上建立架构。

③使用视图记录架构

④评估架构提供系统重要质量属性的能力

⑤架构应当可以增量实现，以避免一次集成所有内容。

关于经验法则：

1.架构应该具有明确的模块，并遵循信息隐藏和功能点分离原则。

2.尽可能使用熟知的架构方式

3.架构不应依赖于商业产品或工具的特定版本

4.生成数据的模块应与使用数据的模块分开。

5.模块和组件一般不是一一对应的

6.应该使每个进程可以轻松地分配给指定的处理器，甚至可以在运行时更改。

7.架构应具有少量简单的组件交互模式。也就是说，系统应该在整个过程中以相同的方式做同样的事情。

8.架构应包含一组特定的且规模较小的资源竞争区域，并且竞争解决方式已经明确指定和维护。