架构视图是一种设计架构、描述架构的核心手段。通过“架构视图”作为分而治之的手段，使架构师分别专注于架构的不同方面、相对独立地分析和设计不同“子问题”。

1、2视图方法

逻辑视图+物理视图。

1.1、逻辑架构

软件的逻辑架构规定了软件系统由哪些逻辑元素组成以及这些逻辑元素之间的关系。具体而言，组成软件系统的逻辑元素可以是逻辑层（Layer）、功能子系统、模块。

设计逻辑架构的核心任务，是比较全面地**识别模块、规划接口，并基于此进一步明确模块之间的使用关系和使用机制。**

1.2、物理架构

软件的物理架构规定了组成软件系统的物理元素，这些物理元素之间的关系，以及它们部署到硬件上的策略。

物理架构可以反映出软件系统动态运行时的组织情况。物理架构定义中所提及的“物理元素”就是进程、线程，以及作为类的运行时实例的对象等，而进程调度、线程同步、进程或线程通信等则进一步反映物理架构的动态行为。

1.3、2视图设计实现

- **逻辑架构中关于职责划分的决策，体现为层、功能子系统和模块等的划分决定，从静态视角为详细设计和编程实现提供切实的指导；逻辑架构还规定了不同逻辑单元之间的交互接口和交互机制。**
- **交互机制是指不同软件单元之间交互的手段。交互机制的例子有：方法调用、基于RMI的远程方法调用、发送消息等。**
- **物理架构，关注软件系统在计算机中运行期间的情况。**

2、5视图方法

5视图适合更大型的软件，更全面地覆盖了架构设计的各个方面。

- **职责划分（逻辑视图）**

分模块、分层、划分垂直功能子系统，为模块/层/子系统定义接口....

- **程序单元组织（开发视图）**

开发语言选型、Application Framework选择、编译依赖关系...

- **控制流组织（运行视图）**

多进程技术、多线程技术、中断服务程序...

- **物理节点安排（物理视图）**

PC机、服务器、单片机的选型与关联...

- 持久化设计（数据视图）

关系数据库、实时数据库、文件、Flash...