随着互联网和软件行业的发展，人们对软件的功能和性能需求越来越高，则软件的复杂性和开发难度也随之增加，则团队合作开发的大型软件成规模式增长。然而，数据表明，大系统软件开发中70%的错误是由需求和软件设计阶段引入的;而且错误在系统中存在的时间越长则越难发现，解决这些错误的代价也越高。

伴随着这些问题的日益突出，整个软件系统结构的设计与规格说明便显得比在早期软件开发中占有重要地位的算法选择和计算问题的数据结构更为重要。由此便有了软件体系结构这一概念。

软件体系结构是软件需求和软件设计之间的一座桥梁，使得软件设计能够很真实地反

映并满足软件的需求，从而提高了软件需求和软件设计的质量。

我们往往比较熟悉的是二层C/S体系结构，即Client/Server结构，后台服务器负责管理数据，前台客户机负责与用户交互，但是该结构因为表现层和事务层都放在客户端，而数据逻辑层和数据存储层则放在服务端，这样的安排带来了很多限制，比如:①客户端很庞大，应用程序升级和维护时十分困难且耗资很大。对应用程序一个小小的改动，就必须通知或亲临每一个客户端去更新;新增或升级一台机器，都要把应用及其相关的文件安装在客户端上;②事务层不能与跨平台的客户端共享;③孤立了不同的逻辑组件;④没有统一的数据逻辑层来提供不同种类的数据存储层;⑤不支持Internet等，为了克服其中的一些缺陷，人们又提出了三层(多层)C/S体系结构。

三层次客户机/服务器(C/S)结构是在常规客户机/服务器(C/S)结构上提出的，系统在客户机和数据库服务器间添加一个应用服务器。三层C/S结构将应用功能分成表示层、功能层和数据层三个部分。表示层是应用的用户接口部分，它承担系统与用户间的对话功能，用于简单检查(不涉及有关业务本身的处理逻辑)用户输入的数据，显示应用的输出数据。在变更用户接口时，只需改写显示控制和数据检查程序，而不影响其他两层;功能层又称业务逻辑层，它将具体的业务处理逻辑编入程序中。表示层和功能层之间的数据传输要尽可能简洁。数据层就是数据库管理系统，负责管理对数据库数据的读写。

三层C/S的解决方案是:对这三层进行明确分割，并在逻辑上使其独立。原来的数据层作为数据库管理系统已经独立出来，所以，关键是要将表示层和功能层分离成各自独立的程序，并且还要使这两层间的接口简洁明了。

与传统的二层结构相比，三层C/S结构具有以下优点:①允许合理地划分三层结构的功能，使之在逻辑上保持相对独立性，从而使整个系统的逻辑结构更为清晰，能提高系统和软件的可维护性和可扩展性。②允许更灵活有效地选用相应的平台和硬件系统，使之在处理负荷能力上与处理特性上分别适应于结构清晰的三层。③应用的各层可以并行开发，各层也可以选择各自最适合的开发语言。④允许充分利用功能层有效地隔离开表示层与数据层，未授权的用户难以绕过功能层而利用数据库工具或黑客手段去非法地访问数据层，这就为严格的安全管理奠定了坚实的基础。

当然了，这和我们平常使用的各种编程框架很相似，写代码时，往往需要考虑自己在写哪一层的代码，从而确定自己的代码风格，比如Web前端往往写一些简单的业务逻辑，对数据库进行增删改查，不需要关心其他层次。从这样的角度看，知识往往都是相同的，特别是计算机领域。