通过参考文献以及网上搜索得到的内容，对软件体系结构有了比较深层次的了解，学习了有关软件体系结构的相关内容，软件体系结构为我们所使用的软件提供了结构和行为的抽象，不仅指定了系统的结构，也表明了系统的需求，以及系统元素之间的关系。我认为规范化，系统化的软件体系结构描述，对理解软件体系结构有很大的帮助作用，但这个过程不是很简单就能完成的，我们都必须经过非形式化的过程，在这个过程中不断积累知识，最后将这个非形式化的过程变得形式化，标准化。

通过短暂的学习了解到了软件体系结构的模型及其作用：

结构模型：用体系结构的构件，连接件来刻画结构，并试图通过结构来反映系统的重要语义内容，包括了系统的配置，约束等。

框架模型：侧重于描写的是整体结构，主要为目标建立只针对该目标的架构。

动态模型：对结构模型，框架模型的补充。

过程模型：顾名思义，构造了系统的过程和主要步骤。

功能模型：一种比较特殊的框架，认为整体架构是按照功能按层次分开的。

4+1试图模型及其作用：

逻辑视图：主要支持系统的功能需求，即系统提供给最终用户的服务。在逻辑视图中，系统分解成一系列的功能抽象，这些抽象主要来自问题领域。这种分解不但可以用来进行功能分析，而且可用作标识在整个系统的各个不同部分的通用机制和设计元素。在面向对象技术中，通过抽象、封装和继承，可以用对象模型来代表逻辑视图，用类图来描述逻辑视图。

开发视图：也称模块视图，主要侧重于软件模块的组织和管理。开发视图要考虑软件内部的需求，如软件开发的容易性、软件的重用和软件的通用性，要充分考虑由于具体开发工具的不同而带来的局限性。开发视图通过系统输入输出关系的模型图和子系统图来描述。

进程视图：侧重于系统的运行特性，主要关注一些非功能性的需求。进程视图强调并发性、分布性、系统集成性和容错能力，以及从逻辑视图中的主要抽象如何适合进程结构。它也定义逻辑视图中的各个类的操作具体是在哪一个线程中被执行的。进程视图可以描述成多层抽象，每个级别分别关注不同的方面。在最高层抽象中，进程结构可以看作是构成一个执行单元的一组任务。它可看成一系列独立的，通过逻辑网络相互通信的程序。它们是分布的，通过总线或局域网、广域网等硬件资源连接起来。

物理视图：主要考虑如何把软件映射到硬件上，它通常要考虑到系统性能、规模、可靠性等。解决系统拓扑结构、系统安装、通讯等问题。当软件运行于不同的节点上时，各视图中的构件都直接或间接地对应于系统的不同节点上。因此，从软件到节点的映射要有较高的灵活性，当环境改变时，对系统其他视图的影响最小。

场景视图：可以看作是那些重要系统活动的抽象，它使四个视图有机联系起来，从某种意义上说场景是最重要的需求抽象。在开发体系结构时，它可以帮助设计者找到体系结构的构件和它们之间的作用关系。同时，也可以用场景来分析一个特定的视图，或描述不同视图构件间是如何相互作用的。场景可以用文本表示，也可以用图形表示

C/S（客户-服务器）风格定义： C/S软件体系结构是基于资源不对等，且为实现共享而提出来的。它定义了工作站如何与服务器相连，以实现数据和应用分布到多个处理机上。有三个主要组成部分：数据库服务器、客户应用程序和网络。

C/S风格优点：

具有强大的数据操作和事务处理能力，模型思想简单，易于人们理解和接受

系统的客户应用程序和服务器构件分别运行在不同的计算机上，系统中每台服务器都可以适合各构件的要求，这对于硬件和软件的变化显示出极大的适应性和灵活性，而且易于对系统进行扩充和缩小

在C/S体系结构中，客户应用程序的开发集中于数据的显示和分析，而数据库服务器的开发则集中于数据的管理，不必在每一个新的应用程序中都要对一个DBMS进行编码。将大的应用处理任务分布到许多通过网络连接的低成本计算机上，以节约大量费用

缺点：开发成本较高。客户端程序设计复杂。信息内容和形式单一。用户界面风格不一，使用繁杂，不利于推广使用。软件移植困难。软件维护和升级困难。新技术不能轻易应用。

12.三层C/S风格： 三层：表示层，功能层，数据层。

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，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；并且这些平台和各个组成部分可以具有良好的可升级性和开放性 3 应用的各层可以并行开发，可以选择各自最适合的开发语言 4 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，为严格的安全管理奠定了坚实的基础

缺点：1 三层C/S结构各层间的通信效率若不高，即使分配给各层的硬件能力很强，其作为整体来说也达不到所要求的性能 2 设计时必须慎重考虑三层间的通信方法、通信频度及数据量。这和提高各层的独立性一样是三层C/S结构的关键问题

B/S风格（浏览/服务器）： 就是三层C/S应用结构的一种实现方式，其具体结构为：浏览器/Web服务器/数据库服务器

优点：1 基于B/S体系结构的软件，系统安装、修改和维护全在服务器端解决。用户在使用系统时，仅仅需要一个浏览器就可运行全部的模块，真正达到了“零客户端”的功能，很容易在运行时自动升级 2 还提供了异种机、异种网、异种应用服务的联机、联网、统一服务的最现实的开放性基础

缺点：1 B/S体系结构缺乏对动态页面的支持能力，没有集成有效的数据库处理功能 2系统扩展能力差，安全性难以控制 3 在数据查询等响应速度上，要远远地低于C/S体系结构 4 数据提交一般以页面为单位，数据的动态交互性不强，不利于在线事务处理(OLTP)应用

HMB风格（基于层次消息总线的体系结构风格）： HMB风格基于层次消息总线/支持构件的分布和并发，构件之间通过消息总线进行通信。 消息总线是系统的连接件，负责消息的分派、传递和过滤以及处理结果的返回。各个构件挂接在消息总线上，向总线登记感兴趣的消息类型。构件根据需要发出消息,由消息总线负责把该消息分派到系统中所有对此消息感兴趣的构件，消息是构件之间通信的唯-方式。构件接收到消息后,根据自身状态对消息进行响应,并通过总线返回处理结果。由于构件通过总线进行连接,并不要求各个构件具有相同的地址空间或局限在一台机器上。