通这学期的软件工程的课程的学习， 我清楚地认知到了软件就是程序加软件工程，软件工程的基本含义。一个项目的好坏不仅仅决定于程序员的代码水平，还有很多的其他方面的因素。同时我了解到针对不同的大型项目有不同的模型。常见的体系架构模式有以下几种：

**一. 分层模式**

这种模式也称为多层体系架构模式。它可以用来构造可以分解为子任务组的程序，每个子任务都处于一个特定的抽象级别。每个层都为下一个提供更高层次服务。

一般信息系统中最常见的是如下所列的4层：

表示层(也称为UI层)

应用层(也称为服务层)

业务逻辑层(也称为领域层)

数据访问层(也称为持久化层)

**二. 客户端-服务器模式**

这种模式由两部分组成：一个服务器和多个客户端。服务器组件将为多个客户端组件提供服务。客户端从服务器请求服务，服务器为这些客户端提供相关服务。此外，服务器持续侦听客户机请求。

**三. 主从设备模式**

这种模式由两方组成;主设备和从设备。主设备组件在相同的从设备组件中分配工作，并计算最终结果，这些结果是由从设备返回的结果。

**四. 管道-过滤器模式**

此模式可用于构造生成和处理数据流的系统。每个处理步骤都封装在一个过滤器组件内。要处理的数据是通过管道传递的。这些管道可以用于缓冲或用于同步。

**五. 代理模式**

此模式用于构造具有解耦组件的分布式系统。这些组件可以通过远程服务调用彼此交互。代理组件负责组件之间的通信协调。

服务器将其功能(服务和特征)发布给代理。客户端从代理请求服务，然后代理将客户端重定向到其注册中心的适当服务。

**六. 点对点模式**

在这种模式中，单个组件被称为对等点。对等点可以作为客户端，从其他对等点请求服务，作为服务器，为其他对等点提供服务。对等点可以充当客户端或服务器或两者的角色，并且可以随时间动态地更改其角色。

**七. 事件总线模式**

这种模式主要是处理事件，包括4个主要组件：事件源、事件监听器、通道和事件总线。消息源将消息发布到事件总线上的特定通道上。侦听器订阅特定的通道。侦听器会被通知消息，这些消息被发布到它们之前订阅的一个通道上。

**八. 模型-视图-控制器模式**

这种模式，也称为MVC模式，把一个交互式应用程序划分为3个部分，

模型：包含核心功能和数据

视图：将信息显示给用户(可以定义多个视图)

控制器：处理用户输入的信息

这样做是为了将信息的内部表示与信息的呈现方式分离开来，并接受用户的请求。它分离了组件，并允许有效的代码重用。

**九. 黑板模式**

这种模式对于没有确定解决方案策略的问题是有用的。黑板模式由3个主要组成部分组成。

黑板——包含来自解决方案空间的对象的结构化全局内存

知识源——专门的模块和它们自己的表示

控制组件——选择、配置和执行模块

所有的组件都可以访问黑板。组件可以生成添加到黑板上的新数据对象。组件在黑板上查找特定类型的数据，并通过与现有知识源的模式匹配来查找这些数据。

**十. 解释器模式**

这个模式用于设计一个解释用专用语言编写的程序的组件。它主要指定如何评估程序的行数，即以特定的语言编写的句子或表达式。其基本思想是为每种语言的符号都有一个分类。