**架构定义**

架构即设计，他包括系统之间，组件之间，类之间的设计和交换。

**架构的目标**

以最小的人力成本去构建和维护系统。

经过架构的系统，开发速度比较平稳。但未经架构的系统，开发速度先快后慢。

**需求与架构**

是需求重要还是架构重要呢？

一个示例：我们优化架构用了1天，做需求用了1天。或者我们直接做需求用了1天（半年后这个时长变成3天，时间越久越长）。很明显，如果我们一直不管架构，那么最后需求也会实现不了。

**编程范式**

目前我们重要有3个编程范式

- 结构化编程

- 面向对象编程

- 函数式编程

**组件**

组件是一个dll或者一个类库，如下是构建组件的基本原则：

- 复用发布等同原则（REP）

软件复用的最小粒度应该等同于其发布的最小粒度。即你想复用多少，你就发布多少，你发布的包应该要有版本号。

- 共同闭包原则（CCP）

我们应该将那些会同时修改，并且为了相同目的而修改的类放到同一个组件中。

- 共同复用原则（CRP）

不要强迫一个组件的用户依赖他们不需要的东西。可以认为是单一职责了，应尽可能小的分割组件。

