201830210026 龙晓怡 第八章作业

8.2影响软件架构设计的主要因素有哪些？

1. 需求分析。需求分析是软件生命周期中的一个关键活动，目的是得到一个正确的、一致并且无二义的需求规约，并将此规约作为后续开发、验证和系统演化的基础。需求分析包括（确定问题空间范围、确定系统用户及他们的职责、从用户的角度去探索整体外部的行为、确定组件和连接件、规范组件和连接件、建立面向软件架构的高层次需求规约、检查规约）

2. 需求到软件架构的映射机制。对相同的需求采用不同的映射机制会得到不同风格的架构。

8.3 什么是软件架构需求？如何从用户需求和软件需求中获得软件架构需求？

软件架构需求是指需求到软件架构的映射，这是一个既复杂又细致的工作，对相同的需求采用不同的映射机制会得到不同风格的架构。在软件架构设计阶段，建立需求规约与架构的映射将相对困难。而把架构概念引人需求分析阶段，有助于保证需求规约、系统设计之间的可追踪性和一致性， 最终有效保持软件质量和软件演化。另外，需求规约是重要的可重用资源，将软件架构概念和原则引人需求分析，可以让我们获得更有结构性和呵重用的需求规约。

在从用户需求获取软件架构的过程中，有两件事情必须要做:

①探索如何用软件架构的概念和描述手段在较高抽象层次上刻画问题空间(用户需求)的软件需求，获得软件需求规约。

②探讨如何从软件需求规约自动或半自动地变换到软件架构设计。

从软件需求向软件架构转换主要关注两个问题:

①如何把软件需求模型转换成软件架构模型。

②如何保证模型转换的可追踪性。针对这两个问题的解决方案根据所采用的需求模型的不同而各异。

8.7 什么是软件架构设计的原则？这些原则为什么重要？

软件架构设计的原则是指软件架构师、软件工程师及其他资深软件开发人员在实践中总结的很多软件架构设计原则。软件架构设计原则有一般(基本)设计原则和关键设计原则两类。一般原则包含商业原则、数据原则、应用程序原则、技术原则等；关键设计原则包含关注分离点、单一职责原则、最少知识原则等。

重要性在于：这些原则对软件架构设计有很好的指导意义，软件开发从业人员人员应该尽可能多地掌握这些原则。

8.8软件架构设计原则和软件设计原则的异同点是什么？

相同点：都是为了达到低耦合，高内聚的目的。

不同点：

软件架构设计的原则有基本原则和关键原则，基本原则包含商业原则、数据原则、应用程序原则、技术原则等；关键设计原则包含关注分离点、单一职责原则、最少知识原则等。

软件设计的五大原则

（1）单一责任原则。就一个类而言，应该仅有一个引起它变化的原因。即，当需要修改某个类的时候原因有且只有一个，让一个类只做一种类型责任。

（2）开放、封闭原则。软件实体（类、模块、函数等）应该是可以扩展的，即开放的；但是不可修改的，即封闭的。

（3）里氏替换原则。子类型必须能够替换掉他们的基类型。即，在任何父类可以出现的地方，都可以用子类的实例来赋值给父类型的引用。当一个子类的实例应该能够替换任何其超类的实例时，它们之间才具有是一个（is-a ）关系。

（4）依赖倒置原则。抽象不应该依赖于细节，细节应该依赖于抽象。即，高层模块不应该依赖于低层模块，二者都应该依赖于抽象。

（5）接口分离原则。不应该强迫客户依赖于它们不用的方法。接口属于客户，不属于它所在的类层次结构。即：依赖于抽象，不要依赖于具体，同时在抽象级别不应该有对于细节的依赖。这样做的好处就在于可以最大限度地应对可能的

变化。