**Chapter 4. Understanding Quality Attributes**

了解：系统的功能需求。功能需求与系统架构的关系。功能需求与质量需求的关系。系统约束。

1. Functionality：系统完成预期工作的能力。
2. 不决定architecture，满足一系列functionality的architecture有很多。
3. Performance，availability，usability
4. 约束是已经做出的设计决策。

理解：系统的质量需求。战术的概念。

1. 上3

2. tactics: 架构师使用一组基本设计技术来实现质量属性响应。pattern是tactics的集合。如果没有要实现的模式（pattern）目标，tactics允许架构师构造一个“first principles”的片段。Tactics使设计更加系统化。

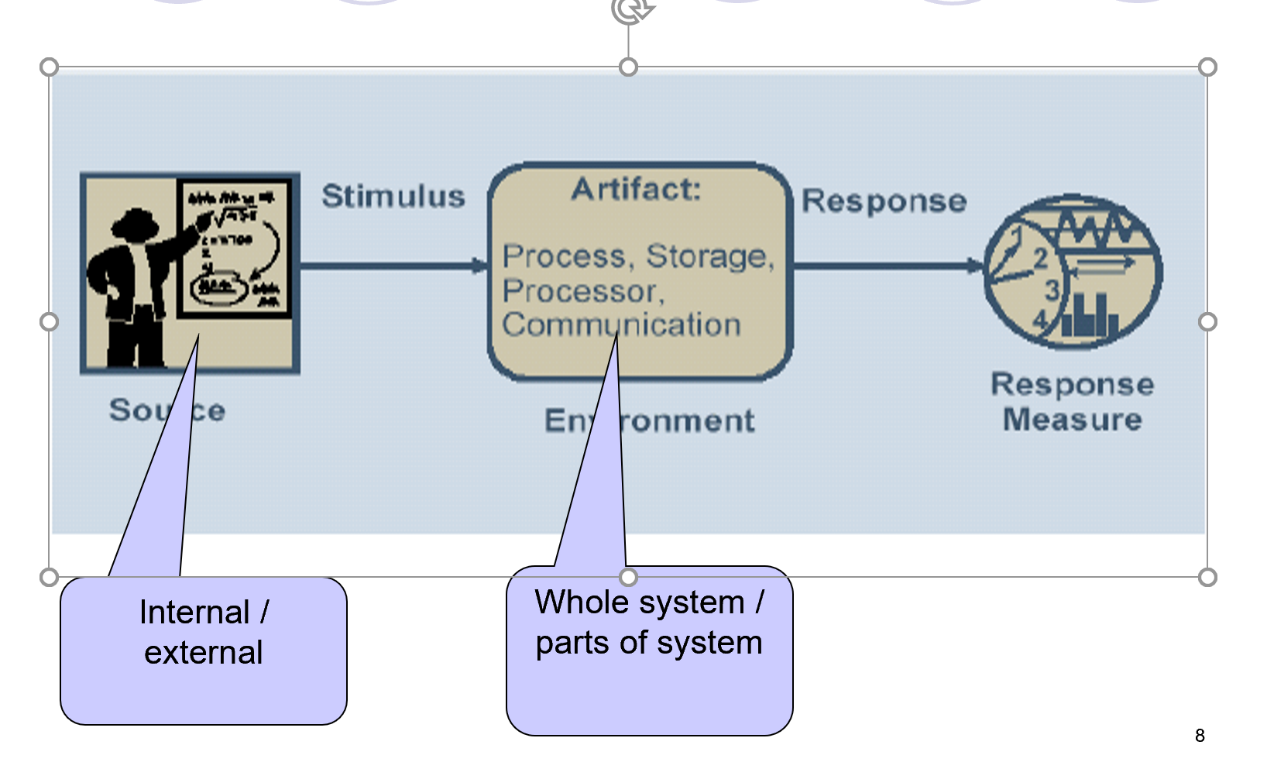
掌握：质量属性场景的概念和举例。质量设计的7种决策。

1. 概念：

Quality attribute **scenarios** has these parts:

Stimulus，Stimulus source，Response，Response measure，Environment，Artifact（刺激，刺激源，响应，响应测量，环境，工件）

1. **Source of stimulus：**产生刺激的某个实体
2. **Stimulus：是一种条件，当它到达一个系统时需要响应。**
3. **Environment：刺激是在此条件下产生的。**
4. **Artifact**.：Some artifact 被刺激.
5. **Response：刺激措施到来时采取的动作。**
6. **Response measure：**响应发生时，它应该以某种方式进行度量，以便可以测试需求。

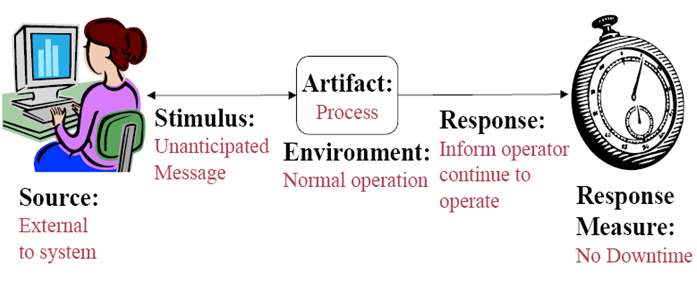


1. 举例：

General scenarios: 独立于系统并且可能适用于任何系统的场景

Concrete scenarios: 那些特定于正在考虑的特定系统的场景

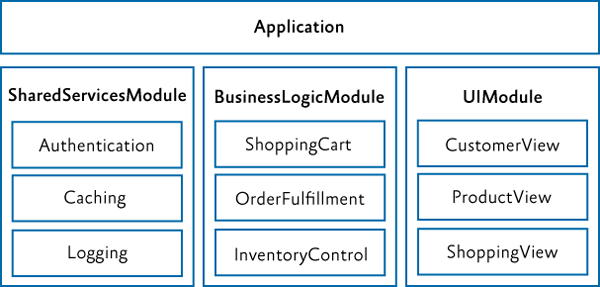
1. general scenario for availability（图文结合于Chapter 5的一般场景中）
2. concrete scenario for availability



1. 决策：

7种决策：Allocation of responsibilities，Coordination model，Data model，Management of resources，Mapping among architectural elements，Binding time decisions，Choice of technology （分配责任，协调模型，数据模型，管理资源，架构元素之间的映射，绑定时间决定，选择技术）

1. **Allocation of responsibilities**: 确定重要的职责，包括基本的系统功能、体系结构基础设施和质量属性的满足。确定如何将这些职责分配给非运行时和运行时元素(即模块、组件和连接器)。



1. **Coordination model:** 识别系统中必须协调或禁止协调的元素。确定协调的属性，如及时性、货币、完整性、正确性和一致性

选择实现这些特性的通信机制（communication mechanisms）。

1. **Data model：**选择主要的数据抽象、操作和属性。对数据进行一致解释所需的元数据（metadata）。数据的组织。这包括确定是否将数据保存在关系数据库、对象集合或两者兼而有之。
2. **Management of resources：**确定：必须管理的资源，每个资源的限制/管理每个资源的系统元素/资源如何共享(出现争用时使用的仲裁策略)/ 饱和度对不同资源的影响。
3. **Mapping among architectural elements：**模块和运行时元素相互之间的映射。将运行时元素分配给处理器/数据模型中的项分配给数据存储/模块和运行时元素到交付单元的映射。
4. **Binding time decisions：**和其他决策的关联：

分配职责：通过参数化的构建脚本模块设计构建时间选择。

协调模型：设计运行时协商的协议。

资源管理：设计在运行时插入（plugged）新的外围设备（peripheral devices）。

技术选择：构建一个APP store ，自动下载APP的适当版本。

1. **Choice of technology：**哪些技术实现在其他类别中做出的决策/支持该技术的工具是否足够/内部熟悉程度和外部对该技术的支持程度/确定副作用/确定新技术是否与现有技术堆栈兼容。