什么是软件架构

架构是一组软件结构

三种架构类别

- **Modules structures （模块结构）**
  - Decomposition structures （分解结构）
  - Uses structures （使用结构）
  - Layer structures （层状结构）
  - Class structures （类结构）
  - Data model（数据模式）
- **Component-and-connrctor structures （元件和连接器结构）**
  - Service structures
  - Concurrency structures （并发结构）
- **Allocation structures （分配结构）**
  - Deployment structures （部署结构）
  - Implentation structures （实施结构）
  - Work assignmennt structures （工作分配结构）

具体每一种结构看PPT

架构模式

Module type pattern：Layered pattern

C&C type pattern：

1. Shared-data pattern
2. Client-server pattern

Allocation pattern：

1. Muti-tier pattern
2. Competence center and platform

什么样的架构算好的架构

1. 只有一个人说话算
2. 质量属性重要，功能次要
3. 文档很重要
4. 要针对质量属性对架构尽早地评价
5. 增量开发
6. 封装、面向接口开发
7. 质量属性应该由相应的架构模式实现
8. 架构不应该依赖于特定商业产品的版本
9. 数据的产生端和消费端通常只有一端会发生变化，因此他们要分离
10. modules和components不可能一一对应

11. 进程应该能在任何硬件上运行
12. 组件之间交互的方式要少
13. 架构要关注特定的资源类型，网络负载、性能

具体详见PPT

质量属性

系统需求

系统需求有多种形式：文本需求、模型、现有系统、用例、用户故事等等。

所有要求都包含以下类别：

1. **功能要求** （Functional）
2. **质量属性要求**（Quality attribute）
3. **约束** （Constraint）

质量属性

Stimulus（刺激）

Stimulus Source（刺激源）

Response

Response Measure

Environment

Atrifact

具体详见PPT

Tatics vs Tradeoffs（权衡）

略

指导质量设计决策

有七种 design decision

1. Allocation of responsibilities （职责分配）
2. Coordination model（协调）
3. Data model
4. Management of resources
5. Mapping among architectural elements （架构之间元素的映射）
6. Binding time decisions
7. Choice of technolog （技术选择）

具体详见PPT

架构决策和模式

架构模式都是经过验证的良好的设计结构的方法

新的模式会依据环境条件自发地出现

通常模式中会包含很多种决策（tactics）

架构模式

注意三点

1. context
2. problem
3. solution

Layered Pattern

分层模式

提高可移植性、可修改性、重用性

缺点是增加前期成本和复杂性 导致系统性能缺失

Broker(经纪人的、代理人) Pattern

优点懒得写了

缺点是增加了复杂性 通信延迟

MVC Pattern

缺点是并不适用所有的情况

Pip-and-Filter Pattern

缺点是对交互系统不友好，增加大量计算开销

Client-Server Pattern

客户端通过请求提供一组服务的服务器的服务进行交互。 n 某些组件可能同时充当客户端和服务器。 n 可能有一个中央服务器或多个分布式服务器。 n 客户端-服务器模式的连接器类型是由用于调用服务的请求/回复协议驱动的数据连接器。

服务器可能是性能瓶颈，也可能是单点故障。 n 关于在何处定位功能（在客户端或在服务器中）的决定通常很复杂，并且在系统构建后更改成本高昂 。

Peer-to-Peer Pattern 点对点模式

看PTT

Service-Oriented Architecture Pattern

看PPt

Publish-Subscribe Pattern

看PPT

Shared-Data Pattern

看PPT

Map-Reduce Pattern、Muti-tier Pattern

这两个pattern都是Allocation pattern，上面的除了第一个都是C&C

###

设计一个架构

架构设计方法的三个关键思想：

1. 分解（decomposition）
2. 根据架构上的重要需求进行设计
3. 生成和测试

Creating the Initial Hypothesis 创建初始假设

1. Existing systems
2. Frameworks.
3. Patterns and tactics
4. Domain decompositi 领域分解
5. Design checklists 设计清单
6. Choosing the Test
7. Generating the Next Hypothesis

ADD 属性驱动设计方法

分5步

1. Choose an element of the system to design. 选择系统的一个元素进行设计
2. Identify the ASRs for the chosen element. 确定所选元素的 ASR
3. Generate a design solution for the chosen element. 为所选元素生成设计解决方案。
4. Inventory (盘点) remaining requirements and select the input for the next iteration. 库存（盘点）剩余需求并选择下一次迭代的输入
5. Repeat steps 1–4 until all the ASRs have been satisfied 重复步骤 1-4，直到满足所有 ASR。

架构评估

分析是架构评估的核心，这是确定架构是否适合其预期目的的过程。

有成熟的评估架构的方法，这些方法使用了我们已经学到的许多概念和技术。

评价因素

评估通常采用以下三种形式之一：

1. 在设计过程中由设计师进行的评估
2. 在设计过程中由同行进行的评估
3. 架构设计完成后由外部人员进行分析

架构权衡分析方法 （The Architecture Tradeoff Analysis Meth） （ATAM）

架构权衡分析方法 n 架构权衡分析方法 (ATAM) 十多年来一直被用于评估从汽车 (汽车) 到金融再到国防 (国防) 领域的软件架构。 n ATAM 的设计使评估者无需熟悉架构或其业务目标，无需构建系统，并且可能存在大量利益相关者。

评估阶段的步骤

Step 1: Present the ATAM

Step 2: Present the Business Drivers.

Step 3: Present the Architecture. 第 3 步：展示架构。

Step 4: Identify Architectural Approaches 第 4 步：确定架构方法

Step 5: Generate Quality Attribute Utility Tree. 步骤 5：生成质量属性效用树。

Step 6: Analyze Architectural Approaches. 第 6 步：分析架构方法。

Step 7: Brainstorm and Prioritize Scenarios. 第 7 步：集思广益并确定场景的优先级。

Step 8: Analyze Architectural Approaches. 第 8 步：分析架构方法。

Step 9: Present Results

每一步干什么还是自己看PPT吧