选择题

All of the following roles are stakeholder except:

**三类质量属性：**

系统的质量属性（可用性，可修改性，性能，安全性，可测试性，易用性）

受构架影响的商业属性（上市时间、成本和收益、所希望的系统生命期的长短、目标市场推出计划、与老系统的集成）

与构架本身相关的一些质量属性（概念完整性、正确性和完整性、可构建性）

All of the following are important system attributes except: attractivity

可用性，可修改性，性能，安全性，可测试性，易用性

Which of the following are elements of a SA: All

Which of the following is not a precondition for architectural review? finished design

**构架的定义：**某个软件或计算机系统的软件构架是该系统的一个或多个结构，它们由软件元素、这些元素的外部可见属性以及这些元素之间的关系组成。

**ATAM的结果（输出）：**

1. 一个简洁的构架表述
2. 表述起初的业务目标
3. 用场景集合捕获的质量需求
4. 构架决策到质量需求的映射
5. 所确定的敏感点和权衡点集合
6. 有风险决策和无风险决策
7. 风险主题的集合

简答题

**What is The Architecture Business Cycle (ABC)?**

ABC is a cycle of influences from the environment to the architecture and back to the environment

架构商业周期是一种相互影响的周期，从环境到架构又返回到环境。

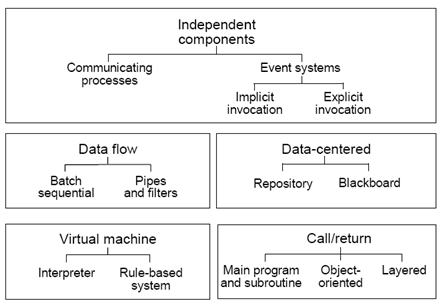
**List 5 architecture patterns/styles.**

1. Data Flow Style数据流

Three example of Data Flow Style:

Batch Sequential批处理、Pipe-and-Filter管道、Process Control进程控制

1. Call/Return Style调用/返回
2. Independent components style独立组件
3. Data-centered style数据中心
4. Virtual machine Style虚拟机



**How is an architectural pattern/style determined?**

a set of component types

一组组件类型

a set of connector types/interaction mechanisms ['mekənizəm] 交互机制

一组连接件类型/交互机制

a topological layout of these components [təu'pɔlədʒi, tɔ-]

这些组件的拓扑分布

a set of constraints on topology and behavior

一组对拓扑和行为的约束

an informal description of the costs and benefits of the style

一些对风格的成本和效益的非正式的描述

**Bass et al's classify all architecture structures into 3 main categories, what are them?**

Model-based模块结构：此处的元素是模块，它们是实现单元。

(Decomposition分解, Users-Layered分层, Class类)

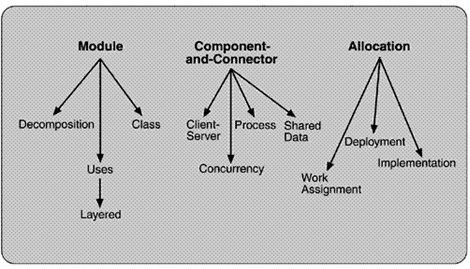
Component-and-connector组件-连接器：此处的元素为运行时组件和连接器

(Client-Server客户机-服务器, Concurrency并发, Process进程, Shared Data共享数据)

Allocation分配：分配结构展示了软件元素和创建并执行软件的一个或多个外部环境中的元素之间的关系。

(Work Assignment工作分配, Deployment部署, Implementation实现)

**List 5 architecture structures according to Bass et al.**



**What is a quality attribute scenario?** [si'nɑ:riəu]

a means to characterize system quality attributes, consists of 6 parts:

1. stimulus source刺激源 ['stimjuləs]
2. stimulus刺激
3. environment环境
4. artifacts affected制品
5. system response系统响应
6. measurement of response响应度量

A unified way to express quality requirements

**Discuss the benefits of architectural reviews（评审）.**

1. financial 节省成本
2. forces preparation for review 推动为评审做准备
3. captured rationale 获取基本原理
4. early detection of problems 尽早地发现问题
5. validation of requirements 确认需求
6. improved architecture 提高体系结构质量

**When can architectural reviews begin?**

早期的评审：

1. 需求分析之后，体系结构还没有确定之前进行
2. 用来理解需求在体系结构方面的隐含内容
3. 检查需求的可行性
4. 为质量目标排序

全面的评审：

1. 体系结构文档可用时进行
2. 评价被评体系结构的质量

**ATAM的目标：**按照质量需求，评价体系结构设计

1. 我们需要一个新方法，让我们能尽早提出正确问题
2. ATAM的目标不是做精确的分析，而是发现体系结构可能带来的风险
3. 我们要发现一些趋势，从体系结构方案语言系统的特性
4. 发现风险，然后做进一步的分析、设计
5. 明显的折中可以被清晰地指出并写入文档

**What is an unplanned architectural review? why should the organization have it?**

答：计划外评估是未曾预料到的，通常是因为项目存在严重的问题，需要采取极端的措施来补救以前的工作。对于项目成员来说，计划外评估更像是一种折磨，因为项目的资源和时间本来就已经很紧张了，但还要抽出资源和时间来进行评估，因此只有当管理层认为项目很有可能会失败，需要在开发过程中进行纠正时，才会进行计划外评估，计划外评估是反应性的，会使项目成员感到非常紧张。

**What is brainstorming?**

头脑风暴是指一群人（或小组）围绕一个特定的兴趣或领域，进行创新或改善，产生新点子，提出新办法。它是一种激发集体智慧产生和提出创新设想的思维方法。

**Explain risk points in architectural decision.**

While risks are potentially problematic architectural decisions…  
风险是有潜在问题的体系结构

Discover risks - alternatives that might create future problems in some quality attribute

发现风险：可能在将来产生质量问题的方案

Example risks

Rules for writing business logic tier of your 3-tier style are not clearly articulated.  
三层架构下，商业逻辑层的规则还没有确定

There is no way of detecting the “live” failure of a critical component.  
没有检测一个关键组件是否正常工作的机制

**Explain non-risk points in architectural decision.**

Non-risks are good decisions relying on implicit assumptions.

在一个可信的假设之下的，非风险是好的方案

Discover non-risks – decisions that promote qualities that help realize business/mission goals

发现非风险：可以提高商业质量决策

Example non-risk

Assuming message arrival rates of once per second，a processing time of less than 30 ms, and the existence of one higher priority process, a 1 second soft deadline seems reasonable.  
假定消息的到达速率是每秒一次，一次处理的时间小于30ms。如果对一个更高优先级的处理的响应时间要求是1秒钟，此系统可行，不会漏掉消息

**Explain sensitivity points in architectural decision.**

Sensitivity points are candidate risks and candidate tradeoff points.  
敏感点是候选的风险和折中

Discover sensitivity points – alternatives for which a slight change makes a significant difference in some quality attribute

发现敏感点：方案中的一个小小变化，就可能让质量完全大变样

Example Sensitivity

Changing the timing scheme from a harmonic framework to a non-harmonic framework would be easy, but due to implied timing dependencies, there would impact far reaching impacts to other modules.  
把定时方法从一个调波的框架移植到一个非调波的框架可能很容易，但是因为各个模块对定时的依赖，可能会极大地影响它们的正常工作

**Explain trade-off points in architectural decision.**

与多个质量属性相关的架构决策

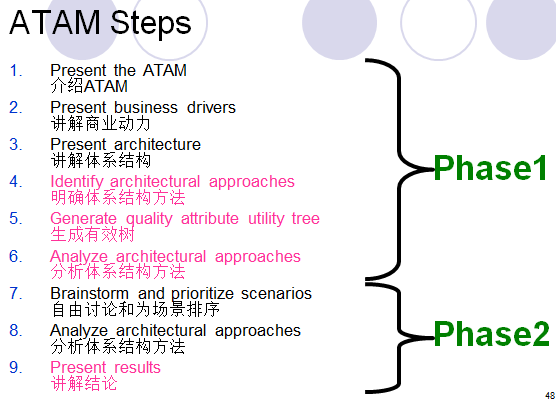
Discover tradeoffs – decisions affecting more than one quality attribute

发现折中：影响一个以上质量的决策

Example Tradeoffs

In order to achieve the required level of performance in the discrete event generation component, assembly language had to be used thereby reducing the portability of this component.  
为了达到性能要求，不得不在离散的事件产生组件中使用汇编语言。此组件不再有移植性

**List ATAM steps：**



**What is a utility tree?**

A utility tree is top-down vehicle for characterizing the “driving” attribute-specific requirements

效用树是一个自顶向下的工具，用来刻画重要的需求

**What is a product line?**

A product line is a group of sharing a common, managed set of features that satisfy specific needs of a selected market, and that are developed from a common set of core assets in a prescribed way.

生产线是一组共享共用的、被合理组织的特性的产品，这些特性满足某个市场的特定需求。

**From a product line, how products can be produced?**

从公共核心资产库中选取合适的构件

使用预定义的变化性机制进行裁剪

必要时增加新的构件

在整个产品线范围内共同的体系结构的指导下，进行构件的组装，形成系统

论述题

**How do the environment influences designs?**

Stakeholder with different, sometimes conflicting concerns;

不同的利益相关者有时候会发生冲突，影响了构架的设计

Business Environment and Organizational Structure (organizational structure, resources, assets, business strategies);

商业环境和开发组织的影响（组织结构、资源、财产、商业策略）

The Architect's Background and Experience

设计师的背景和经验

**How do Architecture Design Change the Business Environment?**

They affect the structure of the organization.

架构影响着开发组织的结构

They affect the business goals of the organization.

架构影响着开发组织的目标

They affect customer requirements for new products.

架构可能会影响客户对下一个系统的要求

They affect how future architectures are designed.

架构影响着未来架构的设计

**What is availability? What tactics can be taken to used for availability?**

可用性是系统正常运行的时间比例，可理解为计算机在任意时刻正常工作的概率。一般将系统可用性定义为：

α = 平均正常工作时间/（平均正常工作时间 + 平均修复时间）

战术：错误预防、错误检测、错误恢复

**What is ADD? describe the steps to carry out ADD.**

ADD是指属性驱动的设计，是一种定义软件构架的方法，该方法将分解过程建立在软件必须满足的质量属性之上。

ADD的设计过程：它是一个递归的分解过程，其中在每个阶段都选择战术和构架模式来满足一组质量属性场景，然后对功能进行分配，以实例化由该模式所提供的模块类型。

1. 选择要分解的模块(最初是整个系统。该模块要求的输入【约束、功能需求、质量需求】都是必须可得的)
2. 根据以下不走对模块进行求精

* 从具体的质量场景和功能需求集合中选择构架驱动因素
* 选择满足构架驱动因素的构架模式
* 实例化模块并用多个视图分配功能
* 定义子模块接口
* 验证用例和质量场景并对其进行求精，使他们成为子模块的限制

1. 对需要进一步分解的每个模块重复以上步骤

**How to create a skeletal system（骨架系统）?**

1. 实现处理构架组建的执行和交互的软件部分
2. 选择把提供功能的那些元素添加到系统中
3. 选择了提供下一个功能增量元素后，采用使用结构，获知应该在系统中采用什么软件来支持该功能
4. 随着该过程的继续，系统的增量越来越大，知道软件开发完毕

**What is the benefits of a product line?**

1. Defect reduction: defect fixed in one product automatically fixed for all future products

减少缺陷：在一个产品中修正的缺陷，自动在未来的产品中也被修正

1. Performance: performance issues addressed for all products

性能：性能问题在所有产品中都被解决

1. Planning: more accurate because all products are produced in the same way

计划：更准确，因为所有的产品都用相同的方式生产

1. Reduction in time to market and staffing

减少产品上市时间和减少人员编制