选择题
All of the following roles are stakeholder except:

All of the following are important system attributes except: attractivity

Which of the following are elements of a SA:
All

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

ATAM outputs include:

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
1.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。

2.List 5 architecture patterns/styles.
unbileveable answer

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

reliable :数据流过程调用事件驱动风格信息共享风格层次风格
3.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
一组连接件类型/交互机制
a topological layout of these components 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

4.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)
Module-based structures: the elements are software modules the implementation units. Includes decomposition , uses , layered , class. Component-and-connector structures: the elements are run-time components. Includes process communication, concurrency parallelism ,shared data production and consumption, and client-server communication.
Allocation Strutures:
5.List 5 architecture structures according to Bass et al.
Decomposition Users Layered class client-server ProcessConcurrencyShared dataDeployment Implement work assignment
6.What is a quality attribute scenario?
a means to characterize system quality attributes, consists of 6 parts:
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

7.Discuss the benefits of architectural reviews.

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

8.When can architectural reviews begin?

the review early “architecture discovery review” is done after requirements are set but before the architecture is firm

需求分析之后体系结构还没有确定之前进行

is used to understand implications of requirements on architecture

用来理解需求在体系结构方面的隐含内容

checks for requirements feasibility

检查需求的可行性

prioritizes architectural goals

为质量目标排序

full architectural review全面的评审：

is done when architectural documentation is available

体系结构文档可用时进行

is used to evaluate qualities of proposed architecture

评价被评体系结构的质量

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

usually occurs when project is in trouble

通常在项目出现问题时采用

often devolves into finger-pointing 导致互相责备

can be painful for project already struggling

对于已经苦苦挣扎了很久的项目而言有些痛苦

An Unplanned evaluation is unexpected and usually the result of a project in serious trouble and taking extreme measures to try to salvage previous effort

other version(ca):

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。

10.What is brainstorming?
头脑风暴的特点是让与会者敞开思想使各种设想在相互碰撞中激起脑海的创造性风暴其可分为直接头脑风暴和质疑头脑风暴法。
other version:
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。

11.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

12.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

13. Explain sensitivity points in architectural decision.
Explain trade-off 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

14. List ATAM steps.

Present the ATAM

介绍ATAM

Present business drivers

讲解商业动力

Present architecture

讲解体系结构
Identify architectural approaches
明确体系结构方法
Generate quality attribute utility tree
生成有效树
Analyze architectural approaches
分析体系结构方法
Brainstorm and prioritize scenarios
自由讨论和为场景排序
Analyze architectural approaches
分析体系结构方法
Present results
讲解结论
15.What is a utility tree?
有效树是一个自顶向下的工具用来刻画重要的需求
16.What is a product line?
A product line is a group of products sharing a common managed set of features that satisfy specific needs of a selected market. 生产线是一组共享公用的、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
17.From a product line, how products can be produced?
Product lines built using a common set of assets epitomize strategic planned reuse.
生产线使用公用的资源构成战略性的、有计划的重用
Individual systems products are modifications of a hypothetical central core system.
每个独立的系统产品修改自一个假想的核心系统
Products are produced by tailoring the architecture 剪裁体系结构 instantiating tailorable components 实例化可剪裁的组件
generating rather than building 产生而不是建造
emphasizing integration not coding 注重集成而不是编码

论述题
1.How do the environment influences SA designs?
Stakeholder with different, sometimes conflicting concerns;
Business Environment and Organizational Structure (organizational structure, resources, assets, business strategies);
The Architect's Background and Experience

2.How do Architecture Design Change the Business Environment?
They affect the structure of the organization.
They affect the business goals of the organization.
They affect customer requirements for new products.
They affect how future architectures are designed.

3.What is availability? what tactics can be taken to used for availability?
是系统正常运行的时间比例。一般将系统可用性定义为

战术:设计决策或策略实现质量属性所采用的方法或解决方案。
可用性战术用于以下几方面
错误检测 Fault Detection
错误恢复 Fault Recovery
错误预防 Fault Prevention
可用性战术阻止错误发展成故障或者把错误的影响限制在一定范围内从而使修复成为可能
4.What is ADD? describe the steps to carry out ADD.
属性驱动的设计将分解过程建立在软件必须满足的质量属性之上
1选择要分解的模块
2根据以下步骤对模块进行求精。
 A. 从具体的质量场景和功能需求集合中选择构架驱动因素。最初是整个系统。该模块要求的所有输入约束、功能需求、质量需求都必须是可获得的
B. 选择满足构架驱动因素的构架模式。
C. 实例化模块并根据用例分配功能使用多个视图进行表示
D. 定义子模块的接口。该分解提供了模块和对模块交互类型的限制。对于每个模块将该信息编写在接口文档中。 E. 验证用例和质量场景并对其进行求精使它们成为子模块的限制。
 3对需要进一步分解的每个模块重复上述步骤
5.How to create a skeletal system?
对构架进行了充分设计后就可以创建骨架系统。首先实现处理构架组件的执行和交互的软件部分。包括实时系统中的调度程序实现规则引擎带有规则的原型集以控制在基于规则的系统中规则的激发实现多进程系统中的进程同步机制或客户/服务器系统中的客户机/服务器的协同。选择把提供功能的哪些元素添加到系统中。可能根据以下因素首先处理问题最多的部分来降低风险或现有开发人员的类型和水平或尽可能快地将有用的产品推向市场。选择了提供下一个功能增量的元素后就可以采用使用结构以获知应该在系统中采用什么软件来支持该功能。随着该过程的继续系统的增量越来越大直到软件开发完毕。任意时刻集成和测试任务都不会很多在每个增量中都很容易找到最近引入的错误源.
6.What is the benefits of a product line?
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 Performance: performance issues addressed for all products e.g. schedulability deadlock distributed system issues
性能性能问题在所有产品中都被解决
 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
Reduction in time to market 减少产品推向市场的时间
Staffing 减少员工

$$\alpha = \frac{\text{平均正常工作时间}}{(\text{平均正常工作时间} + \text{平均修复时间})}$$