

简答预测

按照18-19推测共6道简答题，分值分布 ‘7*5+10=45’ 可能的考点如下：

- 1. Software Architectural 定义 + 一相关点
- 2. Architectural Style 定义 + 一相关点
- 3. Style相关（大概率在 With/out dispatcher / Virtual machine两张图区别 / B/PF 对比 / Data-centered 的两种）
- 4. SPD 定义相关
- 5. QA + 策略
- 6. ATAM部分
- 7. 两个style定义+对比+优缺点（10分）（大概率Batch/PF / Data-center）

1. What is the definition of Software Architecture? Why need we do architecture design? List at least three characteristics of good architecture? What are the values of software architecture?

The software architecture of a program or computing system is the structure or structures of the system, which comprise **software elements**, the externally **visible properties** of those elements, and the **relationships** among them.

An architecture include: component, connector and constraints and it is the result of a set of busiess and technical decisions

Why:

原文	中文翻译
Architecture is the vehicle for stakeholder communication .	架构是利益相关者沟通的媒介。
Architecture manifests the earliest set of design decisions .	架构体现了最早期的一组设计决策。
The Architecture is used as a transfable and re-used model	架构是可复用的模型

Characteristics of a Good Architecture:

原文	中文翻译
Resilient	有弹性
Simple	简洁
Approachable	易于理解
Clear separation of concerns	关注点分离明确
Balanced distribution of responsibilities	职责分配合理
Balances economic and technology constraints	平衡经济与技术约束

Values: (两个部分各选两条)

Organizational Value

原文	中文翻译
Communicate inside organization, and between customers and vendors	支持组织内部与客户/供应商间沟通
Provide the high-level information of systems	提供系统高层信息
Costs and risks evaluating	评估成本与风险
Work allocation and project schedule	分配工作与安排进度

Technical Value

原文	中文翻译
Meet system requirements and objectives	满足系统需求与目标
Specify the constraints of detailed design , construction and testing	明确详细设计、构建与测试的约束
Enable flexible distribution/partitioning of the system	支持系统灵活部署或划分
Reduce cost of maintenance and evolution	降低系统维护与演进成本
Increase reuse and integrate with legacy and third party software	提高复用，便于集成旧系统与第三方

2. What is the definition of Architectural Style? List at least three advantages of Architectural Style

The software architectural style describes the idiomatic paradigm of the organization of software system families in a specific domain, reflects the structural and semantic characteristics shared by many systems in the domain, and guides how to effectively organize various modules and subsystems into a complete system.

Advantages:

- Promote reusability;
- Promote communication among developers;
- Diminish the risks of SA design;
- Improves development efficiency and productivity.

3.Describe the event system style with a dispatcher versus the event system style without a dispatcher. (或者是 All broadcasts vs. Selected broadcast)

类型	原文	中文翻译
Without Dispatcher	Each module allows other modules to declare interest in events. When it sends an event, only interested modules receive it. This is the Observer/Observable style.	无调度器模块：模块之间通过注册事件形成观察者模式，仅向注册模块发送事件。
With Dispatcher	Dispatcher module receives all events and dispatches them using: - All broadcasting: sends to all modules - Selected broadcasting: sends to subscribed modules only	有调度器模块：事件由调度器转发，可选择全广播或选择性广播。

类型	描述（原文）	中文翻译
All broadcasting	The dispatch module broadcasts the event to all modules, but only interested modules fetch the event and trigger their own behavior.	向所有模块广播事件，只有感兴趣的模块会响应。
Selected broadcasting	The dispatch module sends events to those registered modules.	仅将事件发送给已注册模块。

4.What are the types of data-centered styles? Please describe briefly.

类型	原文	中文翻译
Database	The types of transactions in an input stream trigger selection of process to execute.	输入流中的事务类型触发相应过程的执行（数据库驱动方式）。
Blackboard	The current state of the central data structure is the main trigger for selecting processes to execute.	中心数据结构的当前状态触发选择过程（黑板机制）。

5.What is software architecture description? List at least three descriptive principles.

- Software architecture consists of a certain form of structural elements, i.e., a collection of components. Including processing components, data components and connection components.
- Software architecture is the blueprint of the developing system.

Principles:

1. Write from the reader's perspective.
2. Avoid unnecessary duplication.
3. Avoid ambiguity.
4. Use standard organizational structures.
5. Record the reasons.
6. Keep documents current but not updated frequently
7. Review whether the documents meet the requirements

6.Explain the meaning of Usability and briefly provide at least THREE solutions to improve Usability. (六个都一样，策略要写小点)

Answer: Usability means to reduce the difficulty for users to use the software.

Solutions:

Runtime Tactics

- System Anticipates User Tasks
- System Provides Appropriate Feedback to Users
- System Provides Consistent Experience to Users

Design-time Tactics

- Support Undo Operations
- Isolate the User Interface from Other Parts of the System

7.Why we evaluate architecture? What is ATAM? What are the phases of ATAM? What is utility tree? What are the key concepts of ATAM? (力竭了这儿一点不会，自己对着ppt找一下吧)

Why:

Early detection of problems with the existing architecture

Validation of requirements

Forced preparation for the review

Improved architectures

What:

The ATAM is a method that helps stakeholders ask the right questions to discover potentially problematic architectural decisions.

Utility Tree:

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

Key concepts:

A sensitivity point is a property of one or more components (and/or component relationships) that is critical for achieving a particular quality attribute response.

A tradeoff point is a property that affects more than one attribute and is a sensitivity point for more than one attribute.

A risk is a potentially problematic architectural decision.

Non-risks are good architectural decisions that are deemed safe upon analysis.

8.What is Batch architecture? What is Pipe-and-Filte architecture? Compare them and write their advantages and disadvantages.

Answer:

Batch: Computation is performed in separate steps; the output of one step is used as input for the next.

Advantages: - Clear modular separation of processing steps. - Easy to audit and debug using intermediate files.

Disadvantages: - No support for interactive or real-time processing. - High latency between steps.

FP: A pipe-and-filter architecture consists of components called filters and connectors called pipes. Each filter processes its input data and generates output that is passed via pipes to subsequent filters.

Advantages:

- High cohesion
- Low coupling.
- Reusability
- Simple to implement
- Extendibility and flexibility

DisAdvantages:

- Not suitable for interactive applications.
- High overhead from parsing/format conversion.
- Low efficiency when large amounts of shared data are needed.

Campare:

<i>Batch Sequential</i>	<i>Pipe-and-Filter</i>
<i>total</i> (整体传递数据)	<i>incremental</i> (增量)
<i>coarse grained</i> (构件粒度较大)	<i>fine grained</i> (构件粒度较小)
<i>high latency</i> (延迟高，实时性差)	<i>results starts processing</i> (实时性好)
<i>no concurrency</i> (无并发)	<i>concurrency possible</i> (可并发)