

系统构架设计时应考虑的一些必要性因素[4] PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/141/2021\\_2022\\_\\_E7\\_B3\\_BB\\_E7\\_BB\\_9F\\_E6\\_9E\\_84\\_E6\\_c29\\_141402.htm](https://www.100test.com/kao_ti2020/141/2021_2022__E7_B3_BB_E7_BB_9F_E6_9E_84_E6_c29_141402.htm)

2、可维护性 便于在系统出现故障时及时方便地找到产生故障的原因和源代码位置，并能方便地进行局部修改、切割；（可维护性与运行可管理性不同）3、可扩充性：系统方案的升级、扩容、扩充性能 系统在建成后会有一段很长的运行周期，在该周期内，应用在不断增长，应用的层次在不断升级，因此采用的构架设计等方案因充分考虑升级、扩容、扩充的可行性和便利。

4、可移植性 不同客户端、应用服务器、数据库管理系统：如果潜在的客户使用的客户端可能使用不同的操作系统或浏览器，其可移植性必须考虑客户端程序的可移植性，或尽量不使业务逻辑放在客户端；数据处理的业务逻辑放在数据库管理系统中会有较好的性能，但如果客户群中不能确定使用的是同一种数据库管理系统，则业务逻辑就不能数据库管理系统中；达到可移植性一定要注重标准化和开放性：只有广泛采用遵循国际标准，开发出开放性强的产品，才可以保证各种类型的系统的充分互联，从而使产品更具有市场竞争力，也为未来的系统移植和升级扩展提供了基础。5、需求的符合性 从源代码的组织结构看需求的符合型主要考虑针对用户需求可能的变化的软件代码及构架的最小冗余（同时又要使得系统具有一定的可扩展性）。五、写系统构架设计文档应考虑的问题 构架工作应该在需求开发完成约80%的时候开始进行，不必等到需求开发全部完成，需要项目经理以具体的判断来评估此时是否足以开始构建软件构架。给出一致的

轮廓：系统概述。一个系统构架需要现有概括的描述，开发人员才能从上千个细节甚至数十个模块或对象类中建立一致的轮廓。构架的目标应该能够清楚说明系统概念，构架应尽可能简化，最好的构架文件应该简单、简短，清晰而不杂乱，解决方案自然。构架应先定义上层的主要子系统，应该描述各子系统的任务，并提供每个子系统中各模块或对象类的初步列表。构架应该描述不同子系统间相互通信的方式，而一个良好的构架应该将子系统间的通信关系降到最低。成功构架的一个重要特色，在于标明最可能变更的领域，应当列出程序中最可能变更的部分，说明构架的其他部分如何应变。

复用分析、外购：缩短软件开发周期、降低成本的有效方案未必是自行开发软件，可以对现有软件进行复用或进行外购。应考虑其对构架的影响。除了系统组织的问题，构架应重点考虑对于细节全面影响的设计决策，深入这些决策领域：外部软件接口（兼容性、通信方式、传递数据结构）、用户接口（用户接口和系统层次划分）、数据库组织和内容、非数据库信息、关键算法、内存管理（配置策略）、并行性、安全性、可移植性、网络多人操作、错误处理。要保证需求的可追踪性，即保证每个需求功能都有相应模块去实现。构架不能只依据静态的系统目标来设计，也应当考虑动态的开发过程，如人力资源的情况，进度要求的情况，开发环境的满足情况。构架必须支持阶段性规划，应该能够提供阶段性规划中如何开发与完成的方式。不应该依赖无法独立运行的子系统构架。将系统各部分的、依赖关系找出来，形成一套开发计划。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)