

卡内基梅隆大学软件研究所关于软件架构的定义[2] 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__E5_8D_A1_E5_86_85_E5_9F_BA_E6_c29_141463.htm 软件与人类的关系是架构师必须面对的核心问题，也是自从软件进入历史舞台之后就出现的问题。与此类似地，自从有了建筑以来，建筑与人类的关系就一直是建筑设计师必须面对的核心问题。英国首相丘吉尔说，我们构造建筑物，然后建筑物构造我们（We shape our buildings, and afterwards our buildings shape us）。英国下议院的会议厅较狭窄，无法使所有的下议院议员面向同一个方向入座，而必须分成两侧入座。丘吉尔认为，议员们入座的时候自然会选择与自己政见相同的人同时入座，而这也就是英国政党制的起源。Party这个词的原意就是"方"、"面"。政党起源的关键就是建筑物对人的影响。来源

：www.examda.com 在软件设计界曾经有很多人认为功能是最为重要的，形式必须服从功能。与此类似地，在建筑学界，现代主义建筑流派的开创人之一Louis Sullivan也认为形式应当服从于功能（Forms follows function）。几乎所有的软件设计理念都可以在浩如烟海的建筑学历史中找到更为遥远的历史回响。最为著名的，当然就是模式理论和XP理论。架构的目标是什么 正如同软件本身有其要达到的目标一样，架构设计要达到的目标是什么呢？一般而言，软件架构设计要达到如下的目标：来源：www.examda.com 可靠性（Reliable）。软件系统对于用户的商业经营和管理来说极为重要，因此软件系统必须非常可靠。安全行（Secure）。软件系统所承担的交易商业价值极高，系统的安全性非常重要。可扩展性

(Scalable)。软件必须能够在用户的使用率、用户的数目增加很快的情况下，保持合理的性能。只有这样，才能适应用户的市场扩展得可能性。可定制化 (Customizable)。同样的一套软件，可以根据客户群的不同和市场需求的变化进行调整。可扩展性 (Extensible)。在新技术出现的时候，一个软件系统应当允许导入新技术，从而对现有系统进行功能和性能的扩展。可维护性 (Maintainable)。软件系统的维护包括两方面，一是排除现有的错误，二是将新的软件需求反映到现有系统中去。一个易于维护的系统可以有效地降低技术支持的花费。客户体验 (Customer Experience)。软件系统必须易于使用。市场时机 (Time to Market)。软件用户要面临同业竞争，软件提供商也要面临同业竞争。以最快的速度争夺市场先机非常重要。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