一个好的架构能确保软件满足诸如性能、可移植性、可扩展性和可互操作性等关键需求。人们逐渐认识到一个合理的架构是系统设计、开发成功的关键。当前软件体系结构的研究有了很大进展，一个工程的软件架构的基本轮廓逐渐变得清晰，但仍然面临许多挑战和未知。

过去，体系结构的描述大多依赖于非形式化的箱线图，软件体系结构北非一个正式环节，人们逐渐意识到需要一个更严格的方法，于是号召架构设计作为一个需要关注的领域，进而出现了软件架构师。行业有两股潮流，一是认识到在构建复杂软件系统时对方法、技术、模式的共享。设计师用抽象来使整个复杂系统便于理解。第二是对在特定领域利用共性来提供一系列产品的重用框架的关注。这种想法基于相关系统的共同方面可以被提取出来，通过实例化共享的设计来使每个新系统的成本相对较低。

如今，软件开发中体系结构设计越来越重要和明确化，体系结构设计的技术基础已经大为改善，三个重要进步是：（1）体系结构描述语言和工具的发展：非形式化的图不能用于一致性、完整性或正确性分析。对此，人们提出用形式化的符号（正规的表示法）来表示和分析体系结构设计，就是常说的体系结构描述语言（ADLs），典型的有UniCon，Darwin，Rapide，C2，SADL，Wright。（2）产品流水线生产：一个产品线代表一组具有公共的系统需求集的软件系统，将可重用构件与系统独有的部分集成而得到。（3）架构标准的产生：体系结构设计的书籍的出版及有关课程的开设。这些书籍和课程的一个共同主题是使用标准体系结构风格。

软件体系结构如今有了更坚实的基础，但它尚未作为一门学科在整个软件行业普及。该领域的自然发展将导致稳定的进步。网络为中心的计算对软件体系结构有很大影响，在网络无所不在的环境中，系统可能不会集中控制，这样一组新的软件体系结构挑战出现。首先是架构的需要，适应网络的大小和变化。虽然许多相同的架构模式可能会运用，其实现的细节和规范将需要改变。