软件体系结构是具有一定形式的结构化元素，即构件的集合，包括处理构件、数据构件和连接构件。处理构件负责对数据进行加工，数据构件是被加工的信息，连接构件把体系结构的不同部分组合连接起来。

软件体系结构是软件设计过程中的一个层次，这一层次超越计算过程中的算法设计和数据结构设计。体系结构问题包括总体组织和全局控制、通讯协议、同步、数据存取，给设计元素分配特定功能，设计元素的组织，规模和性能，在各设计方案间进行选择等。软件体系结构处理算法与数据结构之上关于整体系统结构设计和描述方面的一些问题，如全局组织和全局控制结构、关于通讯、同步与数据存取的协议，设计构件功能定义，物理分布与合成，设计方案的选择、评估与实现等。

软件体系结构有四个角度，它们从不同方面对系统进行描述：概念角度描述系统的主要构件及它们之间的关系；模块角度包含功能分解与层次结构；运行角度描述了一个系统的动态结构；代码角度描述了各种代码和库函数在开发环境中的组织。

经典的软件体系结构风格之一是管道-过滤器结构，过滤器处理数据流，一个过滤器封装了一个处理步骤。管道连接一个源和一个目的过滤器。每个过滤器都有一组输入集和输出集。过滤器从管道中读入数据流，对输入流进行内部转换和增量计算（丰富，精炼，转换，融合，分解），然后产生输出数据流并写入管道中。

每个过滤器必须是一个独立的实体：过滤器之间无需共享状态，即filter无需知道其输入管道和输出管道所连接的其他过滤器的存在，更不必关注相邻过滤器的实现细节。他仅仅需要对输入数据流进行特定的内部転换和增量计算，筛选出合适的数据。

数据到来是便被处理，不是收集然后处理，即在输入被完全消费之前，输出便产生了。

管道是将数据从一个过滤器的输出端移动到另一个过滤器的输入端，是一个单向流。\*\*不同的管道中流动的数据流，可能具有不同的数据格式。