关于软件体系结构的研究工作主要在国外展开的，国内到目前为止对于软件体系结构的研究尚处在起步阶段。软件体系结构在国内未引起人们广泛注意的原因主要有两点：

1. 软件体系结构从表面上看起来是一个老话题，似乎没有新东西。
2. 与国外相比，国内对大型和超大型复杂软件系统开发的经历相对较少，对软件危机的灾难性体会没有国外深刻，因而对软件体系结构研究的重要性和必要性的认识还不很充分。
3. 应用现状
4. 形成研究热点，仍处于非形式化水平

自20世纪90年代后期以来，软件体系结构的研究成为一个热点。广大软件工作者已经认识到软件体系结构研究的重大意义和它对软件系统设计开发的重要性，开展了很多研究和实践工作。

从软件体系结构研究的现状来看，当前的研究和对软件体系结构的描述，在很大程度上来说还停留在非形式化的基础上。软件构架师仍然缺乏必要的工具，这种工具应该是显式描述的、有独立性的形式化工具。

在目前通用的软件开发方法中，其描述通常是用非形式化的图和文本，不能描述系统期望的存在于构件之间的接口，不能描述不同的组成系统的组合关系的意义。难以被开发人员理解，更不能用来分析其一致性和完整性等特性。

当一个软件系统中的构件之间几乎以一种非形式化的方法描述时，系统的重用性也会受到影响，在设计一个系统结构过程中的努力很难移植到另一个系统中去。对系统构件和连接关系的结构化假设没有得到显式的、形式化的描述时，把这样的系统构件移植到另一个系统中去将是有风险的，甚至是不可能的。

1. 软件体系结构的形式化方法研究

软件体系结构研究如果仅仅停留在非形式化的框图阶段，已经难以适应进一步发展的需要。为支持基于体系结构的开发，需要有形式化建模符号、体系结构说明的分析与开发工具。从软件体系结构研究的现状来看，在这一领域近来已经有不少进展，其中比较有代表性的是美国卡耐基梅隆大学的Robert J．A11en于l997年提出的Wright系统。Wright是一种结构描述语言，该语言基于一种形式化的、抽象的系统模型，为描述和分析软件体系结构和结构化方法提供了一种实用的工具。Wright主要侧重于描述系统的软件构件和连接的结构、配置和方法。它使用显式的、独立的连接模型作为交互的方式，这使得该系统可以用逻辑谓词符号系统，而不依赖特定的系统实例来描述系统的抽象行为。该系统还可以通过一组静态检查来判断系统结构规格说明的一致性和完整性。从这些特性的分析来说，Wright系统的确适用于对大型系统的描述和分析。

1. 软件体系结构的建模研究

研究软件体系结构的首要问题是如何表示软件体系结构，即如何对软件体系结构建模。根据建模的侧重点的不同，可以将软件体系结构的模型分为5种：结构模型、框架模型、动态模型、过程模型和功能模型。在这5个模型中，最常用的是结构模型和动态模型。

1. 结构模型

这是一个最直观、最普遍的建模方法。这种方法以体系结构的构件、连接件和其他概念来刻画结构，并力图通过结构来反映系统的重要语义内容，包括系统的配置、约束、隐含的假设条件、风格、性质。研究结构模型的核心是体系结构描述语言。

1. 框架模型

框架模型与结构模型类似，但它不太侧重描述结构的细节而更侧重于整体的结构。框架模型主要以一些特殊的问题为目标建立只针对和适应该问题的结构。

1. 动态模型

动态模型是对结构或框架模型的补充，研究系统的"大颗粒"的行为性质。例如，描述系统的重新配置或演化。动态可能指系统总体结构的配置、建立或拆除通信通道或计算的过程。这类系统常是激励型的。

1. 过程模型

过程模型研究构造系统的步骤和过程。因而结构是遵循某些过程脚本的结果。

1. 功能模型

该模型认为体系结构是由一组功能构件按层次组成，下层向上层提供服务。它可以看作是一种特殊的框架模型。

这5种模型各有所长，也许将5种模型有机地统一在一起，形成一个完整的模型来刻画软件体系结构更合适。例如，Kruchten在1995年提出了一个“4+1”的视角模型。“4+1”模型从5个不同的视角包括逻辑视角、过程视角、物理视角、开发视角和场景视角来描述软件体系结构。每一个视角只关心系统的一个侧面，5个视角结合在一起才能够反映系统的软件体系结构的全部内容。

1. 发展基于体系结构的软件开发模型

软件开发模型是跨越整个软件生存周期的系统开发、运行、维护所实施的全部工作和任务的结构框架，给出了软件开发活动各阶段之间的关系。目前，常见的软件开发模型大致可分为三种类型：

1. 以软件需求完全确定为前提的瀑布模型。
2. 在软件开发初始阶段只能提供基本需求时采用的渐进式开发模型,如螺旋模型等。
3. 以形式化开发方法为基础的变换模型。

所有开发方法都是要解决需求与实现之间的差距。但是，这三种类型的软件开发模型都存在这样或那样的缺陷，不能很好地支持基于软件体系结构的开发过程。因此，研究人员在发展基于体系结构的软件开发模型方面做了一定的工作。例如，为了形象地表示体系结构的生命周期，北京邮电大学的周莹新博士建立了一个软件体系结构的生命周期模型。

1. 软件产品线体系结构的研究

软件体系结构的开发是大型软件系统开发的关键环节。体系结构在软件生产线的开发中具有至关重要的作用，在这种开发生产中，基于同一个软件体系结构，可以创建具有不同功能的多个系统。在软件产品族之间共享体系结构和一组可重用的构件，可以增加软件工程和降低开发和维护成本。

一个产品线代表着一组具有公共的系统需求集的软件系统，它们都是根据基本的用户需求对标准的产品线构架进行定制，将可重用构件与系统独有的部分集成而得到的。采用软件生产线式模式进行软件生产，将产生巨型编程企业。但目前生产的软件产品族大部分是处于同一领域的。

1. 研究热点

当前，体系结构仍是一个非常新的研究领域，其概念还相当模糊。但软件体系结构作为软件工程领域中的一个组成部分，已经取得了长足的发展，受到大多数软件系统设计和研究人员的重视。

软件体系结构目前较活跃的研究方向包括：（1）软件体系结构形式基础的研究；（2）针对软件体系结构描述中特有的问题研究新的专门的高级语言；（3）建立用于度量和评价软件体系结构的模型和方法；（4）建立面向专门领域的软件体系结构范型库。（5）把软件体系结构从目前的直觉和经验状态过渡到理论。

1. 提供新的软件体系结构描述语言

在提高软件工程师对软件系统的描述和理解能力中，虽然软件体系结构描述起着重要作用，但这些抽象的描述通常是非形式化的和随意的。体系结构设计经常难以理解，难以适于进行形式化分析和模拟，缺乏相应的支持工具帮助构架师完成设计工作。为了解决这个问题，用于描述和推理的形式化语言得以发展，这些语言就叫做体系结构描述语言，ADLs寻求增加软件体系结构设计的可理解性和重用性。

ADL是这样一种语言，系统构架师可以利用它所提供的特性进行软件系统概念体系结构建模。ADL提供了具体的语法与刻画体系结构的概念框架。ADLs使得系统开发者能够很好地描述他们设计的体系结构，以便与人交流，能够用提供的工具对许多实例进行分析。

这种描述语言的目的就是提供一种规范化的体系结构描述，从而使得体系结构的自动化分析变得可能。研究人员已经设计出了近二十种ADLs，比较有影响力的有C2、UniCon、MetaH、Aesop、SADL、Rapide、Wright等。这些语言能够对体系结构连接器进行第一级抽象，同时还能描述模型的结构和内部构件之间的交互作用，并且还引入了一些新的系统分析模式。

1. 对软件体系结构的专门知识的整理

这方面的工作主要是对软件工程师在软件开发实践中得来的各种体系结构的原则、模式的整理和分类。例如，对软件体系结构风格的分类和比较，对体系结构描述语言的综合分析等。就目前看来，国内对软件体系结构的研究主要集中在对软件体系结构的专门知识的整理上。

1. 提供特定领域的体系结构框架

最近，开发特定的领域去为产品提供可重用框架日益受到关注。这些开发基于这样的想法：可以提取相关系统中的共同方面，以便可以通过低成本地把这些共同的设计实例化来构筑新系统。常见的例子有：

1. 编译器的标准分解。这个方法可以使一个本科生在一个学期时间内构造一个新的语言编译系统。
2. 标准化的通讯协议。这个可以使厂家通过在不同层次的抽象上提供服务来互相操纵。
3. 第四代语言。利用4GL开发出商务信息处理的通用范式。
4. 用户界面工具和框架。这个方法为开发者提供了一个可重用框架以及象菜单、对话框这样的可重用构件的集合。

软件体系结构充当一个理解系统构件和它们之间关系的框架，特别是那些始终跨越时间和实现的属性。这个理解对于现在系统的分析和未来系统的综合很有必要。在分析和支持下，体系结构抓住领域知识和实际的一致，促进设计的评估和构件的实施，减少仿真和构造原型。在综合的支持下，体系结构提供了建立系列产品的基础，以可预测的方式利用领域知识构造和维护模块、子系统和系统。

1. 提供软件体系结构的形式化基础

对体系结构设计的推理的形式化表示使得体系结构级的设计更好地被理解、被实现。它的目的是对体系结构设计人员在实践过程中总结出来的一些设计的经验和方法加以总结、概括，从而形成一个形式化的描述，形成一定的理论基础(以代替当前的不精确的研究)。已提出一些形式化机制，如过程代数、偏序集合、化学抽象机等。希望对系统的非功能特性如性能、可维护性等给出形式特征，同时给出软件体系结构的理论。

1. 建立评价软件体系结构的方法

通过分析来预见软件的质量，通过分析来创建、选择、评估与比较不同的体系结构。例如，Kazman等人在2000年提出的ATAM（Architectural Tradeoff Analysis Method）方法。ATAM方法不但能够揭示体系结构如何满足特定的质量需求（例如性能和可修改性），而且还提供了分析这些质量需求之间交互作用的方法。使用ATAM方法评价一个软件体系结构的目的是理解体系结构设计满足系统质量需求的结果。采用ATAM方法的步骤如图3所示，限于篇幅，在此不再详细介绍具体步骤，有兴趣的读者可查阅有关ATAM方法的文献。

1. 发展方向
2. 各种ADLs之间的信息互换

现有的ADLs大多是与领域相关的，所以不利于对不同领域体系结构的说明。但这些针对不同领域的ADLs在某些方面又大同小异，造成资源的冗余。其实，大多数ADLs具有一系列的共同概念。如何用一种公共形式把各种语言综合起来，使得能够交换各种体系结构描述信息，将是今后软件体系结构研究和实践的重点之一。

1. 设计工具和环境

软件体系结构设计既然作为软件工程的一部分，它的计算机辅助实现手段是相当重要的。我们应当开发出一些软件工具来实现体系结构的描述和分析，开发阶段转换工具，以实现阶段成果的自动转换，例如，把需求规格说明自动转换为构件等。目前关于这方面的研究成果很少，特别是可以应用到实际项目开发中的工具和环境就更少。

1. 体系结构再工程

当今软件系统的规模变得越来越大，结构也越来越复杂，同时从头开始构建的大系统数量在急剧地减少，因而很多遗留系统正在被逐步地利用。从遗留系统软件代码和系统中抽取结构信息，经过描述、统一、抽象、一般化与实例化等处理，可总结出系统的体系结构。

在这种情况下，软件再工程变得越来越重要，因为它提供了一条把遗留系统转换为可进化系统的现实可行的途径，是一种可以改进人们对软件的理解和改进软件本身的活动。这类研究的目的是为一些特定的应用领域的软件系统提供一些体系结构框架，如控制系统、移动机器人和用户接口界面等。通过这些框架可以很方便地构造一个新的软件系统。