|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| 武汉大学logo | | | | | |
|  | | | | | |
| 高级软件工程课程报告 | | | | | |
|  | 研究生姓名 | ： | 陈小龙 | |  |
| 学号 | ： | 2019282110194 | |
| 指导教师姓名、职称 | ： | 应时教授 | |
| 专业名称 | ： | 高级软件工程 | |
| **(备注：选课失败，需要应老师您手动添加姓名学号到成绩系统)** | | |
|  | | | | | |
| 二〇一九年十二月 | | | | | |

# 

# 摘要

本文在多个方面对软件工程技术的发展进行了一些讨论，包括软件工程技术发展历程、软件工程学科研究的内容、软件工业化生产模式、软件技术的发展趋势、软件体系结构评估技术。

关键词：软件工程；软件特征；软件复用；评估技术

目录

[摘要 I](#_Toc28869016)

[引言 1](#_Toc28869017)

[1软件工程技术发展历程 2](#_Toc28869018)

[2软件工程的主要研究内容 2](#_Toc28869019)

[3软件技术的发展趋势 4](#_Toc28869020)

[4软件体系结构评估技术 6](#_Toc28869021)

[4.1评价框架简介 6](#_Toc28869022)

[4.2软件体系结构评估概览 6](#_Toc28869023)

[4.4.1 基于场景的评估方法 6](#_Toc28869024)

[4.4.2 基于预测和度量的评估方法 7](#_Toc28869025)

[4.4.2 基于特定软件体系结构描述语言的评估方法 7](#_Toc28869026)

[5结束语 7](#_Toc28869027)

[参考文献 9](#_Toc28869028)

[致谢 10](#_Toc28869029)

# 引言

软件工程作为独立的一门学科，其发展已逾30年。20世纪60年代，由于高级语言的流行，使得计算机的应用范围得到较大扩展，对软件系统的需求急剧上升，从而产生了所谓的“软件危机”，即软件开发从质量、效率等方面均不能满足应用需求。为了解决软件危机这一问题，1968年，在NATO会议上首次提出了“软件工程”这一概念，使软件开发开始了从“艺术”、“技巧”和“个体行为”向“工程”和“群体协同工作”转化的历程。

软件体系结构本质上并不是一个全新的概念，早在20世纪六七十年代，Dijkstra和Parnas等人对软件设计基本原则的研究就是对软件体系结构的最初认识。自20世纪90年代，Perry&Wolf和Garlan&Shaw分别提出了软件体系结构的概念以来，软件体系结构得到了学术界和工业界的广泛重视。

# 1软件工程技术发展历程

软件是客观事物的一种反映，客观世界的不断变化促使软件技术的不断发展，这种事物发展规律促使软件工程的产生和发展。我们仅从解决软硬件的异构性和各种软件之间的异构性角度，就可窥见软件技术发展的一种途径。如，为屏蔽计算机硬件之间的异构性发展了操作系统，为屏蔽操作系统之间和编程语言之间的异构性出现了支撑软件和中间件，为屏蔽不同中间件之间的异构性发展了WebServices技术等等;随着解决问题的不断深入，易用性和适应性要求的不断提升，以及软件技术的不断发展，还会出现更新、更复杂的异构问题，它的解决会促进软件技术的不断发展。从学科角度来看，要不断提炼所要解决问题的概念，建立相应的模型，并寻找处理方法，从而解决这些问题的概念模型和处理问题逻辑间的映射问题。

软件是对客观世界中问题空间与解空间的具体描述，是客观事物的一种反映，是知识的提炼和“固化”。客观世界是不断变化的，因此，构造性和演化性是软件的本质特征。如何使软件模型具有更强的表达能力、更符合人类的思维模式，即如何提升计算环境的抽象层次，在一定意义上来讲，这紧紧围绕了软件的本质特征——构造性和演化性。

在高级语言出现以前，汇编语言(机器语言)是编程的工具，表达软件模型的基本概念(或语言构造)是指令，表达模型处理逻辑的主要概念(机制)是顺序和转移。显然，这一抽象层次是比较低的。高级语言的出现，例如FORTRAN语言、PASCAL语言、C语言等，使用了变量、标识符、表达式等概念作为语言的基本构造，并使用3种基本控制结构来表达软件模型的计算逻辑，因此软件开发人员可以在一个更高的抽象层次上进行程序设计。随后出现了一系列开发范型和结构化程序设计技术，实现了模块化的数据抽象和过程抽象，提高了人们表达客观世界的抽象层次，并使开发的软件具有一定的构造性和演化性。近20年来，面向对象程序设计语言的诞生并逐步流行，为人们提供了一种以对象为基本计算单元，以消息传递为基本交互手段来表达的软件模型。面向对象方法的实质是以拟人化的观点来看待客观世界，即客观世界是由一系列对象构成，这些对象之间的交互形成了客观世界中各式各样的系统。面向对象方法中的概念和处理逻辑更接近人们解决计算问题的思维模式，使开发的软件具有更好的构造性和演化性。目前，人们更加关注软件复用问题，构建比对象粒度更大、更易于复用的基本单元--构件，并研究以构件复用为基础的软件构造方法，更好地凸现软件的构造性和演化特性。易于复用的软件，一定是具有很好构造性和演化性的软件。

# 

# 2软件工程的主要研究内容

从某种角度来说，软件开发的本质就是要实现“高层概念”到“低层概念”的映射，实现“高层处理逻辑”到“低层处理逻辑”的映射。对于大型软件系统的开发，这一映射是相当复杂的，涉及到有关人员、使用的技术、采取的途径以及成本和进度的约束，因此，我们可以把软件工程定义为：软件工程(softwareengineering)是应用计算机科学理论和技术以及工程管理原则和方法，按照预算和进度，实现满足用户要求的软件产品的义、开发、发布和维护的工程或以之为研究对象的学科[1]。软件工程的基本目标是生产具有正确性、可用性及开销合宜(合算性)的产品。正确性意指软件产品达到预期功能的程度;可用性意指软件基本结构、实现及文档达到用户可用的程度;开销合宜意指软件开发、运行的整个开销满足用户的需求。以上目标的实现不论在理论上还是在实践中均存在很多问题有待解决，制约了对过程、过程模型及工程方法的选取。软件工程活动是“生产一个最终满足用户需求且达到工程目标的软件产品所需要的步骤”，主要包括需求、设计、实现、确认以及支持等活动。需求活动是在一个抽象层上建立系统模型的活动，该活动的主要产品是需求规约，是软件开发人员和客户之间契约的基础，是设计的基本输入。设计活动定义实现需求规约所需的结构，该活动的主要产品包括软件体系结构、详细的处理算法等。实现活动是设计规约到代码转换的活动。验证/确认是一项评估活动，贯穿于整个开发过程，包括动态分析和静态分析。主要技术有模型评审、代码“走查”以及程序测试等。维护活动是软件发布之后所进行的修改，包括对发现错误的修正、对环境变化所进行的必要调整等。

围绕工程设计、工程支持以及工程管理，提出以下软件工程基本原则:围绕工程设计、工程支持以及工程管理，提出以下软件工程基本原则:第1条原则是选取适宜的开发风范。以保证软件开发的可持续性，并使最终的软件产品满足客户的要求。第2条原则是采用合适的设计方法。支持模块化、信息隐蔽、局部化、一致性、适应性、构造性、集成组装性等问题的解决和实现，以达到软件工程的目标。

第3条原则是提供高质量的工程支持。提供必要的工程支持，例如配置管理、质量保证等工具和环境，以保证按期交付高质量的软件产品。

第4条原则是有效的软件工程管理。仅当对软件过程实施有效管理时，才能实现有效的软件工程。

由以上软件工程的概念和框架可以看出，软件设计的主要目标就是要实现好的结构，使开发的软件具有良好的构造性和演化性。软件工程学科所研究的内容主要包括:软件开发范型、软件设计方法、工程支持技术和工程管理技术。其中，软件开发范型涉及软件工程的“方向”问题，研究正确的求解软件的计算逻辑;软件设计方法涉及软件工程的“途径”问题，研究“高层概念模型和处理逻辑”到“低层概念模型和处理逻辑”的映射;工程支持技术和过程管理技术涉及工程过程质量和产品质量问题，研究管理学理论在软件工程中的应用。如上所述，软件开发就是实施了一个从“高层概念模型”到“低层概念模型”的映射，从“高层处理逻辑”到“低层处理逻辑”的映射，而且在这一映射中还涉及到人员、技术、成本、进度等要素，那么就必须研究映射模式即软件生产模式问题。

分析传统产业的发展，其基本模式均是符合标准的零部件(构件)生产以及基于标准构件的产品生产(组装)，其中，构件是核心和基础，“复用”是必须的手段。实践表明，这种模式是软件开发工程化、软件生产工业化的必由之路[2]。因此，软件产业的发展并形成规模经济，标准构件的生产和构件的复用是关键因素。

# 3软件技术的发展趋势

Internet无疑是20世纪末伟大的技术进展之一，为我们提供了一种全球范围的信息基础设施。这个不断延伸的网络基础设施，形成了一个资源丰富的计算平台，构成了人类社会的信息化、数字化基础，成为我们学习、生活和工作的必备环境。如何在未来Internet平台上进一步进行资源整合，形成巨型的、高效的、可信的和统一的虚拟环境，使所有资源能够高效、可信地为所有用户服务，成为软件技术的研究热点。

Internet平台具有如下基本特征:无统一控制的“真”分布性;节点的高度自治性;节点链接的开放性和动态性;人、设备和软件的多重异构性;实体行为的不可预测性;运行环境的潜在不安全性;使用方式的个性化和灵活性;网络连接环境的多样性等。因此，Internet平台和环境的出现，对软件形态、技术发展、理论研究提出新的问题，也提供了新的契机。

传统软件的开发基于封闭的静态平台，是自顶向下、逐步分解的过程，因此传统软件的开发，基本都是首先确定系统的范围(即Scoping)，然后实施分而治之的策略，整个开发过程处于有序控制之下。而未来软件系统的开发所基于的平台是一个有丰富基础软件资源但同时又是开放、动态和多变的框架，开发活动呈现为通过基础软件资源组合为基本系统，然后经历由“无序”到“有序”的往复循环过程，是动态目标渐趋稳态。未来软件基本模型由于所处平台的特性和开放应用的需求而变得比任何传统的计算模型都更为复杂，软件生命周期由于“无序”到“有序”的循环而呈现出不同于传统生命周期概念的“大生命周期概念”，程序正确性由于目标的多样化而表现为传统正确性描述的一个偏序集，软件体系结构侧重点从基于实体的结构分解转变为基于协同的实体聚合，软件生产过程和环境的变化导致基于Internet的面向用户的虚拟工厂的形成。

由于软件系统所基于的计算机硬件平台正经历从集中封闭的计算平台向开放的Internet平台的转变，软件系统作为计算机系统的核心，随着其运行环境的演变也经历了一系列的变革。目前，面向网络的计算环境正由Client/Server发展为Client/Cluster，并正朝着Client/Network和Client/VirtualEnvironment的方向发展。那么，未来的基于Internet平台的软件系统又将会呈现出一个什么形态呢?

从技术的角度来看，以软件构件等技术支持的软件实体将以开放、自主的方式存在于Internet的各个节点之上，任何一个软件实体可在开放的环境下通过某种方式加以发布，并以各种协同方式与其他软件实体进行跨网络的互连、互通、协作和联盟，从而形成一种与当前的信息Web类似的SoftwareWeb。SoftwareWeb不再仅仅是信息的提供者，它还是各种服务(功能)的提供者。由于网络环境的开放与动态性，以及用户使用方式的个性化要求，从而决定了这样一种SoftwareWeb，它应能感知外部网络环境的动态变化，并随着这种变化按照功能指标、性能指标和可信性指标等进行静态的调整和动态的演化，以使系统具有尽可能高的用户信赖度。我们将具有这种新形态的软件称为网构软件(internetware)。

网构软件是在Internet开放、动态和多变环境下软件系统基本形态的一种抽象，它既是传统软件结构的自然延伸，又具有区别于在集中封闭环境下发展起来的传统软件形态的独有的基本特征:

(1)自主性:是指网构软件系统中的软件实体具有相对独立性、主动性和自适应性。自主性使其区别于传统软件系统中软件实体的依赖性和被动性;

(2)协同性:是指网构软件系统中软件实体之间可按多种静态连接和动态合作方式在开放的网络环境下加以互连、互通、协作和联盟。协同性使其区别于传统软件系统在封闭集中环境下单一静态的连接模式;

(3)反应性:是指网构软件具有感知外部运行和使用环境并对系统演化提供有用信息的能力。反应性使网构软件系统具备了适应Internet开放、动态和多变环境的感知能力;

(4)演化性:是指网构软件结构可以根据应用需求和网络环境变化而发生动态演化，主要表现在其实体元素数目的可变性、结构关系的可调节性和结构形态的动态可配置性上;演化性使网构软件系统具备了适应Internet开放、动态和多变环境的应变能力。

(5)多态性:是指网构软件系统的效果体现出相容的多目标性。它可以根据某些基本协同原则，在动态变化的网络环境下，满足多种相容的目标形态。多态性使网构软件系统在网络环境下具备了一定的柔性和满足个性化需求的能力。

综上所述，Internet及其上应用的快速发展与普及，使计算机软件所面临的环境开始从静态封闭逐步走向开放、动态和多变。软件系统为了适应这样一种发展趋势，将会逐步呈现出柔性、多目标、连续反应式的网构软件系统的形态。面对这种新型的软件形态，传统的软件理论、方法、技术和平台面临了一系列挑战。从宏观上看，这种挑战为我们研究软件理论、方法和技术提供了难得的机遇，使我们有可能建立一套适合于Internet开放、动态和多变环境的新型软件理论、方法和技术体系。从微观的角度来看，Internet的发展将使系统软件和支撑平台的研究重点开始从操作系统等转向新型中间件平台，而网构软件的理论、方法和技术的突破必将导致在建立新型中间件平台创新技术方面的突破。

归结起来，网构软件理论、方法、技术和平台的主要突破点在于实现如下转变，即，从传统软件结构到网构软件结构的转变，从系统目标的确定性到多重不确定性的转变，从实体单元的被动性到主动自主性的转变，从协同方式的单一性到灵活多变性的转变，从系统演化的静态性到系统演化的动态性的转变，从基于实体的结构分解到基于协同的实体聚合的转变，从经验驱动的软件手工开发模式到知识驱动的软件自动生成模式的转变。建立这样一种新型的理论、方法、技术和平台体系具有两个方面的重要性，一方面，从计算机软件技术发展的角度，这种新型的理论、方法和技术将成为面向Internet计算环境的一套先进的软件工程方法学体系，为21世纪计算机软件的发展构造理论基础;另一方面，这种基于Internet计算环境上软件的核心理论、方法和技术，必将为我国在未来5~10年建立面向Internet的软件产业打下坚实的基础，为我国软件产业的跨越式发展提供核心技术的支持。

当前的软件技术发展遵循软硬结合、应用与系统结合的发展规律。“软”是指软件，“硬”是指微电子，要发展面向应用，实现一体化;面向个人，体现个性化的系统和产品。软件技术的总体发展趋势可归结为:软件平台网络化、方法对象化、系统构件化、产品家族化、开发工程化、过程规范化、生产规模化、竞争国际化。

# 4软件体系结构评估技术

## 4.1评价框架简介

为了系统地对各种软件体系结构评估技术进行说明和比较，需要一个可操作、全面的评价框架。在对软件体系结构评估技术的分析和调查方面，下面两个工作是非常重要并且具有参考价值的:(1)Dobrica和Niemela于2002年首次提出了对软件体系结构分析方法进行刻画和比较的框架，并利用该框架对当时的各种软件体系结构的评估方法进行分析;(2)Babar等人于2004年扩展和细化了Dobrica的框架，详细描述了框架中各个元素，使分析更加具有可操作性，并在此基础上分析和比较了各种基于场景的评估方法。AliBarba框架包括了Dobrica框架中的所有评价项。

随着软件体系结构评估技术的发展，除了基于场景的评估方法,还出现了基于度量-预测的评估方法和基于某种特定软件体系结构描述语言(ADL)的分析方法。为了更好地比较和分析不同的软件体系结构评估技术，本文对Dobrica和AliBarba两种评价框架中的评价项进行了适当的调整和扩充。

## 4.2软件体系结构评估概览

随着时间的推移，新的软件体系结构评估技术不断涌现。本文根据各种评估技术的主要特点，将评估技术分为3类:基于场景的评估方法、基于度量-预测的评估方法和基于特定软件体系结构描述语言的分析方法。

### 4.4.1 基于场景的评估方法

该类评估方法的基本观点是，大多数软件质量属性极为复杂，根本无法用一个简单的尺度来衡量。同时，质量属性并不是处于隔离状态，只有在一定的上下文环境中才能做出关于质量属性的有意义的评判。利用场景技术则可以具体化评估的目标，代替对质量属性(可维护性、可修改性、健壮性、灵活性等)的空洞表述，使对软件体系结构的测试成为可能。所以，场景对于评估具有非常关键的作用，整个评估过程就是论证软件体系结构对关键场景的支持程度。通过对多种基于场景的评估方法的分析，我们认为该类方法具有以下重要的特征:(1)场景是这类评估方法中不可缺少的输入信息，场景的设计和选择是评估成功与否的关键因素;(2)这类评估是人工智力密集型劳动，评估质量在很大程度上取决于人的经验和技术。

### 4.4.2 基于预测和度量的评估方法

测量对于所有科学领域的进步都是至关重要的。在软件工程领域中，软件的度量和预测技术是保证软件质量的重要技术之一。软件体系结构作为软件开发过程中一个早期的设计模型，如果能够度量并预测未来软件产品的质量，那么其预测的结果可以及时给出设计缺陷，这对于减少开发风险和提高软件质量是非常重要的。根据这一思路，出现了一类基于度量和预测的评估方法。

软件体系结构的度量是对软件中间产品的度量，可以更加精确地描述软件体系结构的各种特征;并通过预测去发现软件设计中存在的问题。通过对多种基于度量和预测评估方法的分析和比较，我们认为，该类方法具有以下的重要特征:(1)这些方法的基本思路是将传统的度量和预测技术应用在软件体系结构层次;(2)度量技术需要软件体系结构提供比较细粒度的信息，对模型的要求比较严格;(3)利用度量技术对软件体系结构模型的内部特征(如复杂性、内聚度、耦合性等)进行测量;(4)利用这些度量作为预测指标，对某些软件的外部质量(如可维护性、可演化性、可靠性等)进行预测，但由于预测模型构造的困难，所以这些预测一般只作为一种辅助评估的手段。

### 4.4.2 基于特定软件体系结构描述语言的评估方法

基于特定软件体系结构描述语言(ADL)的评估方法是一类比较特殊的方法，这类方法依赖于某种具体的软件体系结构描述语言，一般是软件体系结构语言研究的附属品。通过对多种基于特定软件体系结构描述语言的评估方法的分析和比较，我们认为该类方法具有以下的重要特征:(1)评估技术与特定软件体系结构描述语言的定义机制和理论基础密切相关;(2)软件体系结构描述语言的定义非常严格，通常是形式化或半形式化的描述语言。

# 5结束语

不同类型的软件体系结构评估技术虽然有不同的研究思路和研究内容，但这些技术之间却并不矛盾，甚至存在着互补关系，如度量技术在基于场景的评估方法中可以作为评估手段来使用。在进行软件体系结构的行为度量时，场景技术则可以为度量提供信息。

当前的软件技术发展遵循软硬结合、应用与系统结合的发展规律。“软”是指软件，“硬”是指微电子，要发展面向应用，实现一体化;面向个人，体现个性化的系统和产品。软件技术的总体发展趋势可归结为:软件平台网络化、方法对象化、系统构件化、产品家族化、开发工程化、过程规范化、生产规模化、竞争国际化。

# 

# 参考文献

[1] 张效祥，主编.计算机科学技术百科全书.北京:清华大学出版社，1998.

[2] 杨芙清，梅宏，李克勤.软件复用与软件构件技术.电子学报，1999，27(2):68−75.

[3] 杨芙清.青鸟工程现状与发展——兼论我国软件产业发展途径.见:杨芙清，何新贵，主编.第6次全国软件工程学术会议论文集，软件工程进展——技术、方法和实践.北京:清华大学出版社，1996.

[4] Bass L, Clements P, Kazman R. Software Architecture in Practice. 2nd ed., Boston: Addison Wesley Professinal, 2003.

[5] Dobrica L, Niemela E. A survey on software architecture analysis methods. IEEE Trans. on Software Engineering, 2002,28(7): 638−653.

# 

# 致谢

本课程报告是基于应教授所授的“高级软件工程”为基础完成。应老师渊博的专业知识，严谨的治学态度，让我获益匪浅。不仅使我对软件工程这一门课程有了更深的理解，而且扩展了我的视野，更重要的是让我对学习知识有了不一样的理解，非常感谢应老师。