软件成本估算旨在预测完成软件项目所需的最现实的工作，因此软件项目管理的成功至关重要。软件成本估算几乎影响所有的管理活动，包括项目招标，资源分配和项目计划。它受到许多因素的影响，比如实施效率，以及在软件开发阶段之前完成的各种评审和研究的成本。准确的成本估算将帮助我们在预算内按时完成项目。准确的估计是重要的，因为它导致了对软件成本估计方法的广泛研究。在这项研究工作中已经研究了一些重要的软件成本估算方法。另外，我们制定了自己的标准，用来比较所有不同的方法。我们也给出了每个评估标准的评分，以便我们可以比较不同的方法进行成本估算的数字。我们的观察表明，最好使用许多不同的估算技术或成本模型，然后在确定任何大的变化的原因之前比较结果。这些方法都不一定比其他方法更好或更差。事实上，我们发现他们的长处和短处往往是相辅相成的。因此，主要结论是，没有一种技术最适合每一种情况，需要仔细考虑一些不同方法的结果，以发现最可能产生现实的估计的东西。

估计软件项目的成本是一个关键的活动，需要使用适当的方法和技术来实现对结果的良好估计。 这是一个具有挑战性的任务，会带来很多障碍。 软件的大小及其准确性对估计的准确性有很大的影响。 项目管理在这些评估过程的指导中也起着至关重要的作用。通过有效的成本估算，进行了一些反映高质量软件需求的研究。

软件工程师必须将理论，工具和方法应用到软件项目中才能解决问题。 但是，他们也必须在预定义的财务约束范围内工作。 与软件项目财务方面密切相关的一个重要问题是所涉及的软件成本的准确估计。 这有助于管理任何软件项目，因为这意味着它将在设定的预算内。软件成本估算在软件项目管理中是一项非常具有挑战性的工作，因为在软件开发的早期阶段预测成本是一个困难的过程。 此外，软件成本的估算受许多因素的影响，包括实施的效率，以及在开发阶段成本之前完成的审查和研究的数量。 软件工作量和软件成本之间存在着很强的关系，因为可以说工作量是软件成本的主要驱动因素。说明软件成本的估算是一个持续的活动，从提案阶段开始，然后在项目的整个生命周期中继续进行，这一点很重要。 在计算项目成本管理时，它包括确保项目在批准预算范围内完成所需的过程。 主要过程包括估算成本（包括自上而下和自下而上的估算，参数化建模等）;

?确定预算（成本基准）;

?控制成本（挣值管理（EVM））

已经设计了许多方法来解决这个软件成本估算过程，科学家和研究人员都试图建立一个精确的成本估算技术。 研究工作提供了广泛的概述。 它将总共涉及五种基本的软件成本估算方法，并将根据评估标准对各种方法进行比较。 这将在整个研究工作中被彻底检查和使用。

2.1。方法1：建设性成本模型（COCOMO）考虑到COCOMO是计算软件成本估计的一个非常重要的模型，它使用一个算法公式来估计软件的成本[17]。因此，这个模型是基于数学和一些实验方程。 Barry Bohem在1981年提出了软件成本估算。它被认为是最完整的方法，并且比其他成本估算模型有更好的记录，如参考文献[18] [19] [20] [21]所示。此外，许多软件工程领域的研究人员正在努力通过保持COCOMO模型的基础来提高效率。而且，由于这个模型的简单性，通常用于算法成本估算技术。 COCOMO由三个模型组成，分别是Basic，Intermediate和Detailed。第一种模式是基本模式，它是计算软件工作量和成本中程序大小的函数。它主要用于中小型软件项目，以便进行快速，粗略的估计。因此，基本的COCOMO被认为在只需要粗略估算的情况下是有效的。估算软件的等式

这个基本模型的努力是:

Effort= a\* SIZE

在这个等式中，SIZE是在千个源代码指令（KLOC，这是一千行代码）中测得的。系数“a”和“b”分别是生产力系数和比例因子系数。这里要指出的是，系数的价值都取决于项目的模式。 Boehm提出的三种不同的模式是有机模式，半独立模式和嵌入模式。第一种是有机地用于小型化项目的2-50KLOC。第二个是半独立模式，适用于50-300 KLOC的中型项目。第三，嵌入模式是针对复杂的，大型的软件工程和应用项目，通常在300 KLOC以上。中间模型被用来根据程序的大小来计算工作量，这套成本动因。这个模型与基本版本略有不同，因为基本的COCOMO没有考虑到中间模型的软件开发环境。中级COCOMO有15个成本驱动因素，为基本的COCOMO增加了一定的准确度。这些成本驱动有四类，即计算机属性，产品属性，项目属性和人员属性。用于估计中间模型的软件工作量的公式是[17]：

努力= a SIZE m X×（2）

在等式中，m（X）表示努力调整因子，这是总共15个努力乘数的乘积。第三个（详细模型）还有两个功能。这些是相位敏感的工作倍增器和3个级别的产品层次结构。这3个级别是模块，子系统和系统，这些都是用来得出一个准确的估计。

2.2。方法2：主成分分析（PCA）的前馈神经网络[22]中的作者提出了称为具有主成分分析（PCA）的前馈神经网络的归约技术。作者的主要目标是使用这个来衡量软件成本估算模型的准确性。所提出的技术基于算法和非算法两种方法。因此，他们使用算法（COCOMO）和非算法（人工神经网络）的组合来估计软件项目的成本。在本段中，作者将简要介绍他们提出的软件成本估计系统的体系结构设计。其主要参数将用作所提出的方法的输入，是尺寸，成本因素和比例因子。这些参数全部来自根据项目规格收集的实际数据集。第二步是应用PCA。这是通过计算相关系数矩阵，以及相关系数矩阵的特征值来完成的。之后可以确定主要成分的数量。这些组件作为输入馈入神经网络系统以训练数据集。然后，输出层将尺寸，工作倍数和比例因子值发送到COCOMOII。通过从神经网络系统COCOMOII发送的这些输入，可以估计软件成本。这个结果是基于研究人员广泛使用的COCOMO样本数据集。它由来自世界各国的161个历史项目组成。结果表明，当PCA和神经网络不适用时，混合技术比相同类型的算法提供更准确的成本估计。

2.3。 方法3：Putnam模型/ SLIM Putnam / SLIM估计方法在20世纪70年代后期由定量软件管理的Larry Putnam开发，如参考文献[23] [24] [25] [26] [27]中所强调的那样。 SLIM（软件生命周期管理）是Putnam给予他的公司QSM，Inc.开发的专有工具套件的名称，其基于他的模型。 这是一个经验性的软件工作评估模型，也是第一个算法成本模型之一。它描述了完成指定大小的软件项目所需的时间和精力。 基于Norden / Rayleigh函数，它通常被称为宏观估计模型（因为它主要用于大型项目）。 Putnam模型的软件方程如下给出。。。在对软件任务进行估计时，解决了实际使用中的软件方程。。。项目完成时软件的估计大小以及组织过程的生产力被使用。 时间 - 努力曲线是通过将努力作为时间的函数绘制来计算的，并且完成项目所需的估计总工作量由曲线上的点表示[27]。这种估计方法对于不确定性非常敏感 规模和生产力的过程。 普特南主张通过校准获得这个过程的生产力[23] [24] [25]：。。。

SLIM的优势

?它利用线性规划来考虑所需的成本和努力的发展限制。

Putnam模型的一个显着特点是，随着项目完成时间的延长，总体工作量减少。这通常由其他参数模型中的调度松弛参数表示。

?SLIM需要较少的参数来生成COCOMO'81和COCOMO'II的估计值。

SLIM的缺点

?此模型基于知道或能够准确估计要开发的软件（在代码行中）的大小。有frei。 M. Keshta DOI：10.4236 / jsea.2017.1010046 829软件工程与应用期刊

对软件的大小有很大的不确定性，导致成本估算不准确。 SLIM的错误率为772.87％

[24]，根据Kemerer的研究，

?此模型对开发时间非常敏感，因为减少开发时间可能会大大增加人员和月份的数量。

?它不适合小项目。

2.4。方法4：功能点分析第一次在文本中使用缩略语和首字母缩写时，即使在摘要中定义之后，也要定义它们。 IEEE，SI，MKS，CGS，sc，dc和rms等缩略语不必定义。除非不可避免，否则不要在标题或标题中使用缩写。诸如COCOMO，Putnam等算法模型需要估计SLOC（源代码行）的数量，以便同时获得人月和持续时间估计。功能点分析是另一种可以用来量化软件系统的大小和复杂度的方法，就功能点能够向用户提供的功能而言。 IBM的Allan Albrecht开发了功能点测量方法，该方法于1976年首次发布[28] [29]。计算功能点[24]涉及两个步骤，它们是：

?计算各种用户功能

?进行调整以处理复杂性

目前，五个用户功能类别为：外部输出类型，外部输入类型，外部接口文件类型，逻辑内部文件类型和外部查询类型。阿尔布雷希特认为，提供给定功能水平所需的努力可能会有所不同，这取决于环境因素。例如，如果将重点放在系统吞吐量或最终用户的便利性上，那么向程序输入事务就比较困难。因此，Albrecht列举了14个处理复杂性特征。这些评级的标度从0（表示无影响）到5（意味着强烈的影响）。所有已经分配的处理复杂点在下一步中总结。这个数字乘以0.01。然后将其加到0.65以获得以下加权：

其中PCA =处理复杂度调整，然后ci =复杂度因子。因此，各种功能点可以相对于原始函数计数??变化±35％。一旦它们被计算出来，这些功能点就可以被用来比较所提出的项目的大小，与以前的项目相比。

I. M. Keshta DOI：10.4236 / jsea.2017.1010046 830软件工程与应用杂志基于功能点分析的模型有许多优点，它们是[30] [31] [32]：

?功能点可以根据需求规格或设计规格进行估算，从而可以在开发初期阶段估算开发成本。

?这些功能点与用于执行的语言，工具或方法无关。

?由于功能点基于系统用户自己的系统外部视图，非技术用户能够更好地理解功能点正在测量的内容。

2.5。方法5：小波神经网络（WNN）作者使用[33]中的小波神经网络（WNN）来估计软件开发工作。对WNN的两个变种，Morlet小波函数[34]，高斯函数和小波神经网络阈值接受训练算法（TAWNN）进行了实验。 （MLP）[35]，径向基函数网络（RBFN）[36]，多重线性回归（MLR）[37]，动态进化神经模糊推理系统（DENFIS）[38]和支持向量机（SVM）[39]。所有这些比较都是根据在加拿大金融“CF”数据集和IBM数据处理服务“IBMDPS”数据集上应用的平均数相对误差“MMRE”来计算的。本研究中使用的WNN由三个相互连接的层组成。这些是输入层，隐藏层和一个单一的单元输出层，如图1所示。WNN也有两种变体，分别是Morlet函数和Gaussian函数。这些都是作为激活功能应用的。需要指出的是，作者受到门槛接受（TA）概念的影响，在这个新概念的解决方案中

图1。 。 小波神经网络WNN.I. M.Keshta DOI：10.4236 / jsea.2017.1010046 831软件工程与应用软件接受是由确定性的标准，而不是一个概率的方法确定。这个想法如下：反向传播的前向部分保持不受干扰，而后 通过使所有权重成为决策变量的向量来完成TA更新的传播。 为了训练WNN，这个TA概念被采用，这就是为什么它被称为阈值接受小波神经网络或TAWNN学习算法。 该训练算法的目标函数为：

易用性是评估各种方法的重要标准。这决定了一个给定方法的简单程度，因为它会试图回答这个方法是多么容易。在小型项目上使用COCOMO模型很容易。然而，由于这些项目的复杂程度以及这些情况下存在多少未知变量，在大型项目中使用它可能会很困难。因此，这种方法在这个特性中的得分是15分之12。对于[22]的作者提出的混合方法（包括算法和非算法方法），使用这种方法并不容易。这是因为它是一种混合方法，它使用算法（COCOMO）和非算法（人工神经网络）来估计软件项目的成本。总分15分中有8分;因为这样的方法显然需要在使用COCOMO模型（即人工神经网络算法）之后进一步的步骤。 Putnam模型是最早开发的模型之一。它是最广泛使用的，因此与COCOMO等模型密切相关。模型的关键之一。 M. Keshta

DOI：10.4236 / jsea.2017.1010046 834“软件工程与应用”“软件工程与应用”大多数软件组织，无论成熟度水平如何，都可以轻松收集过去项目的规模，工作量和持续时间（时间）。因此，给出了15个中的12个。从过去的项目中，单个单位的成本（以美元或小时计）由功能点计算。这种方法使得能够从特定的语言，方法或技术抽象出来，并且更容易理解，也可以解释非技术和外部利益相关者以及用户。因此，功能点分析的得分是十二分之十五。小波神经网络（WNN）有十五分之五，因为它很难使用，并且包含很多参数，需要专家来计算它们。这个分数显然是模型复杂性的结果。它的两个主要功能是Morlet小波函数和用来执行这个计算的高斯函数。这些功能需要更复杂的人来处理。

软件成本估算可以被看作是必须使用正确的方法和技术来完成对结果的良好估计的基本活动。 这就是为什么我们在这项工作中研究了几种成本估算方法，然后评估和比较了其中的五种。 这五种方法在这里需要注意的是，我们引入了不同的评估特性（即易用性，适应性，准确性，一致性，可解释性，可自动化，工具支持，经验验证，敏感性和处理不精确性和不确定性），以便比较 这五个软件成本估算方法之间。我们的观察表明，最好是使用一些不同的估算技术或成本模型为项目经理，然后比较结果，然后确定原因的大变化，并记录任何假设是 作出估计时做出的。