



早期软件规模度量指南

专家级

第二版

2020 年 2 月 27 日

致谢：编辑及评审人员 2020（按字母顺序）

Alain Abran École de Technologie Supérieure Canada	Arlan Lesterhuis COSMIC the Netherlands	Bruce Reynolds Tecolote Research United States
Asma Sellami University of Sfax Tunisia	Hassan Soubra German University of Cairo Egypt	
Sylvie Trudel Université du Québec à Montréal Canada	Francisco Valdés Souto Spingere Mexico	Frank Vogelezang * METRI The Netherlands

* 编辑者

中文版贡献者

翻译组织者	麦哲思科技（北京）有限公司 www.measures.net.cn 电话：400-1780727
初版翻译	徐丹霞 麦哲思科技高级咨询顾问，CMMI 教员，大规模敏捷教练，COSMIC 讲师
校对	郭玲 麦哲思科技高级咨询顾问，COSMIC 讲师

注：对 COSMIC 及本文档的任何疑问或指正之处，请加入 COSMIC 交流 QQ 群——309842452。

本文档变更履历：

日期	评审者	修改/新增
2015 年 7 月	COSMIC 度量实践委员会	本指南第一次发布
2018 年 8 月	相关主题专家	与手册 4.0.2 版保持一致的若干修订，以及相关著作者的评审意见修改。
2020 年 2 月	相关主题专家	根据 2019-2020 年的评审意见所作的修改。

COSMIC 公开发行的文档，包括其他语言的翻译版本，可以从 www.cosmic-sizing.org 下载。您可以在 cosmic-sizing.org/forums 论坛上发布您的问题，并从 COSMIC 全球社区得到回答。答案的质量取决于回答问题者的知识和经验。

目前也有提供相关培训和咨询或工具支持的商业机构——详情请参见 www.cosmic-sizing.org。

DOI 10.13140/RG.2.1.4195.0567

版权所有。保留所有权利。通用软件度量国际联盟（COSMIC）。用于非商业目的的情况下，允许拷贝材料的部分或全部内容，但必须引用文档的标题、版本号和日期，并指明是根据 COSMIC 的授权许可。否则，拷贝需要特殊许可。

COSMIC 方法提供了一个度量软件功能规模的标准化方法。但在实践中，有时得到近似规模就够了、或者只能得到近似规模：

- 项目早期，在功能性用户需求（FUR）还未细化到能够进行精确规模度量之前；
- 当没有足够的时间或资源使用标准方法进行度量，并且快速得到的近似规模是可接受的；
- 当实际需求文档的质量不足以进行精确度量时。

本指南目的

本指南旨在介绍早期或快速 COSMIC 功能规模度量相关的近似方法的最新进展。本文将介绍几种近似方法及其优缺点，推荐的应用领域和有效性。

我们假设本文读者熟悉标准 COSMIC 方法。那些需要在实践中使用近似方法的读者，请阅读：COSMIC 早期软件规模度量指南-实践级。

第 1 章介绍为何需要近似功能规模度量；实际需求描述常见的几种详细程度（称之为“文档化层级”）以及认知和应用近似规模度量方法的几个基本原则。

指南的后续内容划分为四个部分：

- **第一部分** 项目需求阶段的近似方法（2-6 章）
- **第二部分** 项目可行性阶段的近似方法（7-10 章）
- **第三部分** 仍处于研究中的近似方法（11 章）
- **第四部分** 共通概念

本指南读者如果是近似规模度量的初学者，强烈建议先阅读第 1 章近似规模度量的基本原理和第 4 部分共通概念。

前言	5
----	---

目录	6
----	---

1.近似规模度的基本原理	8
1.0 关于术语的注意事项	8
1.1 什么时候需要 COSMIC 近似规模度量?	8
1.2 近似规模度的各种方法	9
1.2.1 通用原则	9
1.2.2 度量的按比例缩放	10
1.2.3 本地化 (校准)	10
1.2.4 按分类和比例系数缩放得到近似规模	10
1.2.5 近似规模度的准确度	11
1.3 实际需求的文档化层级	11
1.4 实际需求的质量	11
1.5 本指南所述近似方法的适用性	12
第一部分	13
项目需求阶段的近似方法	13
2.功能处理均值法	14
3.固定规模分类法	16
4.相等规模带法	18
5.用例均值法	22
6.功能规模度量模式	24
第二部分	29
项目可行性阶段的近似方法	29
7.软件冰山近似法	30
8.早期&快速近似法	33
9.应用模糊逻辑进行近似度量—EPCU 模型	36
10.简单功能点近似法	39

第三部分	41
仍处于研究中的近似方法	41
11.新兴的近似度量方法	42
11.1 对于非正式记录的文字需求的近似度量	42
11.2 基于数据组个数均值的近似法	42
11.3 基于用例名称的近似法	42
11.4 基于 UML 用例图的近似法	43
11.5 等量带近似法	44
11.6 等距带近似法	44
第 4 部分	45
共通概念	45
12.不同的文档详细程度和分解层级	46
12.1 大型软件项目早期的需求演化过程	46
案例 1 在不同的文档颗粒度级别下度量——“Everest”系统	49
案例 2 在软件架构的多个颗粒度级别和分解层级上进行度量	49
12.2 功能规模度量和标准的分解层级	53
13.本地化（校准）指南	56
14.功能变更和范围蔓延的近似规模度量	58
14.1 功能变更的近似规模度量	58
14.2 近似度量与范围蔓延	58
15.近似规模度量方法总结	60
参考文献	61
术语表	64

1. 近似规模度量的基本原理

1.0 关于术语的注意事项

COSMIC 方法度量的是软件的“功能性用户需求”（或简称 FUR）。COSMIC 使用这一术语指代那些详细程度达到能够应用 COSMIC 方法进行功能规模度量的需求。

近似规模度量方法是用于度量那些没有达到此详细程度的需求。在本指南中，我们使用“实际需求”来指代这些需要用近似方法来度量的需求。术语“实际需求”可能包括“系统级”非功能需求。但很多实际需求最初表现为系统的非功能需求，随着项目进展会演化为软件的功能性需求，因而也可以用近似方法度量。

本指南用到的 COSMIC 通用术语，请参见度量书册[2]。本指南的特定术语，请参见最后的术语表。

1.1 什么时候需要 COSMIC 近似规模度量？

在项目生命周期的早期，需求没有详细描述到所有功能。随着时间推进，需求会逐渐细化，有时会随着软件开发的进展而发生变更。而在早期，一般需要对功能规模进行评估以便进行工作量或成本估算。

无论近似度量方法有什么优点，都无法弥补对软件工作内容理解的不足。直到软件规格完全定义清楚时，才能看到完整的解决方案及其对应的软件功能规模。随着我们对需求的理解更完整，并采取适当的风险管控措施，这个过程的不确定性会逐渐减少。这一现象可以图形化为不确定性锥。

不确定性锥的最初概念由美国成本工程师协会在 1958 年提出，是基于美国化工业的工程和建设领域发展而来[4]。在软件行业，Barry Boehm 在 1981 年提出了这一概念[5]。尽管这一现象通常在对数尺度上呈现，表现为一个对称的圆锥形，但我们更喜欢在线性尺度上显示它，以展现出：功能偏多的可能性远远大于功能偏少的可能性。

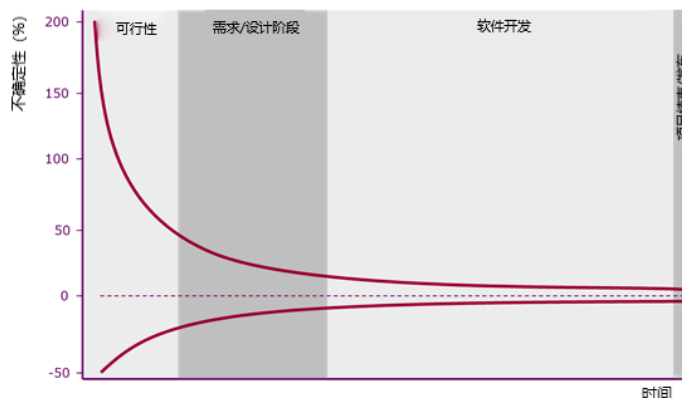


图 1.1 不确定性锥

使用敏捷方法的软件开发项目会打破图 1.1 的平滑曲线，因为项目被分成了多个冲刺（sprint）。每个冲刺适用于该曲线的一些部分。对于一个新软件，敏捷软件开发一般从架构设计开始。一旦确定了架构，其中优先级较高的部分就会立即开始细化，以用户故事的形式进入各个冲刺的待办

事项列表。本指南介绍的近似度量方法，可以在架构层级使用（见第 10 章），也可以在概要的用户故事层级使用。敏捷背景下如何使用 COSMIC 进行规模度量的更多信息请见[6]。

通常，下列三种场景，需要 COSMIC 近似¹规模度量：

- 当需要快速度量软件规模时，如果能用比标准方法快得多的方式得到近似规模，而这样的近似规模是可接受的，这种情况被称为“快速度量”；
- 在项目生命周期的早期，实际需求还未详细到可以进行精确的规模度量时，这种情况被称为“早期度量”；
- 总的来说，当实际需求文档质量不足以支持精确的规模度量时。

当面对较大的软件，或者比如一整套软件时，精确的度量可能会花费太多时间和金钱，并且近似规模是可接受的，这种情况下，“快速度量”是有价值的。

在 FUR 还没有细化到可以进行精确度量之前，项目就需要进行工作量估算。这种情况下，实际需求一般可能包含在各种层级的工作产品中，而这些工作产品可能是没有标准化的。比如，一些实际需求以用例的形式体现，而其他需求已被细化。而用例本身对需求描述的详细程度是互不相同的。

无论精确或是近似，度量人员应该尽量多地获取实际需求的详细描述。进而，可以基于假设得到尽量精确的功能规模。

1.2 近似规模度量的各种方法

1.2.1 通用原则

Steve McConnell 将软件规模的测定分为三个层级[7]：

- 计数级：如果具备足够详细的信息，可以用最准确的方式测定规模，即计数。将此方式映射为 COSMIC 方法，即基于详细到功能处理级别的文档、使用标准方法对数据移动进行计数，也就是遵从标准的度量过程。
即使在这一层级，需求也很少会详细到可以进行精确度量的程度，所以度量人员一般会需要做一些假设。
- 推算级：如果不具备足够详细的信息，则对其中一部分进行精确计数，其他部分使用本地校准的数据进行推算。此方式映射为 COSMIC 方法，即基于较粗的文档描述、使用近似方法的按比例缩放推算出功能处理级的规模。
- 经验判断级：专家可以根据头脑中积累的经验模型，给出一个规模近似值。这种专家经验判断是准确性最低的一种近似方法。如果专家的判断能与具体的规模信息相关联，则会增强准确性。映射到 COSMIC 方法就是：将一些手头已有的“对象”（比如，高层级的实际需求或用例清单）进行分类，并根据分类指定一个规模值，此规模值来自“推算级”建立的比例系数。

根据 McConnell 的定义，本指南 2 至 8 章的方法属于“推算级”规模测定，第 9 章属于“经验判断级”。

¹ 这里与其叫做“近似规模度量”，不如叫“规模估算”方法更准确些。但是，“估算”这个词和项目成本、工作量或者工期估算方法紧密地联系在一起了。为避免混淆，因此我们还是称其为“近似规模度量”。

1.2.2 度量的按比例缩放

按比例缩放：统计每一个实际需求，并将计数或测量值乘以一个数字，即“比例系数”，以确定其 COSMIC 功能规模。比例系数是由一个校准过程确定的，校准过程是对实际需求应用标准 COSMIC 方法计算规模值，并从中选取一个有代表性的数字作为此类需求的比例系数。

所有按比例缩放的近似方法的基本原理都是：找到本地概要层级（高层级）的实际需求描述，想办法度量其近似规模，然后当需求细化为功能处理级时度量其 CFP 个数。“比例系数”就是一个比例数字，可以将概要层级文档的需求个数转换为 CFP 数——见表 1.1。

实际需求的详细程度	规模度量	度量结果
概要层级的需求可能来自： <ul style="list-style-type: none">• 软件需求的概要层级描述• 架构文档• 既存软件的概要视图 以本地自定义单位（可计数）表达的需求，比如用例	COSMIC 度量方法中的近似方法。 经过本地校准	以本地自定义单位表示的规模，或以 CFP 为单位
功能处理级文档描述	COSMIC 度量方法	 以 CFP 为单位的规模

表 1.1 – 在不同的文档层级之间进行规模缩放

比例系数应在本地建立。更多指导步骤参见第 13 章。

1.2.3 本地化（校准）

近似规模度量方法是针对非标准化的文档，且在组织内、不同组织之间其功能详细程度都有所不同。这意味着比例系数需要在本地校准。在本指南中，“本地”意味着定义近似方法比例系数的环境，即代表近似方法的应用环境。

本地化的更多指导步骤参见第 13 章。

1.2.4 按分类和比例系数缩放得到近似规模

分类：将每个实际需求分类并指定一个规模值（即使用缩放比例系数），此规模值即代表这一需求的 COSMIC 功能规模。

分类的通常做法是将待度量的实际需求逐个归类为一个预先定义的类别（或参照模块），每个类别的规模都经过了校准且以 CFP 为单位，即每一类有其自身的比例系数。这样的话，实际需求的每一部分都根据其分类被指定了一个规模值。

所有使用分类的近似方法都建议提供一套客观的规则、标准或典型例子，来帮助进行正确的归类。第 3 章介绍了“固定规模”分类法，第 4 章的分类是“相等规模带”。第 7、8、9、10 章介绍的方法对需求进行了各种角度的分类。

1.2.5 近似规模度量的准确度

任何近似规模度量方法都是度量简便性和速度与准确度损失之间权衡的结果。因此，每种方法的估计误差都应该被记录和报告。更多指导参见第 13 章如何确保准确度。

1.3 实际需求的文档化层级

在只能得到近似 COSMIC 功能规模的情况下，度量人员应当了解所使用的软件文档制品的层级。

所有近似方法都有一个问题：无法找到一个比功能处理层级更高的、可以清晰定义文档化标准的层级。更高层级的文档，可能叫“用例级”、“组件级”、“子系統级”。但这些层级都只能在本地定义清楚，并常常需要给出示例。

此外，研究显示：度量人员（尤其是初学者）经常没有意识到实际需求是在不同层级描述的，且/或无法区分层级。这是度量功能规模时最常遇到的问题，不管是精确度量还是近似度量。

第 12 章主要讲解在实际需求描述处于不同层级文档的情况下，进行度量时要考虑的方面。我们鼓励度量人员在应用任何近似方法之前，特别注意阅读该章节。

1.4 实际需求的质量

对实际需求各个部分的质量等级进行评定，可以帮助确定近似规模度量的准确度，质量等级划分可参考《度量精确性指南》[33]里提供的方案。该质量等级划分方案定义了五个级别，在本指南中我们追加了第六个等级：未提到（未知的未知）。

表 1.2 列出了这六个等级和定义，以及可以应用哪些近似方法。

功能处理质量等级	功能处理定义的质量	可以使用的近似规模度量方法
完全定义	功能处理和其数据移动都有完整定义	使用标准 COSMIC 功能点方法
已文档化	功能处理有描述，但其数据移动描述不清楚	See Chapters 2 – 7 见第 2-7 章
可识别	列出了功能处理，但没有其数据移动的详细描述 ²	见第 2、5、6、7 章
可计数	给出了功能处理个数，但没有详细描述 ²	见第 2、5、6、7 章
隐含的（已知的未知）	没有明确提到该功能处理，而是隐含的	见第 2-6 章，或使用经验判断（见 1.2.1）
未提及（未知的未知）	当前完全不知道该功能处理的存在	基于专家经验（见 1.2.1）或过往经验（见 14.2），为“范围蔓延”增加应急储备

表 1.2 – 功能处理质量等级与相关近似方法

对于本指南，我们**强烈推荐**度量人员不要拿着里面的近似方法直接当作“菜谱”一样操作。要记得：

² 第 2-7 章描述的方法都是在“推算”层级测定规模，第 8 章的方法是在“经验判断”级测定规模，如 1.2.2 节所述。

1. 仔细检查待度量的实际需求，了解其文档化层级、需求的完整性和质量，然后再应用近似方法开始度量；
2. 尝试从软件相关专家那里获得更多需求细节，至少让你可以列出所有功能处理的名字；
3. 使用本地需求和度量结果验证该近似方法（将精确规模和近似规模做对比），确保在你们的环境中得到的规模结果足够准确。如有必要，在本地使用自己的实际数据校准该方法。见第 13 章如何本地化。

1.5 本指南所述近似方法的适用性

本指南介绍的大多数实际经验是对第 2、3、4 章近似方法的应用，度量的是新开发业务应用类软件的实际需求规模。

对于实时嵌入式软件：第 4 章的一个案例介绍了对一个非常复杂的航空电子软件应用近似方法的结果。第 12.1 节有一个案例是关于一个复杂的电信软件架构的近似规模度量，该架构的文档描述详细程度处于多个层级。

本文介绍的近似方法可以应用于如下实际需求：

- 新开发软件和对既有软件的功能增强，后者需包含完整的新功能；
- 任何领域的软件，比如业务类、实时嵌入式或基础设施类。

我们还未看到任何将近似方法应用于包含大量变更（新增、修改和删除）的功能增强型软件的度量报告。但是，近似方法可以用于这类功能增强的规模度量，只是需要进行慎重的校准。更多信息见 13 章。

为了实际应用，我们将近似方法分为三部分：

第一部分 项目需求阶段的近似方法，当需求不含“未知的未知”时。

第二部分 项目可行性阶段的近似方法，当需求完整性尚待考虑时。

第三部分 仍处于研究中的近似方法。截至我们编写本指南之时，这些方法还没有实际使用的公开报告。

第四部分 对一些共通问题进行更详细的讨论。

第一部分

项目需求阶段的近似方法

2.功能处理均值法

近似方法的起源和原理

功能处理均值法在 COSMIC 方法 2.2 版[8]里被首次提出。这个方法是获得软件块近似规模的最简方法。可以用于软件的实际需求只详细到功能处理级但尚未达到数据移动级的情况。

A 确定比例系数

1. 抽取一组实际需求的样本，其功能处理和数据移动已被详细定义，并且其特征与待度量软件的实际需求相似。
2. 识别样本需求的功能处理。
3. 使用标准 COSMIC 方法精确度量样本需求功能处理的规模。
4. 确定样本需求的功能处理平均规模（CFP）（例如，平均规模 = 8CFP）。“8”即是该方法的比例系数。
5. 计算标准差。

B 使用比例系数进行近似度量

1. 识别并统计待度量软件实际需求的所有功能处理个数（比如，有 40 个功能处理）。
2. 对于一组需求来说：其功能规模的近似估算结果为（功能处理数 x 比例系数）= $40 \times 8 \text{ CFP} = 320 \text{ CFP}$ [1]。
3. 对于某一具体需求来说：其近似规模在平均值 ± 1 倍标准差的范围内。

根据以上步骤，如果平均值是 8CFP，标准差是 2CFP：

- 此功能处理的规模范围为：[6,10CFP]；
- 40 个功能处理的规模范围为：[240, 400CFP]

适用性和使用报告

在已经开始进行 COSMIC 度量实践的组织里，这个方法被用来初步计算出一个大概的规模近似值。

2005 年，据 Vogelezang 报告[10]，不同行业的功能处理的平均规模值都不一样，也就是所说的比例系数不同。因此我们建议使用近似度量方法应该进行本地校准。

优点和缺点

优点：容易使用。

缺点：

- 功能处理均值是（假设是）领域相关的。
- 这种方法需要对功能处理平均值进行抽样和计算，而且必须基于相同的本地定义的度量元（见第 13 章）。而这一信息通常还无法得到。

建议应用范围

只要有充分的理由证明用于计算功能处理平均规模的样本是代表了所要进行近似功能规模度量的软件，那么这个近似度量值就是有效的。详见第 13 章。

因为这个原因，比较好的实践做法是从数据集中把与待估算需求不相似的数据剔除。同时也要注意：该方法在对称分布的数据集中效果最好，且标准差（ σ 值）最好明显小于功能处理规模均值。

研究进展

2009 年，Van Heeringen 等人对采用功能处理均值法的度量结果和采用标准 COSMIC 方法的精确度量结果的准确性进行了比较。在[11]中，他们对比了来自不同组织的 24 个软件块的近似度量值和精确度量值。

2013 年，De Marco 等人报告了采用这种方法进行网站应用开发工作量估算的结果非常好[12]。

2014 年，Del Bianco 等人对该方法做了实验性评估，该近似方法没有达到理想的预测能力[13]。通过计算样本的普通最小二乘公式进行估计，可以提高其预测能力。他们还提交了一个类似的方法，基于包含的数据组个数而不是数据移动个数。该方法的详情见 10.2 节。

2019 年，Lavazza 和 Morasca 得出结论：该方法一般可以在早期，快速得到一个合理的近似规模值，但在一些案例中它的误差过大，不可接受[14]。

功能增强类项目的实践应用

大多数项目不只是需要开发新的功能处理，也需要修改现有的功能处理。我们已观察到以下实践做法：

1. 对一个项目进行度量时，每个功能处理都被标注为“新增”或“修改”。
2. 对新建和修改的功能处理的规模均值是分别计算的。所得到的均值会随着领域不同而变化。但通常修改的功能处理规模均值是新建的功能处理规模均值的一半左右。

当在项目早期阶段进行规模近似度量的时候，新建和修改的功能处理需要分别被识别、计算并乘以其各自的规模均值。

3. 固定规模分类法

近似方法的起源和原理

固定规模分类近似法在 COSMIC 方法 3.0 版[15]的“高级&相关主题”里被首次提出。

此方法取决于从待度量软件块中识别功能处理规模的典型分类。然后为每个功能处理指定其对应类别的规模或比例系数。

需要对实际需求的描述进行分析以识别功能处理，并根据其规模分为三个或者更多类别，比如小、中、大。表 3.1 展示了在某个商业组织中实际使用的一组规模分类示例。表格中的行代表此组织的三种规模分类，以及每类功能处理对应的 CFP 数（比如，如果识别了一个小的功能处理，其指定的比例系数为 5，那么其规模就是 5CFP）。为了迫使度量人员认真地选择规模，不同分类之间的跨度要足够宽，设为 5CFP。

表格中的四列（#E、#X、#R 和 #W）解释了每类功能处理指定的 CFP 数是如何组成的。例如，小型功能处理是假设其包含一个输入、一个读、一个写和一个输出。对于中型或大型功能处理，则假设其包含更多数据移动。第五列的“错误消息”代表错误/确认消息的一个输出。

分类	规模 (CFP)	#E	#X	#R	#W	错误消息
小	5	1	1	1	1	1
中	10	2	2	3	2	1
大	15	3	3	4	4	1
...						

表 3.1 – 固定规模分类表[15]

如果需要在开发过程的早期对需求进行近似度量，那么首先将需求拆分成一个或多个功能处理，然后指定每个功能处理的规模分类和对应的近似值。使用表（如 3.1）可以帮助度量人员更快地为功能处理指定规模分类。必要时，可以将此表格进行扩展，包含更多分类，比如：“非常大：20CFP”。如果校准得当，此方法可以提供比第 2 章的平均规模近似法更准确的功能规模。

适用性和使用报告

这种方法在荷兰已经被一个大型商业组织广泛使用并取得成功。除了这个组织以外，还没有其他关于这种近似度量方法的使用和准确度的公开信息。

优点和缺点

优点：

- 容易使用。

- 实施方法简单。
- 比例系数有记录，即可以验证。
- 因为近似规模值是根据兴趣对象（也就是数据移动）的预计数量得到的，知道这个因素可以帮助度量人员决定为功能处理分配哪个类别。

缺点：

- 规模分类的定义是（假设是）与领域相关的。
- 把功能处理分配一个规模分类，是这种近似度量方法的一个主观因素，会削减其优势。参见第 1 章。

建议应用范围

只要有充分的理由证明分配的规模类别代表了所要度量的软件，那么这个近似度量值就是有效的。详见第 13 章。强烈建议定义客观的本地化规则，以协助度量人员进行正确分类。

研究进展

2000 年，Santillo 深入研究了不同量级的功能处理分类，即在量级上呈递增顺序的：功能处理、大型处理、巨型处理。在此方法中，每一类处理又被分为小、中、大，只是间距未知（而非本方法中的间距已知）。这种方法发展为早期快速近似法，详见第 8 章[31]。

2019 年，Lavazza 和 Morasca 实验了固定规模分类法的两种替代方案[14]：等量带法（见 11.5 节）和等距带法（见 11.6 节）。

等量带法中，来自参考数据集的功能处理先被排序，然后划分为功能处理个数相同的几个区间带。每个带的规模估算值就是其内功能处理的平均值（详见 11.5 节）。

等距带法中，先算出参考数据集中最小的功能处理和最大的功能处理规模之差，将这个差值划分为等量的几个区间带。每个带的规模估算值等于其内最小和最大功能规模的平均值（详见 11.6 节）。

Lavazza 和 Morasca 认为这些替代方案的预测效果比固定规模分类法更好。根据他们的总结，总体来说：基于区间带的方法比功能处理均值法更准确，只要做分类的人有足够的技能并且区间带个数合理。

4.相等规模带法

近似方法的起源和原理

相等规模带法在 COSMIC 方法 2.2 版[8]里被首次提出。

在相等规模带法中，功能处理被划分为几个规模带。使用校准过程确定规模带的区间，使得每个规模带内功能处理的累计规模是相同的。

A 确定比例系数

1. 确定一个实际需求的样本，其功能处理和数据移动已详细定义，其特征与待度量软件的实际需求相似。
2. 识别这些样本需求的功能处理。
3. 使用标准 COSMIC 方法精确度量这些样本需求的功能处理的规模。
4. 将所有功能处理从小到大排序，以图形方式展示其累加规模。
5. 使用累积分布图，将样本划分成若干个总规模相等的区间带（包含的功能处理个数可能不同）。比如：如果分成三个区间，那么每个区间中所有功能处理规模累计将占被度量软件总规模的 33%。
6. 确定各区间带功能处理的平均规模（单位为 CFP），作为每个区间带的比例系数。算出的比例系数通常都不是整数。

B 使用比例系数进行近似度量

1. 对每个将要近似度量的功能处理，识别其属于哪个规模带。
2. 将此规模带的比例系数分配给该功能处理。
3. 将所有功能处理的近似规模累加起来，得到整个软件块的功能规模近似值。

适用性和使用报告

Vogelezang 和 Prins 报告了用 37 个业务应用类开发项目的度量数据进行校准的度量结果，每个项目的总规模都大于 100CFP[16]。他们决定使用 4 个规模带来区分相对较小的功能处理、中等规模的功能处理、大的以及非常大的功能处理。来自 37 个项目的 2427 个功能处理被分布在四个带，每个带的规模均值如下：

规模带	功能处理的平均规模	功能规模累加占比%	功能处理个数占比%
小	4.8	25%	40%
中	7.7	25%	26%
大	10.7	25%	19%
非常大	16.4	25%	15%

表 4.1 – 由 37 个业务应用程序得到的 4 个等规模带示例[16]

同样的方法被用于校准一个重要的实时航空电子系统（总规模 10875CFP）中的一个组件，结果如下：

规模带	功能处理的平均规模	功能规模累加占比%	功能处理个数占比%
小	5.5	25%	49%
中	10.8	25%	26%
大	18.1	25%	16%
非常大	38.8	25%	7%

表 4.2 - 航空电子系统主要组件的 4 个等规模带示例

请注意：尽管软件类型完全不同，但四个规模带的功能处理数量是相似的。然而，各个规模带的功能处理的平均值是不同的，特别是：大和非常大的规模带。这反映了规模在本地校准的必要性。

度量一个新软件块的规模：首先识别软件的功能处理，按“小”、“中”、“大”或“非常大”进行分类。下一步，使用每个规模带的平均规模（如上文所列，但最好是在本地校准）分别乘以对应的功能处理的数量，得出近似度量的规模总数。

这种度量方法的优势在于：在完成某个新软件的近似规模度量后，度量人员可以检查一下新软件每个规模带的功能处理的规模之和占比是否接近“相等规模带”法里设定的 25%。如果是的话，说明对于这次新的度量来说校准值很适合；如果不是，那度量人员需要考虑一下校准值是否足够准确。

规模分类校准过程的准确性对这种方法非常关键，直接影响到规模度量的准确性，因为功能处理的规模通常表现为偏态分布，正如以上两组数据所示。换言之，软件系统一般都有大量小规模的功能处理和少量较大规模的功能处理。因此，为了获得准确的总规模，应该更关注如何准确的度量占比较少的“大”规模甚至更少的“非常大”规模的功能处理。

优点和缺点

优点：

- 容易使用。
- 此方法已经被证实对业务应用类和实时领域的软件都适用。

- 规模带划分的越多，近似结果越精确。

缺点：

- 注意选择规模带的数量，应保证规模带的间隔足够大。
- 规模带划分的越多，给功能处理指定一个分类时越可能出错。因此，规模带划分需要保证度量人员能够正确识别。
- 在确定规模带之前，需要精确度量大量的功能处理。
- 在进行近似规模度量时，给每个功能处理分配一个分类是个主观判断。
- 当“非常大”规模带里只有很少的功能处理时，度量这个规模带的平均规模需要非常谨慎，因为此规模带的比例系数可能造成最后的近似规模有很大差别。

建议应用范围

此方法推荐用于近似度量那些功能处理规模分布有严重偏态的软件规模。

只要有充分的理由证明分配的规模分类代表了所要近似功能规模度量的软件，那么这个近似度量值就是有效的。详见第 13 章。

强烈建议定义客观的本地化规则，以协助度量人员进行正确分类。

从以上结果我们可以得出结论：偏态程度越大，与第 2 章和第 3 章的方法相比，用此方法进行度量的准确度优势就越大。

研究进展

2009 年，Van Heeringen 等人对 24 个来自不同组织的软件块的近似度量值和精确度量值进行了比较[11]。结果是：近似度量规模和采用标准 FSM 度量方法得到的规模平均来说只相差 1.26%。这意味着对于本研究中 90%的值，相等规模带法相对于标准 COSMIC 方法得到的结果偏差在-15%到 +25%范围内。

2012 年 Valdes Souto 和 Abran 使用模糊逻辑模型对 Vogelezang 和 Prins 在 2005 年完成的一项研究[16]进行了测试[20]，该研究是对 C-注册系统进行功能规模近似度量的案例分析。这个测试显示如果功能处理能像 C-注册系统案例分析里一样被清楚的表述并记录下来，那么相等规模带法比实验中采用的 EPCU 模糊逻辑模型法更好。关于 EPCU 模型的介绍见 8.2 节。

2013 年，在 Almakadmeh 的博士论文“关于改进软件功能规模近似度量比例系数的框架研发”里，他声称将相等规模带法和 COSMIC 度量准确性保证指南错误!未找到引用源。里的质量评定机制相结合可以设计出一个可靠的近似度量框架。

2016 年，Luigi Lavazza 和 Sandro Morasca 对历史数据集和待度量新软件的功能处理规模分布之间的相关问题进行了分析。他们的报告显示：如果用于校准的历史数据集分布与新软件的分布是相似的，那么相等规模带法和功能处理均值法的准确度基本一致。这一结果说明使用相等规模带法存在一个风险：在给新软件的每个功能处理进行分类之后，可能发现分类工作是无意义的，因为功能处理均值法可能给出相同的结果，但更快且成本更低。相反地，如果分布是不同的，那么相等规模带法给出的结果肯定更准确。

2019 年，Lavazza and Morasca 发表了一份对早期和快速规模估算法的经验评估报告[14]。总体结论是：基于区间带的方法比功能处理均值法更准确，只要做分类的人有足够的技能并且区间带个数合理。

5.用例均值法

近似方法的起源和原理

平均用例近似度量方法在 COSMIC 方法 3.0 版[15]的“高级&相关主题”被首次提出。

本方法近似估算的原理类似于第 2 节的功能处理均值法，但应用于更高层级的文档，即用例中。

通过本地化校准，确定一个用例（本地定义）平均包括多少个功能处理，比如 3.5 个，一个功能处理的平均规模是 8CFP（如：第 2 章示例）。那么，用例的平均规模为： $3.5 \times 8 = 28$ CFP。

对于一个新项目，如果有 12 个用例，其软件规模就是： $12 \times 28 = 236$ CFP。

因此，在开发项目早期，通过这种校准方法识别用例的数量，可以为软件规模提供初始估算的功能点数。这种方法得到的近似规模的不确定性比第 2 章中所提及方法要高。这是因为比例系数 28 是另外两个比例系数（8 和 3.5）的乘积，而后两者是估算出来的。（因此这个结果这样表述更合适：比如 $240 \pm x\%$ ，其中 $x\%$ 可以通过适当的分析获得）。

适用性和使用报告

关于这种近似度量方法的使用报告没有文献记载。

优点和缺点

优点：

- 如果在本地有关于用例定义的标准，特别是对用例的颗粒度级别具体描述的规定，本方法使用起来会比较容易。

缺点：

- 该方法的问题是：不同的组织和对“用例”这一概念的理解不同，所以一个用例包含的功能数量可能有很大差异[13]。有证据显示：如果组织内没有用例编写的相关标准、无法保证用例规模的一致性的话，该方法是行不通的。
- 由于此方法比例系数是另外两个比例系数的乘积，而另外两个比例系数是通过估算得到的，这增加了近似度量结果的不确定性。
- 用例的平均规模需要在本地校准，校准需要充分的历史数据。因为用例是非标准化的，验证历史数据的同质性十分重要。

建议应用范围

只要有充分的理由证明分配的用例平均规模类别代表了所要度量的软件，那么这个近似度量值就是有效的。

同时需注意：该方法在对称分布的数据集中效果最好，且标准差（ σ 值）最好明显小于功能处理规模均值。

研究进展

2013 年，Gencel 和 Symons 在一个大型的软件工厂中进行的实践研究 [23] 表明，不同部门对于什么是用例有非常不同的看法。其中一部分人对于每个用例所含功能处理比例的看法非常一致，而另一部分人对这个比例的分歧非常大。在采用这种近似方法进行本地校准时，应该考虑此发现。

2018 年，Ecar 等人提出了 COSMIC 用户故事标准，以便帮助克服对用户故事的不同理解。他们给出了一个用户故事模板并进行了测试，该模板可以辅助对用户故事的度量，并且保证其描述在同一层级[24]。

6. 功能规模度量模式

近似方法的起源和原理

功能规模度量（FSM，即 Functional Size Measurement）模式[25]在 2016 年首次被提出，旨在帮助缺乏经验的度量人员快速学习应用 COSMIC 方法，帮助他们建立方法规则和度量结果之间的映射关系。

COSMIC 专家已发现：度量结果中的一些模式会反复发生。在[25]中，对此有正式定义：

定义 – FSM 模式

为解决某一特定背景下重复发生的度量问题而预定义的通用软件模型。

原则 – FSM 模式

某一给定软件领域的功能用户需求经常呈现出一定的模式。因此对其功能规模的度量也遵循一定的模式。

其中定义了四种模式，分别是：“微型 FSM 模式”，“基本 FSM 模式”，“复合 FSM 模式”，“多元复合 FSM 模式”。见图 6.1。

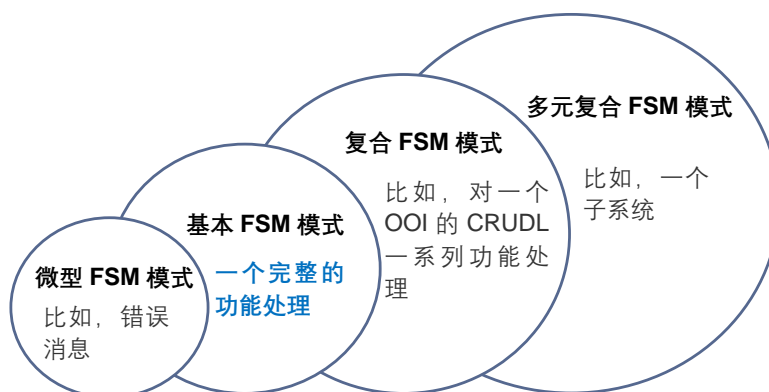


图 6.1 – 四种度量模式

微型 FSM 模式： 功能处理的一个片段，包含一个或多个数据组。例子：显示一条错误消息。

注意：此片段在做 COSMIC 详细度量时没有其他单独的实例，除了度量变更时。

基本 FSM 模式：一个完整独立的 COSMIC 功能处理。

复合 FSM 模式：一组基本 FSM 模式，组合起来代表更高层级的功能。一个复合 FSM 模式由若干功能处理组成。对某个或多个兴趣对象（OOI）数据进行维护的 CRUDL（新增、查询、修改、删除、列表）一系列功能处理，是复合 FSM 模式的一个例子。

多元复合 FSM 模式：一组复合/基本模式，其功能有一定联系。一个多元复合 FSM 模式由多个功能处理组成，处理若干兴趣对象的数据。在业务应用软件中，一个多元复合 FSM 模式可以代表整个模块、或分布式应用的一个组件、甚至是整个应用程序。在实时嵌入式系统中，它可能是针对一个设备群的一组后端子系统功能。

适用性和使用报告

每种模式都描述了以下三项特征：

- **问题：**简要介绍本模式想要解决的问题；
- **背景：**介绍问题发生的背景情况；
- **解决方案：**介绍在这一特定背景下问题如何解决。

同时需要给该模式命名。

A) 微型 FSM 模式的例子：

模式名：显示简单的错误消息

问题：如何近似度量 FUR：显示一条或多条错误消息。

背景：功能处理对人类功能用户的输入信息有一个或多个校验。

解决方案：对于显示简单的错误消息：

功能处理	数据组	数据移动	功能规模（CFP）
<功能处理>	错误消息	X	1
总计：			1

a. 图例：E=输入；X=输出；R=读；W=写；

B) 基本 FSM 模式的例子：

模式名：简单的新建功能处理

问题：如何度量 FUR：对一个独立事件的响应处理过程，包含一个数据组的持久化和一个简单的数据是否存在的校验。

背景：信息系统中保存一个数据以便后续使用的功能很常见。FUR 一般如此描述：【功能用户】输入关于某<兴趣对象>的数据（1 个输入）并保存；【软件】确保此输入数据在系统中未录入过（1 个读），并将此次输入的实例持久化（1 个写）；如果无法保存或其他校验失败，显示错误消息（1 个输出）。

解决方案：对于简单的新建：

功能处理	数据组	数据移动	功能规模 (CFP)	备注
新建一个<兴趣对象>	<兴趣对象>	ERW	3	新建一个实例
	错误消息	X	1	
总计：			4	

- a. 图例：E=输入；X=输出；R=读；W=写；
b. 加粗的“E”代表此功能处理的触发输入。

C) 复合 FSM 模式的例子：

模式名：复合型 CRUDL-3OOI

问题：如何度量一段 FUR（比如 CRUDL），其处理过程与处理一个给定兴趣对象的一系列功能处理是类似的，只是处理过程涉及到另外两个兴趣对象。

背景：信息系统中常见一些 FUR 重复出现、软件预期行为相同，只是兴趣对象不同。此类 FUR 与前面的例子相比描述的更概要，一般只是笼统地描述：【系统】要提供对<兴趣对象 1>的新建、查询、修改、删除和列表功能。<兴趣对象 1>的【属性 X】与系统中既存的<兴趣对象 2>相关联，其【属性 Y】与系统中既存的<兴趣对象 3>相关联，属性均以列表选项形式展现在屏幕上。

解决方案：

功能处理	数据组	数据移动	功能规模 (CFP)	备注
新建 <兴趣对象 1>	<兴趣对象 1>	ERW	3	新建一个实例
	<兴趣对象 2>	RX	2	读取和展示选项列表
	<兴趣对象 3>	RX	2	读取和展示选项列表
	错误消息	X	1	小计：8 CFP
查询<兴趣对象 1>	<兴趣对象 1>	ERX	3	选择、读取和展示已保存的实例
	<兴趣对象 2>	RX	2	需要读取其 ID 以便展示其名称
	<兴趣对象 3>	RX	2	同上
	错误消息	X	1	小计：8 CFP
更新<兴趣对象 1>	<兴趣对象 1>	ERW	3	更新已保存实例
	<兴趣对象 2>	RX	2	读取和展示选项列表
	<兴趣对象 3>	RX	2	读取和展示选项列表
	错误消息	X	1	小计：8 CFP
删除 <兴趣对象 1>	<兴趣对象 1>	ERW	3	删除一个实例，先读取它，不涉及其他兴趣对象
	错误消息	X	1	小计：4 CFP
列出<兴趣对象 1>	<兴趣对象 1>	RX	2	读取和展示列表
	筛选条件	E	1	包含所有兴趣对象的筛选条件
	<兴趣对象 2>	RX	2	读取/展示列表（筛选的）
	<兴趣对象 3>	RX	2	同上
	错误消息	X	1	小计：8 CFP
总计：			36	本 FSM 模式

- a. 图例：E=输入；X=输出；R=读；W=写；加粗的“E”代表此功能处理的触发输入。

实时嵌入式软件中，与 CRUDL 等价的模式是对外部功能用户/兴趣对象的状态变换；每个状态的转换意味着一个触发事件，触发了一个单独的功能处理。

D) 多元复合 FSM 模式

模式名：多元复合模块-300I

问题：如何度量一个完整的模块，涉及到两个主要兴趣对象，其功能与类似模块的功能处理相同，只是涉及到另外一个次要兴趣对象、三个参数型兴趣对象和四个事务型兴趣对象。

背景：信息系统中常见一些模块重复出现，软件预期行为相同，只是兴趣对象不同。模块级的 FUR 可能笼统地描述：【系统】要有对<兴趣对象 1>和<兴趣对象 2>的“管理功能”（这里的“管理”等同于 CRUDL），同时也管理第 2、3、4 个参数型兴趣对象等等，直到覆盖该模块所有预期功能。

解决方案³：

FSM 模式	分类	功能规模 (CFP)	举例
CRUDL-300I	复合	36	管理“客户”
CRUDL-100I	复合	20	管理“销售代表”
CRUDL-100I	复合	20	管理“客户类型”
CRUD-200I	复合	22	管理“账户分龄参数”
CRUD-300I	复合	26	管理“发票参数”
CRUD-300I	复合	26	管理“现金入账参数”
事务处理-700I	基本	12	录入手工发票
事务处理-600I	基本	10	录入手工现金入账
事务处理-800I	基本	14	录入对发票或现金入账的调整
报告-300I	基本	7	客户销售情况报表
报告-400I	基本	9	客户分龄报表
报告-500I	基本	11	客户对账单
里程碑-200I	基本	10	应收账款月结
	总计：	223	整个 FSM 模式

优点和缺点

优点：

- 有助于减少度量工作量。
- 可帮助相对缺乏 COSMIC 经验的使用者。
- 帮助避免共通的度量错误，提供更准确的规模度量结果。
- 这一系列 CRUDL 的各种组合模式，在给出“简单”、“一般”、“复杂”几种功能处理的规模后会更详细。另外，“报告”也被加入到 CRUDL 系列里。这组模式的早期版本是在[27]中记载的。这一经验告诉我们：早期估算的需求可以通过在组织内形成一套基本和复合级的标准模式得到解决。如果需要估算多元复合型 FUR，其规模可以由低层级模式的本地数据累计而成，并参考经验判断。

³ 请注意：这里对此类模块只有一个概要描述。完整描述参见[25]。

- 有一套定义完善、且基于组织内部软件需求特征调整过的模式，将是对本指南第 8、9 章所述方法的高级应用（定义完善的模式也可以与这些方法融合使用）。模式能够改善早期规模估算的可重复性。

缺点：

- FSM 模式及其应用，还未收到与 COSMIC 方法对比的评价数据。需要更多案例研究。
- FSM 模式需要描述的很清晰，以便缺乏经验的度量者使用。

建议应用范围

本方法适合于如下类型软件：业务和实时类软件，且指定了具体的领域。

研究进展

2017 年，关于本方法的实验开始在波兰农业重组和现代化机构展开。

基于各种背景的 COSMIC 支持工具和模式定义的研究和开发一直在持续。FSM 模式需要描述的很清晰，以便缺乏经验的度量者使用。

第二部分

项目可行性阶段的近似方法

7. 软件冰山近似法

近似方法的起源和原理

本近似方法由 Abran 和 Vedadi [42]提出，是基于：

- A. The iceberg analogy 冰山类比
- B. ISO-IEEE standard 29148 on Requirements Engineering. ISO-IEEE 29148关于需求工程的标准

如图 7.1 所示的软件冰山类比中，水平面上方冰山的可见部分（左）好比是早期的软件需求描述，这些需求逐渐细化，直到最后软件需求所有细节都完全显露出来（水平面以下的冰山部分逐渐变得可见）。



图7.1 软件冰山类比：从最初可见的功能（左）到全部可见（右）

在物理学中，根据大量的经验测量和科学观察，浮式冰山的水下部分与水上部分重量之比是众所周知的一个常数。

在软件开发中，虽然没有已知的恒定比率，但根据众多 COSMIC 案例的观察和度量数据发现，在本地环境中，经过一系列步骤就能得出适用于本地的近似比率。

在所有软件项目中，功能可见性在整个开发生命周期中都会变化，因此描述软件功能的文档也会随生命周期阶段变化。

ISO/IEC/IEEE 29148 需求工程标准定义了在整个系统和软件生命周期过程中的需求各种来源、类型和详细程度的相关概念。

最初的需求源于两方面：业务干系人和其他干系人，这些都是“系统”需求。系统的功能性需求，一部分分配给软件需求（以及硬件需求，有时甚至是手工操作步骤）。这些来源提供了系统背景需求，包括：系统目的、系统范围和系统概览。根据这些上下文信息，可以识别下列需求：

- 系统功能性需求（其中一些将分配给软件），
- 系统非功能性质量要求（其中一些将分配给软件）。

ISO-IEEE 29148还指出，除了清晰识别出了一些软件功能外也识别了接口，但接口尚未明确定义，同时也包括一些仍然较概要的质量需求。

适用性和使用报告

在 Abran 和 Vedadi [42]的报告中，这些概念都适用于 COSMIC 的两个案例分析：

1. Course Registration System CRS（课程注册系统 CRS）
2. RestoSys（Resto 系统）

在这些 COSMIC 案例分析中：

A)文档有以下三种详细水平：

- Level 1：（业务）功能（“系统功能列表”）。
- Level 2：（业务）功能分配到软件功能处理（“软件功能”列表）。
- Level 3：软件功能处理的具体功能描述（分配到软件的详细功能描述）。

B)功能分类

案例分析中对功能处理的每条功能描述都被归类为以下五类之一（对应案例分析的文档详细水平）：

- a. 从业务需求分配到软件的功能 - Level 2
- b. 来自业务需求的更详细的功能 - Level 3
- c. 实现业务需求的操作级功能 - Level 3
- d. 源自系统需求且分配到软件的功能 - Level 2
- e. 与其他软件系统接口相关的功能 - Level 1或2

图 7.2 展示了 CRS 案例中的 COSMIC 规模分布。

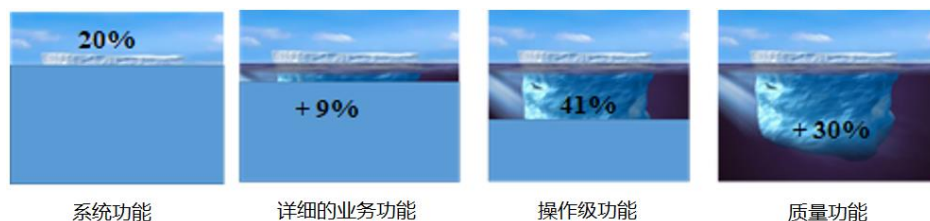


图7.2 CRS案例研究 – 功能处理的比例系数[42]

此冰山比率可以用作生命周期各个阶段和不同层级文档的比例系数。

举例来说：如果 COSMIC 度量是在生命周期的非常早期进行的，那时只有系统功能可以被识别和用于 COSMIC 度量，那么 20%的比例系数可按以下方式使用：

如果在那一时刻，系统功能分配到软件部分的度量规模是 30CFP，并且配有功能描述的文档，那么此规模只占最终规模的 20%：即是说此功能描述的缩放比例是 1：5：

- 即在生命周期早期度量的 1CFP，可能代表最终的 5CFP。

因此，30CFP 的规模，以 1：5 的比例推算，将在开发完成后变成 150CFP。

建议应用范围

在一个新系统开发的最早期阶段，项目被正式定义前，即 ISO-IEEE21948 所定义的系统级，需求可能只有一个大致轮廓。在这个阶段，可以通过其他现存软件的已知大小（比如 CRS 或 Resto 系统案例分析），使用冰山近似法来确定新软件的近似大小，但是要使用本指南中描述的其他近似度量方法就为时尚早。

在[42]中展示的比例系数只针对对应的案例，但冰山近似法可以适用于绝大多数组织，只需要收集历史项目、对功能进行分类并对文档详细程度进行识别。

8.早期&快速近似法

近似方法的起源和原理

COSMIC 早期&快速近似法是对早期&快速功能点法[28]的改编版本。之后，早期&快速度量法将应用领域扩展到了 COSMIC 方法[31]。这一方法利用了来自本地或全球度量数据库（比如 ISBSG 基准数据库和其他数据库[10][16]）的数据做了优化。

COSMIC 早期&快速近似法将比例系数和分类法融合在一起。该方法允许系统各分支的文档描述在不同详细水平，分支也可以在不同分解层级。总体规模近似值（该值是三点估算的结果，即最小、最大和最可能值）等于每个组件近似规模之和。

COSMIC 早期&快速近似度量方法，要求度量人员能够将实际需求归类为一些特定的功能类别。然后，度量人员参考适合的换算表，找到对应的 CFP 平均值（根据标准 COSMIC 方法，这一步骤应在软件架构的每一层分别进行）。按照量级递增和组成元素数量递减的顺序，功能处理分为：功能处理、典型处理、大型处理或巨型处理。

- a) 在早期&快速近似法里，功能处理⁴（FP，即 Functional Process）是最小的处理，可以被划分为小、中、大或非常大，取决于它的数据移动的估计数量。这种分类与第 3 章中讨论的“固定规模分类法”类似。
- b) 典型处理（TP，即 Typical Process）是四种基本的用户操作的组合：对某一具体兴趣对象的数据进行创建（Create）、检索（Retrieve）、更新（Update）和删除（Delete），即 CRUD。这些典型的处理经常出现在业务应用类软件中。
- c) 大型处理（GP，即 General Process）是一组中等规模的功能处理的组合，可以看作是应用程序的操作级子系统。根据所包含的功能处理的估计数量，GP 可以分为小、中或大。
- d) 巨型处理（MP，即 Macro-Process）是一组中等大小的通用处理的组合，可以看作是用户环境中整个信息系统的一个完整子系统。根据所包含通用处理的估计数量，MP 可以分为小、中或大。

每一层都是在前一层的基础上定义的。度量人员参考适合的换算表，找到对应的 CFP 平均值。换算表参见 7.2 节。

⁴ 早期&快速近似度量方法里对功能处理的定义与标准的 COSMIC 方法不同。计划在未来版本里采用 COSMIC 标准定义。两种方法中的定义将会是一致的。

为了算出估计值，度量人员必须（在完成标准 COSMIC 方法的最初几步——定义应用的边界、层和度量范围后）⁵将各部分的实际需求归类到以上某一分类的某一层级。从换算表中找出其对应的规模值。通过这种方法，不仅可以量化功能树的叶子节点，还可以量化中间分支。

适用性和使用报告

COSMIC 早期&快速近似度量方法要求度量人员能够把实际需求归类到所属的具体功能类别上。实际需求的每部分都要被归类，按照量级和组成元素数量递增的顺序，分为：功能处理、典型处理、大型处理或巨型处理。然后，度量人员参考换算表 7.1，为每个处理分配一个 CFP 平均值（在每个分解层级分别进行）。

最新发布的参考值参见表 8.1[31]。

类型	层级	功能范围/与 COSMIC 等价	最小 CFP 值	最可能 CFP 值	最大 CFP 值
功能处理 (FP)	小	1 - 5 数据移动	2.0	3.9	5.0
	中	5 - 8 数据移动	5.0	6.9	8.0
	大	8 - 14 数据移动	8.0	10.5	14.0
	非常大	14+ 数据移动	14.0	23.7	30.0
典型处理 (TP)	小	CRUD (小/中型处理) CRUD + 列表 (小处理)	15.6	20.4	27.6
	中	CRUD (中/大型处理) CRUD + 列表 (中型处理) CRUD + 列表+ 报告 (小型处理)	27.6	32.3	42.0
	大	CRUD (大型处理) CRUD + 列表 (中/大型处理) CRUD + 列表 + 报告 (中型处理)	42.0	48.5	63.0
大型处理 (GP)	小	6 - 10 一般的 FP	20.0	60.0	110.0
	中	10 - 15 一般的 FP	40.0	95.0	160.0
	大	15 - 20 一般的 FP	60.0	130.0	220.0
巨型处理 (MP)	小	2 - 4 一般的 GP	120.0	285.0	520.0
	中	4 - 6 一般的 GP	240.0	475.0	780.0
	大	6 - 10 一般的 GP	360.0	760.0	1,300

表 7.1 - 早期&快速方法功能分类的估计值

对实际需求的每一部分进行归类，尤其是归类为比功能处理更高的级别，是非常主观的。该方法给出了辅助正确分类的详细指南[30]。本方法的准确度主要取决于度量人员把较为概要的文档进行归类的能力以及其接受的培训。

优点和缺点

优点：

- 当实际需求的大部分都不够详细、无法识别功能处理时，适用此方法。
- 可以处理文档详细程度不同、分解层级不同的需求。

缺点：

- 将功能处理进行归类，是个主观因素。

⁵ 注意：此方法用到的“最初几步”对应 COSMIC 方法的“度量策略”阶段，但不像 COSMIC 方法定义的那么严格。

- 本方法的不确定性只取决于估算者对需求归为表格中哪一类的判断能力，与校准过程无关。对于大型处理（“应用程序的操作级子系统”）和巨型处理（用户环境中整个信息系统的一个完整子系统）的定义，不同的度量人员可能有不同的理解。
- 不是为度量功能增强（功能变更）的近似规模而设计的。

建议应用范围

本方法最适合应用于实际需求的详细程度无法识别功能处理的情况。本方法应谨慎使用，且需提供方法相关培训。

研究进展

方法提出者定期对方法进行改进，包括校准各分类元素的权重，以及如何报告整个程序的结果。

Almakadmeh 报告了一个实验[8]，评价该方法的可再现性和准确性。实验结论是：当在实验背景下由不同度量人员对同一组需求进行近似度量时，其结果的可再现性很低。

其他背景下进行的实验结果也有陆续报告[32]。该方法可以应用于差别很大的分解层级和文档详细程度，其不确定性在分解层级和文档化程度较低时变高，反之变低。

9.应用模糊逻辑进行近似度量—EPCU 模型

近似方法的起源和原理

2012 年，Valdés 等人提出了一个采用模糊逻辑模型的方法，被称作不确定环境下的项目估算——EPCU 模型，该方法不需要使用本地历史数据也可以进行规模近似估算[20]。

EPCU 模型需考虑如下方面：

- 估算专家在估算领域（此案例中为近似度量）采用的基于经验的语言变量，以及
- 专家将这些语言变量和近似功能规模度量结合起来的方式。

Valdés 的报告称：在实际试验中，大多数项目的 EPCU 估算过程比采用“专家判断”的估算方法明显要好很多[34]。另外，值得注意的是 EPCU 模型能够进行系统的复制：无论给输入变量分配值的人处于什么技能水平，EPCU 模型产生的估算结果都比“专家判断”方法偏差更小，这在给规模分配等级存在内在主观性的时候特别有用。

使用 EPCU 模型的六个步骤：

1. 识别输入变量；
2. 输出变量（即估算的功能规模）的规格说明；
3. 生成推理规则；
4. 模糊化；
5. 推理规则的评估；
6. 去模糊化。

前三步是关于估算过程的配置和建立估算流程，生成“EPCU 环境”。EPCU 环境是“一组输入和输出变量，及作用于一个指定项目或者一组类似项目的关系式[20]。

这种方法用于软件开发项目的早期阶段，即大多数实际需求还是用自然语言编写的时候。因为这时的实际需求还不完全清楚，所以这种估算通常是在一个不确定的环境下进行的。

在 EPCU 环境中进行近似规模度量需要考虑如下两个输入变量：

1. 变量 1：功能处理规模，
2. 变量 2：功能处理移动数据的兴趣对象数量。

2014 和 2015 年，某实际项目用该方法进行了实验。提供给度量人员的实际需求是一份用例清单，这种情况在软件项目生命周期早期阶段很常见，即实际需求不够详细。从真实项目得到的数据分

析可以看出，EPCU 法得到的近似规模比相等规模带法更好，而两种方法得出的规模都比实际小 [21]。

对 EPCU 近似法的相关研究，在 FUR 文档的详细程度上，主要聚焦于两类：功能处理级，用例级。

适用性和使用报告

根据 EPCU 的定义，第一步是定义一组输入和输出变量及其关系式（EPCU 环境），EPCU 应用程序用于度量功能规模近似值，EPCU 环境定义用到两个输入变量，影响到待度量的功能处理或用例的规模，后者定义为输出变量。

输出变量定义为连续型数据，根据 Vogelezang 等人在基于相等规模带法的一个案例中的研究数据 [16]，输出变量，即规模上限（截止点）为 16.4 CFP。2015 年，Valdés 等人提出另一个版本的模糊逻辑规模近似法，增加了一个案例，其规模上限（截止点）为 44。

2017 年，Valdés 使用非参数检验对比发现：当文档详细到功能处理级时，EPCU 环境更接近实际规模分布。在该研究中，使用统计学方式展示了：

- 对于 180 各功能处理的样本，近似值分布（上限 16.4 CFP）与实际值分布（使用标准 COSMIC 方法）接近；
- 对于 293 个用例的大样本，近似值分布（上限 44 CFP）与实际值分布（使用标准 COSMIC 方法）接近 [19]。

鉴于该研究的发现，对于详细程度处于用例级的 FUR，可以建立 EPCU 环境，并且推荐截止点为 44 CFP。另一方面，当文档处于功能处理级，推荐使用截止点为 16.44 CFP 的 EPCU 环境。

从 2015 年开始，EPCU 近似法在墨西哥有广泛应用，并且作为墨西哥国家数据库的基本要素，建立了功能规模和工作量以及成本的关系。

优点和缺点

强项：

- 当实际需求的大部分都不够详细、无法识别功能处理时，适用此方法。
- 可以处理文档详细程度不同、分解层级不同的需求。
- 不需要本地历史数据就可以提供规模估计结果，尤其是目前大部分近似方法都需要使用历史数据进行校准，以便在本地环境得到更准确的结果。收集这些数据代价很大且耗费时间，而没有本地数据校准的此类方法基本没有价值。
- 研究显示出很好的度量结果，即使使用者不熟悉 COSMIC 方法。
- 在墨西哥得到了广泛且有效的使用。

缺点

- 应用这一方法包含一系列步骤，需要经过训练的输入，这使其在软件工程项目中的使用具有挑战性。

- 使用 EPCU 的简单方法是购买其商业版本，该工具由方法提出者研发。

建议应用范围

当实际需求（其中一部分）不够详细、没有描述出完整的功能处理或用例时，最适合使用该方法。此时只能识别出有哪些功能处理或用例，也就是在软件开发生命周期的早期阶段。

该方法也可以在没有历史数据库（用于校准近似法系数）的情况下使用。

研究进展

2012 年，Valdes Souto 和 Abran 在一份报告[20]中使用模糊逻辑模型对一个软件的功能规模进行近似度量，其 FUR 就如课程注册案例一样全部已知且描述完整。该研究显示：在相似情况下，相等规模带近似法比 EPCU 模糊逻辑模型的效果更好。

2014 年，有进一步研究专门针对早期近似度量，让参与者使用 EPCU 模型对一个只有用例名称列表的实际项目进行度量。这一案例研究证实：EPCU 近似法不需要本地校准，没有历史数据也可以使用。对于一个真实的项目、而非参考样例软件，EPCU 法得到的结果优于相等规模带法，而两种方法得到的规模都偏小。

2015 年，Valdés 等人提出另一个版本的模糊逻辑近似法[17]。其定义了输出变量是一个连续型数据，并根据 Vogelezang 等人研究[16]中的数据集，将输出变量的 Q4（四分之四位，即上限）定为 44 CFP。在该研究报告中，针对一个实际项目的用例级文档，截止点设为 44 CFP 的 EPCU 法[17]得到的结果优于相等规模带法和截止点设为 16.4 CFP 的 EPCU 法[21]，这二者得到的规模都偏小。另一方面，截止点设为 44 CFP 的 EPCU 法得到的结果更接近现实。

对 EPCU 规模近似法的研究主要集中在两类文档详细程度上[20]：功能处理级、用例级（使用两种 EPCU 环境定义）；但是，对于什么时候使用哪一种 EPCU 环境（EPCU16.4，EPCU44）、对每一类文档详细程度哪个环境能得到更好的结果，并不清楚。2017 年，Valdés 使用非参数检验进行了对比[18]，当文档详细到功能处理级时，哪一种 EPCU 环境更能代表实际规模分布。

用例没有标准定义，但我们观察到一个用例通常包含多个功能处理，所以当功能需求描述处于用例级时，截止点设为 44 CFP 的 EPCU 法能得到更好的结果这一结论听起来比较合理，而需求处于用例级也是业界常见情况。2017 年，一项类似的研究专门使用非参数检验进行了对比，在文档处于用例级时，是否截止点设为 44 CFP 的 EPCU 法更接近实际规模分布[19]。

基于[18]的发现，可确定的结论是：截止点设为 16.4 CFP 的 EPCU 法更适合详细到功能处理级的文档，截止点设为 44 CFP 的 EPCU 法更适合详细到用例级的文档。

10. 简单功能点近似法

近似方法的起源和原理

简单（EASY，即 EArly&SpeedY 的缩写形式）功能点近似度量方法在 2012 年被首次提出，其原理是基于一个较为概要的软件快速近似度量技术（SMART），用来度量模糊的实际需求[41]。在“SMART”方法里，对于任何功能，度量人员可以根据他/她对实际需求所描述功能的理解自行假定一个或者多个值的“概率”。

举例：这个报告“最可能”有 5 个数据移动（60%），但是它“可能”有 2 个额外的数据移动（30%），或者甚至有 4 个额外的数据移动（待确认）（10%）。那么这个功能的近似度量值是所有可能值的加权总和（权重就是相对应的概率。比如，在这个案例里就意味着近似规模为 $5 \times 0.6 + 7 \times 0.3 + 9 \times 0.10 = 6.0$ CFP）。（对于一个功能，所有项的概率之和必须是 100%）。

需注意的是最可能的值不一定总是位于“中间”。这取决于度量人员对所有可能的值分配的概率。这种方法不同于之前章节介绍的“平均”度量方法，在平均度量方法中“总是”将平均值或中间值当作最可能的值。

然而，“SMART”方法可能比较耗时，因为对于每个待度量的功能，度量人员要分配不只一个可能的值，而且要为每一个值分配相应的概率。

简单近似度量法为度量人员提供了最典型的概率分布以供其选择。该方法也允许近似规模和精确规模混合使用。（精确规模，就是根据标准度量方法所度量的规模，相当于给规模分配一个接近 100% 的概率值）。

下表 10.1 描述了在业务应用领域，最常见情况的规模近似值的典型概率分布（FP 代表“功能处理”）[41]。

功能处理类别	描述详细程度	CFP (最小)	CFP	CFP (最大)	近似 CFP	概率
小型 FP	少数未知	2 (10%)	3 (75%)	5 (15%)	3.2	>80%
小型 FP	未知（没有 FUR）	2 (15%)	4 (50%)	8 (35%)	5.1	<50%
中型 FP	少数未知	5 (10%)	7 (75%)	10 (15%)	7.25	>80%
中型 FP	未知（没有 FUR）	5 (15%)	8 (50%)	12 (35%)	8.95	<50%
大型 FP	少数未知	8 (10%)	10 (75%)	12 (15%)	10.1	>80%

大型 FP	未知（没有 FUR）	8 (15%)	10 (50%)	15 (35%)	11.45	<50%
复杂 FP	少数未知	10 (10%)	15 (75%)	20 (15%)	15.25	>80%
复杂 FP	未知（没有 FUR）	10 (15%)	18 (50%)	30 (35%)	21	<50%

表 10.1 – 业务应用领域近似值的概率分布

以上这几个功能处理案例的概率分布的不同选择，以及最小和最大 CFP 值的不同，会导致简单近似度量方法的不同应用结果。实时软件领域可以做出类似的分类表。

类似的方法、不同的规模值和概率设置，可以用于功能变更（针对功能增强型项目）规模的近似度量，变更的类别可以是从小到“大”。

适用性和使用报告

没有文献记载关于这种近似度量方法的使用报告。

优点和缺点

优点：

- 可以和标准度量方法混合使用。
- 可以被缩放至不同的文档层级。
- 适用于功能增强型项目（度量变更规模，而不是功能本身规模）。

缺点：

- 可能比较耗时（建立校准值和使用时）。
- 依赖于把模糊的实际需求映射到哪个“典型”案例上。

建议应用范围

随着需求描述的逐渐详细，此近似度量方法在实际需求的整个演变过程中都可以使用。

研究进展

正在收集使用数据用于验证和改进此方法。

第三部分

仍处于研究中的近似方法

11. 新兴的近似度量方法

本章节介绍的近似度量方法还处于早期开发阶段。我们相信这些方法在不远的将来有潜力发展成可用于估算的近似度量方法或工具。

11.1 对于非正式记录的文字需求的近似度量

对于非正式记录的文字需求的近似度量方法[33]是以对正式的需求规格说明进行自动化 COSMIC 功能规模度量为基础建立的：

- 首先要挑选大量非正式文字需求来对同一个功能处理进行描述，然后对其进行人工度量。
- 每个功能处理的文字需求和对应规模被储存在数据库里作为参考。
- 然后以整个数据集的四分位为边界，将所度量的功能处理分成四个模糊的规模等级。
- 然后对其进行文本挖掘，可以从数据集里面提取语言特征并形成文本分类算法，该算法可以自动地将一组新的文字需求分配至四个模糊规模等级之一。

优点：经过准备阶段之后，它可以对文字需求自动生成规模估算。试验已对不同来源的文本进行了测试。

缺点：不同环境里的语言特征不同，不同语言容易互相复制，这意味着对每个环境都要求本地校准。

11.2 基于数据组个数均值的近似法

2014 年，Del Bianco 等人报告称：根据他们收集的数据，每个功能处理的数据组个数均值比数据移动个数均值的预测效果更好[13]。他们报告称：在一个功能处理中，每个数据组的数据移动个数是个准常数，在该数据库中，以下公式可以给出很好的 COSMIC 功能规模估算：

$$\overline{CFP} = \text{每功能处理的数据组均值} * 1.8 * \text{功能处理个数}$$

使用该近似方法的前提是：组织中的 COSMIC 度量执行得很成熟，能够收集到足够得历史数据以便推演出该公式。

11.3 基于用例名称的近似法

2016 年，Ochodek 提出了一种基于用例名进行搜索的自动化近似度量方法，用于某一特定环境[35]。目前最新的用例编写指南都建议：将用例命名为可以准确表达用户目的的简单句子，句子中隐含一个主语。这样的命名可以被分析并归类为以下十三个类别中的一个：

• 检查对象	4.25 CFP
• 异步检索	5.10 CFP
• 删除	5.87 CFP
• 新建	7.01 CFP
• 动态检索	7.56 CFP
• 检索	7.84 CFP
• 变更状态	8.23 CFP
• 复杂内部活动	9.00 CFP
• 删除链接	9.00 CFP
• 转移	9.19 CFP
• 链接	13.70 CFP
• CRUD	17.60 CFP

要强调的是：这些类型是用例的特征，而不是单个数据移动的特征。这些类型中，有些含义是不言自明的，比如新建或删除。其他的则在目的上有细微的差别。

例如，复杂内部活动指的是含有复杂运算/处理过程的内例。通常，此类用例场景看起来很琐碎，但从 COSMIC 方法的角度来分析立刻就非常清晰：它们可能包含一个或多个输入数据移动（经常是对各种参数和输入数据进行处理），同样也可能包含读和写数据移动。

关于以上用例类型的详细说明参见[35]。

这些分类对应的功能规模是根据历史数据确定的。作者对分类的完备性和可识别性做了验证，论证了这些分类可以代表不同行业的各种用例。尽管大部分用例是遵循用例编写指南编写的，还是会发现一些用例的命名不符合规则。目前观察到有以下现象：

- 4%的用例名会有误导，其字面意思并不代表用例的场景。
- 还有 2%的用例名字叫 CRUD（新建、查询、修改、删除）或者其中一部分，但其实只是包含 CRUD 中的一个操作。

11.4 基于 UML 用例图的近似法

2017 年，Haoues 等人提出了一种基于从 UML 用例图提取的动作进行近似度量的方法[36]。尽管此方法的度量公式是基于网页和移动类应用场景提出并测试的，其概念对所有软件的 UML 用例图都通用。

该方法的提出是基于以下发现：一个功能处理的功能规模（functional size）的最大和最小值限定在如下范围内：

$$2 \leq FS(FP) \leq FS(A) + FS(S) + FS(E/C)$$

其中，2 是一个功能处理的规模最小值，其最大值取决于三者规模之和：用户的动作—FS(A)，系统的动作—FS(S)，错误处理和确认消息—FS(E/C)。

在调研用的数据库中，该方法度量出的规模保持在软件总规模的 77-100%。如果经过历史数据校准，该方法可以用于 UML 图的自动化近似度量。

该方法也可以应用于变更规模的近似度量，不论变更发生在软件生命周期的哪一时刻。变更在系统需求里可以是功能性的、或是技术性的。功能上的变更会影响到FUR，而技术上的变更可能影响到NFR或PRC。

变更经常以自然语言表达，由变更提出者（比如客户、用户、开发组等）编写。[36]中提出的方法可以结合机器学习算法给出变更的规模近似值。将这一方法与机器学习结合的好处是：让决策者能快速监控到各层级的变更请求。

11.5 等量带近似法

2019 年，Lavazza 和 Morasca 实验了固定规模分类法的两种替代方案[14]。他们报告称这些替代方案的预测效果比固定规模分类法更好。

等量带法是将所有功能处理划分为个数相同的几个区间带（Band），即如下：

$$\forall i, j \ |Band_i| = |Band_j| \wedge i < j \Rightarrow \forall l, m \ FP_l \in Band_i \wedge FP_m \in Band_j \Rightarrow size(FP_l) < size(FP_m).$$

每个区间带的参考规模值（用于后续估算）设置为本区间带内功能处理规模的平均值。

实践中，实际的区间带只是大致划分的，因为功能处理个数不一定是区间带个数的倍数。假设我们有 98 个功能处理，如果想划分 4 个区间带：最理想的做法是：

- 两个区间带有 25 个功能处理
- 两个区间带有 24 个功能处理。

11.6 等距带近似法

2019 年，Lavazza 和 Morasca 实验了固定规模分类法的两种替代方案[14]。他们报告称这些替代方案的预测效果比固定规模分类法更好。

等距带法是将区间带设置为等距的。

如果有一组数据，其功能处理最小规模是 3CFP，最大规模是 44CFP，我们想设置 4 个区间带，那么每个区间带的宽度是 $\frac{44-3}{4} = 10.25$ CFP。各区间带的边界点就是：13.25 CFP，23.5 CFP 和 33.75 CFP。

因为功能处理规模必须是整数，则小型的功能处理在[3,13]CFP 这个范围；同理，其他区间带的功能处理分别在[14,23]，[24,33]和 [34,44] CFP 的范围内。

每个区间带的参考规模值（用于后续估算）设置为其中点。比如，某区间带包含的功能处理在 [24,33] CFP 范围内，其参考规模值是 $\frac{24+33}{2} = 28.5$ CFP。

第 4 部分

共通概念

12.不同的文档详细程度和分解层级

为了保证对不同详细程度（颗粒度）和分解层级的实际需求文档进行规模度量的结果是可比的，本章将讨论功能规模度量需要注意的一些方面。这一问题可能在以下情况中凸显：当功能规模度量必须在某一大型软件项目的早期阶段进行；或者总体上需要保证不同部分实际需求的规模度量结果是可比的时候。

功能规模度量需要注意的方面应在度量策略阶段予以考虑，此阶段的描述参见度量手册。这些问题并不是 COSMIC 特有的，从理论上说任何功能规模度量方法都存在。

12.1 大型软件项目早期的需求演化过程

在一个大型软件项目的早期，当最初的实际需求被提出，后续的处理方式可能是以下二者之一：

- 第一种，实际需求不断被细化定义；
- 第二种，实际需求被分成多个小一些的、界限清晰、易于管理的“块”，以便多个小组并行开发。这些分开的“块”随后会发展成独立的软件块，即独立的“子系统”。

结果可能是（并且在实践中已经观察到多次）当需要进行规模度量时，实际需求处于各种“文档详细程度”和“分解层级”。这两个概念很容易混淆。因此将其定义描述如下：

定义 — 分解层级

将软件块分解为构件而得到的任意层级（例如，称为“1 级”），然后将构件分解为子构件（“2 级”），再将子构件分解为子-子构件（“3 级”），等等。

注 1：不要和“文档详细程度”混淆。

注 2：对软件块构件的规模度量只可以与处于同一分解层级的构件直接比较。

当实际需求处于不同文档详细程度或分解层级时，度量人员应该仔细检查所处级别之后再建立比例系数，如第 1 章所述。

定义 — 文档详细程度（颗粒度级别）

对于一个软件块任意组成部分的描述（例如：对需求的陈述，或者对软件块结构的描述）的任意扩展级别，每一次深入扩展，对软件块的功能性描述也更加细化并具有一致的详细级别。

注：度量人员应该意识到当需求在软件项目的早期演化过程中，在任何时候，需要的软件功能的不同部分通常以不同的颗粒度级别被文档化。

精确的 COSMIC 功能规模度量要求实际需求的详细程度达到可以识别功能处理及其数据移动的水平（即我们所说的达到了“功能性用户需求（FUR）”的程度）。“功能处理级详细水平”定义如下：

定义—功能处理级详细水平（功能处理颗粒度级别）

对软件块描述的一个颗粒度级别，在该级别上：

- 功能用户（类型）是单独的人、工程设备或软件块（而不是它们的任何组合），并且
- 软件块应响应的是单个的事件（类型）（而不是定义为事件组的任何颗粒度级别）

注 1：在实践中，软件文档对功能性需求的描述在不同部分通常具有不同的颗粒度级别，尤其当文档还在演化时。功能处理可以在其中任意颗粒度级别显现出来。

注 2：人、工程设备或软件块的组合（功能用户），举例来说，可能是一个“部门”，它的成员处理多种类型的功能处理，或者是一个有多种仪器的“控制面板”，或者是“中央系统”。

注 3：“事件组”，举例来说，如在一个高颗粒度级别的功能性需求描述中提到的会计软件系统中“销售交易”的输入事件流，或者航空电子软件系统中“飞行命令”的输入事件流。

规则—功能处理级详细水平

- a) 精确的软件块功能规模度量要求其 FUR 的颗粒度达到能够识别功能处理及数据移动子处理的级别。
- b) 如果必须要对一些还没有达到足够详细程度的需求进行度量，可采用近似方法度量需求。这些方法定义了如何在高颗粒度级别度量需求。在高颗粒度级别的度量中运用缩放系数得出功能处理及其数据移动子处理颗粒度级别的近似规模。见“早期或快速 COSMIC 功能规模近似度量指南”

本指南介绍的方法，即实现规则 b) 的方法。

如 1.3 节所述，所有近似方法都有一个共同的问题：无法找到一个比功能处理层级更高层级的、可以清晰定义的文档化标准层级。进而使得度量人员，尤其是缺乏经验的度量人员，经常没有意识到实际需求是处于不同的详细程度的，而且/或无法区分不同的详细程度。这是度量功能规模时最常见的问题，无论是精确度量还是近似度量。

为了帮助度量人员评估需求的可度量性，COSMIC 编写了度量精确性指南[37]。根据该指南，需求可以被评价为以下几个等级：

- A. 描述完整
- B. 部分有描述
- C. 仅能识别
- D. 仅能计数
- E. 隐含的（“已知的未知”），没提到或缺实（“未知的未知”）

以下是各种详细程度的需求描述的例子，以及在进行近似度量前应该如何分析的提示。

需求描述	文档详细水平及如何分析	分级
“软件系统应该控制洗衣机的所有流程包括洗涤周期、烘干、注水、排水、用户控制面板接口，等等。”	非常粗的文档描述。没有经验的度量人员甚至很难进行近似度量。但是，有经验的、了解洗衣机硬件知识的度量人员，应该能够列出功能处理的数量，进而使用某一近似方法进行度量。	E
“软件可以让人事部人员维护所有常驻员工的数据。”	“维护”这个词隐含的功能处理通常至少包括新建、读取、修改和删除数据（记住首字母缩写为“CRUD”）。可以采用近似度量方法进行度量，但是度量人员必须检查一下列出的功能处理是否完整。（可能需要几种类型的查询和修改功能处理。）	C
“我想查询订单库，而且这个库必须是实时更新的。”（这有可能是敏捷开发的一个“用户故事”案例）	这个实际需求好像只定义了一个查询功能处理。而且从上下文中甚至可以判断出：可以对功能处理进行精确的度量。但是实际需求并不清晰。这个故事有可能隐含了其他功能处理。那么度量人员必须要询问，比如： a) “订单库”是什么意思？查询需要多详细-按订单、按客户、按产品、按下单时间，等等？ b) 为了保持订单库实时更新需要什么功能？	B
“客户通过网络访问系统应该遵守行业标准安全策略，要求使用邮件地址作为 ID,并输入密码。” ⁶	大多数度量人员应该能列出这个实际需求的功能处理，比如：允许新客户和现有客户访问系统，处理忘记密码的情况，提供修改密码的功能，等等。所以至少可以进行近似规模度量。	D
“销售经理生成按分支机构、地区和国家的月报。”	这个实际需求不清晰。可能规定了三个功能处理，但是因为没有给出数据移动的具体细节，它也有可能只是按分类过滤的操作。没有更多细节，近似规模度量可能会有很大范围的不确定性。	E
“软件必须每 10 秒读取和显示外部温度，并更新温度历史记录。”	充分说明了功能处理及其数据移动，可以被精确度量。	A

⁶ 安全需求可能被认为是一个“非功能需求”（NFR）。但是像很多其他非功能需求一样，这个需求需要通过软件来实现，并且能够使用 COSMIC 方法度量。

为了说明度量人员所面临的是什么样的困难，我们提供两个案例。

1. 第一个案例，我们简要的回顾下那个熟知的网上订货系统的案例，度量手册中把它称作“Everest”系统，并对其进行了详细的讨论。这个案例展示了在不同的文档颗粒度级别下度量实际需求的困难。
2. 第二个案例，我们思考一个电信软件架构的案例。这个案例描述了在度量这样的实际需求时遇到的困难：需求定义的越来越详细，同时需求被分解为较小的“块”、同步细化。

案例 1 在不同的文档颗粒度级别下度量——“Everest”系统

在度量手册 4.0.2 的 2.4.3 节里面描述了 Everest 系统的案例。对 Everest 系统的分析和给出的描述被高度简化，并且“只涵盖了 Everest 用户可用的功能。而不包括为给客户供货而必须要具备的功能，比如：Everest 员工、产品供应商、广告商、支付服务商可用的功能，等等”。

如果在最概要层级描述整个 Everest 应用的话，我们可能会将其列为一组功能域，“顾客订货”是其中的一个域。其他域可能有：内部处理（例如生成账单）；产品供应；管理；广告；支付服务；系统维护等等。我们可以在这个层级将整个应用进行“分解”，然后单独考虑每个功能域。

对于要去度量某个域的实际需求的人来说，他们的任务是随着实际需求越来越详细而去理解需求。度量范围可以被定义为“顾客订货功能域”。度量人员不需要考虑在此范围内再“分解”。

我们将顾客订货功能域的实际需求“放大”到较低颗粒度级别，来回顾一下这个案例的一些观察结果：

- 待度量功能域的范围不变，
- 功能用户（下订单的个人用户）不变。在所有颗粒度级别，顾客都能“看见”该功能域的所有功能。

进一步，并且最重要的观察结果是：“在实践中，当采用自上而下的方法分析某些功能时，不能认为图表中处于某个“级别”的功能就对应着 COSMIC 方法中定义的同一“颗粒度级别”。（此定义要求：在任一颗粒度级别，功能都“处于一个同等的详细程度”）。”

如同度量手册里对 Everest 订货系统一个分析图所示，在实践中，在这样的图表中，功能处理可能存在于多个层级中。因此，度量人员必须检查整个系统“树”的每个主要分支、小分支、或者叶子节点，以制定出适合那部分的比例系数。在实践中，在任何时刻，一个功能模型的所有部分不会都演化成同一颗粒度级别，不能对所有部分应用同一个比例系数。

案例 2 在软件架构的多个颗粒度级别和分解层级上进行度量

本节的案例阐述了当软件实际需求越来越详细时，度量实际需求的方法，该案例不同于上一节所描述的“Everest”系统的方法。这个案例也来自于不同的软件领域，即一个复杂的实时电信软件架构。本案例由一个电信设备的主要制造商提供，作为他们目前实践的一个示例。

以下描述使用了该制造商的术语[8]。

本案例目的在于随着实际需求的演变度量其功能规模，并将此作为项目估算方法的输入。

软件架构的描述和分析

图 12.1 a)展示了该软件架构内的一个“逻辑网络元素”（或称 LNE），以及将其功能向下分解为两层较低的颗粒度级别，即图 12.1 b)的“系统构件”（SC）层级和图 12.1 c)的“子系统”（SS）层级。

图 12.1 所展现的模型是该电信公司在三个阶段（不同的颗粒度级别）分别建立的，其目标是在每个阶段估算开发整个 LNE 的项目工作量。因此，这里所描述的分析过程是“早期近似规模度量”的一种形式。但是这个案例及分析也帮助阐明了其他问题。

这个电信架构的描述和分析方式跟上一节的“Everest”案例比较，第一个关键区别是，在每个颗粒度级别，度量范围被拆分为“构件”以便对各层级的每个“构件”都可以单独度量。（需要记住的是，在“Everest”案例里面，随着分析视角放大到较低颗粒度级别，其度量范围是不变的）。因此这个方法除了将功能“放大”以外，在分析时还涉及到对功能的“分解”。

对一个软件块（即对其实际需求和度量范围）的分解必然会出现新的功能用户。比如：如果一个软件块被分解成两个互相关联的构件，那么这两个构件则互为功能用户，并且互相交换数据移动。因此，如果需要度量几个构件的总规模的话，度量人员必须考虑规模汇总的规则（见度量手册）。

LNE 的不同度量范围如图 12.1 显示，LNE 层级用实线表示，SC 层级用横杠虚线表示，SS 层级用点状虚线表示。

逻辑上，在每个颗粒度级别，构件间似乎都可以直接交互。（当然，实际上，构件之间是通过操作系统交互；这一点在颗粒度最低的 SS 层级比较明显，该层级是物理构件所在的层级）。因此为了进行规模度量，架构中每个颗粒度级别的构件都可以看作互为功能用户。

图 12.1a)显示的是处在最高颗粒度级别的唯一一个功能处理：逻辑网络元素 1（LNE1）。就这个功能处理而言，它有两个功能用户，与 LNE1 处在同一颗粒度级别，即 LNE2 和 LNE3。这些用户是对等软件块。一些数据从 LNE2 输入到 LNE1，一些数据由 LNE1 发送给 LNE2 和 LNE3。LNE1 也会发送数据给持久存储介质或从其中读取数据。

在下一层颗粒度级别，图 12.1b)显示 LNE1 被分解成 SC1 到 SC4 四个系统构件。在这个层级，每个系统构件的功能用户要么是 LNE1 内的其他系统构件，要么是 LNE2 和 LNE3 内的系统构件（图中没有显示后者）。

图 12.1b) 显示，LNE 层级唯一的这个功能处理已经被分解成了三个功能处理，SC1，SC2 和 SC4 每个系统构件内各一个。（我们现在能够看出 SC3 没有参与到 LNE 层级的功能处理中。）

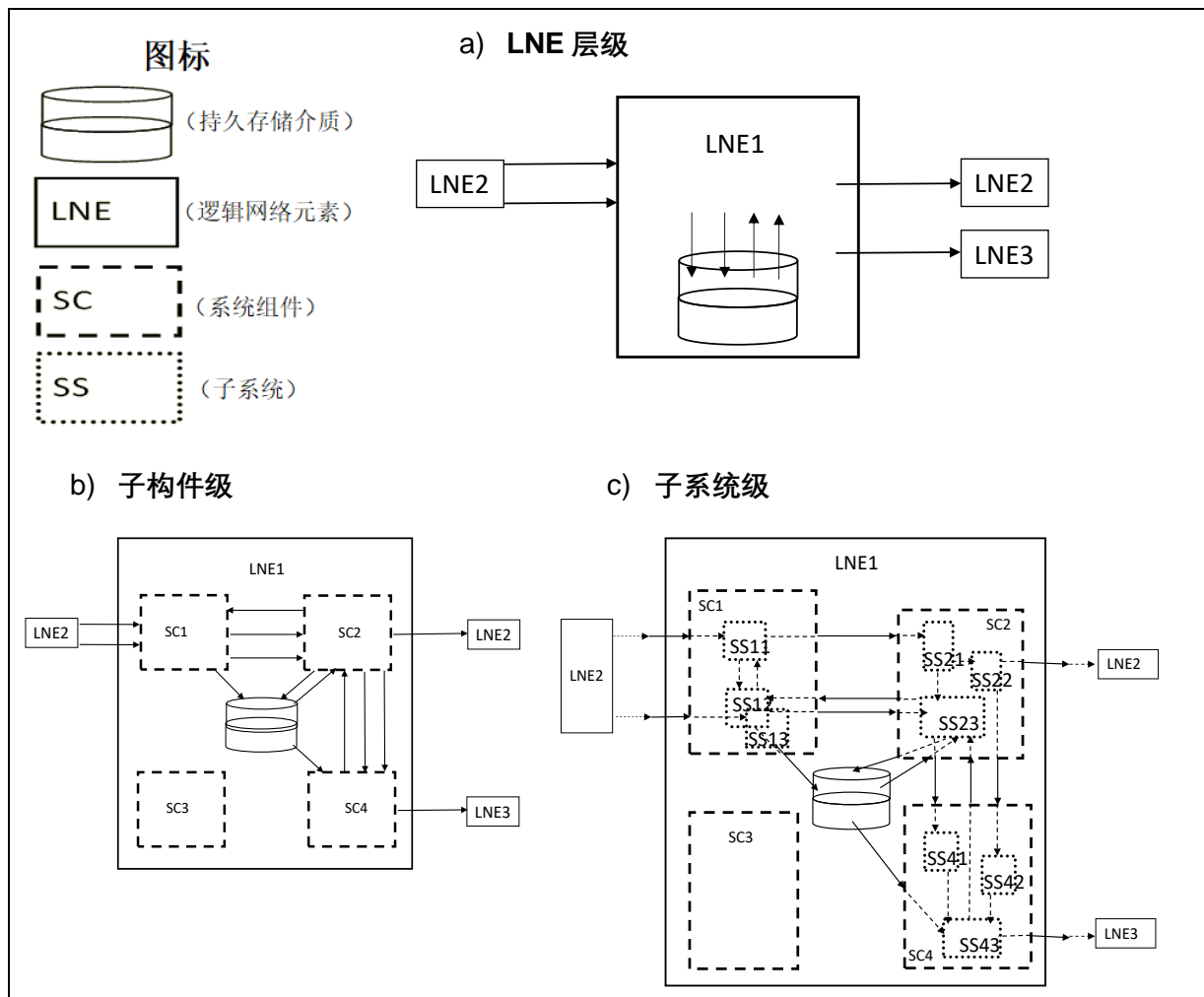


图 12.1 – 逻辑网络元素及其在两个颗粒度级别的分析

在下一层颗粒度级别，图 12.1b)显示 LNE1 被分解成 SC1 到 SC4 四个系统构件。在这个层级，每个系统构件的功能用户要么是 LNE1 内的其他系统构件，要么是 LNE2 和 LNE3 内的系统构件（图中没有显示后者）。

图 12.1b) 显示，LNE 层级唯一的处理已经被分解成了三个功能处理，SC1, SC2 和 SC4 每个系统构件内各一个。（我们现在能够看出 SC3 没有参与到 LNE 层级的功能处理中。）

在最低一层颗粒度级别，图 12.1c)显示每个系统构件被分解成许多子系统（SS）。在这个层级，任何一个子系统的功能用户要么是 LNE1 内的子系统，要么是其他 LNE 内的子系统（后者在图里没有显示）。在 LNE 层级的唯一功能处理现在已经被分解成了九个最低颗粒度级别的功能处理，每个子系统一个功能处理。

在每个颗粒度级别，一些数据被移动到持久存储介质，一些数据从持久存储介质里取出。图 12.1 显示，当我们分解到较低颗粒度级别的时候，这些功能所涉及到的 LNE1 构件。（为了方便画图，持久存储介质被显示为“共享资源”，与颗粒度和分解层级无关。严格地说，正如此图所示，每个颗

粒度级别的功能都被分解了，导致产生了新的度量范围，持久存储介质应该出现在其被使用的每一个范围内。)

为供参考，LNE1 子系统颗粒度级别的数据移动 (DM) 的物理顺序如下所示：

- a) 对 (LNE1 的 SC1 的) SS11 的触发输入来自于 LNE2(由其中一个 SC 内的一个 SS 发送);
- b) SS 间的数据移动交互后 (可能是同一个或者不同的 SC 的一部分)，LNE1 发送一个输出 (通过 SC2 的 SS22) 给 LNE2 内的一个 SS (比如，请求从功能处理的最初发起者处获取更多信息)；
- c) 然后，LNE2 内的一个 SS 用另一个输入 (与触发输入不同) 回应 LNE1 (确切地说是给 SC1 中的 SS13)；
- d) 同样地，在一些内部数据移动之后，LNE1 发送 (通过 SC4 中的 SS43) 一个最终的输出给 LNE3 的一个 SS；
- e) 除此之外，在过程中也会有读和写操作；

如以上所述，项目团队是在子系统层级开始实际的软件开发；子系统属于自主应用。这一点非常重要，因为提供这个案例的电信软件公司希望在这个颗粒度级别分别进行项目估算。

度量逻辑网络元素的规模

通过这种分析方法，在较低颗粒度级别随着更多的构件和功能处理显现出来，图 12.1 显示的功能规模明显有所增长。

这种“增长”和我们在地图所看到的情况类似。随着我们把地图从一个大的比例尺放大为一个较小的比例尺以便显示更多的道路时，即使两个地图在每个比例尺上的度量单位是一样的 (比如公里)，道路网路的规模看起来也是增长的。

图 12.1 各个颗粒度级别的功能规模如下：

- LNE 颗粒度级别 (一个功能处理) = 8 CFP
- SC 颗粒度级别 (三个功能处理) = 20 CFP
- SS 颗粒度级别 (八个功能处理) = 32CFP

请注意：作为对这个案例的度量结果的核查，任何一个颗粒度级别的规模都能用比它低一个颗粒度级别的规模累加而得到，但要去掉这一较低层级内部构件间的所有输入和输出。

LNE 案例分析的进一步探讨

从这个案例分析中首先观察到的关键点是，当使用电信设备制造商自己的术语进行描述时，12.1 节所给出的“功能处理颗粒度级别”的定义会有问题，这个定义说的是：

“对软件块描述的一个颗粒度级别，在该级别上：

- 功能用户是单独的人、工程设备或者软件块 (而不是它们的任何组合)，并且 (等等)。”

在这个场景中对此定义进行解释的困难之处在于：LNE 案例里的所有功能用户都是“软件块”，而无论如何也不可能找到一个普遍适用的方法来定义什么是软件块。

该电信设备公司在这里所使用的分析方法和术语导致的结果是，每个颗粒度级别的度量范围被重新定义了，并且功能处理被定义在三个颗粒度级别上，而不是按照定义假设的只有一个标准级别。

通过改变电信公司的术语可以避免此问题，比如“功能处理”只能用于子系统级别。在较高级别，可采用“超级处理”和“超超级处理”之类的术语。

但是还有一个更基本的问题是：无论用什么名字，“功能处理颗粒度级别”在该公司就会有特定定义。这个定义对于本公司来说解释起来相对容易。因为根据定义，这个层级是公司组建项目团队去开发“单个的……软件块（而不是这些的任意组合）”的层级。但是，这个方法不能保证这家公司的“一个软件块”和另外一家公司的意思是一致。软件可以被汇总或分解成多个颗粒度级别，而且软件构件类型根据使用的技术也会有不同。“一个软件块”对一个公司来说可能是一个完整的逻辑网络元素；对另外一个公司来说则有可能是一个可复用的对象（也许是个可售卖产品），其颗粒度级别显然是不同的。

当场景里包含的功能用户是“单个的人和/或工程设备”时，“功能处理颗粒度级别”的定义便不会产生问题，因为它们是可以被准确识别的。在这种场景下，如果也有“软件块”作为功能用户，那么这些软件用户的颗粒度级别以及和被度量软件之间的数据交互，应该跟人类或工程设备功能用户的颗粒度级别相同。

推算 LNE 案例的“比例系数”

在 LNE 案例里，假设我们当前所处的阶段是已经完成其规格说明，一部分处于最高的 LNE 颗粒度级别，一部分处于 SC 级别，并且我们希望在子系统这一最低颗粒度级别上确定最终软件的规模，并以此作为项目估算方法的输入。为此，我们需要用比例系数乘以 LNE 或 SC 的度量规模，得出在最低 SS 层级的度量规模。

如果我们采用前一节给出的 LNE1 的规模度量结果去校准一个近似度量方法，然后用于度量在 SS 层级的其他 LNE 和 SC 的规模，我们可以得出以下比例系数：

- 将处于 LNE 层级的度量规模缩放至 SS 层级的度量规模，可以用 LNE 规模乘以 4.0 (32 / 8)
- 将处于 SC 层级的度量规模缩放至 SS 层级的度量规模，可以用 SC 规模乘以 1.6 (32 / 20)

实际上，当一个项目估算需要功能规模作为输入时，实际需求的开发可能会处于不同的颗粒度级别。在这种情况下，度量人员必须练习判断何时采用实际度量和缩放度量相混合的方法，以便得到在规定颗粒度级别的估算规模。

12.2 功能规模度量和标准的分解层级

本节简要地讨论一下在上文中提到过的问题，即为了确保度量结果是可比较的，需要将分解层级标准化，并采用标准的功能处理颗粒度级别的文档。在一个大型分布式软件系统中，当需求分成多个部分在并行细化时，做到这一点可能很困难。

建立需求的详细程度（即颗粒度级别，由客户驱动）和软件分解层级（由系统架构驱动）之间的关系是个复杂的过程，最好按以下四个步骤执行：

出发点： 我们想要度量一个正在演化中的、大型软件系统的规模，目的是估算工作量。该软件的整体架构已经建立起来了，但需求仍在演化中。

第一步：识别架构中想要单独度量的每个软件块的分解层级，并且确定其度量范围。

以下是软件分解层级的案例，以及可能的度量范围：

- 12.2 节的主题，电信系统，可以在 LNE 层级、系统构件或子系统层级（如图 12.1 所示）、或者甚至是更低层级（比如子系统主要构件，可复用构件层级等）中的任何一级进行度量。
- 在业务应用领域，软件可以在整个应用层级、或主要应用构件（比如包含在一个“多层”架构中）层级、或一个对象类、或复用构件等层级进行度量。（但是需要注意的是，在某些行业如银行，其软件系统已经成长演变了几十年，定义和区分其“应用”可能比较困难。）

第二步：识别每个软件块的实际需求的颗粒度级别，其度量范围在第一步中已定义。

第三步：识别第一步中软件块的功能用户，此功能用户不能被再分解（单个的人或硬件块）。根据实际的度量目的，单个人或者硬件与软件块的交互，只涉及到有限的几个分解层级。（举例来说，对于人类用户可见的那部分业务应用软件规模，只度量两个分解层级就够了，即整个应用、或者一个多层应用程序的用户接口部分。）

因此，我们能准确地定义出人或硬件功能用户与其所交互的软件分解层级的几种组合关系。由此，我们能准确地识别这些组合的功能处理颗粒度级别的需求描述，因为功能用户必须响应或产生的各类事件都可以被识别出来。所以，我们能准确的识别功能处理并得到功能规模度量结果，并且能够把不同来源组合的度量结果放心地互相比较。同理，对于那些比功能处理级别更高颗粒度的实际需求，我们也能采用本指南所描述的近似度量方法。

第四步：对于第一步中识别的软件块，识别有哪些其他软件块是它的功能用户。他们可能存在于多个分解层级。软件没有标准的分解层级。另外，待度量软件的实际需求及其功能用户的实际需求的描述都可能处于多个颗粒度级别。因此，在实践中，当被度量软件和其所有的功能用户都是软件块时，为功能处理颗粒度级别定义一个国际统一的标准是很困难的。这也意味着，找到一种识别和度量软件功能处理的方法，使得这种“软件/软件”组合的各种来源的度量结果都是可比的，也同样困难。同样地，将本指南所述的近似度量方法应用在这种“软件/软件”组合上，也很困难。

只要为软件分解层级和需要进行近似规模度量的颗粒度级别定义本地标准，这些困难都可以在组织内部或存在协作的组织间得到克服。

为了加大规模度量的可比性，COSMIC 建议，提供功能规模度量服务或工具的供应商应明确定义软件分解的标准层级，符合这一标准层级的规模度量结果才是可接受的。

COSMIC 建议，在业务应用软件领域，以下分解层级可以作为标准化案例。

- 一个“完整的应用”；

- 完整应用中的一个主要构件；
- 一个可复用的对象类。

所有这些分解层级对应的功能用户可能是人或其中任何一个分解层级的其他软件块，例如：一个完整的应用可能是一个 SOA 构件的功能用户，反之亦然。但是对于一个可复用的对象类的人类用户可能情况就不是这样了。

这些标准的“分解层级”同时也是度量手册里给出的关于度量范围的典型案例。

13.本地化（校准）指南

建立一个本地定义方法进行 COSMIC 近似规模度量，可以遵照以下指南：

- a) 本地组织应该定义一个或者多个（类型）的软件工作产品，其颗粒度级别比能识别出功能处理及其数据移动的颗粒度级别要高，以该定义方式描述的软件功能是可被度量的（即至少可以被识别和计数）。
- b) 在 a) 步中选择的软件工作产品所描述的实际需求，如果将其颗粒度级别按升序排列的话，可能依次是功能处理级、“用例”级，子系统级等。（需注意的是功能处理颗粒度级别⁷以上的层级是不存在标准术语定义的。）因此，当找到适合进行近似度量的工作产品，且其颗粒度级别高于功能处理颗粒度级别时，对其的定义一定要特别谨慎。
- c) 选作进行比例系数校准度量的高层级工作产品，必须能够代表以后要进行近似度量的软件。
- d) 以上所选择的软件工作产品应该具有大致相似的规模，或可以按照相似规模区间（类）进行分类（见第 3、4、9、10 章节所描述的近似度量方法）。这样就可以为每个或每类工作产品分配一个平均值作为本地比例系数。
- e) 应用本地定义的近似方法度量了一些高层级的实际需求后，当需求细节明确时，核算近似度量结果的精确度、并吸取经验是非常重要的。可以按以下步骤进行：
 - 首先，至少抽取一个详细的实际需求进行精确度量。
 - 然后，将近似度量值和精确度量值相比较，来检查所采用的比例系数是否合理。例如，当某个项目的实际需求最终确定后，发现其精确度量值比近似度量结果明显大很多。其中的偏差可能是由于采用了不恰当的比例系数和/或项目的“范围蔓延”导致的。这一结果可以作为影响因素之一，用于将来对近似度量方法的调整。（对于“范围蔓延”的处理详见 14.2 章节。）
- f) 考虑到近似度量方法的不确定性，在报告近似度量值的时候，应该给出一个范围或精度提示，基于 e) 中描述的与详细规模度量值对比核算过的精确度。
- g) 核算近似规模度量精确度的流程也应该在本地建立。每次度量的精确度取决于下面两个因素。
 - 需求的详细程度和不确定程度，它们会随着项目进展状态而发生明显变化（见 1.5 节的需求质量、12.1 节的案例及 14.2 节的“范围蔓延”指南）；
 - 规模度量所采用的具体近似方法。（比如：第 4 章的方法更加详细，对于相同质量的需求，如果校准得当，它的规模度量结果比第 2 章方法的结果更准确。）

⁷ 像“用例”这样的术语当然会被定义，但是尽管有定义，实践证明这也不能保证对于给定的实际需求，两个分析师将会得出同等数量的用例。因此，对于一个用例的构成，每个组织必须建立自己的概念。

- h) 估算规模的最佳实践是之后用“三点”估算法，这三点指的是最小规模估算、最有可能的规模估算和最大规模估算。对决策者以及必须估算项目工作量的人来说，给出这三个数字作为近似规模度量结果以便展示一个不确定的范围，比报告一个最可能的估算值更有价值。
- i) 第 8 章介绍的早期&快速估算法用到了三点估算。
- j) 三点估算的更多信息请见 [Wikipedia](#)。

14. 功能变更和范围蔓延的近似规模度量

14.1 功能变更的近似规模度量

对既存软件的变更需求的近似度量，大致方法与新开发软件一样。如果已经存在一个现有功能处理及其规模（或规模分类）的清单，那么可以从以下两种方法中选择一种：

- a) 如果可能，基于变更的功能性用户需求，判断影响到哪些相关功能处理的数据移动，并计算这些数据移动个数。
- b) 或者，估计每个功能处理必须被改变的数据移动数量或比例。比如：如果一个被变更的功能处理的规模是 12CFP，估计其变更影响了 25% 的数据移动，那么其变更规模就是 3CFP。

用变更功能点个数或占比代替被更改的功能处理的总规模，作为前面近似方法的补充。

如果没有现成的功能处理清单，那么第一件事就是识别出实际变更需求影响到哪些功能处理，然后从上面提到的近似度量方法中选择一种进行度量。

14.2 近似度量与范围蔓延

经验表明：在软件开发项目的早期阶段，软件的功能规模往往会随着项目实际需求的演化而增加，演化过程是从大纲性需求到详细的需求，再到功能规格等。这种现象通常被称为“范围蔓延”，发生的原因可能是：

- 范围延展超出了原定计划，涵盖了其他功能；
- 和/或，随着细节变得更清晰，之前要求的功能比原本设想的更广泛（例如，每个功能处理要有更多的数据移动）；
- 非功能需求可能会（部分）由软件实现[38]。

（当然，范围比本来计划的有所减少也是有可能发生的，比如由于削减预算。）

在本指南所描述的近似规模度量方法里面没有明确地将范围蔓延考虑进去。当采用这些近似度量方法进行早期度量的时候，潜在的范围蔓延应该被当作一个额外的因素考虑。如果忽略潜在的范围蔓延，将会有低估最终软件规模以及项目工作量的风险。

对一个具体项目范围蔓延的预估不在本指南范围。但是，考虑以下问题可能会有所帮助：

- 这个项目的实际需求是从开始就非常不确定吗？如果是，考虑到可能的范围蔓延，需要对近似规模值进行纠正（或增加“应急储备”）吗？

- 如果范围蔓延是组织内的普遍现象，那么我们可以使用历史的度量数据来帮助量化这一现象。比如：在一个使用特定开发流程的组织中，对开发过程存在很多度量数据，可能会找到一个重复的模式，例如“在第 3 阶段结束时，规模通常比第 1 阶段结束时增加 30%”。

15. 近似规模度量方法总结

在软件项目的早期，近似规模度量方法是可行且非常有价值的，并且/或与采用标准 COSMIC 精确度量方法相比，近似规模度量方法可以节省时间和工作量。当实际需求不明确的时候，可能也需要采用近似规模度量方法。但是近似规模度量方法需要谨慎使用。

- 无论使用近似规模度量方法的目的是什么，当有进一步的信息可以进行更精确和/或准确的规模度量时，度量人员都应该完善和更新度量结果。特别是在把度量结果当作估算的一个输入（比如工作量预测）的时候——由于存在误差传播现象[40]，更应该进行更新。
- 由于类似的原因，在与合同相关的场合或类似情况下，不能使用近似规模推断实际值，因为此时要求精确并且准确的数据——由于这种限制，所以在项目的末期，应该用标准度量值代替最初的近似规模度量值。

当需要采用近似规模度量时，组织应该：

- 根据是否有数据可用于校准、可用的时间及对近似度量的精度要求，选择一个能满足度量目的的最佳方法；
- 使用本地软件样本的精确度量数据来校准该方法，选择的样本与将要被近似度量的软件具有可比性；
- 当实际需求不清晰或不完整时，至少要设法识别出所有功能处理；
- 特别注意识别那些大型的功能处理，并为这些功能处理确定一个适当的比例系数，因为尽管它们数量很少但却对整个规模影响很大；
- 在发布近似规模值的时候，考虑一下是否要为“范围蔓延”和非功能需求可能导致的影响留一些余量（或者“应急储备”）；
- 估算和记录此近似规模值的正负偏差范围（即不确定性），说明一下为范围蔓延准备的应急储备量；在与合同相关的场合下，对近似规模不确定性的估算尤其重要。

参考文献

参考文献

- [1] C.R. Symons and A. Lesterhuis, "Introduction to the COSMIC method of measuring software", version 1.1, January 2016.
- [2] F.W. Vogelezang, C.R. Symons, A. Lesterhuis, R. Meli and M. Daneva, "Approximate COSMIC Functional Size: Guideline for approximate COSMIC Functional Size Measurement", 23rd International Workshop on Software Measurement and 8th International Conference on Software Process and Product Measurement – IWSM-MENSURA, Ankara, Turkey, 2013.
- [3] C.R. Symons and A. Lesterhuis, "The COSMIC Functional Size Measurement Method Version 4.0.2 Measurement Manual, the COSMIC implementation guide for ISO/IEC 19761:2017", December 2017.
- [4] H.C. Bauman, "Accuracy Considerations for Capital Cost Estimation", Industrial & Engineering Chemistry, April 1958.
- [5] B. Boehm, "Software Engineering Economics", Prentice-Hall, 1981.
- [6] E. Berardi, C.R. Symons and S. Trudel, "Guideline for the use of COSMIC FSM to manage Agile projects", September 2011.
- [7] S. McConnell, "Software Estimation, Demystifying the Black Art", Microsoft Press, 2006.
- [8] K. Almakadmeh, "Development of a scaling factors framework to improve the approximation of software functional size with COSMIC - ISO19761", PhD thesis, Ecole Technologie Supérieure - Université du Québec, Montréal, Canada, 2013.
- [9] A. Abran, J-M. Desharnais, S. Oligny, D. Saint-Pierre and C.R. Symons, "COSMIC-FFP Measurement Manual version 2.2, the COSMIC implementation guide for ISO/IEC 19761:2003".
- [10] F.W. Vogelezang, "Early estimating using COSMIC-FFP", Proceedings of the 2nd Software Measurement European Forum, March 16-18, 2005, Rome.
- [11] H.S. van Heeringen, E.W.M. van Gorp and T.G. Prins, "Functional size measurement – accuracy versus costs – is it really worth it?", Software Measurement European Forum - SMEF, Rome, Italy 2009.
- [12] L. de Marco, F. Ferrucci and C. Gravino, "Approximate COSMIC size to early estimate Web application development effort", 39th Euromicro Conference on Software Engineering and Advanced Applications - SEAA, Santander, Spain, September 4-6, 2013.
- [13] V. del Bianco, L. Lavazza, G. Liu, S. Morasca and A. Zaid Abualkashik, "Model-based early and rapid estimation of COSMIC functional size – An experimental evaluation", Information and Software Technology, Volume 56, Issue 10, Oct. 2014, pp 1253-1267.
- [14] L. Lavazza and S. Morasca, "Empirical evaluation and proposals for bands-based COSMIC early estimation methods", Information and Software Technology, Vol.109, May 2019, pp 108-125.

- [15] A. Lesterhuis and C.R. Symons, "The COSMIC Functional Size Measurement Method version 3.0 Advanced and Related topics", December 2007.
- [16] F.W. Vogelezang and T.G. Prins, "Approximate size measurement with the COSMIC method: Factors of influence", Software Measurement European Forum - SMEF, Roma, Italia, 2007.
- [17] F. Valdés-Souto and A. Abran, "Improving the COSMIC approximate sizing using the fuzzy logic EPCU model", Joint International Workshop on Software Measurement and International Conference on Software Process and Product Measurement IWSM-MENSURA, Cracow, Poland, 2015.
- [18] F. Valdés-Souto, Analyzing the performance of two COSMIC approximation sizing techniques at the functional process level, Sci. Comput. Program. 135 (2017) 105–121.
- [19] F. Valdés-Souto, "Analyzing the Performance of Two COSMIC Sizing Approximation Techniques using FUR at the Use Case Level", Joint International Workshop on Software Measurement and International Conference on Software Process and Product Measurement - IWSM-MENSURA, Beijing, China, September 2018.
- [20] F. Valdes Souto and A. Abran, "Case Study: COSMIC Approximate Sizing Approach Without Using Historical Data", Joint 22nd International Workshop on Software Measurement and 7th International Conference on Software Process and Product Measurement – IWSM-MENSURA, Assisi, Italy, 2012.
- [21] F. Valdés-Souto and A. Abran, "COSMIC Approximate Sizing Using a Fuzzy Logic Approach: A Quantitative Case Study with Industry Data", Joint International Workshop on Software Measurement and International Conference on Software Process and Product Measurement - IWSM-MENSURA, Rotterdam, the Netherlands, 2014.
- [22] L. Lavazza and S. Morasca, "An empirical evaluation of two COSMIC Early Estimation methods", Joint 26th International Workshop on Software Measurement and the 11th International Conference on Software Process and Product Measurement – IWSM-MENSURA, Berlin, Germany, October 2016.
- [23] C.R. Symons, and C. Gencel, 'From requirements to project effort estimates: work in progress (still?)', Requirements Engineering for Software Quality (REFSQ 2013) conference, Essen, Germany, April 2013.
- [24] M. Ecar, F. Kepler and J.P.S. da Silva, "COSMIC User Story Standard", International Conference on Agile Software Development (XP 2018), Agile Processes in Software Engineering and Extreme Programming, pp 3-18.
- [25] S. Trudel, J-M. Desharnais, J. Cloutier, "Functional Size Measurement Patterns: A Proposed Approach", 26th International Workshop on Software Measurement and the 11th International Conference on Software Process and Product Measurement – IWSM-MENSURA, Berlin (Germany), 5-7 October 2016.
- [26] C.R. Symons, Guideline for 'Measurement Strategy Patterns', Ensuring that COSMIC size measurements may be compared, March 2013.
- [27] C.R. Symons, 'Software sizing and estimating: MkII FPA', John Wiley & Sons, 1991.
- [28] R. Meli, "Reference Manual release version 1.1 of IFPUG E&Q FP method release 3.1", april 2012, dpo.it/wp-content/uploads/2018/01/EQFP-IFPUG-31-RM-11-EN-P.pdf
- [29] R. Meli, "Early Function Points : a new estimation method for software projects", ESCOM 97, Berlin (Germany) 1997.

- [30] T. Iorio, R. Meli and F. Perna, "Early & Quick Function Points® v3.0: Enhancements for a Publicly Available Method", Software Measurement European Forum (SMEF 2007), Roma, Italia, 2007.
- [31] L. Santillo, "Early & Quick COSMIC-FFP Analysis Using the Analytic Hierarchy Process." 10th International Workshop on Software Measurement, *Lecture Notes in Computer Science* 2006, (pp. 147-160), Berlin (Germany) 2000.
- [32] R. Meli, "Early & Quick Function Point Method - An Empirical Validation Experiment", International Conference on Advances and Trends in Software Engineering (SOFTENG 2015), Barcelona, Spain, April 2015.
- [33] I. Hussain, L. Kosseim and O.Ormandjieva, "Approximation of COSMIC functional size to support early effort estimation in Agile", *Data & Knowledge Engineering* 85 (2013) 2-14.
- [34] F. Valdés, A. Abran, "Comparing the Estimation Performance of the EPCU Model with the Expert Judgment Estimation Approach Using Data from Industry", chapter 15, in 'Software Engineering Research, Management and Application 2010, published in 'Studies in Computational Intelligence', Volume 296, Springer-Verlag, Berlin, pages 227-240.
- [35] M. Ochodek, "Functional size approximation based on use-case names", *Information and Software Technology*, Volume 80, 2016, pp 73-88.
- [36] M. Haoues, A. Sellami and H. Ben-Abdallah, "A Rapid Measurement Procedure for Sizing Web and Mobile Applications based on COSMIC FSM Method", 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement – IWSM-MENSURA, Gothenburg, Sweden, October 2017, pp 130-138.
- [37] A. Abran, A. Lesterhuis, and C.R. Symons, "Guideline for Assuring the Accuracy of Measurements", version 1.1, July 2018.
- [38] F. AbuTalib, D. Giannacopoulos and A. Abran, "Designing a Measurement Method for the Portability Non-Functional Requirement", 23rd International Workshop on Software Measurement & 8th International Conference on Software Process and Product Measurement – IWSM-MENSURA, October 2013, Ankara (Turkey).
- [39] F.W. Vogelesang, C.R. Symons and A. Lesterhuis, "Web Advice Module – COSMIC Case Study", published January 2014 by NESMA (www.nesma.org).
- [40] L. Santillo, "Error Propagation in Software Measurement and Estimation", 16th International Workshop on Software Measurement - IWSM, October 2006, Potsdam (Germany).
- [41] L. Santillo, "EASY Function Points – 'SMART' Approximation Technique for the IFPUG and COSMIC Methods", 22nd International Workshop on Software Measurement & 7th International Conference on Software Process and Product Measurement – IWSM-MENSURA, November 2012, Assisi (Italy).
- [42] A. Abran, S. Vedadi and O. Demirors, "Development of COSMIC Scaling Factors using Classification of Functional Requirements", 29th International Workshop on Software Measurement & 14th International Conference on Software Process and Product Measurement - IWSM-MENSURA, October 2019, Haarlem (the Netherlands).
- [43] ISO Guide 99: International vocabulary of basic and general terms in metrology (VIM), International Organization for Standardization – ISO, 2019.

术语表

术语表

本术语表中术语专属于 COSMIC 近似规模度量指南。其他 COSMIC 术语，见度量手册 V4.0.2 的术语表。

准确度 [43]

度量值与被度量对象真值之间的一致性。

注 1: “度量准确度”这一概念不是一个量，不给出有数字的量值。但度量提供较小的度量误差时，我们说该度量是更准确的；

注 2: 术语“度量准确度”不应与“度量正确度”、“度量精度”相混淆，尽管它与这两个概念有关；

注 3: “度量准确度”有时被理解为多个度量值之间可归因于测量的一致程度。

近似规模度量

1. 规模的近似度量值。
2. 应用近似方法对规模的度量。

校准

为了得到尽可能准确的近似度量结果，根据本地化环境确定所用近似方法的比例系数或分类值，而不是本指南等参考文献中发布的比例系数或分类值。

分类

将实际需求的各部分归为对应的分类（参考需求），每一类对应一个已校准的 CFP 值。

本地化（参阅校准）

在使用近似度量方法的环境中，找到有代表性的案例对比例系数和分类值进行校准。

精度

表征量值的精密度或可辨别程度（ISO/IEC 24765:2010）

举例：精度为 2 位小数 vs 精度为 5 位小数。

比例系数

用于规模转换的常数，将一组条件下（如，实际需求在某一详细水平）的度量规模转换至另一组条件下（如，同一实际需求在另一个详细水平）的度量规模。