COSMIC 功能规模度量方法 4.0.1 版

COSMIC 软件度量方法简介

1.1 版

2016 年 1 月

致谢

1.0 版本和 1.1 版本的评审者		
Alain Abran École de Technologie Supérieure, Université du Québec Canada	Diana Baklizky Métricas Brazil	Lindsay Davies Freelance Editor United KIngdom
Peter Fagg Pentad United Kingdom	Cigdem Gencel Free University of Bolzano-Bozen Italy	Arlan Lesterhuis [1)] The Netherlands
Bernard Londeix Telmaco United Kingdom	Hassan Soubra [2)] Ecole Supérieure des Techniques Aéronautiques et de Construction Automobile France	Charles Symons [1)] United Kingdom
Monica Villavicencio ESPOL Ecuador	Frank Vogelezang [2)] Ordina The Netherlands	Chris Woodward CW Associates United Kingdom

1) 该文档的作者

2) 1.1 版本的评审者

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：400-1781727
初版翻译	周雍正 麦哲思科技高级咨询顾问，COSMIC 讲师
校对	郭玲 麦哲思科技高级咨询顾问，COSMIC 讲师
	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

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。

版本控制

下表概述了本文档的更新履历。

日期	评审者	修订/添加
2000 年	COSMIC 核心团队	简介、综述、幻灯片展示、补充说明
2007 年 9 月	来自 7 个国家的 14 位评审员	COSMIC V3.0 方法综述的首个版本
2014 年 5 月	COSMIC 度量实践委员会	COSMIC V 4.0 软件度量方法简介 1.0 版
2016 年 1 月	COSMIC 度量实践委员会	更新 COSMIC 度量手册到 4.0.1 版本，参考 COSMIC 在其网上发布的新内容。其他一些已经修订和编辑的改动见附录 A

前言

COSMIC 方法是一种国际标准化方法（ISO 19761，见[1]），用于度量大多数软件领域的功能需求规模，包括业务应用(或管理信息系统)软件、实时软件、基础设施软件和某些类型的科学/工程软件。

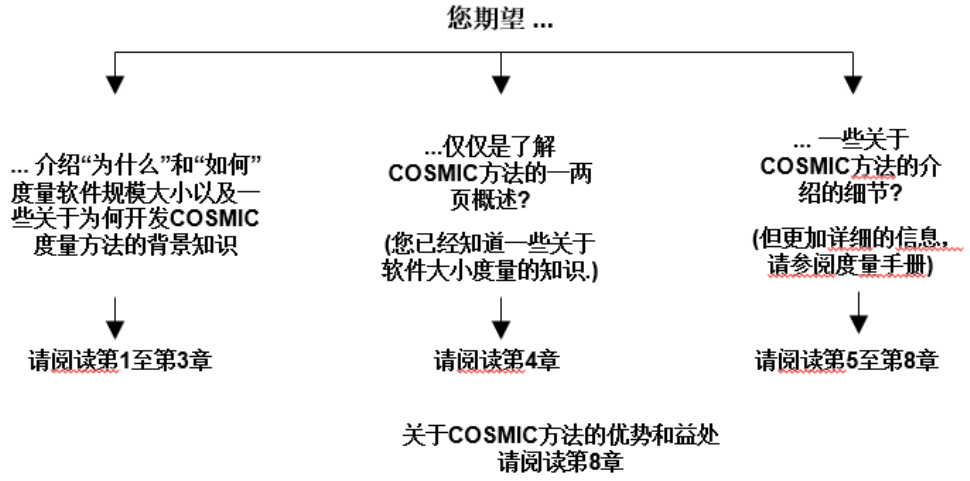
COSMIC 是通用软件度量国际联盟的缩写。该组织是由一组来自澳大利亚、欧洲和北美的软件度量专家于 1998 年成立的，目的是开发一种基于成熟的软件工程原理和计量标准的全新的软件规模度量方法。它的出版物是完全开放的，可以免费下载。

目前，该方法在世界范围内得到了广泛的应用。在其规划的所有领域，都被用于软件合同中规模大小的度量，并成功地应用于项目性能度量、基准对比和估算。

本"简介"文档的目的

本文档的目标读者是那些需要通过该简介来了解软件规模度量方法并使用它的人，以及那些想要了解 COSMIC 方法概要，但不希望了解其全部细节的人。[1]

可通过如下图表来决定阅读的章节。



COSMIC 方法相关文献

除了 ISO 19761 标准之外，所有 COSMIC 方法相关文档都可以从 COSMIC 网站 www.cosmic-sizing.org.的知识库下载。

定义该方法的主要文档是:

- ISO 19761 标准("软件工程-COSMIC-功能规模度量方法")，其中包含了该方法的定义和基本规则。(截至撰写本文时，该标准 2012 版尚未更新到方法的 v4.0.1 版本。)。
- "COSMIC 方法 4.0.1：度量手册"，提供所有的原理、规则以及术语表。并给出了进一步的说明和更多的例子，以帮助度量人员理解和应用该方法。这是度量人员在实践中需要的主要"工作文档"。

下图展示了 COSMIC 相关文档的结构。在本简介出版时，这些文件大部分是可获取的；星号表示该文档仍在开发中。所有内容都在更新，使他们与该方法的 4.0.1 版本保持一致。

[1] 该简介代替了"方法概述"文档，并适用于方法的 4.0.1 版本。

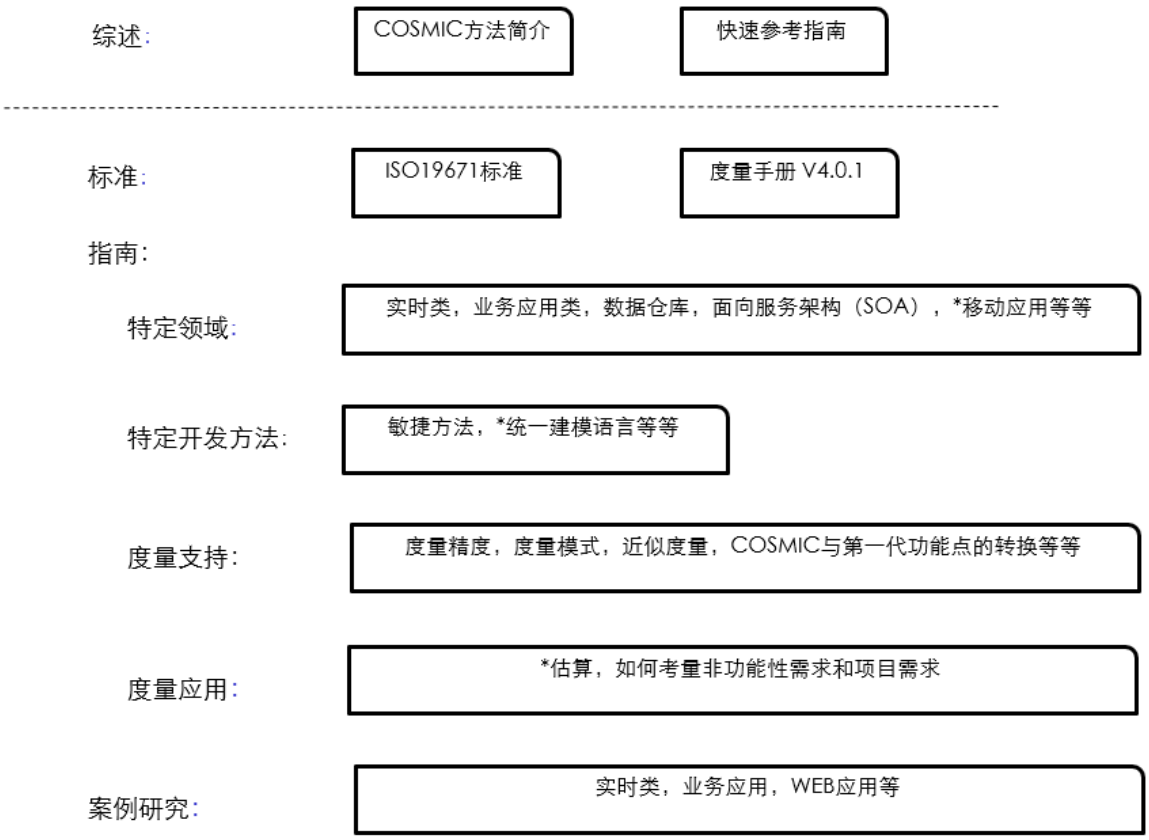


图 - COSMIC 相关文档结构

除英文外，《度量手册》还有 11 种语言版本。所有这些都可以在 COSMIC 网站 www.cosmic-sizing.org 上找到。

该网站提供了更多的背景信息，包括功能规模度量的方法和用途、COSMIC 组织及其活动、COSMIC 相关服务的供应商、COSMIC 认证考试、COSMIC 时事新闻、如何贡献和获得 COSMIC 基准数据以及支持度量的工具、COSMIC 相关的研究论文等，均可以免费下载。

COSMIC 度量实践委员会

2016 年 1 月

目录

1 为什么要度量软件规模? 8
1.1 为什么会有人想要度量软件规模? 8
1.2 软件规模度量在其他方面的应用 8
1.3 谁通常会从这些度量中受益? 8
2 如何度量软件规模? 9
2.1 和其他度量单位一样，你需要标准 9
2.2 度量软件规模最佳方法是什么? 9
2.3 统计代码行数 9
2.4 度量软件需求 9
2.5 其他度量软件规模的方法 9
2.6 深入理解软件需求 - 功能性需求和非功能性需求 10
2.7 度量方法所需的技术 10
3 度量方法简史 11
3.1 一切源何而起? 11
3.2 国际标准化组织（ISO）的介入 11
3.3 COSMIC 方法的诞生 11
3.4 ISO 一锤定音：“让市场来决定” 12
4 COSMIC 方法简要概述 13
4.1 方法的适用范围 13
4.2 COSMIC 功能规模度量过程的三个阶段 13
4.3 第一阶段：度量策略 13
4.4 Phase 2: Mapping 第二阶段：映射 13
4.5 第三阶段：度量 14
5 COSMIC 方法 - 度量策略阶段 15
5.1 为什么需要一个“策略”? 15
5.2 需要确定的五个关键策略参数 15
5.3 软件层级 15
5.4 度量目的是如何影响其他度量策略参数的案例 16
5.5 在开始度量前，还应该考虑哪些? 17
6 COSMIC 方法 - 映射阶段 18
6.1 通用软件模型 18
6.2 关键关系：事件/功能用户/功能处理 18
6.3 FUR 和功能处理的结构 19
6.4 数据运算的统计 20
6.5 四种类型的数据移动 20
6.6 持久存储介质 21
6.7 一个数据移动移动了单一数据组，描述了单个兴趣对象 21
6.8 映射阶段 21

6.9 映射的几个简单示例 22
6.10 从这些示例中得到的启发 26
7 COSMIC 方法 - 度量阶段 27
7.1 COSMIC 度量原则 27
7.2 规模汇总 27
7.3 需求变更的规模 27
8 COSMIC 方法的优势和益处 29
9 参考文献 31
10 附录 A. 从 V4.0 到 V4.0.1 的主要变化 32
11 附录 B-COSMIC 的变更请求和建议程序 33

-

1

为什么要度量软件规模?

1.1 为什么会有人想要度量软件规模?

想度量软件规模最可能的原因是，你需要估算其开发的工作量。你的第一想法可能是'这个软件有多大？'而开发工作量通常取决于软件的规模。

打个比方:如果你让供应商估算一项工作的工作量，比如给浴室墙壁贴瓷砖，供应商首先会想知道要贴瓷砖的墙壁的表面积（即规模）。然后已知通常贴瓷砖的速度是多少，也就是每小时能贴多少瓷砖，我们称之为"生产率"，供应商就可以对工作量做出初步估计。这种估算的出发点总是基于:

估算工作量=估算的规模除以生产率

此初步估算可能需要根据浴室特殊的转角或窗户而进行细化，但工作量的主要"成本因子"是需要贴瓷砖的面积（即估算的规模））。

在估算开发或变更某些软件的工作量时，估算过程是相似的。我们需要度量或估算出待开发或变更的软件规模。而生产率数据可以使用与待开发软件项目使用相似技术的项目的数据，可以从以下渠道获得:

- 组织内已完成的项目的生产率数据，
- 基准数据，如可公开获得的 ISBSG 行业数据库，网址为 www.isbsg.org。

（另外，要注意本简介中提到的"软件规模"和"项目规模"的区别。项目可能包括开发软件以外的其他活动；项目规模的度量取决于工作时间，人员水平，周期等。）

1.2 软件规模度量在其他方面的应用

除了项目估算外，度量软件规模对于很多其他方面来说都是非常有价值的。例如:

- **对比。**组织可能希望将采用敏捷方法进行项目管理的生产率与采用传统的瀑布方法管理的生产率进行对比。为此，必须使用相同的方法来度量所有项目类型的软件。
- **控制范围，预算和进度。**随着需求的演变，跟踪规模有助于项目经理控制范围蔓延，从而控制项目预算，并根据预算控制进度。
- **控制缺陷密度。**当项目完成时，可能需要跟踪运行第一个月发现的缺陷，并进行汇报，会用到比如"缺陷密度"，即每单位规模的缺陷数量。

有关软件规模度量的更多应用，请参见第 8 章。

1.3 谁通常会从这些度量中受益?

主流的商业软件供应商经常度量软件的规模，并将它们用于新项目的工作量估算和生产率的度量。这对于管理风险和保持盈利能力至关重要。软件客户则可以通过使用这些度量数据来控制项目范围、供应商的价格/性能、交付质量等。

2

如何度量软件规模?

2.1 和其他度量单位一样，你需要标准

软件的规模可以通过多种度量方法，并在软件项目生命周期的不同阶段进行度量。

如果你出于多种目的度量软件规模，并希望应用于多个活动时，那么很明显，你必须采用一种标准的方法来度量软件规模。COSMIC 方法就是一个国际标准的功能规模度量方法(FSM)，已被命名为 ISO 19761 国际标准。

2.2 度量软件规模最佳方法是什么?

度量软件规模的方法主要有三种:

- 你可以统计为实现软件需求而编写的源代码行数(SLOC)。
- 你可以度量软件需求的规模。
- 你可以使用与软件开发方法或阶段相关的方法。

2.3 统计代码行数

统计 SLOC（代码行）是最早度量软件规模的方法之一。统计 SLOC（代码行）的优点是可以通过源代码分析工具自动计算 SLOC（代码行）。但 SLOC（代码行）计数有很明显的缺点:

- SLOC（代码行）计数没有公认的标准，计数可能因自动计数工具的不同而结果不同。
- 当对同样的软件需求进行编程时，SLOC（代码行）的行数将取决于所使用的编程语言以及程序员的技能。因此，在比较跨项目的生产率，特别是在使用不同的编程语言时，存在本质上的困难。
- 只有在软件程序完成时才可以准确获得 SLOC（代码行）的规模。因此，很难在项目生命周期的早期使用 SLOC（代码行）来估算工作量。如果使用 SLOC 来估算规模，项目必须进展到某个阶段，并对设计和程序结构有一定的了解，同时需要一些经验猜测或类推来进行 SLOC（代码行）估算。
- 一些基于参数设置及选项的编程语言和工具可能无法识别源代码行。

尽管如此，当软件的物理规模与工作量是相关的，并且在某些已经积累了多年度量经验的软件领域中，仍然使用 SLOC（代码行）计数。此外，一些著名的软件项目估算方法，例如 COCOMO II[2]（结构化成本估算模型），也使用了 SLOC（代码行）进行了校准。.

2.4 度量软件需求

度量软件功能需求的方法，被称为“功能规模度量”(或“FSM”)方法，其中之一是 COSMIC 方法，它在对项目进行估算时具有明显的优势， 只要知道“功能性用户需求”(FUR)，就可以使用它。大多数 FSM 方法也有缩放系数，在功能性用户需求详细描述出来之前就可以进行大致的估算。

FSM 方法的另一个优点是，度量的规模与实现软件的技术无关。此外，对于一些 FSM 方法，它们的度量单位是国际标准化的。FSM 方法的度量单位是软件行业最接近标准单位的等价单位。(类似测量长度的米)。

2.5 其他度量软件规模的方法

还有许多其他度量软件规模的方法，但它们几乎都与特定的开发方法或在开发的特定阶段度量有关。

例如：UML 的“系统用例”、敏捷方法的“用户故事”、面向对象方法的“对象个数”等等。这些方法中都没有得到国际用户组织的定义和支持，也无法跨开发平台使用。它们都不是国际标准化的。它们中的一部分，如用户故事，实际上是非常主观的。

2.6 深入理解软件需求 - 功能性需求和非功能性需求

仔细研究软件需求可以发现，软件需求有两种类型，即功能性用户需求(“FUR”)和非功能性需求(“NFR”)。简单地说：

- FUR 说明软件必须为用户提供的任务和服务（ISO 14143-1）。
- NFR 通常是指适用于整个硬件/软件系统的约束。

通过如下例子对 FUR 和 NFR 进行阐述：

业务应用软件案例：某公司的人事系统。

- *FUR 会详细说明该软件必须可以录入和维护公司员工的所有信息，包括他们的姓名、地址、出生日期、就业时间、级别、职位、部门、资质、家属、职业发展和评价记录等。该软件还必须针对存储的数据提供查询。*
- *对于该人事系统，NFR 则为：安全访问控制、系统可用性、软件使用的技术、响应时间要求等等。*

实时软件案例：某嵌入式软件，可控制一个简单的复印机。

- *复印机的 FUR 将详细说明该软件必须支持所有的用户的命令，例如在打开电源后初始化系统，当用户输入所需复印的数量、选择黑白或彩色复印、放大复印件等等各种选项时进行响应。然后在用户按下“开始”按钮后按部就班执行复印。软件还必须响应传感器的信号，如出现卡纸情况、纸张或墨水耗尽等。*
- *复印机的 NFR 则为：系统的时间限制、软件的零缺陷目标、系统可用性标准等。*

请注意，许多最初以非功能性形式出现的系统需求会随着项目的推进演变为软件功能性需求。

例如：对于可审计性或可用性的系统需求可能在项目早期以非功能的形式出现，但是，随着项目的进展，将被转换为软件功能的需求。这些需求可以像其他功能性需求一样采用 COSMIC 方法度量。

在“非功能性和项目需求的指南”中建议了如何在软件项目性能度量、基准对比和估算[14]中考虑这些类型的需求。

2.7 度量方法所需的技术

具备需求工程师或系统分析师所需的技能即可以使用 COSMIC 方法准确地度量规模。而对于使用 COSMIC 规模度量结果进行项目性能对比、制定基准和估算，最好具有统计方法的基础知识。

3

度量方法简史

本章描述了 COSMIC 方法发展的起因和历程。

3.1 一切源何而起?

20 世纪 70 年代中期，IBM 公司的 Allan Albrecht 受命在 IBM 中对于使用多种编程语言的软件项目的生产率进行度量。考虑到 SLOC（代码行）作为交付软件的规模度量的缺点，他想到了一个好办法，即开发一个与所使用的技术无关的软件需求规模度量方法。

Albrecht 的方法于 1979 年首次发表（见参考文献[3]），被称为“功能点分析-FPA”。后来，该方法的后续开发由国际功能点用户组接管，被称为“IFPUG FPA”。

尽管 IFPUG 方法可能仍然是业务应用软件领域使用最广泛的 FSM 方法，但是该方法有几个缺点，其中最重要的是：

- 将 Albrecht 定义的功能类型映射到现代软件需求建模方式变得越来越困难。尤其对于基于服务构建的软件，以及实时类和基础设施软件领域。
- 该方法所定义的功能类型只能限定在一个有限的规模范围中，这意味着该方法对实际软件中存在的极值规模不敏感。而作为一个度量刻度通常应该是线性和开放式的。

3.2 国际标准化组织（ISO）的介入

到 1990 年左右，对于需要一个符合 ISO 标准的 FSM 方法的呼声越来越高。但就当时存在的方法（IFPUG 方法和其他方法）是否为合适的候选方法并没有达成一致。因此，ISO 成立了一个工作组[2]来研究和定义 FSM 的原则。该标准的第一个版本 ISO 14143/1（参见参考文献[4]）于 1998 年发布。

新的标准有助于提高对 FSM 的理解，但没有解决现有方法的缺陷。市场需要一种新的 FSM 方法。

3.3 COSMIC 方法的诞生

ISO 职责在于从现有信息中找寻大家一致认可的标准，而不是开发新方案。因此，一个由来自澳大利亚、欧洲和北美的软件度量专家组成的非正式小组决定在 1998 年底着手开发一种基于 ISO 14143/1 原则的“第二代”FSM 方法。该组织被命名为“COSMIC”，即通用软件度量国际联盟。

COSMIC 的目标
COSMIC 小组的目标是开发一种基于软件工程基本原理和度量理论的软件功能性用户需求度量方法，并获得市场的认可，使其适用于度量业务应用类、实时类和基础设施类软件。

COSMIC 同时还是一个完全自愿的、由来自行业内和学术界的软件度量专家组成的国际小组。

必须强调的是，***自从 1999 年该方法首次发布以来，度量的基本原理始终没有改变***，但 COSMIC 一直致力于根据实践经验对该方法的定义和解释进行完善。

[2] ISO JTC1/SC7/WG12

3.4 ISO 一锤定音：“让市场来决定”

在 2000 年初的 ISO 标准中，仍然没有国际标准的 FSM 方法，也没有对任何现有的方法达成一致。因此，ISO 最终同意了一项政策即“让市场来决定”。所以，现在有五种符合 ISO 标准的 FSM 方法（IFPUG，COSMIC 和其他三种）可供选择。

COSMIC 方法的 ISO 标准（ISO 19761）于 2003 年初首次发布。该标准 2011 年的最新版本可从 www.iso.org 获取。

4

COSMIC 方法简要概述

本章旨在对 COSMIC 方法进行初步的、高度概括。

在所有章节中，COSMIC 方法相关的关键字都是**粗体**。有关关键字的正式定义，请参见“度量手册”[5]的术语表。

4.1 方法的适用范围

COSMIC 方法旨在度量业务应用程序(或“信息管理系统”)、实时和基础设施软件以及某些类型的科学/工程软件的**功能性用户需求**（FUR），适用于软件架构的任何层，以及软件分解的任何级别。

4.2 COSMIC 功能规模度量过程的三个阶段

COSMIC 度量过程如图 4.1 所示。这三个阶段将在下一节中解释。

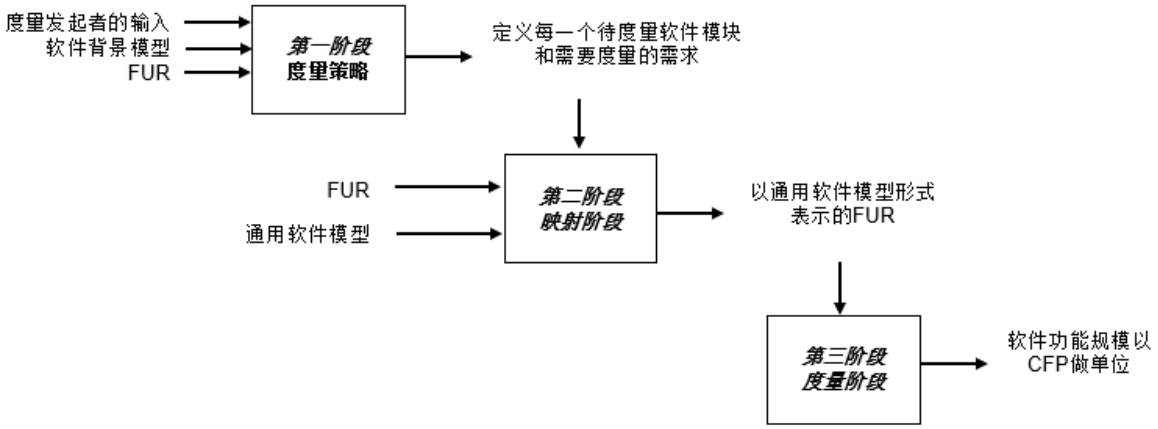


图 4.1-COSMIC 度量过程

4.3 第一阶段：度量策略

我们必须首先确定要度量什么。软件的规模取决于我们将谁或什么定义为**功能用户**，即与软件交互的人、硬件设备或其他软件。为了度量软件的规模，我们必须首先确定度量的**目的**，由此再确定度量的**范围**（待度量的软件 FUR 的范围）和功能用户，还有一些其他的参数[3]。

必须将度量策略的参数文档化，以便将来其他使用者能够正确地理解度量的结果。

4.4 第二阶段：映射

映射阶段的任务是构建 FUR 的 COSMIC 模型，从任何可用的软件制品开始，如概要或详细的需求说明、设计模型、实际已上线的软件等等。为了构建模型，我们依照 COSMIC **通用软件模型**的原则来对 FUR 进行分析。

软件的 FUR 模型遵循四个主要原则：

1. 软件功能由多个**功能处理**组成。每个功能处理的任务是响应软件**功能用户**所触发的**事件**。

[3]度量策略所需参数的基本原则在“软件环境模型”中进行了定义。本简介中没有关于该模型的描述。请参考度量手册。

2. 功能处理由子处理组成。它们只做两件事：移动和运算数据。**数据移动**子处理将数据从功能用户移动到功能处理，并将数据返回给功能用户，分别称为**输入**和**输出**。将数据移动到**持久存储介质**和从**持久存储介质**移出数据的子处理分别称为写和读。图 4.2 展示了四种类型的数据移动。

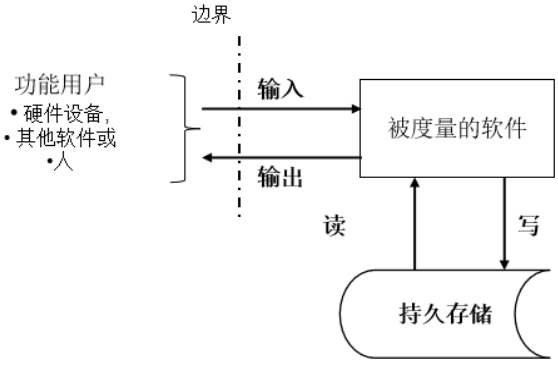


图 4.2 - 四种类型的数据移动

3. 每个**数据移动**（输入、输出、读或写）都会移动单个**数据组**，该数据组的**属性**[4]描述一个单一的“事物”（**兴趣对象**）。
4. 数据运算子处理由它们关联的数据移动来负责。数据运算不单独识别。

一个功能处理在响应了该事件所要求的所有事情后，结束执行。

4.5 第三阶段：度量

COSMIC 方法的度量单位是“COSMIC 功能点”(CFP)。每个数据移动计为 1 CFP。

在度量阶段，我们通过识别每个功能处理的所有数据移动（输入、输出、读和写），并对它们求和，以此来度量一个软件块的规模。

为了提供最小且完整的服务，一个功能处理必须至少包含两个数据移动（一个输入加一个输出，或一个输入加一个写）。因此，一个功能处理的最小规模是 2 CFP。功能处理的规模没有上限。

对现有软件的变更的度量，需要识别所有的新增、修改和删除的数据移动，并将这些移动进行累计汇总。功能处理的变更的最小规模是 1 CFP。

[4] 在其他一些 FSM 方法中称为“数据元素类型”或“DETs”。

5

COSMIC 方法 - 度量策略阶段

COSMIC 功能规模度量过程必须遵循三个阶段。在第一个度量策略阶段，度量人员必须与度量发起者就度量目的达成一致，通常还有一些其他的度量参数都依赖于度量目的。

在本章及后续的章节中，每章中首次出现的 COSMIC 关键字将以**粗体**显示。关键字的正式定义请见度量手册的术语表。度量手册中包括了 COSMIC 方法的所有原则和规则，并给出了许多例子[5]。这篇简介只包含非正式的定义。

5.1 为什么需要一个“策略”?

需要与度量发起者商定并记录度量的目的和其他各种参数，以便将来每个人都能理解度量的规模和如何使用。

实际上，在组织中度量不同类型的软件时，只需要确定少数几个反复出现的参数“模式”。“COSMIC 度量策略模式指南”[6]定义了一些最常见的模式及其用途，可供参考。

5.2 需要确定的五个关键策略参数

- 度量**目的**。目的有助于确定以下所有参数。
- 待度量软件块的**范围**。一个项目可能需要交付多个软件块，或者要度量的功能可能在某种程度上受到限制。功能中包含哪些，不包含哪些。
- 待度量软件块的**分解层级**。例如以下这些是不同的层级：“整个应用程序”（“层级 0”），或分布式系统的主要部件 （“层级 1”），或 SOA 架构中的可重用组件（“层级 2”）。
- 待度量软件块的**功能用户**。这些人或“事物”（硬件设备或其他软件块）是预期的数据发送者或接收者。你所要度量的是他们“看到”的功能。
- 在整个软件架构中软件所处的**层**。待度量的软件块必须仅限于同一层。

将每次度量用到的这些参数文档化，有助于确保将来这些数据只会在“类似的”基础上进行比较和使用。

5.3 软件层级

大多数度量策略参数易于理解。但是“层”这个术语在软件行业中有不同的用法。图 5.1 展示了支持业务应用软件的一个典型计算机系统“分层体系结构”。

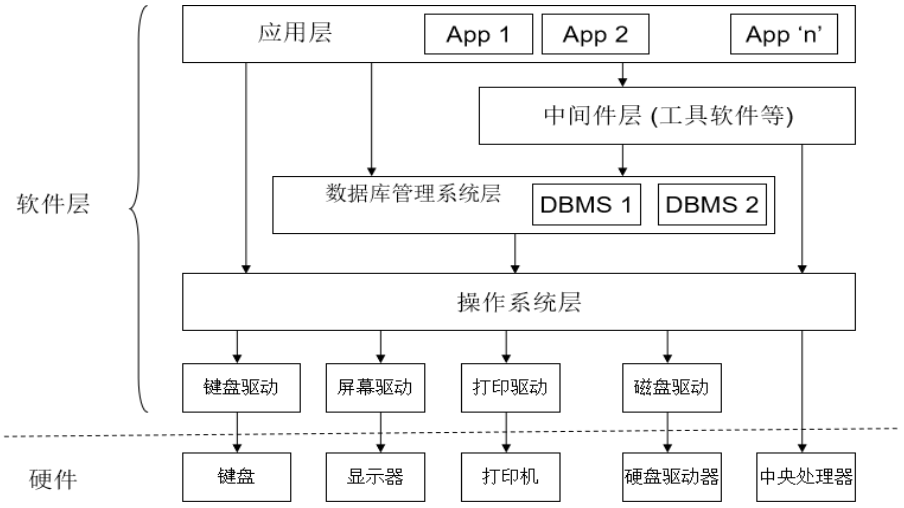


图 5.1-业务/MIS 计算机系统的典型分层软件体系结构

图 5.2 显示了图 5.1 中的应用层可以细分为其他层，这取决于软件架构师的“视图”（显而易见，也依赖于待度量软件的功能用户）。

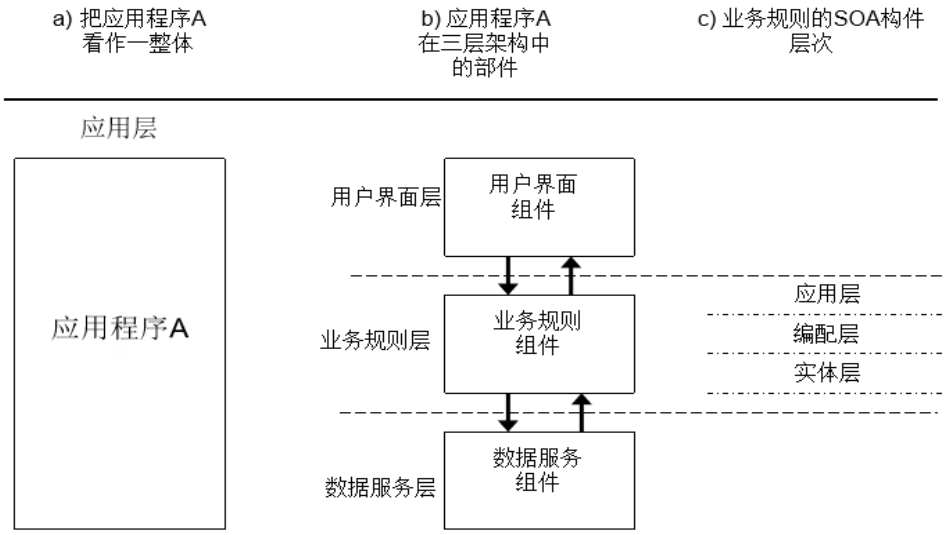


图 5.2-软件块应用程序的三层视图

5.4 度量目的是如何影响其他度量策略参数的案例

业务应用软件示例：假设要开发和度量的软件是一个分布式的三层业务应用程序系统。度量背景是用于与供应商签订合同，以支付为目的，那么软件规模将在整个应用程序级别进行度量，忽略任何组件结构。

案例一

目的：度量已交付应用的规模，以用于合同支付。

范围：一个应用程序的 FUR

功能用户：人和任何其他与其有接口的应用程序

分解层级：无（层级 0）

层：应用程序，即视图 a，如图 5.2 所示

案例二

目的：度量分布式应用程序的每个主要组件的规模，以便供应商能够估算项目工作量，因为每个组件都将使用不同的技术来开发。

范围：每个组件分别度量（即有三个度量范围）。

功能用户：参见图 5.2 视图 b 所示三个层

用户界面组件的功能用户包括人类用户和业务规则组件

业务规则组件的功能用户包括用户界面组件和数据服务组件

数据服务组件的功能用户包括业务规则组件和任何其他接口程序。

分解层级：应用程序的一级分解(层级 1)

层：三个层，如图 5.2 视图 b 所示

实时软件示例：由人类用户使用的嵌入式系统软件的功能，例如一台组合的电脑打印机/复印机，可以从两种功能用户类型的角度来度量。（在这两种情况中，假设我们不考虑任何组件结构及嵌入式软件可能使用的固件。）

案例一

目的：度量人类用户可用的功能规模（面向市场的"消费者可用功能"），以便与其他竞争产品进行比较。

范围：可供人类操作员使用的功能（即不包括操作员无法控制或无法"查看"的功能，例如打印机与电脑通信所需的某些功能。）

功能用户：人类操作员

案例二

目的：度量嵌入式软件开发人员必须为设备运行所提供的功能

范围：嵌入式软件的所有功能

功能用户：软件必须与之交互的所有硬件设备（如键盘、控制按钮、屏幕、打印驱动装置、纸张传输装置等，以及与打印机通信的电脑和打印机驱动软件。）

5.5 在开始度量前，还应该考虑哪些?

可以获取到哪些软件制品作为度量的 FUR，确定这一点是非常重要的。在实际应用中，可用的制品可能无法满足任何一种 FSM 度量所需的精确度，所以度量人员通常在推导 FUR 时必须做一些假设。最好咨询待度量软件的需求专家，有助于理解软件，并使度量尽可能准确。

可能遇到的一些典型问题的例子：

- 如果在项目的早期就需要进行规模度量，需求可能还没有文档化，从而无法进行精准的 COSMIC 度量。针对这种情况，我们有一个指南[7]，介绍了 COSMIC 的缩放系数，用于近似规模估算；
- 有时，软件需求定义得较为概括（宏观），而后逐渐细化（微观），我们称之为**颗粒度级别**。为了确保可比性，规模度量必须在标准的"功能处理"颗粒度级别进行（请参阅以下内容）。如有需要，可使用近似规模度量的方法来对宏观的需求进行度量并缩放至标准的颗粒度级别；
- 有时，需要度量已安装在系统中的软件规模，这些软件已不存在需求。在这种情况下，度量人员需要从可用的制品，如屏幕、用户文档、报告、用户界面等来进行"逆向工程"，确定 FUR。

COSMIC 指南介绍了不同类型的软件或开发方法如何推导和分析 FUR。这些指南都可以在 www.cosmic-sizing.org 网站上找到。

最后，度量人员可能需要估算度量一个特定软件块所需的时间。使用 COSMIC 方法度量的平均速度与其他标准 FSM 方法度量的速度相似。而实际速率可能与这一平均值相差很大。度量所需的工作量往往会因下列原因而增加：

- 可供度量的制品的质量非常差；
- 度量发起人所要求度量的准确性和记录的详细度非常高；
- 度量人员度量此类软件的经验缺失和 COSMIC 方法经验不足。

6

COSMIC 方法 - 映射阶段

映射阶段的任务是通过使用 COSMIC“**通用软件模型**”的原则，从待度量软件的可获取制品中提取该软件的**功能性用户需求(FUR)**。我们首先阐述这些原则，然后详细地描述模型的元素，最后介绍映射过程。

6.1 通用软件模型

原则 - COSMIC 通用软件模型
a) 软件块跨越**边界**与功能用户交互、并与边界内的**持久存储介质**进行交互。
b) 被度量软件块的 FUR 能够被映射到唯一的一组功能处理。
c) 每个功能处理由一系列子处理组成
d) 一个子处理可以是一个**数据移动**或者是一个**数据运算**。
e) 有四类数据移动：**输入**，**输出**，**写**和**读**。 输入从功能用户移动一个数据组到功能处理。 输出从功能处理中移出一个数据组到功能用户。 写从功能处理移动一个数据组到持久存储介质。 读从持久存储介质移动一个数据组到功能处理。
f) 一个数据移动仅移动单个**数据组**
g) 一个数据组描述一个单一的**兴趣对象**，它由唯一的一组**数据属性**集组成。
h) 每个功能处理都是通过**触发输入**数据移动来启动的。触发输入移动的数据组由功能用户响应**触发事件**生成
i) 一个功能处理包括至少一个输入数据移动，以及一个写或输出数据移动，即一个功能处理应该包含至少两个数据移动。一个功能处理中数据移动的数量没有上限。
j) 作为对度量目的的一种近似处理，数据运算子处理不单独度量。任何数据运算的功能被假定已经计算在相关的数据移动内了。

重要提示：上述原则中的所有 COSMIC 关键词（除了“持久存储介质”）都应该以“类型”结尾。例如，“子处理”实际上应该写为“子处理类型”，“输入”应写作“输入类型”。所有的 FSM 方法都是为了识别功能或数据的类型，而非实例。然而，为了便于阅读，我们省略了所有这些关键字中的“类型”，除非需要区分“类型”和“实例”。

原则 a)简单概括了软件所做的一切。其他原则将在本章后续部分解释。

6.2 关键关系：事件/功能用户/功能处理

原则 b)和 h)告诉我们，软件的任务是响应在其功能用户的世界中发生的事件。功能用户通知软件事件发生，并发送事件相关的数据。作为对该事件的响应，软件必须为功能用户做些有用的事情。我们将这种“有用的事情”称为功能处理。 所有软件的 FUR 都可以用功能处理来表示。

存在于功能用户世界中的事件与软件的功能处理之间的关系如图6.1所示。（**边界**是正在度量的软件与其功能用户之间的接口。）

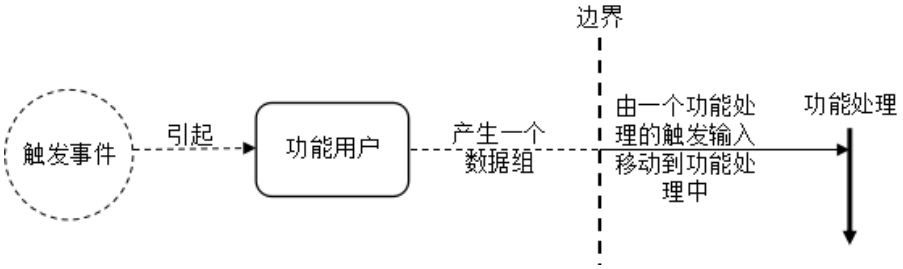


图 6.1-事件、功能用户和功能处理之间的关系

这个图通常的解释是，*事件引发功能用户生成消息（数据组），该消息由"触发输入"移动到其功能处理中，从而启动功能处理。*（请注意，当人类功能用户决定对现有数据进行查询时，才真正生成该事件，然后生成消息。）

事件就是"发生的事情"。触发事件要么已经发生，要么没有发生；在待度量软件的 FUR 中，它不能被再拆分。

案例：一场足球比赛。三个不同的软件程序的 FUR 可能对比赛中发生的事件有完全不同的视图。

应用程序 A 允许记者在报纸上刊登足球比赛的结果。FUR 识别的唯一事件是"比赛结束"。

应用程序 B 是一个"现场直播"系统，它允许记者通过网络向应用程序的在线用户发送在比赛期间记者认为重要的事件的评论，如开球、进球、犯规、受伤等。FUR 识别的唯一事件是"记者向应用程序 B 输入比赛时发生的任何事件的评论"

应用程序 C 允许对球员的表现进行实时监控。每个球员都携带一个 GPS 定位装置和一个心率监控器，以非常短的时间间隔定期传送数据。 FUR 识别的唯一事件是时钟的"节拍"，它负责将每个球员在每个"节拍"时刻所处的位置和心率数据传输到应用程序 C。

（请记住，对于以上这些 FUR，我们实际上应该记录"事件类型"。这三个 FUR，每一个都只识别了一个事件类型，但这三个事件类型又完全不同。但就事件实例而言 ， App A 预计每场比赛只发生一次，App B 预计每场比赛可能发生几十次，而 App C 则可能每场比赛发生成千上万次。）

请注意，图6.1没有说明各种概念之间关系的程度（或"基数"）。例如，一个事件可能被相同或不同的软件块的多个功能用户检测到。 （例如地震可被多个传感器检测到）；一个软件块功能用户也可以感知到多种类型的事件（例如，人类与软件的交互）。

适用于图6.1中关系基数的唯一一条规则是（这是一个非常重要的规则）："任何触发待度量软件块的触发输入可能只启动该软件的一个功能处理"。

6.3 FUR 和功能处理的结构

原则 b)、c)和 d)描述了 FUR 的理论结构，也就是它分解为功能处理和子处理的结构，如 6.2 左图所示。

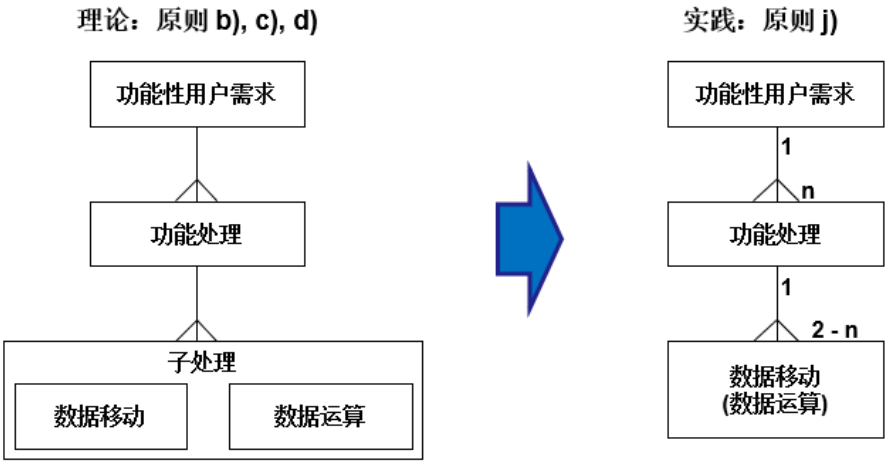


图 6.2-功能性用户需求的结构

["鱼尾"符号表示两个相邻概念之间的推导关系。原则 i)表明一个功能处理可以有 2 个（最少）至 n 个数据移动。]

正确识别功能处理是映射阶段最重要的步骤。所以必须完全理解它的定义。

定义- 功能处理
a) 体现了被度量软件的功能用户需求基本组成部分的一组数据移动，这些数据移动在该 FUR 中是唯一的，并能独立于该 FUR 的其他功能处理进行定义。 b) 一个功能处理可能只有一个触发输入。每个功能处理在接受到由其触发输入数据移动所移动的一个数据组后，开始进行处理。 c) 一个功能处理的数据移动的集合是响应触发输入的所有可能的功能性需求所需要的集合。 注 1：实现时，一个功能处理实例，在收到一个触发输入实例移动的数据组实例时，才开始执行处理。 注 2：除了触发输入外，一个功能处理的 FUR 可能需要一个或多个其他的输入。 注 3：如果功能用户发送了包含错误的数据组，例如，由于传感器失灵，或者人输入的命令存在错误，通常是由功能处理来判断事件是否确实发生、和/或输入的数据是否有效、以及如何响应。

6.4 数据运算的统计

COSMIC 方法没有刻意度量数据运算，因为目前还没有被普遍接受的度量数据运算的方法，因此它可以与数据移动的度量相结合，生成有意义的功能规模。因此，我们引用原则 j)，假设每个数据移动都可以承担与之相关联的数据运算，如图 6.2 右边部分所示。事实证明，这个假设对于所有实践中的度量目的是合理的，正如设计该方法的初衷：用于项目性能度量和估算，并且在该方法的常用领域都行得通。如果出现这样的合并假设无法充分代表数据运算的场景时，COSMIC 度量方法支持对方法的本地化扩展，以克服其局限性。

6.5 四种类型的数据移动

原则 e)如图 6.3 所示。

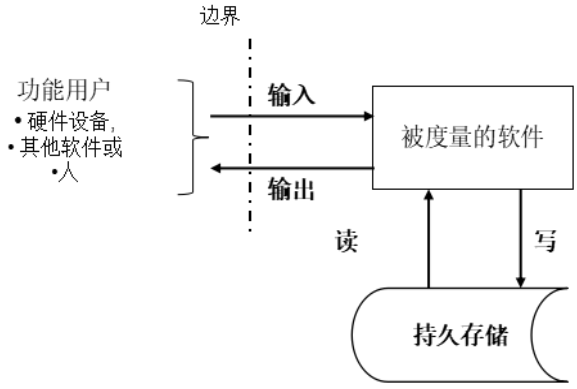


图 6.3-四种类型的数据移动

输入和**输出**分别将数据从功能用户移入或移出软件。

读和**写**分别将数据从**持久存储介质**移出或移入软件。

6.6 持久存储介质

持久存储介质(如图 6.3 所示)是通用软件模型的一个抽象概念。在模型中，如果需要存储数据或检索存储的数据，软件的任何层都可以访问该存储介质。 当一个功能处理写入了一些数据到持久存储介质后，该“持久数据”就可以被其他功能处理所用，或者被写入它的功能处理的另一个实例所用。

从这个概念可以推断，如果您正在度量一个必须存储数据或检索存储数据的应用程序，则不必考虑底层软件或硬件如何从物理层面处理这些数据。只需分析通过写和读需要存储或检索数据的 FUR。

如果在度量策略阶段已经把物理硬件设备定义为软件的功能用户，那么只需要把持久存储介质看作物理磁盘或内存来考虑。而功能用户总是通过输入和输出与要度量的软件进行交互。

6.7 一个数据移动移动了单一数据组，描述了单个兴趣对象

那么，我们如何区分两个不同的输入数据移动，以及区分输出、读和写呢?

兴趣对象是待度量软件必须为之处理或存储数据的“事物”(物理或概念上的)。**数据移动**会移动由一个或多个**数据属性**(在其他 FSM 方法中称为“数据元素类型”)组成的单个**数据组**。数据组中的所有属性都描述同一个兴趣对象。

根据度量目的，不需要识别数据属性。提到它们的唯一原因是，有时通过检查数据属性有助于区分不同的数据组及兴趣对象。

图 6.4 用两个例子展示了这三个概念之间的关系。

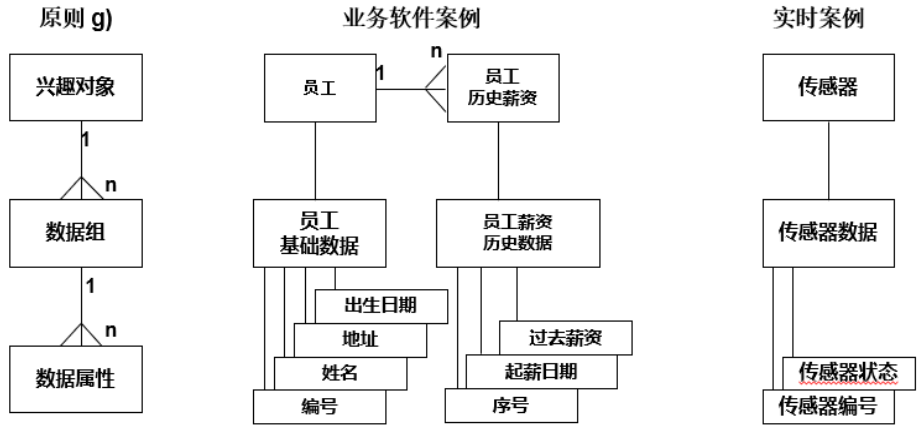


图 6.4 兴趣对象、数据组和数据属性之间的关系

为了帮助您理解这些概念:

- 如果您熟悉在业务应用领域通常使用的数据分析方法，那么在实体关系分析中找到的实体类型，以及在关系数据分析中找到的第三范式的主体都是兴趣对象。 但这些分析方法通常只适用于*存储*(或“持久”)数据组结构。对于 COSMIC 度量来说需要应用这些分析方法来区分兴趣对象，从而区分功能处理的输入和输出中的数据移动。同时，任何一个输入或输出移动一个*瞬态*数据组时，也都是兴趣对象。
- 兴趣对象与 UML 分析中的对象类是一对一的关系，尽管它们是不同的概念。
- 在实时软件领域，通常不需要考虑兴趣对象。如图 6.4 的实时案例，传感器可以看作是发送关于自身数据的功能用户，即功能用户扮演兴趣对象的角色，因此它将在度量策略阶段的早期被识别出来。（除非这样做有助于度量过程，否则做这种区分是没有意义的。）

6.8 映射阶段

如果软件的可获取制品达到功能处理颗粒度级别，我们便可以通过 COSMIC 模型推导出 FUR。这个过程的步骤（记住我们一直说的是*类型*）如下:

- 识别软件必须响应的功能用户世界中的独立事件，即“触发事件”。

- 识别这些触发事件是由哪些功能用户响应的，后者会生成一个由触发输入移动的数据组。
- 为每个触发输入识别一个功能处理。
- 在每个功能处理中，识别响应该触发输入的、满足 FUR 所需的所有其他数据移动，包括输入、输出、读和写。

对于最后一步，需要识别每次数据移动所移动的数据组和所描述的兴趣对象。

6.9 映射的几个简单示例

我们现在可以使用通用软件模型分析一些简单的示例，以便将需求概述映射到 COSMIC 功能处理和数据移动。

业务软件示例：一个简单的人事系统

需求概述：需要一个系统，使人事部职员能够保存和维护员工的数据，包括他们的工资和历次薪资变化。 [该需求还描述了要记录的每个员工的数据(属性)及其验证规则。但在这个简易示例中，度量人员基本不需要考虑。] 每个月都需要一份报告，列出所有员工的姓名和当前薪资、员工总数和当前薪资成本总额。

度量策略参数

度量目的：对人事系统进行精确的功能规模度量，用于项目工作量的估算。

度量范围：需求说明书中规定的整个系统。

功能用户：人事部职员。

层：应用层。

分解级别："0 级"，即没有分解。

映射阶段

一些假设：

- *适用于图 6.4 中业务示例的数据结构。*
- *每个员工将由人事部职员分配一个唯一的 ID。"员工薪资历史记录"记录的主键是【员工 ID，起薪日期】。*
- *需求概述中"维护"一词通常意味着每个兴趣对象必须有创建、读取、更新和删除功能处理（还记得"CRUD"（即增删改查）缩写吗？）。"更新"意味着将允许对数据组中除主键外的任何属性进行修改。*
- *这里有两个兴趣对象("员工"和"员工薪资历史记录")，必须保存这些对象的持久数据。我们需要四个"CRUD"功能处理来维护员工基础数据。.*
- *我们还假设员工在入职时就创建了员工薪资历史记录。之后，员工的工资可以随时更新，即不只是在更新员工基础数据时才更新。因此，员工的薪资历史记录，不需要单独的"创建"或"删除"功能处理。但是，需要"更新员工薪资"的功能处理 和独立的"读取"功能处理，用于查询员工的薪资数据。加上生成月报的处理，意味着需要一共 7 个功能处理才可以满足要求（我们在下面展示了了其中四个的分析。）*
- *在实际情况中，显示从员工薪资历史记录后，可能还需要一个"读取"功能处理来单独查询员工的基础数据。此外，当员工离职时，可能需要"删除"功能处理来归档员工的基本数据和薪资历史数据，而不是实际删除它。为了简单起见，我们忽略了这些需求。*
- *还需要注意的是，在度量在线业务应用程序时，应该忽略那些仅仅是辅助导航和提供选项的"菜单"以及输入屏幕上的"空白"数据。根据度量目的，必须识别输入、输出、读和写的数据移动。*

1. "创建员工"的功能处理分析。

FUR 是输入一名新员工的数据。

（示例显示了每个数据移动和对应的数据组）。

功能处理：“创建员工”。触发事件：雇佣了一个新员工。

触发输入：员工的基础数据

输入：初始薪资及起薪日期

读：员工基础数据（检查输入的员工 ID 是否已经存在）

写：员工基础数据

写：员工薪资历史记录（第一次输入薪资时将创建一条新的记录）

输出：错误/确认消息（对于各种验证失败必须有错误提示信息，对于成功的输入数据也必须有某种形式的确认，我们用一个输出来涵盖所有这些消息。）

2. “读取和更新员工数据”处理分析，包括可能的薪资更新

我们假设用户在输入对一个或多个属性（可能包括新的薪资）的更改之前，首先希望检索和显示员工的基础数据。这个过程需要两个功能处理，第一个的触发事件是用户决定显示现有数据；它是“读取员工基础数据”的功能处理。第二个的触发事件是员工的一个或多个属性在现实世界中发生了变化；这是“更新员工基础数据”功能处理。这两个功能处理是：

功能处理“读取员工数据”。触发事件：决定显示现有数据

触发输入：员工 ID

读：员工基础数据

读：员工历史薪资

输出：员工基础数据

输出：员工薪资历史记录

输出：错误/确认消息（如果输入了不存在的编号）

功能处理“更新员工数据”。触发事件：员工数据在某些方面发生了变化

触发输入：更新员工基础数据（用于更新一个或多个属性）

输入：更新薪资及其起薪日期

写：员工基础数据（更新后的记录）

写：员工薪资历史记录（如果薪资已更新，将创建一条新的记录）

输出：错误/确认消息（输入了无效的数据或可能更新失败）

3. 人资部门编制月报的处理分析

功能处理“员工月报”。触发事件：月底

触发输入：月底的时间标志（每个功能处理都必须有一个触发输入，即使这里没有传递任何变量）。

读：员工基础数据（获取员工 ID 和姓名）

读：员工薪资记录（获取当前薪资）

输出：员工当前薪资（每名员工一行，包括其 ID、姓名和薪资）

输出：每月雇员统计数据（包括：员工总人数及薪资总额）

注意：最后的输出会移动一个描述兴趣对象“所有员工”的数据组。没有存储任何有关此兴趣对象的数据。因此数据组是临时的，而兴趣对象是一群真实的人，即功能用户世界中的真实的“事物”。

我们没有计算此功能处理的错误消息，因为应用程序没有任何理由必须生成这样的消息。（如果找不到数据，操作系统可能会生成错误提示信息， 但这不是应用程序的一部分。）

实时软件示例：一个简单的家庭警报系统

需求概要说明

我们通过实际操作了解如何使用该系统，以推断出对普通住户有哪些可用的功能，以及这其中分配给软件的功能。我们不关注为警报系统维护工程师提供的功能以及首次安装时设置的功能。

警报系统的主要用途是：当警报系统被激活时，如果传感器检测到屋内动静或前门被打开，则发出一到两次警报（产生很大噪音的设备）。

该警报系统软件通过一个键盘和红/绿 LED 灯进行人机交互。该软件还可以接收来自设备的数据，该设备可感知房子的正门是否打开，以及接收来自几个内部运动传感器的数据。（警报系统可处理最多 10 个探测器，这个数字对于本次分析并不重要，因为这些探测器都是相同的），警报系统还控制内部和外部警报器。

警报系统总是"通电状态"，但不是"活动状态"，即运动传感器和前门传感器不工作，除非系统由住户激活。当系统激活时，软件要么处于等待接收来自这些传感器的信号的状态，要么软件轮询这些传感器以获得它们的状态。我们不知道使用的是哪一种模式，但这与功能规模度量无关。

若要激活和关闭警报系统，住户必须在规定的时间内输入正确的 PIN（PIN：个人识别码）。PIN 由软件存储，可以更改，因此肯定存在某种持久存储介质。当输入 PIN 的第一个数字时，内部警报将启动；此警报在输入所有正确的 PIN 数字时停止。如果输入三次错误的 PIN，或者没有在规定的时间内输入正确的 PIN，则外部警报也将启动。

如果电源供电失败，有一个蓄电池可以持续供电，所以必须有一个电源电压检测器。

当电源打开时，绿色 LED 灯就会亮起来。如果警报器启动，或者电源发生故障，绿色 LED 灯就会关闭，红色 LED 灯就会亮起来。

由于某些功能必须在预定的时间内完成，所以必须有计时机制。例如，如果警报系统在离开房子之前被激活，住户必须在预先设定的秒数内离开并关闭前门；否则，警报器就会启动。外部警报器的持续时间不得超过 20 分钟的法定时限。

我们不知道计时功能是如何实现的，但是为了简单起见，我们假设有一个软件实现，只要有需要它就可以启动。监控流逝时间的功能是一种数据运算，我们可以忽略它。

度量策略参数

度量目的：度量可供住户正常使用的嵌入式应用软件的功能处理。

度量范围：警报系统嵌入式应用软件功能，可供住户正常使用。（我们不考虑操作系统层面）

功能用户：下文图表显示了所有硬件功能用户以及他们如何与软件交互。请注意，检测器在功能上都是相同的，因此不需要区分。报警系统的人类用户（称为"住户"）不是功能用户；他/她只通过键盘、音频和视觉信号与应用程序交互。

层：应用层。

分解级别："0 级"，即没有分解。

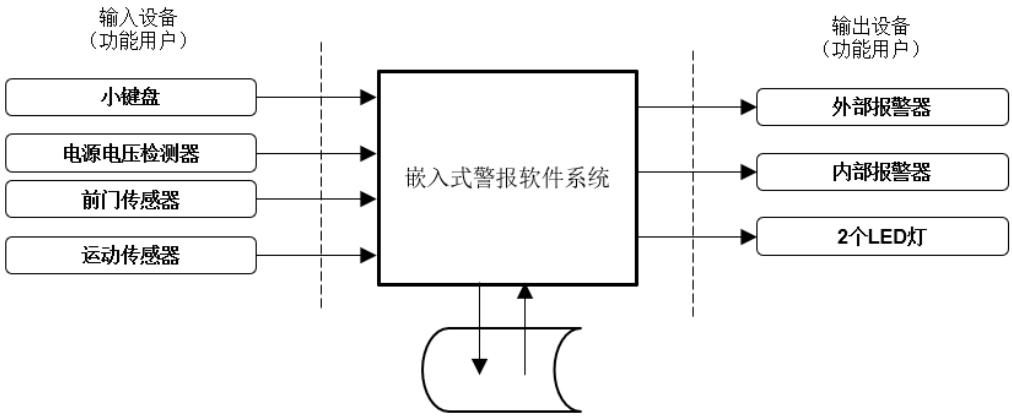


图 6.5 - 家庭警报系统环境图

功能处理：*初始设置完成后，警报系统应用程序为住户提供 9 个功能处理。可以通过软件必须响应的事件来识别。*

1）住户希望变更现有的 PIN。

2）住户希望离开房子时启动警报系统。

3）前门传感器检测门已经打开，同时警报系统已激活。

4）住户希望在家中时启动警报系统，例如，在夜间休息时，或者超出运动传感器感知范围。

5）住户希望在屋内关闭警报系统，例如早上起床后进入运动传感器的范围以前。

6）当警报系统被激活时，运动传感器发出信号（启动内部警报器）。

7）住户希望在事件 3）或事件 6）发生后通过输入正确的 PIN 来关闭警报系统。

8）电源电压检测器发出电源故障信号。

9）电源电压检测器发出电源恢复信号。

一个功能处理实例分析：

我们分析上面列表中的事件 3)（前门被打开，同时警报系统已激活）。当前门传感器检测到此事件时，内部警报器启动；然后必须在规定的时间内输入正确的 PIN，以关闭系统并停止内部警报。如果在规定的时间内没有输入 PIN，或输入错误的 PIN 超过三次，外部警报器就会启动。该功能处理有以下数据移动。

功能处理：检测到可能的入侵者。触发事件：门打开，同时警报系统已激活。

触发输入：来自前门传感器的“开门”消息

读：从持久存储中获取 PIN

输出：将绿色 LED 灯从“打开”切换到“关闭”的消息

输出：将红色 LED 灯从“关闭”切换到“打开”的消息

输出：启动内部警报的消息

输入：输入 PIN 码（如果输入了错误的 PIN，用户可以再输入两次 PIN，但过程是相同的，所以只度量一次）。

*— *：将红色 LED 灯从“打开”切换到“关闭”的消息（PIN 输入成功）*

*— *：将绿色 LED 灯从“关闭”切换到“打开”的信息（PIN 输入成功）*

输出：停止内部警报的消息（PIN 输入成功）

输出：启动外部警报的消息（在输入三次错误的 PIN 之后，或者没有及时输入 PIN）

输出：停止外部警报的消息（20 分钟后，这是法定要求）

*注意：星号这些是重复出现的 LED 的输出实例，但有着不同的数据值（从“打开”切换到“关闭”，或者相反）。

6.10 从这些示例中得到的启发

- 实现细节通常与映射过程无关。例如，人事系统的功能处理可以通过多种方式实现，无论哪种方式应用 COSMIC 模型都会得到相同的数据移动。 同样，我们不需要知道“开门”信息是如何触发警报系统进程的。（当软件激活时，可以轮询前门传感器和移动探测器来确定它们的状态，然后将它们的状态发送到软件中。）
- 将功能处理中的数据移动大致按照执行的顺序进行描述，可以帮助理解。但实际的操作顺序会更加复杂。例如，人事系统中对输入数据的验证可以与错误消息穿插在一起。
- 这些示例说明了功能处理定义的一部分，即功能处理的所有数据移动集（其中包括触发输入）是响应该触发输入的满足 FUR 的所有可能响应的集合。在人事系统中更新员工数据时，只有在薪资发生变化时才会创建新的薪资历史记录。在报警系统的功能处理中，是否发送消息到 LED 和/或警报器将取决于用户是否输入了正确的 PIN。 度量人员唯一的任务是识别功能处理所需的所有数据移动，是否满足 FUR 对于接收到（源自于输入或读）的数据所做的所有可能响应。度量人员不必担心数据移动的顺序，也不必担心在功能处理中是否需要这些数据移动，这取决于输入的数据值。
- 功能处理的数据移动集是类型的集合，而不是实例的集合。
- 警报系统这个例子，每个数据组输入或输出的兴趣对象也是发送或接收该组数据的功能用户（即功能用户正在发送或接收有关自身的数据）。在这些示例中，在度量策略阶段确定功能用户的同时也就确定了兴趣对象。

7

COSMIC 方法 - 度量阶段

在映射阶段结束时，度量人员将生成被度量软件块 FUR 的 COSMIC 模型 （通用软件模型的一个实例）。然后我们可以将度量阶段的规则应用到该模型，来度量软件 FUR 的功能规模。

7.1 COSMIC 度量原则

图 6.2 右边部分所示的模型显示了 COSMIC 的度量原则.

COSMIC 度量原则
软件块的功能规模等于其数据移动的数量

功能规模度量的单位称作“COSMIC 功能点”，简称为“CFP”。单个数据移动的规模定义为 1CFP(输入、输出、读或写)。

7.2 规模汇总

规模可以在不同的级别上进行汇总。

- 功能处理的规模等于其数据移动的数量
- 软件块的规模等于其功能处理的规模的总和
- 只要遵循度量手册中给出的汇总规则，就可以从其组件规模推导出整个软件规模。

下表展示了使用度量手册[5]附录 A 中给出的矩阵，如何记录 6.9 节中人事系统四个功能处理的分析结果。

	数据组名称							数据移动个数				
人事系统功能处理	员工基础数据	员工ID	员工历史薪资	错误/确认消息	月底时钟信号	员工当前薪资	员工总数	输入	输出	读	写	总计
创建员工	E, R, W		E, W	X				2	1	1	2	6
读取员工数据	R, X	E	R, W	X				1	3	2		6
更新员工数据	E, W		E, W	X				2	1		2	5
月底报告	R		R		E	X	X	1	2	2		5
							人事系统总计:	6	7	5	4	22

第 6.9 节分析了家庭报警系统的功能处理，包括 2 个输入、1 个读和 6 个输出。因此，其总规模为 9 CFP。

7.3 需求变更的规模

对现有软件块进行需求变更的规模，例如功能增强类项目引起的变更，度量如下:

- 对一个数据移动（新增、修改或删除）变更的规模，约定为 1CFP。（“修改”可能包括对数据移动和/或该数据组任何属性相关联的数据运算的变更）。

- 因此，对功能处理的变更的最小规模是 1 CFP。
- 一个软件块变更的规模等于其所有功能处理中新增、修改或删除的数据移动之和。

8

COSMIC 方法的优势和益处

COSMIC 作为度量软件需求功能规模的方法，是第一个具有如下特征的方法：

- 根据软件工程的基本原理设计，
- 适用于业务应用、实时和基础设施领域以及某些科学/工程类型的软件，适用于软件体系结构的任何层，适用于从整个应用程序到分解到最小组件的任何层级，
- 无论是软件开发或功能增强都能度量，独立于其所使用的各种技术和开发方法，
- 由国际软件度量专家小组设计和维护
- 设计符合 ISO 14143/1 标准关于功能规模度量的原则要求，
- 是完全开放的，所有的文件都可以从其网站 www.cos-sizing.org 免费下载。
- 已经被认可为国际标准（ISO 19761）。

与“第一代”功能规模度量方法（见第 3 章）相比，COSMIC 方法：

- 由于它的基本设计原则自该方法首次发表以来没有改变，因此非常稳定。这意味着组织对现有度量的投资得到了保障，并且该方法将适用于未来的软件范例。
- 遵从度量理论原则，具有开放式度量量程，这意味着可以用 COSMIC 度量出的规模进行所有的数学运算。
- 没有针对工作量的校准系数，因此保证是纯粹的基于规模的度量。[5]

COSMIC 方法提供以下方面的支持：

- 完善的方法定义；“度量手册”已翻译成十多种语言。
- 描述如何将该方法应用于特定类型软件的指南，例如数据仓库或 SOA 软件，或特定的项目管理方法，例如敏捷方法。
- 关于如何收集和报告度量数据的案例研究和工具。
- 全面的基准数据参见 www.isbsg.org。众多不同领域的软件项目度量结果显示，以 COSMIC 方法度量的规模与项目的工作量紧密相关。
- 供应商服务，包括培训、咨询、估算工具等。
- 入门级认证考试。
- 确保度量的准确性和可比性的指南。
- 当需求的所有细节还没有确定、或为了进行快速估算时，可以在项目生命周期的早期使用 COSMIC 近似估算法[7]。
- 用统计相关性分析方法将第一代 FSM 方法度量的规模转换为 COSMIC 规模的文献[16]。
- 在 LinkedIn 上的 COSMIC 用户和 Twitter 上 COSMIC_FSM 的活跃用户群。
- 论坛地址 www.cos-sizing.org，用于提问和讨论，以及发布新闻。

COSMIC 方法在世界范围内成功地应用于项目性能度量、估算、项目范围控制等。下面的思维导图显示了功能规模度量的一系列用途。

[5] 最近研究[15]表明，COSMIC 与 MkII FP 方法度量同一软件的规模具有很好的相关性。后一种方法是将规模与项目工作量相关联来进行校准的。 COSMIC 与其良好的相关性意味着符合其设计目的（作为衡量项目性能的一个组成部分，并作为项目工作量估算的输入），即使它没有针对工作量进行校准。

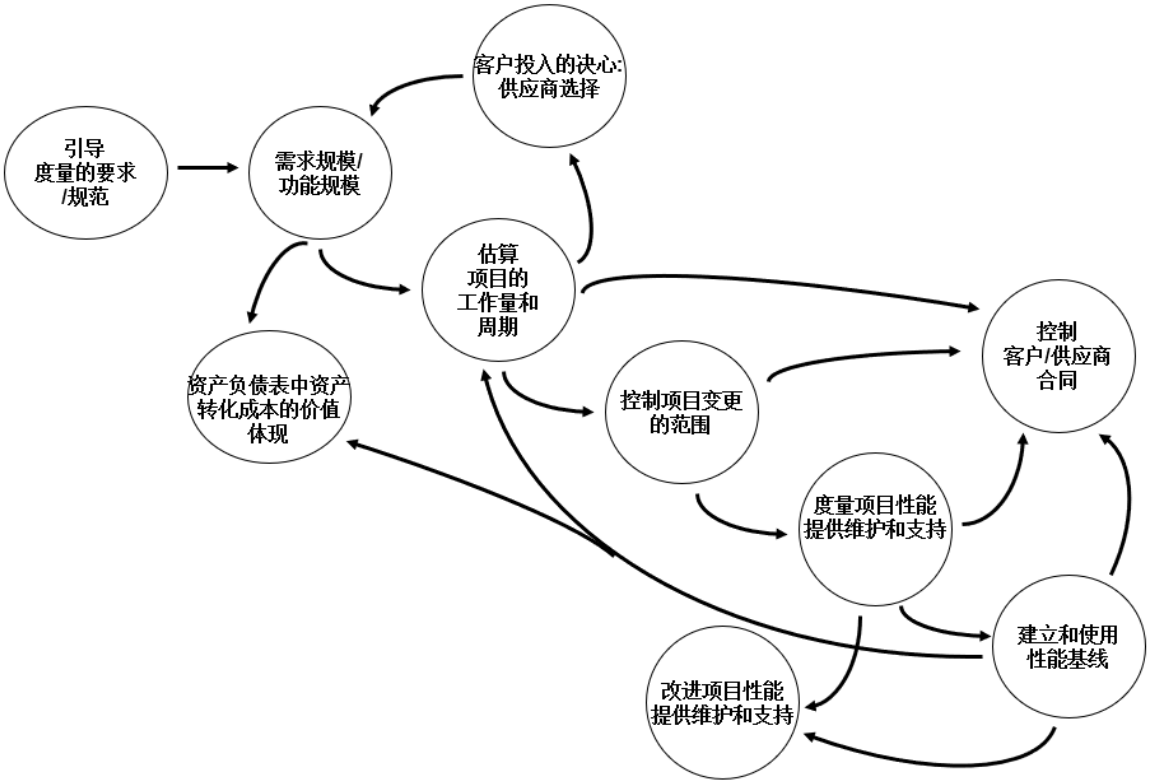


图 8.1-功能规模度量多种用途的思维导图

COSMIC 方法正被学术界广泛研究。其中值得关注的研究是可以通过多种方法实现 COSMIC 度量的自动化，例如：以 UML 方式对需求进行建模，以及从可执行程序中进行自动化度量。

www.cosmic-sizing.org 网站有一个庞大的研究和会议论文库。

参考文献

[1] ISO 19761:2011 – Software Engineering – COSMIC: a functional size measurement method' obtainable from www.iso.org.

[2] The COCOMO II Estimating Model', www.csse.usc.edu/csse/research/COCOMOII

[3] Albrecht, A.J., 'Measuring Application Development Productivity', IBM Applications Development Symposium, Monterey, October 1979

[4] ISO 14143-1:2012 – Software and Systems Engineering – Software measurement – Functional size measurement – Definition of concepts

[5] The 'COSMIC Functional Size Measurement Method, version 4.0.1: Measurement Manual'. (The COSMIC Implementation Guide for ISO 19761: 2011), April 2015

[6] Guideline for 'Measurement Strategy Patterns': Ensuring that COSMIC size measurements may be compared, version 1.0, March 2013

[7] Guideline for early or rapid COSMIC functional size measurement by approximate approaches', July 2015

[8] Guideline for sizing Business Application software, version 1.1, May 2008

[9] Guideline for sizing Real-time software, version 1.0, June 2012

[10] Guideline for sizing Data Warehousing application software, version 1.0, May 2009

[11] Guideline for sizing Service-Oriented Architecture software, version 1.0, April 2010

[12] Guideline for the use of COSMIC FSM to manage Agile projects, version 1.0, September 2011

[13] Guideline for mapping from the Unified Modeling Language to COSMIC FSM (in preparation)

[14] Guideline on Non-functional and Project requirements: how to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating, v1, November 2015

[15] A, Dasgupta, C. Gencel, C., Symons, 'A process to improve the accuracy of MkII FP to COSMIC size conversions: insights into the COSMIC method design assumptions', IWSM 2015, Krakow, Poland.

[16] 'Guideline on how to convert 'First Generation' Function Point sizes to COSMIC sizes' (to be published early 2016)

附录 A. 从 V4.0 到 V4.0.1 的主要变化

V4.0 章节	变更
通用	网址 www.cosmic-sizing.org 取代了 www.cosmicon.com
通用	其他 COSMIC 出版物的参考资料已更新到截至 2015 年底版本。
2.6	COSMIC 对非功能性需求(NFR)的定义不再包括项目需求和约束。现在这些问题在[14]中针对 NFR 进行了明确定义和处理。这一变化导致对 2.6 节中的示例作了微小改动。
6.9	在映射阶段对简单人事系统示例的假设描述进行了一些小幅编辑改动，以更加明确。
6.9	某大型家庭实时报警系统的实例需要澄清和修正，其中最重要的如下: • 文本可能会混淆“警报系统 alarm system”和“警报 alarm”(发出巨大噪音的东西)。后者被重新命名为“警报 siren”。 • 重新编写了报警系统功能描述的一些方面，使其更加清晰和完整。重新编写事件列表，以明确调用或感知事件的功能性用户。 • 识别到需要增加一个额外的功能处理(第 7 个)。
8	增加了一些最近关于功能规模度量转换(脚注 5)和 COSMIC 度量自动化方面的研究成果，表明了 COSMIC 方法进一步的潜在优势。

附录 B-COSMIC 的变更请求和建议程序

COSMIC 度量实践委员会（MPC）非常愿意接受反馈、建议，以及对此指南的变更请求（如果必要的话）。本附录展示了如何与 COSMIC MPC 联系。

所有与 COSMIC MPC 的联系都应该通过电子邮件的方式发送到下面的地址：

mpc-chair@cosmic-sizing.org

非正式的反馈和建议

关于“度量手册”的非正式评论或反馈意见，比如理解或应用 COSMIC 方法的任何困难，一般性改进的建议等，都可以通过电子邮件发送到上面的地址。消息会被登记下来，并且通常在收到后的两周内给予答复。度量实践委员会不保证对一般性意见给予答复。

正式的变更请求

如果“度量手册”的读者本指南的读者如发现某个文字缺陷，或不足之处需要澄清，或者一些文字需要加强，那么可以提交一个正式的变更请求（“CR”）。正式的 CR 会被登记下来，并在收到后的两周内给予答复。每个 CR 将分配一个序列号，在 COSMIC MPC 的成员中循环传递，COSMIC MPC 是一个世界范围内 COSMIC 方法的专家组。他们的正常评审周期最少需要一个月，如果变更请求很难解决，可能需要更长的时间。评审的结果可能是 CR 被接受，或者拒绝，或者“未决待进一步讨论”（后面这种情况，例如本 CR 要依赖另一个 CR），结果会尽快地反馈给提交者。

只有包含了下面的所有信息，一个正式的 CR 才会被接受。

- 提交 CR 的人员的姓名、职位和单位
- 提交 CR 的人员的详细联系方式
- 提交日期
- 关于 CR 的目的的总体陈述（如“需要改进文本……”）
- 需要变更、替换或删除的实际文字（或澄清引用出处）
- 建议增加或替换的文字
- 关于变更的必要性的充分解释

提交 CR 的表格可以从门户网站 www.cosmic-sizing.org 获得。

COSMIC MPC 对 CR 的评审结果，以及在哪个版本的“度量手册”中应用这个 CR（如果被接受了的话）的决定是最终的决定。

关于 COSMIC 方法应用的问题

COSMIC MPC抱歉不能回答与COSMIC方法应用相关的问题。有商业机构能够提供本方法的培训和顾问工作，或者支持工具。详细情况请咨询www.cosmic-sizing.org网站。