



COSMIC 度量手册
ISO 19761

第二部分：
指南

5.0 版本
2020 年 5 月

前言

COSMIC 度量手册（ISO 19761:2011）包括三个部分：

第一部分：原则、定义&规则

第二部分：指南

第三部分：COSMIC 概念和度量实践的举例

本文档为第二部分，包含了 COSMIC 小组为了确保 COSMIC 度量结果准确、可重复和可再现，制定的一系列指南。

本版本（2020.5）变更：

- 修改 3.7 节标题，“规则 21-24 的指南”修改为“规则 24 的指南”。
- 在“如何识别数据移动的修改”中增加了三个要点（来自度量手册 4.0.2 版本）。

度量手册基于 ISO 19761:2011，并在 2019 年得到再次确认。新版 COSMIC 度量手册 5.0 版本将取代 4.0.2 版本：该版本只是对 4.0.2 版本进行了简化，并没有修改度量的定义、规则和指南方面的内容。

COSMIC 度量手册和技术报告的通用版本，包括其它语言的翻译版，可以在门户网站 www.cosmic-sizing.org 上找到。

编辑：

Alain Abran，魁北克大学（加拿大）
Peter Fagg（英国）
Arlan Lestherhuis（荷兰）

COSMIC 度量实践委员会的其他成员：

Diana Baklizky，（巴西）
Jean-Marc Desharnais，魁北克大学（加拿大）
Cigdem Gencel，博尔扎诺自由大学（意大利）
Dylan Ren，麦哲思科技（北京）有限公司（中国）
Bruce Reynolds，Telecote 研究所（美国）
Hassan Soubra，开罗德国大学（埃及）
Sylvie Trudel，魁北克蒙特利尔大学（加拿大）
Frank Vogelesang，Metri 集团（荷兰）

中文版翻译者&校对者：

郭玲，麦哲思科技（北京）有限公司

任甲林，麦哲思科技（北京）有限公司
夏思文，易才人力资源公司
张坤，上海千杉网络技术发展有限公司
高艳，无锡农村商业银行
程敏，中汇信息技术（上海）有限公司
徐妍玲，麦哲思科技（北京）有限公司

版权 2020。版权所有。通用软件度量国际联盟（COSMIC）。非用于商业目的情况下，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。

目录

1. 介绍	5
1.1 本文档的目的	5
1.2. COSMIC 度量过程.....	5
2. 度量策略阶段	5
2.1 度量策略阶段概览	5
2.2 确定 FSM 的目的和范围	6
2.3 从软件制品中识别 FUR.....	6
2.4 非功能性需求	7
2.5 识别层	8
2.6 识别功能用户	9
2.7 分解层级.....	9
2.8 环境图	9
2.9 识别颗粒度级别	9
3. 映射阶段	10
3.1 把功能性用户需求（FUR）映射为通用软件模型	10
3.2 识别功能处理	10
3.3 识别数据组.....	11
3.3.1 识别数据组.....	11
3.3.2 识别兴趣对象和数据组.....	12
3.3.3 不适合作为数据候选对象的数据或数据组.....	12
3.3.4 识别数据属性（可选）	13
3.4 识别数据移动	13
3.5 度量分布式软件系统的组件.....	15
3.6 软件的复用	15
3.7 度量软件变更的规模.....	15
4. 其他话题	16
4.1 扩展 COSMIC 度量方法——本地扩展	16
4.2 COSMIC 在敏捷中的应用	16
5. 度量阶段	16
6. 度量报告	17

1. 介绍

1.1 本文档的目的

本文档的目的是为使用 COSMIC 即 ISO19761 功能点方法的实践者提供指导。

第二部分是第一部分的补充，第三部分将包括很多案例。

1.2. COSMIC 度量过程

在第一部分展示过的 COSMIC 度量过程，如图 1.1

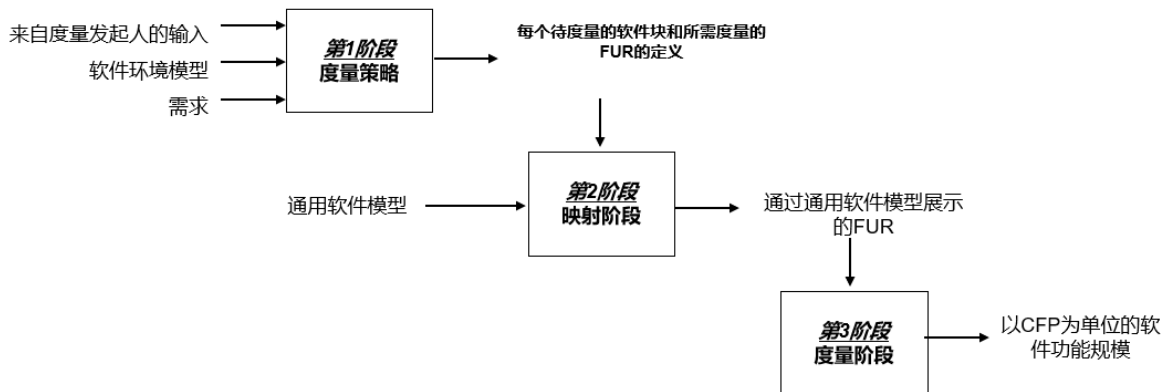


图 1.1 - COSMIC 方法度量过程

2. 度量策略阶段

2.1 度量策略阶段概览

图 2.1 度量策略阶段示意图。

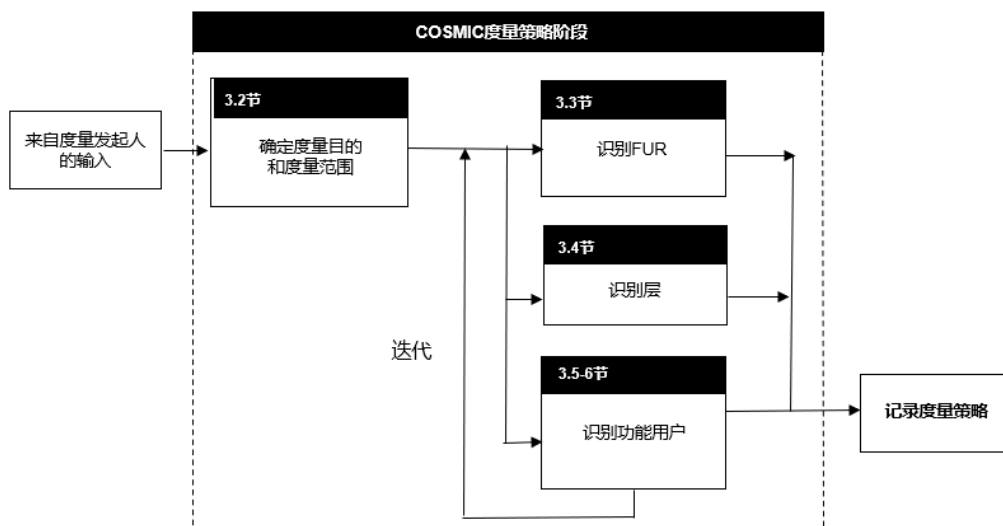


图 2.1 - 确定度量策略阶段的过程

2.2 确定 FSM 的目的和范围

“目的”这一术语采用其通常的英语含义。它将帮助度量者确定：

- 度量的范围以及度量所需的制品。
- 功能用户。
- 功能变更。
- 在项目生命周期中进行度量的时间点。
- 度量的精确性，以及是否应该使用标准的 COSMIC 方法，或是近似度量技术（例如，在项目生命周期早期，FUR 被完整定义之前）。

为了帮助确定度量策略，“度量策略模式”指南为不同类型的软件提供了一组用于度量软件规模的标准参数集，称为“度量策略模式”（简称“度量模式”）。始终如一地使用相同的度量模式，可以帮助度量者确保在相同目的下的度量活动能一致地执行，可以有把握地与其他相同模式的度量进行对比，也有助于所有日后的使用者对度量的正确解读。使用标准模式的另一个好处是大大减少了再次确定度量策略参数的工作量。

在使用标准模式之前，COSMIC 组委会强烈建议度量者学习并掌握 COSMIC 方法，特别是度量策略参数。

2.3 从软件制品中识别 FUR

在把 FUR 映射为 COSMIC “软件模型”前，度量者通常需要从软件的已有制品中提取 FUR。

如图 2.2 所示，在软件开发出来之前，可以从软件工程制品中导出 FUR。因此，软件的功能规模可在软件实现前进行度量。

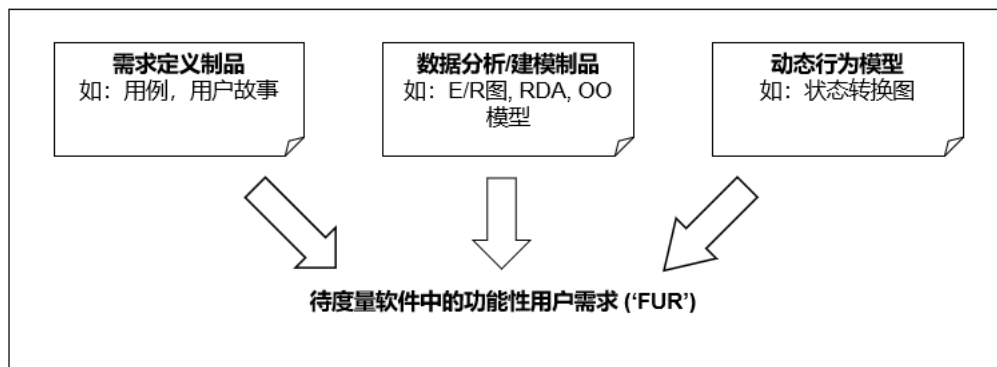


图 2.2 - 功能性用户需求在软件实现前的来源

注：部分软件可能需要在没有或只有少量可用架构或设计制品的情况下进行度量，并且功能性用户需求也可能没有文档化（如：历史软件）。在这种情况下，即使软件已实现，仍然可以从下列软件制品中获取 FUR，如图 2.3。

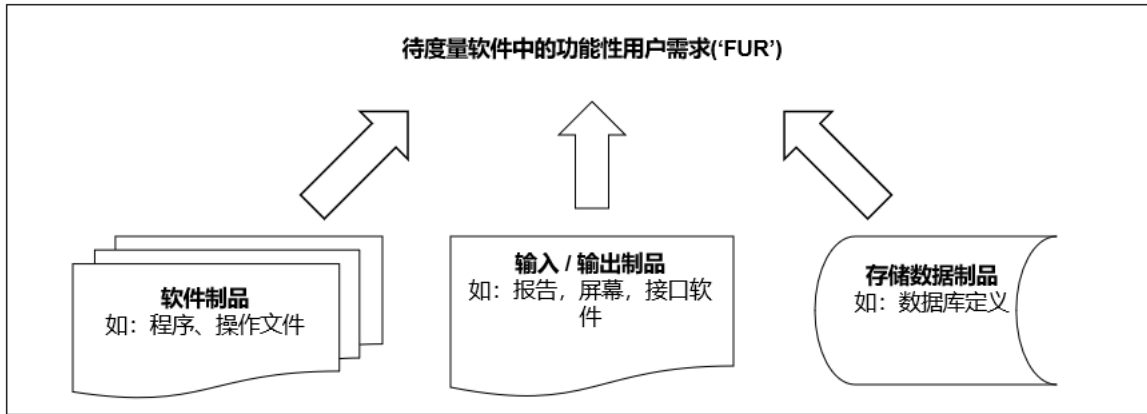


图 2.3 - 功能性用户需求在软件实现后的来源

从不同类型的软件工程制品和从已安装的软件中提取 FUR 的过程和花费的工作量显然是不同的；这些提取 FUR 的过程不在度量手册中进行讨论。根据度量目的，COSMIC 方法假定待度量软件的 FUR 要么存在，要么可以从软件制品中提取或推导出来。

如果度量者理解了这两个模型，即使在度量的过程中由于缺少信息或信息模糊不得不做出假设，度量者也是可以从被度量软件中提取 FUR。

2.4 非功能性需求

非功能性需求（NFR）对于软件项目来说非常重要。在某些极端的情况下，软件密集型系统的需求定义中对 NFR 的描述可能与功能性需求的一样多。COSMIC 方法可以用来度量某些最初呈现为非功能的需求，但随着项目进展演化为在软件功能中可以实现的混合需求，而另外一部分则是真正的非功能需求或约束。参照图 2.4。

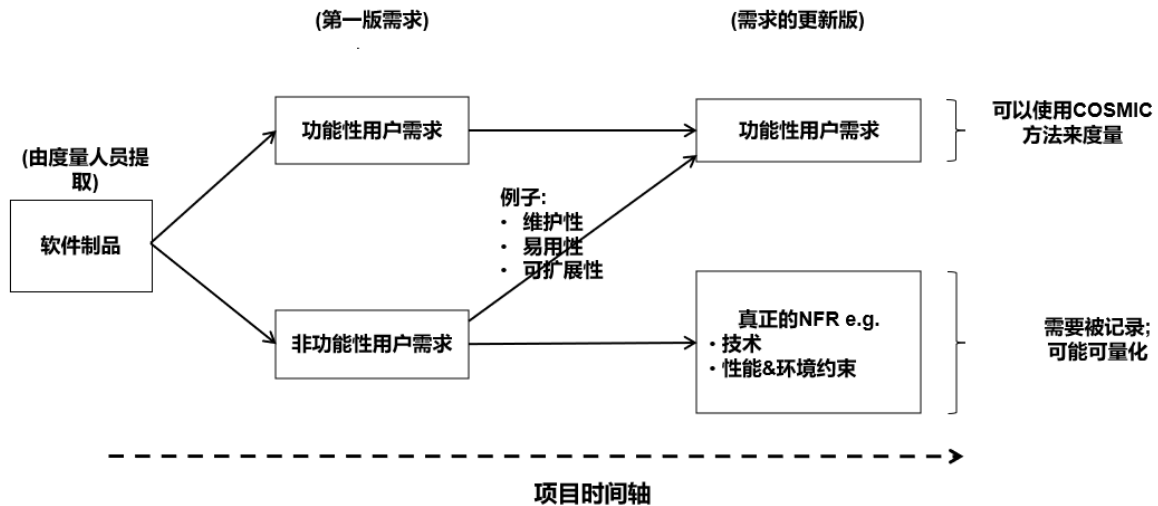


图 2.4 - 许多最初呈现为 NFR 的需求随着项目的推进会发展成 FUR

对于许多系统质量和约束而言确实如此，如响应时间、易用性和可维护性等。这些在项目初期“藏”在 NFR 中的软件功能，一旦被识别，就可以像对待其他的软件功能一样，用 COSMIC 方法度量其规模。软件规模会随着项目的进展不断增长的原因之一就是没有识别出这些“隐藏”的功能规模。

更多关于 NFR 的详细说明请参考“非功能性需求和项目需求指南”。

2.5 识别层

软件架构可能非常复杂，而 COSMIC 只需使用一个非常简单的软件架构视图，便足以应对度量目的。度量者负责对架构进行简化。本节将为度量者提供指导。

因为待度量的软件块范围必须限制在一层中，因此定义 FSM 范围时需要度量者首先确定软件架构中有哪些层次。本节我们将讨论 COSMIC 方法中使用的软件“层”的概念。理由如下：

- 度量者可能需要在历史的软件环境中进行度量，这些软件已经非常久远，并且没有按照清晰的架构来设计。因此度量人员需要一些指导，以按照 COSMIC 方法对层进行区分；
- 在软件领域中，“层”和“分层体系结构”的使用经常不一致。当度量者要度量一个所谓的“分层体系结构”的软件时，最好先确认一下该架构中的“层”是否与 COSMIC 方法中的定义一致。因此，度量者应该在“分层体系结构”中的特定架构对象和本手册定义的层的概念之间建立对应关系。

在软件体系结构中，每一层可能具备以下特点：

- a) 一层中的软件会根据已定义的准则提供一组内聚的服务，而且其他层里的软件无需了解这些服务是如何实现的也能利用它们。
- b) 任意两层中软件之间的关系可用“通信规则”来定义，有两种情况：
 - “分层的”，如 A 层的软件可以使用 B 层软件提供的服务，但反之则不成立（通常指客户端与服务器端）。
 - “双向的”，如 A 层的软件可以使用 B 层的软件，反之亦然（通常指 P2P）。
- c) 一层的软件通过相应的功能处理与另一层的软件交换数据组。
- d) 一层的软件没有必要使用其他层的软件提供的所有功能服务。
- e) 在一个已定义的软件体系结构中属于同一层的软件，根据另外一个不同的体系架构可能被划分为其他的层。

规则 4 的指南：识别层

如果 FSM 的整体范围涉及到多个层，则度量人员应按照如下步骤进行：

- 如果待度量软件存在于一个已建立的层次结构中，此结构又可以映射到如上定义的 COSMIC 分层特点中，那么就可以用该结构来识别层以满足度量目的。
- 但是，如果度量目的要求度量某些并非按照 COSMIC 分层特点结构化的软件，度量者应该尝试使用上文定义的原则将软件分解为层。
- 按照惯例，基础设施软件包如数据库管理系统、操作系统或设备驱动等提供的服务都可以被处于其他层的软件调用，而他们本身则处于单独的层中。

通常在软件架构中，“顶层”，即在层次结构中不从属于其他任意层的层，被称为“应用层”。在应用层的软件依赖于其他层提供的服务。处于顶层的软件本身可以再进行分层，如在用户界面、业务规则和数据服务组件的三层架构中。

识别后，每一层可在度量报告中进行记录并加以标识。

2.6 识别功能用户

关于规则 7 的指南：识别功能用户

功能用户的识别由待度量的功能“用户”需求和度量目的决定。

2.7 分解层级

软件块的规模度量只能直接与处于相同分解层级的组件进行对比。这一点很重要，因为根据第五章的规则，在不同分解层级的软件块的规模不能简单的叠加。因此，在比较开发不同软件块的项目性能（如：生产率=规模/工作量）时，必须是这些软件块都位于同一分解的层次上才有可比性。此外，不同软件项目的性能（如生产力=规模/工作量）也只有处于相同分解层级时才能比较。

软件不同的分解层级可能对应的是软件层的不同“视图”，如图 2.4 中的第三部分。然而，无论软件设计是否使用分层架构模型，软件都可以被分解为多个层级。

2.8 环境图

在定义 FSM 的范围和功能用户时，为待度量软件绘制环境图很有帮助。环境图展示了软件块和功能用户（人类，硬件设备或其他软件）之间的数据流，并展示了软件块和持久性存储介质之间的数据流。

环境图是将度量模式应用到待度量软件的一个实例。图 2.5 列出了环境图所用到的主要符号及含义：


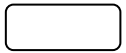
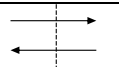
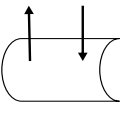
符号	含义
	待度量软件块（加粗框），即度量范围的定义。
	任何待度量软件的功能用户。
	箭头代表跨越功能用户和待度量软件间的边界（虚线）的所有数据移动。
	箭头表示待度量软件与“持久存储介质”间的所有数据移动。 (代表“数据存储”的标准流程图标志强调持久存储介质是一个抽象的概念。使用此标志表明软件并不直接与物理硬件存储器交互。)

图 2.5 - 环境图中的符号

2.9 识别颗粒度级别

COSMIC 方法要求 FUR 以足够详细的程度表示，以便建立 COSMIC 度量模型，这被称为颗粒度级别。

根据 ISO19761 的规则，使用 COSMIC FSM 方法得到功能规模，需要需求达到可以识别功能处理和其数据移动的程度。当需求模糊时，可以使用“早期软件规模度量方法：实践者指南”，进行近似规模度量。

注 1：在软件开发的初始阶段，实际需求通常处于高颗粒度级别，即比较概括或是缺少细节。随着项目的进展，实际的需求逐渐细化（如需求版本从 v1.0 升级到 v2.0, v3.0 等），需求中包含了越来越多的细节，即变为低颗粒度级别。需求的不同详细程度便被称为不同的“颗粒度级别”。

注 2：度量者应该意识到：随着项目的进展，在不同的时期，项目的需求通常处于不同的颗粒度级别，并通过不同的形式体现。

关于度量不同颗粒度级别和分解等级的案例，请见“使用近似方法进行早期快速 COSMIC 功能规模度量”中的通信系统的案例。

3. 映射阶段

功能处理由移动了数据（数据移动）和可能运算了数据（数据运算）的子处理组成。

3.1 把功能性用户需求（FUR）映射为通用软件模型

图 3.1 展示了把已有的软件制品中的功能性用户需求（FUR）映射为 COSMIC 通用软件模型所需格式的过程。

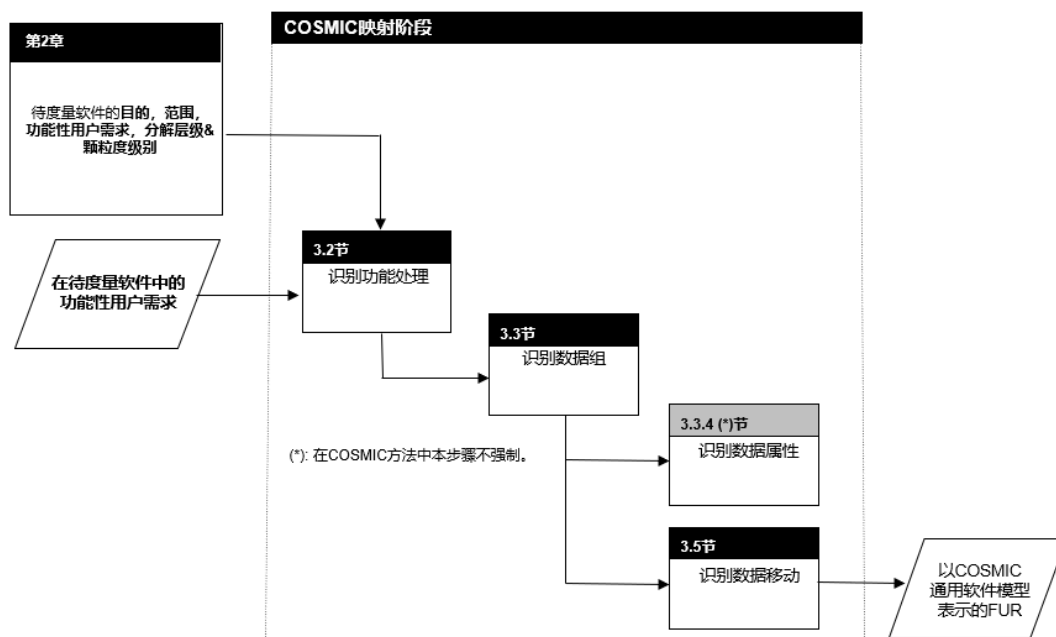


图 3.1 - COSMIC 映射过程的通用方法

目前已有一些指南描述了如何将不同领域中各种数据分析和需求定义方法映射到 COSMIC 方法：

- 业务应用软件规模度量指南
- 数据仓库软件规模度量指南
- SOA 软件规模度量指南
- 实时软件规模度量指南

对于业务应用领域和实时领域，也可参考“快速参考指南”，包含了一些度量过程方面的简单介绍。

3.2 识别功能处理

度量阶段的第一步是从 FUR 中识别待度量软件块的功能处理。

触发事件、功能用户和触发了待度量功能处理的数据移动之间的关系如图 3.2 所示：触发事件引起功能用户生成一个数据组，该数据组由功能处理的触发输入所移动，启动了功能处理。

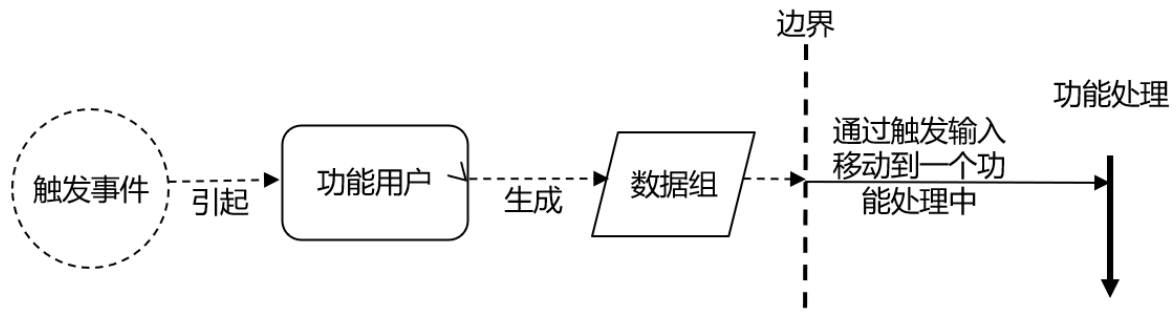


图 3.2 - 触发事件、功能用户及功能处理间的关系

注 1：为了便于阅读，在说明功能用户生成了触发功能处理的触发输入时，或者进一步简洁，功能用户启动了功能处理时，省略了对数据组的引用。

注 2：图 3.2 中所有概念之间（触发事件/功能用户/数据组/触发输入/功能处理）可能是一对多、多对一或多对多的关系，只有一个例外，即由触发输入移动的数据组可能只启动其所属的功能处理。

规则 10 的指南：识别功能处理

根据图 3.2 的关系图，在识别功能用户后，识别功能处理的流程如下：

1. 识别功能用户世界中必须响应的独立事件，即“触发事件”。

注：可以在状态图或实体生命周期图中识别触发事件，因为一些状态转换或实体生命周期转换对应着软件必须响应的触发事件。

2. 识别出每个触发事件必须响应的功能用户。
3. 识别出每个功能用户对事件进行响应的数据组（即触发输入）。
4. 识别出每个触发输入启动的功能处理。

使用以下规则来检查被识别出的功能处理是否正确：

1. 是否所有已识别的功能处理都处于同一层？
2. 是否所有已识别的功能处理都包括了一个输入和至少一个写或输出数据移动？

3.3 识别数据组

3.3.1 识别数据组

在识别了功能处理之后，需要识别数据移动。接下来的指南将有助于识别数据组，也就是兴趣对象，特别是功能处理输出的兴趣对象。

规则 11 的指南：识别同一个功能处理中被移动的不同的数据组

对于功能处理的输入、输出、读、写中的所有数据属性：

- a) 具有不同实例频率的数据属性集描述的是不同的兴趣对象；

b) 具有相同实例频率但不同关键属性的数据属性集，描述的是不同的兴趣对象。按照指南 a) 和 b) 判断后归为一个集合的数据属性属于同一数据组，除非在 FUR 中明确说明了此功能处理中可能有不止一个数据组描述了同一个兴趣对象。（请见数据移动唯一性中的规则 13 和 14，例子 b) 和 c)）。

注 1：待度量软件的功能用户可能是功能用户发送/接受的数据组的兴趣对象。

注 2：理论上讲，从功能性用户需求的角度来看，如果一个属性足以描述兴趣对象的话，那么，数据组就可能只包含一个数据属性。实践中，这样的情况通常出现在实时应用软件中（如一个时钟节拍数据组或表示传感器状态的数据组），而在业务应用软件中并不多见。

数据组的起源可以有多种形式，例如：

- a) 在硬件存储设备上的物理记录（文档、数据库表、ROM 等）。
- b) 计算机暂存空间中的物理结构（动态分配或通过预先分配的内存空间块分配的数据结构）
- c) 输入/输出设备（显示屏、打印报告、控制面板显示等）上功能相关的数据属性的合集展示。
- d) 在设备和计算机或网络等之间传输的信息。

3.3.2 识别兴趣对象和数据组

为适用于尽可能广泛的软件，兴趣对象和数据组的定义、原则刻意地被放宽。这一点有时会导致在度量一个特定软件块时，很难应用这些定义和原则。请参考第三部分的案例，以帮助理解如何将原则应用在特定场景中。

当一组数据属性被移入或移出一个功能处理，或者被一个功能处理移入或移出持久存储介质，在分析它们时，关键是要确定这些属性传递的数据是否全部是关于一个单一的“兴趣对象”的。按 COSMIC 方法的定义，数据移动所移动的是数据组，因此“兴趣对象”决定了“数据组”的数量。

例如，如果输入到一个功能处理中的数据属性是属于三个独立的兴趣对象，那么必须要识别三个输入数据移动。在分析业务应用软件的功能处理的输出时，判断数据组的数量尤其困难，可能包含如下情况：

- 多个数据组，每个数据组描述的兴趣对象各不相同，比如在一张报表中，显示了不同级别的汇总总数；
- 根据输入条件变化而变化的查询结果；
- 数据组之间各自不相关，如，一张发票包含了其他无关的广告信息。

当分析复杂输出时，例如：某报告描述了多个兴趣对象，可以把每个数据组当作是由某功能处理单独输出的。当度量复杂的报告时，按照此方法识别的每一个数据组类型也需要分别计数。

3.3.3 不适合作为数据候选对象的数据或数据组

任何出现在输入输出屏幕或报告中、与功能用户兴趣对象无关的数据，都不应识别为数据移动的对象。

COSMIC 通用软件模型假设在一个功能处理中，所有的数据运算都关联到了一种数据移动（输入、输出、读、写）。因此，在功能处理中，除了这四种数据移动类型，没有额外通过数据运算识别的数据组。

3.3.4 识别数据属性（可选）

在 COSMIC 方法中，并不强制要求识别数据属性。但是，弄清楚“数据属性”的概念对理解如何“度量变更”是必要的。某需求要求变更数据属性，对于数据属性所属的数据移动也将认为发生了变化。

另外，分析和识别数据属性有助于区分兴趣对象和数据组。

3.4 识别数据移动

本步骤包括识别每个功能处理的数据移动（输入、输出、读和写）。

规则 12 的指南：数据移动

一般来说，事物的“类型”是所有具有某些共同特征的事物的抽象类别，因此归属于同一个 FUR。（类型的同义词：种类、类别）。

事物的“实例”是指通过为其属性赋值而使其发生。在面向对象世界，也叫“实例化”——产生一个实例。

下列指南将有助于确认输入数据移动的状态：

规则 16 的指南：输入（E）

- a) 触发输入的数据组可能只包含了一个数据属性，该数据属性仅仅通知软件“事件 Y 发生了”。
- b) 时钟节拍作为触发事件，从功能用户识别了一个输入，在该例中，功能用户是时钟。
- c) 除非需要特殊的功能处理，否则从系统时钟获得日期和/或时间不可以被认为是一个输入或任何其他数据移动。

注：通常，尤其是在业务应用软件中，触发输入的数据组有多个数据属性，除了告知某事件发生了之外，还包含了关于这个事件的一些数据。

规则 17 的指南：输出（X）

- a) 对于输出是固定文档的查询功能（“固定”指的是所包含的信息没有变量），要为固定文本输出识别一个输出。
- b) 在识别输出时，忽略所有帮助人类用户理解输出数据的字段及其他表头。

规则 18 的指南：读（R）

当待度量软件的 FUR 把软件或硬件功能用户作为数据组来源时，或要从功能用户读取或存储数据组时，不应识别为读。

规则 19 的指南：写（W）

当待度量软件的 FUR 把软件或硬件功能用户作为数据组的输出对象或存储对象时，不应识别为写。

注：当 FUR 需要把数据存入持久性存储介质，或从持久性存储介质中检索数据时，度量者必须确认该数据是否在其边界范围内，即向/从“持久性存储介质”，或者数据的存储/检索是否需要待度量软件的功能用户的帮助（有时会通过其他软件块或是硬件设备检索/存储数据）。

规则 16-19 的指南-与数据移动关联的数据运算

与数据移动关联的数据运算不包括成功完成数据移动后所需的数据运算，也不包括与其他数据移动关联的数据运算。

下面的指南包含了最常见的场景（指南 a），以及其他可能的情况（指南 b 和 c）：

- 在 a) 中，数据组的实例从属于同一个 FUR：因此识别一个数据组和一个数据移动。
- 在 b) 和 c) 中，对于不同的数据组也同样适用：因此，为每一个数据组类型识别一个数据组和一个数据移动。

规则 13 和 14 的指南 - 数据移动唯一性

a) 除非 FUR 如指南 b) 或 c) 中所述，否则描述任意一个兴趣对象且需要输入到某个功能处理中的数据都需要被识别为一个数据组即一个输入。

注 1：一个功能处理当然可以有多个输入，每一个移动的数据描述了不同的兴趣对象。

注 2：该指南也适用于读、写、输出。

b) 如果 FUR 中明确了在某功能处理中，需要输入不同的数据组，且每个数据组来自不同的功能用户，但每个数据组描述的是同一个兴趣对象，则需要为每个数据组识别一个输入。

注 1：该指南也适用于向多个功能用户输出数据。

注 2：任何一个功能处理应该只有一个触发输入。

c) 如果 FUR 中明确了在一个功能处理中，必须要从持久性存储介质中读取多个数据组，每个数据组描述的是同一个兴趣对象，则需要为每个数据组识别一个读。

注 1：该指南同样适用于写。

注 2：该指南类似于规则 b)， “读” 多个描述了同一个兴趣对象的数据组时，它们很有可能来自不同的功能用户。而 “写” 多个描述了同一个兴趣对象的数据组时，它们也很可能被不同的功能用户读取。

规则 16-17 的指南：从功能用户中获取数据的功能处理

a) 当功能处理不需要告知功能用户要发送什么数据时，为每个兴趣对象识别一个输入就够了。

b) 当功能处理需要告知功能用户要发送什么数据时，在识别了输入之后还要识别一个输出。

规则 16-19 的指南：带有人机界面的应用程序中的控制命令

带有人机界面的应用程序中“控制命令”应该被忽略，因为它们不包含关于任何兴趣对象的数据移动。

错误和确认消息是输出的特殊形式，规则同样适用。

规则 16-19 的指南：错误确认消息和其他错误提示

a) 根据其 FUR，对于待度量软件的任意一个功能处理来说，所有的错误确认消息都合并为一个输出。

b) 如果发送给人类功能用户的错误确认消息，除了提示输入数据已收到或输出数据有错误之外，还输出了其他数据，那么需要再额外识别一个数据组作为输出。

- c) 所有其他的数据，由待度量软件向其硬件或软件功能用户发送/接收的数据，都应该根据其 FUR，按照常规的 COSMIC 原则，识别为输出或输入，而不考虑数据是否为错误提示。
- d) 对于报告错误提示的情况，也需要考虑是否是读或写。对于待度量功能处理，如果错误提示与持久性存储介质的读或写有关，则不需要识别为输入。
- e) 在使用待度量软件的过程中可能发生某些错误提示信息，但并不需要软件进行任何操作，在这种情况下不识别为输入或输出。比如操作系统发出的错误信息。

3.5 度量分布式软件系统的组件

当度量目的是要分别度量分布式软件系统中各构件的规模时，则要为每个构件定义单独的范围。在这种情况下，度量每个构件的功能处理遵循上文所述的一般规则。

每个度量过程（定义范围、识别功能用户和边界等）遵循以下规则：如果一个软件块由两个或多个构件构成，每个构件的度量范围之间不能有重叠。每个构件的度量范围必须定义一套完整的功能处理。例如，一个功能处理不能一部分在一个范围内，一部分在另一个范围内。同样地，即使两个构件有信息交互，在度量范围内的一个构件的功能处理也不可能有任何关于度量范围内的另一个构件的数据。

每个构件的功能用户是通过检查该构件功能处理的触发事件的位置来确定的（触发事件只可以由功能用户触发）。

3.6 软件的复用

在同一待度量的软件中，两个或多个功能处理可能有某些功能完全一样或非常相似，在需求中的多个地方分别进行了描述。这种现象被称为“功能共性”或“功能相似性”。

但是，每个功能处理的定义、建模和度量都是独立的，即不需要参考待度量软件中的其他功能处理的 FUR。

因此，如果给定功能处理的 FUR 在需求的其他地方被引用，则被引用的功能的规模将被包括在被度量的功能处理中。

3.7 度量软件变更的规模

对现有软件的功能变更，转换成 COSMIC 语言为“添加新的数据移动或对现有数据移动或数据运算的修改或删除。”因此在描述功能变更时，我们经常用到术语“增强”和“维护”。

关于软件变更，可能有如下情况：

- 新增 FUR（即对于现有功能的增加），或
- 对 FUR 的变更（可能包括增加、修改和删除），或
- 为修正一个缺陷所需的“维护”。

这些变更的度量规则是相同的，但度量者在进行性能度量和估算时要注意区分各种情况。

一个数据移动在功能方面的修改如下所述：

规则 24 的指南 - 修改数据移动

- a) 如果以下情况至少一种适用，则认为数据移动在功能上被修改了：
 - 移动的数据组被修改了。

- 关联的数据运算被修改了。

b) 如果以下情况至少一种适用，则认为数据组被修改了：

- 在数据组中添加了一个或多个新属性。
- 从数据组中删除了一个或多个属性。
- 数据组中的一个或多个属性被改变，例如意义或格式上的改变，而不是属性值改变。

c) 如果数据运算发生功能性的改变，则认为数据运算被修改了。

d) 当关联了数据移动的数据运算被修改，或是所移动的数据属性的数量或类型被修改，都属于该数据移动被修改，度量变更时都是 1CFP，与该数据移动中实际修改的数值无关。

e) 如果修改了某数据组，但移动该数据组的数据移动其功能并没有受到影响，则不识别为变更。

注：对于输入或输出屏上任何数据的改变，如与功能用户的兴趣对象无关，则不识别为变更。

f) 常规的度量习惯是，如果为了使软件与其 FUR 保持一致，而必须要对软件修改以修正缺陷，那么软件的功能规模并没有变更；而如果修改了 FUR，则软件的功能规模也将变更。

修改的数据移动对于软件块的规模并没有影响，因为它们在修改前后都存在。

当软件块被完全替代时，比如重新编写某软件块，不管是否扩展和/或省略了功能，该变更的规模都是被替换的软件规模，并按照通用规则度量新软件。

注意：通常情况下，本章中讨论的功能变更的规模和软件的功能规模变更是不同的。

4. 其他话题

4.1 扩展 COSMIC 度量方法——本地扩展

COSMIC 方法无法度量软件规模的方方面面。因此在设计 COSMIC 方法时，就没有考虑要单独度量数据运算。数据运算对规模的影响，实际已通过对软件领域的简化假设进行了考虑。

但是，就方法针对的目的和应用领域来说，COSMIC 规模度量方法仍然被认为是一个优秀的估算方法。对于希望通过一种本地化的标准来使用这个方法的组织，COSMIC 度量方法提供了本地化扩展。当使用本地化扩展时，度量结果必须按照第六章的要求来记录和汇报。

4.2 COSMIC 在敏捷中的应用

COSMIC 方法可以成功地度量敏捷软件开发中的用户故事的规模，即使其中的数据移动有时可能很少。但这种实践是行之有效的，很多报告都记录了在汇总单个用户故事后得到的敏捷迭代或冲刺中的功能点规模。并且功能规模与迭代工作量的相关性很高，该相关性比工作量与故事点个数的相关性更强。详见(<https://cosmic-sizing.org/publications/guideline-for-sizing-agile-projects-with-cosmic/>)

5. 度量阶段

规则 23 的指南：功能规模的累加

- a) 只有在 FUR 的同一功能处理颗粒度级别进行度量时，软件块的规模或者软件变更的规模才可以累加。
- b) 只有对度量目的有意义时，同一层或不同层的软件规模或软件变更的规模才可以进行累加。
- c) 一个软件块的规模是通过累加其构件（不管它是如何被分解的）的规模来获得的，并且构件间的数据移动需要被剔除。

在每一个层中，可以多层次累加功能规模。可以累加各个功能处理的规模，也可以累加软件中所有的功能处理的规模，取决于每次度量活动的目的和范围。

6. 度量报告

必须报告并记录度量结果，以确保对结果的解读清晰明确。COSMIC 度量结果将根据以下约定进行报告和存档。

规则 25 的指南：COSMIC 度量标签

COSMIC 度量结果表示为 ‘ x CFP (v)’，其中：

- ‘ x ’ 表示功能规模的数值
- ‘ v ’ 表示用于获得功能规模数值 ‘ x ’ 的 COSMIC 方法的标准版本的标识。

注：如果度量活动使用的是本地近似估算方法，而不是标准 COSMIC 方法，那么也需使用上述的标识，但应该在其他地方说明使用的是近似方法。

规则 25 的指南：COSMIC 本地化扩展的标识

使用本地化扩展的 COSMIC 度量结果表示为：

‘ x CFP (v .) + z 本地 FP’，其中：

- ‘ x ’ 表示使用 v 版本的标准 COSMIC 方法得到的汇总所有单个度量结果后得到的数值，
- ‘ v ’ 表示用于获得功能规模数值 ‘ x ’ 的 COSMIC 方法的标准版本的标识。
- ‘ z ’ 表示使用 COSMIC 方法本地化扩展得到的汇总所有单个度量结果后得到的数值。