



中华人民共和国国家标准

GB/T 42449—2023/ISO/IEC 20926:2009

系统与软件工程 功能规模测量 IFPUG 方法

Systems and software engineering—Functional size measurement—
IFPUG method

(ISO/IEC 20926:2009, Software and systems engineering—Software
measurement—IFPUG functional size measurement method 2009, IDT)

2023-03-17 发布

2023-10-01 实施

国家市场监督管理总局 发布
国家标准化管理委员会

目 次

前言 I

引言 II

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 7

5 测量过程 7

附录 A（资料性） 复杂度和功能规模的汇总表 18

附录 NA（资料性） 本文件应用案例 20

参考文献 23

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件等同采用 ISO/IEC 20926:2009《软件与系统工程 软件测量 IFPUG 功能规模测量方法 2009》。

本文件做了下列最小限度的编辑性改动：

- 为与现有标准文件协调，将标准名称改为《系统与软件工程 功能规模测量 IFPUG 方法》；
- 增加了附录 NA(资料性)本文件应用案例。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本文件起草单位：中国电子技术标准化研究院、深圳赛西信息技术有限公司、北京软件造价评估技术创新联盟、北京高质系统科技有限公司、国家应用软件产品质量检验检测中心、道普信息技术有限公司、广西达译科技有限公司、山东省计算中心(国家超级计算济南中心)、上海市软件行业协会、中国航天系统科学与工程研究院、山东山科数字经济研究院有限公司、重庆市小苹果科技有限公司、中国石油天然气股份有限公司规划总院、浙江迪捷软件科技有限公司、上海宝信软件股份有限公司、云南电网有限责任公司信息中心、中冶赛迪重庆信息技术有限公司、浙江省电子信息产品检验研究院、上海同思廷软件技术有限公司、上海计算机软件技术开发中心、北京华宇信息技术有限公司、北京中基数联科技有限公司、广东省科技基础条件平台中心、苏州洞察云信息技术有限公司、上海旋思智能科技有限公司、北京可信华泰信息技术有限公司、成都四方伟业软件股份有限公司、北京软件和信息服务交易所有限公司。

本文件主要起草人：张旻旻、李文鹏、严亮、苏伟、杨根兴、王海青、楼莉、韩庆良、周鸣乐、李刚、贺瑜亮、李玲璠、刘华林、吴迪龙、廖易宇、许宗敏、李旺、季永炜、董冠涛、康烁、张元元、毛玲燕、张艺、车江涛、王瑞、胡芸、郝琳、徐敏霞、吕雪、胡宇辉、邓日洁、冯宽、魏志伟、代寒玲、福德鹏、于英利、韩德隆、余剑、庄园、姚宝敬、韩明军、马文、鲁仁元、杨昕、龚家瑜、尹榕慧、赵明、黄琳芳、刘林、欧阳树生、杜君、韩勇、刘芬、刘永超。

引言

为有效支持 GB/T 18491《信息技术 软件测量 功能规模测量》的落地和实施,定义一种符合 GB/T 18491 的功能规模测量方法,制定本文件。

自 20 世纪 70 年代中期以来,以功能点用于测量软件功能规模的方法,已经从几个兴趣组织发展成为一个使人印象深刻的全球组织。艾伦·阿尔布雷希特是第一个公开发布软件规模的测量方法的人,该方法被称为功能点计数方法。随着功能点使用的增加,该方法的应用和使用也越来越广泛。自 1986 年成立以来,国际功能点用户组(International Function Point User Group, IFPUG)不断改进阿尔布雷希特的软件功能分级方法。本文件定义的功能规模测量方法促进了对功能规模测量的一致性解释。IFPUG 功能规模的测量方法也称为功能点计数方法,其功能规模的单位称为功能点。

组织可以应用本文件来衡量以下方面的软件产品的规模:

- 支持质量和生产率分析;
- 估算软件开发、增强和维护所需的成本和资源;
- 为软件对比提供标准化因子;
- 通过对应用程序包中包含的所有功能进行功能分级,确定所采购的应用程序包的规模;
- 通过确定特定于其需求的功能规模,来帮助用户确定应用程序包对其组织的好处。

功能点分析通过量化软件主要基于逻辑设计向用户提供的任务和服务(即功能)来衡量软件。功能点分析的目的是测量:

- 在软件中实现的用户请求和接收的功能;
- 不依赖于实现其技术测量软件开发和维护的规模。

功能点分析的过程为:

- 足够简单,以尽量减少测量过程的成本;
- 在各个项目和组织中的测量具有一致性。

系统与软件工程 功能规模测量

IFPUG 方法

1 范围

1.1 目的

本文件规定了 IFPUG 功能规模测量(FSM)方法的定义、规则和步骤。

1.2 一致性

本文件遵循 ISO/IEC 14143-1:2007 中的规定。

1.3 适用性

本文件适用于所有的功能领域。

注：IFPUG 组织持续发布白皮书,为不断演变的环境和领域提供指南。

本文件与 IFPUG 规模测量方法的先前版本完全兼容。

IFPUG 功能点分析人员已经确定了不同的交付率(交付单个功能点的时间),这些交付率与在不同功能领域中构建的应用程序有关,这些功能领域针对不同的项目规模和软件复杂度进行了校准。

1.4 用户

本文件适用于任何需要测量功能规模的用户。对于有经验的用户,本文件可提供有用的参考。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

ISO/IEC 14143-1:2007 信息技术 软件测量 功能规模测量 第1部分:概念定义 (Information technology—Software measurement—Functional size measurement—Part 1: Definition of concepts)

注：GB/T 18491.1—2001 信息技术 软件测量 功能规模测量 第1部分:概念定义 (ISO/IEC 14143-1:1998, IDT)

3 术语和定义

下列术语和定义适用于本文件。

3.1

适应性维护 adaptive maintenance

对交付后的软件产品以适应已变更或正在变更的环境而进行的修改。

注：适应性维护提供了必要的改进,以适应软件产品必须运行于其中的环境的变更。为了与不断变更的环境保持同步应做出相应的变更。例如,操作系统可能升级并且可以做某些变更,以适应新的运行系统。

[来源:ISO/IEC 14764:2007, 3.1]

3.2

应用程序 application

由一个或多个部件、模块或子系统组成,支持业务目标的自动化过程和数据的内聚合集。

示例:应付账款、应收账款、工资单、采购、车间生产、装配线控制、空中搜索雷达、目标跟踪、武器发射、航班安排和乘客预订。

3.3

应用程序功能规模 application functional size

由应用程序功能点计数确定,用功能点表示应用程序的大小。

注:通过应用程序功能规模,还可以确定支持已实现应用程序所需的工作量。

3.4

应用程序功能点计数 application function point count

应用本文件测量应用程序功能规模的活动。

3.5

安排 arranging

事务功能中对属性排序的活动。

3.6

关联实体类型 associative entity type

包含进一步描述两个其他实体类型之间的多对多关系属性的实体类型。

3.7

属性实体类型 attributive entity type

进一步描述另一个实体类型的一个或多个属性的实体类型。

3.8

基本功能部件 base functional component;BFC

功能规模测量方法中为了测量目标而定义和使用的用户功能需求的基本单位。

示例 1:功能用户需求是“维护客户”,它可由下列基本功能部件组成:“添加新客户”“报告客户购买情况”和“更改客户详细信息”。

示例 2:“客户详细信息”之类的在研的软件所维护的逻辑相关业务数据的汇集。

[来源:ISO/IEC 14143-1:2007,3.1]

3.9

边界 boundary

在研软件与其用户之间的概念上的界面。

注:ISO/IEC 20926:2003 使用了术语“应用程序边界”。

[来源:ISO/IEC 14143-1:2007,3.3]

3.10

一致状态 consistent state

过程已完全执行,功能用户需求已得到满足,无需做更多的工作的节点。

示例 1:功能用户要求是打印支票并将适当的账户标记为已付款。如果只完成了功能用户需求的一部分(例如只打印支票或只将其标记为已付款),则应用程序将不处于一致状态。打印支票而不将账户标记为已付款,会导致应用程序中的不一致,正如标记为已付款而不打印一样。

示例 2:当功能用户需求是一个接受输入文件来更新数据存储、生成生产控制报告并将错误报告返回给发送应用程序的批处理过程时,除非所有部分都完成,否则该过程不会处于一致状态。

示例 3:功能用户需求是将员工调动到新的工作,并验证其安全许可级别时,要完成此操作,将向安全应用程序发送一个实时请求(该应用程序维护政府安全许可,而不是应用程序安全),并在完成传输之前收到响应。所有步骤都需要形成一致状态。与安全应用程序的交互不是一个独立的步骤或操作。它本身不会发生,没有它,调动员工的事务也不会处

于一致状态。

3.11

控制信息 control information

通过指定要处理的数据的内容、时间或方式来影响基本过程的数据。

3.12

转换功能 conversion functionality

用于转换数据和(或)提供其他用户指定转换要求的事务或数据函数。

注：转换功能只存在于应用程序开发或增强过程中。

3.13

纠正性维护 corrective maintenance

软件产品交付后为纠正发现的问题而进行的反应性修改。

注：修改是修正软件产品，以满足要求。

[来源：ISO/IEC 14764:2007，3.2]

3.14

计数范围 counting scope

功能点计数中包含的一组功能用户需求。

3.15

数据元素类型 data element type; DET

唯一的、用户可识别的、非重复属性。

3.16

数据功能 data function

用户所见的数据的最基本形式，用于控制、记录和传输信息。

注：一般具备唯一、用户可识别、非重复属性。

3.17

派生数据 derived data

除了直接从数据函数中检索和验证信息之外，还包括其他步骤过程中所产生的数据。

3.18

开发项目 development project

开发和提供软件应用程序第一个版本的项目。

3.19

开发项目功能规模 development project functional size

用开发项目的功能点计数的方法来测量软件第一个版本提供给用户的功能。

注：开发项目功能规模可包括转换功能的规模。

3.20

开发项目功能点计数 development project functional point count

应用本文件来测量开发项目功能规模的活动。

3.21

基本过程 elementary process

对用户有意义的最小活动单元。

3.22

增强项目 enhancement project

开发和提供适应性维护的项目。

注：增强项目也可开发和提供纠正性、完善性维护，这些并不会增加增强项目的功能规模。

3.23

增强项目功能规模 enhancement project functional size

用增强项目功能点计数的方法来测量完成增强项目时的添加、更改或删除的功能。

注：增强项目的功能规模可包括转换功能的规模。

3.24

增强项目功能点计数 enhancement project functional point count

应用本文件来测量增强项目功能规模的活动。

3.25

关联实体 entity dependent

〈实体〉在没有其他实体存在的情况下,对业务本身没有意义或不重要,以致:

——实体 X 的发生与实体 Y 的发生有必要关联;

——删除一个实体 Y 的发生会导致删除所有相关的实体 X 的发生。

3.26

独立实体 entity independent

〈实体〉在没有其他实体存在的情况下,对业务本身是有意义的或重要的。

3.27

外部输入 external input; EI

处理或控制来自应用程序边界之外的数据或信息的基本过程。

注：外部输入是一种基本功能部件。

3.28

外部查询 external inquiry; EQ

向应用程序边界之外发送数据或控制信息的基本过程。

注 1：外部查询是一种基本功能部件。

注 2：应用程序在此过程中生成一个大小完全确定的输出,不需要进行加工处理。

3.29

外部接口文件 external interface file; EIF

用户可识别的一组(被测量的应用程序所引用,但是在另一个应用程序的边界内维护)逻辑相关数据或控制信息。

注：外部接口文件是一种基本功能部件。

3.30

外部输出 external output; EO

向应用程序边界之外发送数据或控制信息,包括外部查询之外的额外处理逻辑的基本过程。

注：外部输出是一种基本功能部件。

3.31

引用文件类型 file type referenced; FTR

被事务处理维护或者读取的内部逻辑文件(ILF)或者外部逻辑文件(ELF)。

3.32

功能复杂度 functional complexity

使用本文件定义的规则为一项功能进行特定复杂度评级。

3.33

功能规模 functional size

通过功能用户需求进行量化导出的软件规模。

[来源:GB/T 18491.1—2001, 3.6]

3.34

用户功能需求 functional user requirements

指定软件在任务和服务方面应该做什么的用户需求子集。

注 1：功能用户需求包括但不限于以下内容：

- 数据迁移(例如：输入客户数据、发送控制信号)；
- 数据转换(例如：计算银行利息、计算平均温度)；
- 数据存储(例如：保存客户订单、随时记录环境温度)；
- 数据检索(例如：列出当前雇员，检索飞机位置)。

注 2：非用户功能需求包括但不限于：

- 质量约束(例如：易用性、可靠性、效率和可移植性)；
- 组织约束(例如：操作位置、目标硬件、标准依从性)；
- 环境约束(例如：互操作性、信息安全性、隐私和安全)；
- 实现约束(例如：开发语言、交付计划)。

[来源：ISO/IEC 14143-1:2007，3.8]

3.35

功能点 function point; FP

本文件中界定的功能规模的测度单位。

3.36

功能点分析 function point analysis; FPA

本文件中界定的功能规模测量的方法。

3.37

功能点计数 function point count

应用本文件规则测量应用程序或项目功能规模的活动。

注：功能点计数有三种类型：应用程序、开发项目和增强项目。

3.38

功能类型 function type

本文件界定的基本功能部件的类型。

注：本文件定义了五种功能类型：外部输入、外部输出、外部查询、内部逻辑文件和外部接口文件。

3.39

内部逻辑文件 internal logical file; ILF

一组用户可辨认的在被测应用程序边界内维护的逻辑相关数据或控制信息。

注：内部逻辑文件是一种基本功能部件。

3.40

维护 maintain

通过基本过程添加、修改或删除数据。

3.41

有意义的 meaningful

用户可识别并满足功能用户需求。

3.42

完善性维护 perfective maintenance

软件产品交付后为检测并纠正软件产品中潜在故障所作的修改。

注 1：改自 ISO/IEC 14764:2007 的 3.7。

注 2：完善性维护给用户提供的增强性的程序文档改进和重编码，以改进软件性能、维护性或其他软件属性。

注 3：相对：适应性维护、纠正性维护。

3.43

主要意图 primary intent

最重要的意图。

3.44

处理逻辑 processing logic

用户为完成基本过程(如验证、算法或运算,以及读取或维护数据功能)而特别要求的需求。

3.45

计数目的 purpose of the count

进行功能点计数的原因。

注:见 5.3a)。

3.46

记录元素类型 record element type;RET

数据功能中用户可识别的数据元素类型子集。

3.47

自包含 self-contained

非事前或后续处理步骤来启动或完成功能用户需求。

示例: 功能用户需求描述需要同时添加和更新员工。员工的完整信息可能由多个部分组成,能由单独的物理屏幕、窗口或选项卡表示,如:

- 员工身份;
- 员工地址;
- 关联信息;
- 薪资信息;
- 教育情况。

添加员工时,要根据业务规则完成一个或多个选项卡。在输入所有强制性信息前,添加过程不是自包含的。

更新员工时,可在任何给定时间更新一个或多个选项卡,但它们都是满足更新员工的功能用户需求的过程步骤。在任何一个选项卡上添加、更改或删除信息不是一个单独的基本过程,而是更新员工的过程步骤。即使可以将更多的信息输入到员工记录中,所有的信息一起被认为是单个基本过程的一部分:更新员工。

添加员工和更新员工是一个自包含过程。

3.48

分类 sorting

事务功能中行或记录的排序活动。

3.49

事务功能 transactional function

提供给用户用以处理数据功能的基本过程。

注:事务功能是外部输入、外部输出和外部查询。

3.50

用户 user

在任何时刻与软件通信或交互的人或事物。

注:“事物”包括但不限于软件应用程序、动物、传感器或其他硬件。

[来源:ISO/IEC 14143-1:2007,3.11]

3.51

用户可识别 user recognizable

用户和软件开发者共同商定和理解的过程和/或数据需求。

3.52

用户视角 user view

用户描述的功能用户需求。

注：开发人员将用户视角转换为软件，以便提供解决方案。

4 缩略语

下列缩略语适用于本文件。

BFC:基本功能部件(Base Functional Component)

DET:数据元素类型(Data Element Type)

EI:外部输入(External Input)

EIF:外部接口文件(External Interface File)

EO:外部输出(External Output)

EQ:外部查询(External inQuiry)

FP:功能点(Function Point)

FPA:功能点分析(Function Point Analysis)

FTR:引用文件类型(File Type Referenced)

ILF:内部逻辑文件(Internal Logical File)

RET:记录元素类型(Record Element Type)

5 测量过程

5.1 通则

为进行功能点计数,应通过以下活动对基本功能部件(ILF、EIF、EI、EO、EQ)进行识别和分类:

- a) 依据 5.2 收集可用的文档;
- b) 依据 5.3 确定计数范围和边界,并识别功能用户要求;
- c) 依据 5.4、5.6 和 5.7 测量数据功能;

注 1: 依据 5.6 测量转换功能(如适用);依据 5.7 测量增强功能(如适用)。

- d) 依据 5.5、5.6 和 5.7 测量事务功能;

注 2: 依据 5.6 测量转换功能(如适用);依据 5.7 测量增强功能(如适用)。

- e) 依据 5.8 计算功能规模;
- f) 依据 5.9 记录功能点计数;
- g) 依据 5.10 报告功能点计数的结果。

注 3: 图 1 为功能点计数过程的图形化概述。

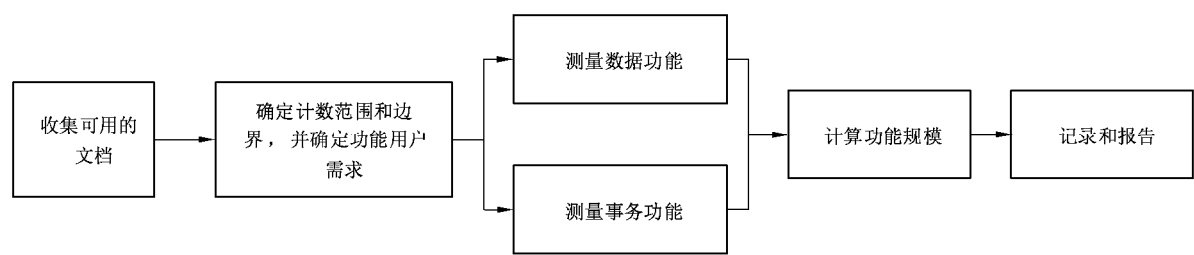


图 1 功能点计数过程

5.2 收集可用的文档

支持功能点计数的文档应描述软件所交付的功能,或被测量的软件项目所影响的功能。
应获取足够的文档来进行功能点计数或与主题专家联系,这些专家能够提供额外的信息来解决文件中的任何空白。

注: 合适的文档可包括需求、数据/对象模型、类图、数据流程图、用例、过程描述、报告框架、屏幕布局、用户手册和其他软件开发文件。

5.3 确定计数范围、边界并确定功能用户需求

为了确定每个应用程序的计数范围、边界,并确定功能用户需求,应进行以下活动。

- a) 识别计数目的。
 - 注 1: 功能点计数是为业务问题提供解决方案,同时它也是一个有明确目的的业务问题。
 - 注 2: 计数目的决定了计数范围。
 - 示例 1: 计数的目的可能是确定特定软件版本的规模。
 - 示例 2: 计数的目的可能是确定某个应用程序的规模,作为组织确定其软件组合规模的一部分。
- b) 基于目的识别计数类型,包括:
 - 1) 开发项目功能点计数;
 - 2) 应用程序功能点计数;
 - 3) 增强项目功能点计数。
- c) 基于目的和计数类型确定计数范围。
- d) 基于用户视角而非技术视角确定计数范围内每个应用程序的边界。
 - 示例 3: 如果计数目的是估算人力资源和福利应用程序的增强成本,相关活动如下:
 - 计数类型是增强项目计数;
 - 计数范围包括人力资源和福利应用新增、更改或删除的事务功能和数据功能,以及每个应用程序的任意转换需求;
 - 从用户视角来看,人力资源和福利应用是不同的功能区域,因此它们是不同的应用;
 - 在人力资源和福利应用之间存在边界,每个应用及其用户之间也存在边界。
- e) 用户需求可以同时包括功能需求和非功能需求,识别出其中的功能需求,并排除非功能需求。

5.4 测量数据功能

5.4.1 通则

数据功能满足功能用户需求中数据存储和/或引用的要求。在计数范围内的所有数据功能均应被评估,以识别出每个逻辑数据组。

为了测量数据功能,应执行如下活动:

- a) 根据 5.4.2 识别所有的逻辑数据并按数据功能分组；
- b) 根据 5.4.3 将数据功能分类成内部逻辑文件或者外部接口文件；
- c) 根据 5.4.4 统计每个数据功能的数据元素类型；
- d) 根据 5.4.5 统计每个数据功能的记录元素类型；
- e) 根据 5.4.6 测定每个数据功能的功能复杂度；
- f) 根据 5.4.7 测定每组数据功能的功能规模。

5.4.2 识别全部逻辑数据并按照数据功能分组

注 1：采用逻辑数据模型来识别数据功能是最容易的一种方式。当然除了这种方式外，我们也不能排除在（特定）环境中采用其他测量方式，比如说数据或对象建模技术。数据建模技术是一种识别数据功能规则并将其文档化的工具，当然还有一些其他工具也可以达到相同的目的。

为了识别数据功能，应执行以下活动。

- a) 在计数范围内，找出所有逻辑相关且用户可识别的数据或控制信息。
- b) 排除没有被任何应用程序维护的实体。
- c) 将有实体依赖关系的相关实体整合。

注 2：能够独立存在的实体被视为独立的数据逻辑组。

- d) 排除以下被称为代码数据的实体：
 - 1) 包含代码、解释性名称或描述的可替代数据实体；
 - 2) 包含一个或多个很少改变的属性（如果有的话）的单一出现实体；
 - 3) 包含的数据基本上都是静态或者很少修改的实体；
 - 4) 实体的属性全部都是默认值；
 - 5) 包含可供选择或验证可用值的有效值实体；
 - 6) 包含了一组用于验证的范围数值的实体。

注 3：使用了以上编码数据的实体，可能也包含了一些用于审计以及定义生效日期的数据属性。但是，这些属性的存在，并不会改变实体本身的性质。

- e) 排除不包含用户所需属性的实体。
- f) 删除包含用户不需要的附加属性的关联实体，和只包含外键的关联实体；将外键属性与主实体分组。

注 4：外键属性是用户与其他数据函数建立关系所需的数据。

5.4.3 将每个数据功能识别为内部逻辑文件或外部接口文件

一个数据功能应按如下分类：

- a) 如果它被应用程序使用了同时也被维护了，就可以认为是 ILF；
- b) 如果被测量的应用程序只是引用了它，并没有维护，且它被一个或多个其他应用程序识别为内部逻辑文件，就可以认为它是 EIF。

5.4.4 计数每个数据功能的数据元素类型

为了统计数据功能的数据元素类型（DETs），应执行以下活动：

- a) 在计数范围内，通过执行所有基本过程，将每个独特的、由用户识别的、不重复的、在数据功能中维护或检索的属性计数为一个数据元素类型；

注 1：例如，在一个内部逻辑文件或外部接口文件中，将 12 个重复的月度预算金额字段计数为一个 DET。计数一个额外的数据元素类型以确定适用的月份。

- b) 当两个或多个应用程序维护或引用相同的数据功能时,只计数那些被正在被测量的应用程序使用的数据元素类型;
注 2: 被测量的应用程序未引用到的属性不被计数。
- c) 为每一个用户所需的与其他数据功能建立关联关系的属性,计数一个数据元素类型;
- d) 通过查看相关属性,来确定它们是否被归为一类并计做一个数据元素类型,还是被计数成多个数据元素类型。归类将取决于基本过程如何使用应用程序内的属性。

示例: 属性(名、中间名、姓)按如下归类方式进行计数:

- 如果这些属性经常被结合使用,则识别为数据元素类型姓名(名、中间名、姓);
- 如果有时姓氏单独使用,则识别为数据元素类型名(名和中间名)和姓;
- 如果这三者可以单独使用,则识别为数据元素类型名、中间名、姓。

5.4.5 为每个数据功能统计记录元素类型

为数据功能计数记录元素类型(RETs),应执行如下活动:

- a) 为每个数据功能计数一个记录元素类型(例如,默认情况下,每个数据功能都有一个子组的数据元素类型被统计成一个记录元素类型);
- b) 为以下每个包含多个数据元素类型的附加逻辑子组(在数据功能内)计数一个额外的 RET:
 - 1) 具有非关键属性的关联实体;
 - 2) 子类型(第一个子类型除外);
 - 3) 在非强制性 1-1 的关系中的属性实体。

注 1: 强制性 1-1 关系反映两个实体之间的关系,其中每个实体都与相关实体的一个且仅一个实例相关。

注 2: 如果没有数据模型,请查找重复的数据组,以便标识记录元素类型。

示例 1: 发票具有客户信息的标题部分和采购信息(例如物料编号、说明、价格、重量)行记录。标题部分计为一个 RET。而行记录是一个重复组,计数为一个额外的记录元素类型。

示例 2: 单个重复属性(如客户的多个账号)不算作重复组,它被计为单个数据元素类型而非记录元素类型。

示例 3: 一个同时发生的多组数据元素,如年、月和预算金额可计为重复组,它被计为三个数据元素类型而不是一个记录元素类型。

5.4.6 为每个数据功能确定功能复杂度

表 1 说明使用记录元素类型以及数据元素类型的个数确定每个数据功能的复杂度。

表 1 数据功能复杂度

RETs	DETs		
	1~19	20~50	>50
1	低	低	中
2~5	低	中	高
>5	中	高	高

5.4.7 为每个数据功能确定功能规模

表 2 说明使用类型和功能复杂度来确定每个数据功能的功能规模。

表 2 数据功能规模

功能复杂度	数据功能类型	
	ILF	EIF
低	7	5
中	10	7
高	15	10

5.5 测量事务功能

5.5.1 通则

事务功能满足处理数据的功能用户需求。应对计数范围内的所有事务功能进行评价,以识别每个唯一的基本过程。

为了测量事务功能,应执行以下活动。

- 遵循 5.5.2 识别用户需要的每个基本过程;
- 遵循 5.5.3 将每个事务功能划分为外部输入(EI)、外部输出(EO)或外部查询(EQ);
- 遵循 5.5.4 对每个事务功能所引用的文件类型(FTRs)进行计数;
- 遵循 5.5.5 对事务功能的数据元素类型(DETs)进行计数;
- 遵循 5.5.6 确定每个事务功能的功能复杂度;
- 遵循 5.5.7 确定每个事务功能的功能规模。

5.5.2 识别基本过程

5.5.2.1 为了识别每个基本过程,应执行以下活动。

- 将用户功能用户需求组合和/或分解为最小的活动单元,活动单元满足以下所有条件:
 - 对用户是有意义的;
 - 构成完整事务;

注:对于本文件的用户来说,第 2)项不是一个变更,而是为了专门增强理解一致性的改进。

- 是自包含的;
- 使被计数的应用程序的业务处于一致状态。

示例 1: 单个功能用户需求可表述为提供维护员工信息的一个功能。该需求被分解为更小的工作单元,如添加员工、更改员工、删除员工、查询员工。

示例 2: 单个需求可能需要添加不同类型的员工信息(如地址、工资及相关信息),但对用户有意义的最小活动单元是添加员工。

- 识别满足 a)中所有准则的每个活动单元的基本流程。

5.5.2.2 为识别特定基本过程,应执行以下活动。

- 与已识别的基本过程相比,如果满足下列条件,将两个类似的基本过程计数为同一个基本过程:
 - 需要相同的 DETs 集合,
 - 需要相同的 FTRs 集合,
 - 需要相同的处理逻辑来完成基本过程[参见 b)的相关说明]。

注 1: 测量事务功能的活动是按顺序描述的;实际上却是迭代的。FTR 和 DET 是根据 5.5.4 和 5.5.5 进行识别的,但比较两个相似的基本过程是必要的;

注 2：一个基本过程可能包括不同的 DET 或 FTR 以及处理逻辑的多种选择、变化或事件。

示例：当确定了添加员工的基本过程时，不会分解成两个基本过程来说明员工是否有从属关系这一事实。基本过程依然是添加员工，为了说明从属关系，处理逻辑和数据元素类型会有一些差异。

b) 不要将具有多种处理逻辑形式的基本过程拆分为多个基本过程。

注 3：接受并验证来自用户的数据，读取和过滤来自 ILF 的数据，并排序后将结果返回给用户，这个基本过程不能被拆分为多个基本过程。

5.5.2.3 完成基本过程的各种形式的处理逻辑如表 3 所示。

表 3 处理逻辑的形式

1) 执行验证 示例 1：向组织添加新员工时，该过程将验证员工类型(DET)。
2) 执行数学公式并计算 示例 2：当汇总一个组织内的所有员工信息时，该过程包括计算带薪员工、钟点员工和所有员工的总数。
3) 等价转换 示例 3：使用表格将员工年龄转换为年龄范围组。使用指定条件筛选和选择用于比较多个数据集的数据。
4) 使用指定条件筛选和选择用于比较多个数据集的数据 示例 4：要根据员工的工作分配生成员工列表，基本过程会比较职务编号以选择并列出具具有该任务的适当员工。
5) 分析条件以确定哪些条件适用 示例 5：在添加员工时，基本流程执行的处理逻辑将取决于员工的工资是按月发放的工资还是根据工作时间计算的。在本例中根据不同选择(工资或小时)DETs 的输入(以及由此产生的处理逻辑)是基本过程的一部分。
6) 更新一个或多个 ILF 示例 6：添加员工时，基本过程会更新员工 ILF 来维护员工数据。
7) 引用一个或多个 ILF 或 EIF 示例 7：添加员工时，将引用货币 EIF 以获得正确的美元汇率，从而确定员工的小时工资。
8) 检索数据或控制信息 示例 8：为了查看员工列表，从数据功能中检索员工信息。
9) 通过转换现有数据以创建其他数据的方式生成衍生数据 示例 9：为了确定(导出)患者的注册号(例如，SMIJO01)，需要串联以下数据： ——患者姓氏的前三个字母(如 Smith 的 SMI)； ——患者名字的前两个字母(如 John 的 JO)； ——唯一的两位数字序列号(从 01 开始)。
10) 改变应用程序的行为 示例 10：当支付员工工资从每月 15 日和最后一天支付变为每隔一个星期五支付时，支付员工工资的基本过程的行为也被改变；其结果是每年 24 个发薪期变为 26 个发薪期。
11) 生成并呈现边界外的信息 示例 11：格式化员工列表并向用户展示。
12) 有能力接收进入应用程序边界的数据或控制信息 示例 12：用户输入信息以向应用程序添加客户订单。
13) 排序或排列一组数据。这种处理逻辑的方式不影响数据的类型的识别，也不影响基本过程的唯一性；即数据的展示位置不构成唯一性 示例 13：员工列表按字母顺序或位置顺序排列。 示例 14：在订单输入屏幕界面上，订单标题信息位于屏幕顶部，订单详细信息位于下方。 注：ISO/IEC 20926:2003 错误地表述了“重新排序”和“重新排列”这两个术语；使用“排序”和“排列”这两个术语是一种更正，而不是一种变更。

5.5.3 将每个基本过程分类为事务功能

5.5.3.1 对于每个基本过程：

- a) 主要目的应被识别为以下类别之一：
 - 1) 改变应用程序的行为；
 - 2) 维护一个或多个 ILF；
 - 3) 向用户呈现信息。
- b) 应从 5.5.2.3 的列表中确定完成基本过程所需的处理逻辑方式。

5.5.3.2 每个基本过程应被分类为：

- a) 一个 EI, 假如它：
 - 1) 包括了为接受进入应用程序边界的数据或控制信息所进行的处理逻辑；
 - 2) 具有以下两个类别之一的主要目的：
 - 维护一个或多个 ILF, 或者
 - 应用程序的行为。
- b) 一个 EO, 假如它具有向用户呈现信息的主要目的, 且包括至少一项以下处理逻辑方式：
 - 1) 执行了数学计算；
 - 2) 更新了一个或多个 ILF；
 - 3) 生成了衍生数据；
 - 4) 改变了应用程序的行为。
- c) 一个 EQ, 假如它具有向用户呈现信息的主要目的, 并且它：
 - 1) 引用一个数据功能以检索数据或者控制信息；
 - 2) 不满足被分类为一个 EO 的条件。

注 1：表 4 总结了主要目的和事务功能类型之间的关系。

表 4 主要目的和事务功能类型之间的关系

功能	事务功能类型		
	EI	EO	EQ
改变应用程序的行为	PI	F	N/A
维护一个或多个 ILF	PI	F	N/A
向用户呈现信息	F	PI	PI
其中： PI ——事务功能类型的主要目的； F ——事务功能类型的一个有时会呈现但不是主要目的的功能； N/A ——事务功能类型不允许执行此功能类型。			

注 2：表 5 总结了处理逻辑方式和事务功能类型之间的关系。

表 5 处理逻辑和事务功能类型之间的关系

处理逻辑方式	事务功能类型		
	EI	EO	EQ
1. 执行验证	c	c	c
2. 执行数学计算	c	m*	n
3. 等价转换	c	c	c
4. 使用指定条件筛选和选择用于比较多个数据集的数据	c	c	c
5. 分析条件以确定哪些条件适用	c	c	c
6. 更新一个或多个 ILF	m*	m*	n
7. 引用一个或多个 ILF 或 EIF	c	c	m
8. 检索数据或控制信息	c	c	m
9. 产生了衍生数据	c	m*	n
10. 改变应用程序的行为	m*	m*	n
11. 生成信息并呈现到边界外	c	m	m
12. 接受进入应用程序边界内的数据或控制信息	m	c	c
13. 排序或排列一组数据	c	c	c
其中： m —— 事务功能类型处理逻辑方式是强制性的； m* —— 事务功能类型执行至少一个处理逻辑方式(m*)是强制性的； c —— 事务功能类型能执行该处理逻辑方式,但不是强制性的； n —— 事务功能类型不能执行该处理逻辑方式。			

5.5.4 计数每个事务功能的 FTR

对于每个事务功能,通过事务功能访问(读取和/或写入)的每个唯一数据功能应计数为一个 FTR。

注：测量事务功能的活动是按顺序描述的,但是它们实际上是迭代的。为了比较两个相似的基本过程,如 5.5.2.2 所述,需要识别 FTR。

5.5.5 计数每个事务功能的 DET

要计数每个事务功能的 DET,应执行以下活动。

a) 审查所有通过(输入和/或退出)边界的操作。

b) 对于在事务功能处理过程中通过(进入和/或退出)边界的每个唯一用户可识别、不重复的属性计数为一个 DET。

示例 1：通过边界的 DET 包括：

—— 用户通过屏幕输入以及在报告或屏幕上显示的属性，

—— 进入应用程序边界的属性,这些属性要求指定数据检索或生成数据的时间、内容和/或方式，

—— 由事务功能的用户提供或呈现的属性，

—— 进入或退出边界的电子文件中的属性。

c) 每个事务功能发送应用程序响应消息,即使有多个消息也只能计数一个 DET。

示例 2：如果向用户显示多条错误/确认消息,则只计数一个数据元素类型。

d) 每个事务功能启动操作,即使有多种方法也只能计数一个 DET。

示例 3: 如果用户能通过单击“确定”按钮或按功能键来触发生成报告,但只计数为一个数据元素类型。

- e) 下列条目不能计数为 DET:
- 文字,如报告标题、屏幕或面板标识符、列标题和属性标题;
 - 应用程序生成的戳,如日期和时间属性;
 - 分页变量、页码和定位信息,如“211 行中的第 37 行到第 54 行”;
 - 导航助手,例如使用“上一个”“下一个”“第一个”“最后一个”和对应的图形在列表中导航;
 - 通过事物功能在边界内产生并且保存到内部逻辑文件而没有离开边界的属性;
 - 从内部逻辑文件或外部接口文件中检索或引用的,参与没有退出边界的处理过程的属性。

注: 测量事务功能的活动是按顺序描述的;但是,它们实际上是迭代的。为了比较两个相似的基本过程,如 5.5.2.2 所述,需要确定 DET。

5.5.6 确定每个事务功能的功能复杂度

遵循表 6 或表 7,应通过 FTRs 和 DETs 的数量来确定每个事务功能的功能复杂度。

表 6 EI 功能复杂度

FTRs	DETs		
	1~4	5~15	>15
0~1	低	低	中
2	低	中	高
>2	中	高	高

表 7 EO 和 EQ 功能复杂度

FTRs	DETs		
	1~5	6~19	>19
0~1	低	低	中
2~3	低	中	高
>3	中	高	高
注: 一个 EQ 至少有一个 FTR。			

5.5.7 确定每个事务功能的功能规模

应通过表 8 所示的事务功能类型和复杂度来确定每个事务功能的功能规模。

表 8 事务功能规模

功能复杂度	事务功能类型		
	EI	EO	EQ
低	3	4	3
中	4	5	4
高	6	7	6

注：为方便读者使用，本文件在附录 A 中给出了表 1、表 2、表 6、表 7 和表 8 的合集。

5.6 测量转换功能

开发或增强项目的功能计数范围也可以包括项目所需的转换功能的功能规模。所需转换的事务功能和数据功能(未计入)按照 5.4 和 5.5 进行计算。

注：计数范围规定了是否对转换功能进行计数。

5.7 测量改进功能

增强项目可以包括对现有功能的添加、更改和删除。改进功能应按照规定进行测量：

- a) 不要修改已经建立的增强项目应用程序的边界；
- b) 按照 5.4 对添加、更改或删除的数据功能进行计数；
- c) 根据 5.5 对添加、更改或删除的事务功能进行计数；
- d) 应用程序功能规模可能会更新，以反映：
 - 1) 增加功能，增加应用程序功能规模；
 - 2) 变更功能，可能增加、减少或对应用程序功能规模没有影响；
 - 3) 删除功能，减少了应用程序功能规模。

注 1：对数据功能的更改可涉及添加、更改或删除 DETs 和/或 RETs；

注 2：对事务功能的更改可涉及添加、更改或删除 DETs 和/或 FTRs 和/或更改处理逻辑。

5.8 计算功能规模

选择和使用适当的公式计算功能规模时，应考虑目的和计数范围。

开发型项目功能规模应使用公式(1)计算：

$$DFP = ADD + CFP \quad \dots\dots\dots (1)$$

式中：

DFP ——开发型项目功能点计数；

ADD ——开发型项目交付用户的功能规模；

CFP ——转换功能的规模。

开发型项目完成后或者在应用程序生存周期内任何时间测量的应用程序功能规模应使用公式(2)计算：

$$AFP = ADD \quad \dots\dots\dots (2)$$

式中：

AFP ——应用程序功能点数；

ADD ——开发型项目交付用户的功能规模(不包含转变功能的规模)，或者在任意时间对应用程序计数时都存在的功能规模。

增强项目功能规模应使用公式(3)计算：

$$EFP = ADD + CHGA + CFP + DEL \quad \dots\dots\dots (3)$$

式中：

EFP ——增强项目功能点计数；

ADD ——增强项目新增的功能规模；

CHGA ——增强项目实施后或者将要实施后变更的功能规模；

CFP ——转变功能的规模；

DEL ——增强项目删除的功能规模。

增强项目完成后的应用程序功能规模应使用公式(4)计算：

$$\text{AFPA} = (\text{AFPB} + \text{ADD} + \text{CHGA}) - (\text{CHGB} + \text{DEL}) \quad \dots\dots\dots (4)$$

式中：

AFPA ——增强项目完成后的应用程序功能点数；

AFPB ——增强项目完成前的应用程序功能点数；

ADD ——增强项目新增的功能规模；

CHGA ——增强项目实施后或者将要实施后变更的功能规模；

CHGB ——增强项目实施前或者曾经实施前变更的功能规模；

DEL ——增强项目删除的功能规模。

注：依照本文件测量过程的应用案例见附录 NA。

5.9 记录功能点计数

功能点的计数要记录如下内容：

- a) 计数的目的和类型；
- b) 计数的范围和系统边界；
- c) 计数的日期；
- d) 所有数据功能和事务功能的列表,包括它们的类型、复杂度和被分配的功能点数量；
- e) 计数结果(见 5.10)；
- f) 所有做出的假设和解决的问题。

功能点计数的文件还能包括如下内容：

- a) 确定计数所依据的原始文件；
- b) 确定参与者的角色和资格；
- c) 针对每一个数据功能,DETs 和 RETs 的数量；
- d) 针对每一个事务功能,DETs 和 FTRs 的数量；
- e) 所有数据功能与事务功能的交叉引用；
- f) 所有数据功能与源文件中抽象概念的交叉引用；
- g) 所有事务功能与源文件中抽象概念的交叉引用。

注 1：和客户协商文件的级别,并告知客户相关的成本和收益。

注 2：一个完整的功能点计数文档有助于可追溯性、可用性以及可维护性；但是相较于过程文档客户可能更关注结果。

5.10 功能点计数的结果报告

注：一致的功能点计数结果报告,使读者能够识别它们所遵循的标准。

5.10.1 遵循本文件的结果应报告如下：

$$\text{S FP (IFPUG-GB/T)}$$

其中：

S ——功能点计数的结果；

FP ——IFPUG 功能规模测量方法的大小单位；

GB/T ——国家标准(GB/T 42449—2023)。

示例：250 FP (IFPUG-GB/T 42449—2023)

5.10.2 遵循本文件的本地定制结果应报告如下：

$$\text{S FP (IFPUG-GB/T-c)}$$

其中：

c——一个或多个字符,表明结果不完全遵循本文件。

示例：250 FP (IFPUG-GB/T 42449—2023-a)

附 录 A
(资料性)

复杂度和功能规模的汇总表

表 A.1～表 A.5 中给出了本文件中所有的功能复杂度和功能规模表。

表 A.1 数据功能复杂度

RETs	DETs		
	1～19	20～50	>50
1	低	低	中
2～5	低	中	高
>5	中	高	高

表 A.2 数据功能规模

功能复杂度	数据功能类型	
	ILF	EIF
低	7	5
中	10	7
高	15	10

表 A.3 EI 功能复杂度

FTRs	DETs		
	1～4	5～15	>15
0～1	低	低	中
2	低	中	高
>2	中	高	高

表 A.4 EO 和 EQ 功能复杂度

FTRs	DETs		
	1～5	6～19	>19
0～1	低	低	中
2～3	低	中	高
>3	中	高	高
注：一个 EQ 至少有一个 FTR。			

表 A.5 事务功能规模

功能复杂度	事务功能类型		
	EI	EO	EQ
低	3	4	3
中	4	5	4
高	6	7	6

附 录 NA
(资料性)
本文件应用案例

NA.1 收集可用的文档

银行柜员在柜员终端系统为客户办理贷记卡开户时,需要增加支持开立“借贷一体卡”的功能,“借贷一体卡”的功能用户需求描述如下。

- a) 新增功能:银行柜员终端系统支持开立“借贷一体卡”。
- b) 使用用户:银行柜员。
- c) 业务处理流程包括。
 - 1) 修改银行柜员终端系统的“贷记卡”开户功能,页面增加“借贷一体卡”标志(1——普通卡;2——借贷一体卡),默认值为“1——普通卡”,允许银行柜员修改此字段为必输。该字段由银行柜员终端系统中的“个人卡申请信息登记表”进行维护。
 - 2) 如果银行网点柜员选择“2——借贷一体卡”,调用卡片合法性校验公共模块进行如下校验:
 - 如果所开卡的卡种不是贷记卡,则报错;
 - 如果不是个人卡,则报错;
 - 如果不是银联卡,则报错。
 - 3) 开卡交易成功后,打印凭条信息时,增加返回如下卡标志信息:
 - 主卡的借贷一体卡标志(1——普通卡;2——借贷一体卡);
 - 副卡的借贷一体卡标志(1——普通卡;2——借贷一体卡);
 - 附属卡的借贷一体卡标志(1——普通卡;2——借贷一体卡)。

NA.2 确定计数范围、边界并确定功能用户需求

NA.2.1 识别计数目的

计数的目的是测量银行柜员终端系统增加支持开立“借贷一体卡”功能的软件规模。

NA.2.2 识别计数类型

项目是在存量的“贷记卡开户”功能中,增加“借贷一体卡”标识,因此属于“增强项目功能点计数”。

NA.2.3 识别计数范围

计数范围包括在银行柜员终端开立贷记卡页面增加“借贷一体卡”标识,对卡号进行合法性校验,打印凭证增加显示“借贷一体卡”卡标识。

NA.2.4 识别系统边界

系统边界为银行柜员终端系统。

NA.3 分析数据功能

NA.3.1 识别全部逻辑数据并按照数据功能分组

该功能涉及修改银行柜员终端系统中存量表:“个人卡申请信息登记表”,该表记录的数据可以被用

户所识别,并且具有逻辑相关性,因此可以被识别为逻辑数据文件。

NA.3.2 将每个数据功能识别为内部逻辑文件或外部接口文件

“个人卡申请信息登记表”被柜员终端系统的“开立贷记卡”功能所调用并被维护,增加了“借贷一体卡”字段,因此该表被识别为 1 个内部逻辑文件(ILF)。

NA.3.3 计数每个数据功能的数据元素类型

在“个人卡申请信息登记表”中,为主卡、副卡、附属卡三种卡片类型各增加“借贷一体卡”字段,具有独特的、用户可以识别的、不重复的数据属性,共计为 3 个数据元素类型(DET)。

NA.3.4 计数每个数据功能的记录元素类型

为每个数据功能计数一个记录元素类型,本案例中计数为 1 个记录元素类型(RET)。

NA.3.5 确定每个数据功能的功能复杂度

根据表 1“数据功能复杂度”矩阵,使用记录元素类型以及数据元素类型的个数确定所识别数据功能的复杂度,识别本案例中内部逻辑文件(ILF)的复杂度为“低”。

NA.3.6 确定每个数据功能的功能规模

根据表 2“数据功能规模”矩阵,使用类型和功能复杂度来确定所识别数据功能的功能规模,识别本案例中内部逻辑文件(ILF)的功能规模为 7 FP。

NA.4 分析事务功能

NA.4.1 识别基本过程

该功能涉及银行柜员在终端系统开立“借贷一体卡”和打印凭证时增加显示“借贷一体卡”标识,是对用户有意义的两个独立完整事务过程,因此被识别为两个基本过程。

NA.4.2 将每个基本过程分类为事务功能

根据事务功能类型的定义,开立“借贷一体卡”的主要目的是维护内部逻辑文件 ILF“个人卡申请信息登记表”,增加了“借贷一体卡”字段,因此被识别为内部输入(EI)。

打印凭证时增加显示“借贷一体卡”标识,主要目的是向用户呈现信息,没有执行数学计算,没有更新数据文件,没有生成衍生数据,没有改变应用程序行为,因此被识别为外部查询(EQ)。

NA.4.3 计数每个事务功能的 FTR

对于每个事务功能,通过事务功能访问(读取和/或写入)的每个唯一数据功能应计数为一个 FTR,本案例中每个事务功能各计数为 1 个记录文件类型(FTR)。

NA.4.4 计数每个事务功能的 DET

审查所有通过(输入和/或退出)边界的操作,对于在事务功能处理过程中通过(进入和/或退出)边界的每个唯一用户可识别、不重复的属性计数为一个 DET,本案例中每个事务功能各计数为 1 个记录元素类型(DET)。

NA.4.5 确定每个事务功能的功能复杂度

根据表 6 和表 7 矩阵,使用 FTR 和 DET 的数量确定每个事务功能的功能复杂度。

识别本案例中内部输入(EI)的复杂度为“低”。

识别本案例中外出查询(EQ)的复杂度为“低”。

NA.4.6 确定每个事务功能的功能规模

根据表 5“事务功能规模”矩阵,使用类型和功能复杂度来确定所识别事务功能的功能规模。识别本案例中内部输入(EI)的功能规模为 3 FP,本案例中外出查询(EQ)的功能规模为 3 FP。

NA.5 计算功能规模

本案例属于开发型项目,根据公式(1)计算得到本案例调整前的功能点规模。

$$\begin{aligned} \text{DFP} &= \text{ADD} + \text{CFP} \\ &= (\sum \text{EI} + \sum \text{EO} + \sum \text{EQ} + \sum \text{ELF} + \sum \text{ILF}) + \text{CFP} \\ &= (3 + 0 + 3 + 0 + 7) + 0 \\ &= 13 \text{ FP} \end{aligned}$$

参 考 文 献

- [1] ISO/IEC 14764:2007 Software engineering—Software life cycle processes—Maintenance
-