

IFPUG 功能点估算方法使用指南

排版整理：旦丰

终稿

目录

1. 引言	1
1.1. 目的	1
1.2. 预期读者	1
1.3. 术语表	1
1.4. 参考资料	2
2. 功能点分析法概论	2
2.1. 功能点分析方法的目标	2
2.2. 功能点方法的收益	2
2.3. 功能点分析法的步骤	2
2.3.1. 决定分析的类型	2
2.3.2. 识别分析范围和应用边界	3
2.3.3. 确定未经调整的功能点数（Unadjusted Function Point Count -- UFPC）	3
3. 分析流程	5
3.1. 决定分析的类型	5
3.1.1. 定义：功能点分析的类型	5
3.2. 识别分析范围和应用边界	6
3.2.1. 识别分析范围和应用边界中的定义	6
3.2.2. 定义应用边界	7
3.3. 分析范围以及应用边界的规则和流程	7
3.3.1. 边界识别的规则	8
3.3.2. 分析范围和应用边界流程：	8
3.3.3. 边界识别的一些技巧：	8
3.4. 计数数据功能	8
3.4.1. 定义	9
3.4.2. 计数流程概述	9
3.4.3. ILF 识别规则	10

3.4.4. EIF 识别规则	10
3.4.5. 复杂度和贡献的定义和规则	10
3.4.6. ILF/EIF 计数流程	11
3.4.7. 复杂度和贡献确定流程	12
3.4.8. 数据功能计数技巧	13
3.5. 计数交易功能	14
3.5.1. 定义	14
3.5.2. EI,EO,EQ 计数规则	18
3.5.3. 复杂度和贡献的定义和规则	20
3.5.4. EI,EO,EQ 的计数流程	22
3.5.5. 复杂度和贡献确定流程	23
3.5.6. 交易功能计数技巧	25
3.6. 决定调整系数	27
3.6.1. 调整系数的决定	27
3.6.2. 确定 VAF 的流程	27
3.6.3. 通用系统特性及其影响程度的评定	28
3.7. 计算调整功能点	37
3.7.1. 开发项目功能点的计算	37
3.7.2. 升级项目功能点的计算	38
3.7.3. 应用功能点的计算	39

1. 引言

1.1. 目的

本文档描述功能点估算的方法。

1.2. 预期读者

1. 想使用功能点方法进行规模估算的及想了解这种估算方法的人员；

1.3. 术语表

- 功能点方法 – 功能点方法是度量规模的一种，它是主要从逻辑设计的角度出发对提供给客户的功能进行量化的方法
- 数据功能：ILF 和 EIF 的统称
- 交易功能 – EI,EO 和 EQ 的统称
- ILF – Internal Logical File – 内部逻辑文件
- EIF – External Interface File – 外部接口文件
- EI – External Input – 外部输入
- EO – External Output – 外部输出
- EQ – External Query – 外部查询
- VAF – Value Adjustment Factor – 调整系数
- DET – Data Element Types – 数据元素类型
- RET – Record Element Types – 记录元素类型
- FTR – File Type Referenced – 引用文件类型
- UFPC – Unadjusted Function Point Count – 未经调整的功能点数
- DI – Degree of Influence – 影响程度
- TDI – Total Degree of Influence – 整体影响程度

1.4. 参考资料

无

2. 功能点分析法概论

2.1. 功能点分析方法的目标

功能点方法度量的是软件的规模，它是主要从逻辑设计的角度出发对提供给客户的功能进行量化的方法。功能点分析方法的目标是：

- ◆ 度量用户要求和能够接收到的功能；
- ◆ 提供一种与具体实施方法和技术无关的对软件开发和维护进行度量的手段。

除了满足上述的目标之外，功能点分析方法还是：

- ◆ 一种相对来说比较简单的对规模进行度量的手段；
- ◆ 在不同的项目和组织之间能够保持一致的度量方法。

2.2. 功能点方法的收益

功能点方法在组织中的应用包括：

- ◆ 用来从功能角度度量一个采购软件的规模；
- ◆ 帮助用户从提供的功能角度判断一个软件对他们的好处；
- ◆ 为一个组织判断自己的质量和生产率提供“分母”；
- ◆ 帮助软件开发组织从规模出发判断一个软件项目的日程、人力和成本；
- ◆ 提供对软件进行横向比较的基本判断依据。

2.3. 功能点分析法的步骤

功能点分析方法的主要步骤包括：

2.3.1. 决定分析的类型

功能点分析的第一个步骤是确定分析的类型。功能点分析的方法既可以应用在项目上，也可以应用在应用上。功能点分析有如下三种类型：

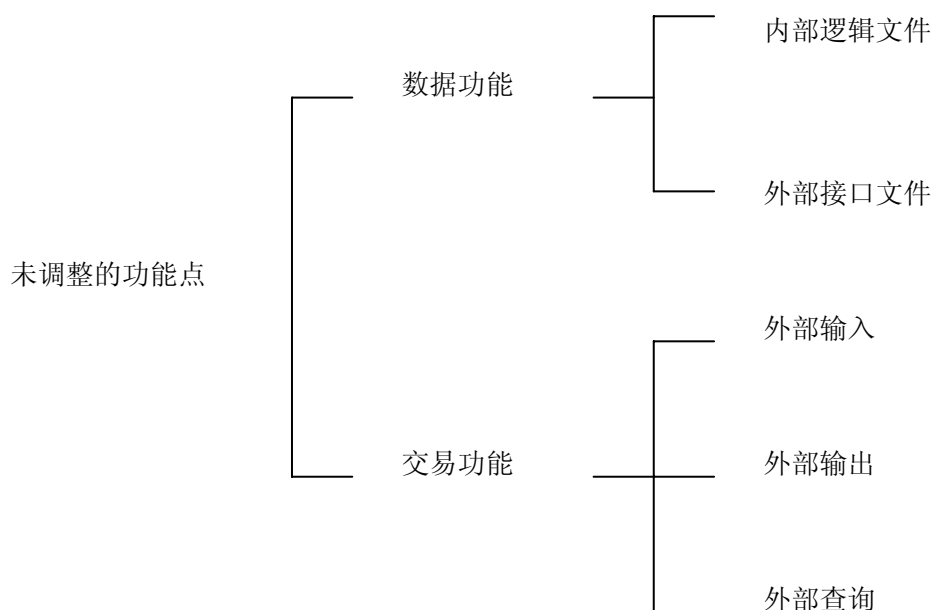
-
- ◆ 开发项目功能点分析
 - ◆ 升级项目功能点分析
 - ◆ 应用的功能点分析

2.3.2. 识别分析范围和应用边界

分析范围的界定限制了在一次分析中所应该包含的功能范围。应用边界的界定划出了被分析的应用与用户之间的界限

2.3.3. 确定未经调整的功能点数（Unadjusted Function Point Count -- UFPC）

未经调整的功能点数反映了应用向用户提供的功能的数量。计量用户功能的原则是数出“**该应用向客户提供了什么功能**”而不是“**这些功能是怎么提供的**”。在计数过程中只计入用户要求和定义的组件。未经调整的功能点数（UFPC）包括两个大类：数据和交易。这两个类别的进一步定义在下图中：



2.3.3.1. 数据功能的计数

数据功能是指向用户提供的满足内部或者外部数据需求的功能。数据功能有两类，内部逻辑文件和外部接口文件：

- ◆ 内部逻辑文件：

内部逻辑文件（Internal Logical File -- ILF）是指一组用户能够识别的，存在内在逻辑关联的数据或者控制信息。这些数据或者信息应该是在本应用的边界之内被控制的。ILF 的主要目的是**容纳一组在本应用中由一个或者一组基本处理来维护的数据**。

◆ 外部接口文件：

外部接口文件（External Interface File-- EIF）是指一组用户能够识别的，在本应用中被引用的，以及存在内在逻辑关联的数据或者控制信息。与内部逻辑文件不同，这些数据或者信息是在本应用的边界之外被控制的。EIF 的主要目的是容纳一组在本应用中由一个或者一组基本处理引用到的数据。这就意味着在本应用中的 EIF 必须是另外一个应用中的 ILF。

2.3.3.2. 交易功能的计数

交易功能指的是向用户提供的用来处理数据的功能。

◆ 外部输入：

外部输入（External Input -- EI）是指一个处理来自本应用边界之外的一组数据或者控制信息的基本处理。外部输入的基本目的是为了维护一个内部逻辑文件（ILF）或者改变系统的行为。

◆ 外部输出：

外部输出（External Output -- EO）是指一个向应用边界之外发送数据或者控制信息的基本处理。外部输出的基本目的是为了向用户展示一组经过了除了提取之外的其他逻辑处理的数据或者控制信息。这里的其他处理包括**至少一个数学演算或者对衍生数据的生成**。外部输出也可能包括对内部逻辑文件（ILF）的维护或者对系统行为的改变。

◆ 外部查询：

外部查询（External Query -- EQ）是指是指一个向应用边界之外发送数据或者控制信息的基本处理。外部查询的基本目的是为了向用户展示提取的数据或者控制信息。外部查询的逻辑处理里面不包含数学公式或者计算以及对衍生数据的生成。外部查询不维护内部逻辑文件（ILF），也不会引起系统行为的改变。

2.3.3.3. 确定调整系数

调整系数（Value Adjustment Factor -- VAF）反映的是应用给用户提供的功能的概况。VAF 包含了 14 个基本系统特征（General System Characteristic -- GSC），每一个特征都有特定的规则描述来帮助使用者确定该特征对本应用影响的大小。这些影响值从 0 到 5，分别表示对系统从无影响到具有强烈影响的程度。

2.3.3.4. 计算经过调整的功能点

经过调整的功能点（Adjusted Function Point）是针对不同类型的使用（开发、升级、应用）使用不同的公式计算得来的。在后面的章节里还会介绍这些公式。

3. 分析流程

3.1. 决定分析的类型

3.1.1. 定义：功能点分析的类型

功能点的分析既可以应用在项目上也可以应用在应用上。以下是三种功能点分析的类型：

- ◆ 开发项目
- ◆ 升级项目
- ◆ 应用

以下是对这三种类型的定义：

开发项目：

这种类型的分析为用户提供应用第一次安装的时候所提供的功能多少的度量。

升级项目：

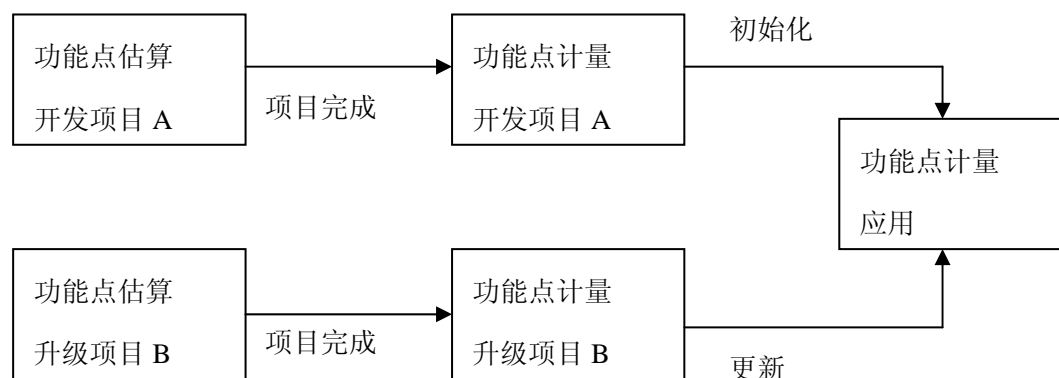
这种类型的分析为用户提供升级项目对目前系统所提供的功能修改、增加和删除的功能数量的度量

当升级的应用被安装的时候，应用的功能点度量应该得到相应的更新。

应用：

对应用的功能点分析是对一个已经安装的应用系统进行度量的手段。它也被称为基线功能点或者安装功能点。这个度量揭示了当前应用为用户所提供的功能数量。它在应用第一次被安装的时候确立，在系统升级的时候得到更新。

下图说明了这三种类型的分析之间的关系：



估算和最终计量的关系：

在项目开始的时候所进行的功能点分析是对应用将要给用户提供的功能的一种预测，在项目进展过程中，对功能、范围的调整和细化都有可能最终发布给用户的系统的功能与最初的估算不完全相符。这种现象有时候被称为范围的偏移（Scope Creep）。

在这种情况下，在项目结束的时候对功能点进行重新的计算是非常必要的，只有这样，应用的功能点才能真正反映发布给客户的功能状况。

3.2. 识别分析范围和应用边界

3.2.1. 识别分析范围和应用边界中的定义

本节定义的目的是对分析范围和应用边界进行定义，并且揭示此二者是如何受到分析目的的影响的。

对分析目的的定义：

功能点分析的目的是提供对业务问题的答案。该目的：

- ◆ 决定功能点分析的类型并且通过在划定的范围内进行功能点分析来获得对所需要解决的业务问题的答案；
- ◆ 影响了应用软件及其周边的界限的界定。例如，如果人力资源管理模块是作为一个软件包的一部分来提供的，那么分析者可能决定将人力资源管理作为一个单独的应用来看待。

分析的目的的例子有：

- ◆ 通过进行功能点分析，为项目估算开发第一个版本所需要的人力投入的估算提供输入；
- ◆ 为已经安装的软件提供一个规模的基线；

-
- ◆ 对不同供应商所提供的软件包进行功能上的比较。

定义计数范围

计数范围定义了在一次功能点分析中所需要涵盖的功能范围。

计数范围：

- ◆ 定义了被计数的软件的一个集合或者子集；
- ◆ 是由这一次分析的目的所决定的；
- ◆ 识别出为了满足分析目的所需要提供的答案中应该包括的功能；
- ◆ 可能**包含了不止一个应用**。

为了不同的计数目的，该范围的定义如下：

- ◆ 在升级项目中，该范围包括所有被**增加，改变和删除**的功能。
- ◆ 开发项目的范围包括**所有牵涉到的功能，包括被创建的或者被客户化的功能**。
- ◆ 对应用的功能点分析的范围可能包括被用户使用到的功能或者该应用所提供的所有的功能。

3.2.2. 定义应用边界

应用边界表示所度量的软件 and 用户之间的边界。

应用边界：

- ◆ 定义了对于应用来说什么是外界
- ◆ 是对应用的“内部”和“外部用户世界”之间接口的概念性定义
- ◆ 是交易所处理的数据（EI, EO 和 EQ）进入和送出该应用的一层“细胞膜”
- ◆ 封装了由该应用所维护的数据集合（ILF）
- ◆ 帮助识别出该应用所使用，但并非是由该应用所维护的数据（EIF）
- ◆ 是由用户对该应用的商业视角所决定而非技术/实施考虑决定的

3.3. 分析范围以及应用边界的规则和流程

本节定义了识别分析范围以及定义应用边界时需要遵循的规则和流程。应用的边界之所以重要，是因为它直接影响到分析的最终结果；同时也是因为边界的划定区分开了进入和输出该应用的数据（EI 和 EO）。

3.3.1. 边界识别的规则

识别边界的时候必须应用如下的规则：

- ◆ 边界的定义必须基于用户的视角，边界必须是用户能够理解和描述的；
- ◆ 相关应用之间的边界是由用户看到的不同功能区域来划分，而不是由技术考虑来划分的；
- ◆ 应用之间初始的边界不会因为功能点分析而改变。

3.3.2. 分析范围和应用边界流程：

步骤	行动
1	建立分析的目的；
2	识别分析范围；
3	识别应用的边界；
4	记录如下条目： <ul style="list-style-type: none">● 分析目的● 分析范围● 应用边界● 所有与以上条目有关的假设

3.3.3. 边界识别的一些技巧：

- ◆ 使用系统的用户需求或者获得一个系统的流程图，在系统的周围画一个圈，以此区别系统内部和外部的元素；
- ◆ 察看数据的维护方式；
- ◆ 察看其他数据，例如成本，人力，缺陷等。功能点分析中的应用范围应该和其他的度量数据的应用范围是一致的。

3.4. 计数数据功能

数据功能指的是提供给用户的用来满足内部和外部数据需求的功能。数据功能分为内部逻辑文件（ILF）和外部接口文件两种（EIF）。在这里，文件的概念**并非传统意义的文件，而是一组逻辑上相关联的数据的集合**。

本章将介绍对 ILF 和 EIF 的定义，并且解释计数这些功能的规则和流程。

3.4.1. 定义

◆ 内部逻辑文件：

内部逻辑文件（Internal Logical File -- ILF）是指一组用户能够识别的，存在内在逻辑关联的数据或者控制信息。这些数据或者信息应该是在本应用的边界之内被控制的。ILF 的主要目的是容纳一组在本应用中由一个或者一组基本处理来维护的数据。

◆ 外部接口文件：

外部接口文件（External Interface File-- EIF）是指一组用户能够识别的，在本应用中被引用的，以及存在内在逻辑关联的数据或者控制信息。与内部逻辑文件不同，这些数据或者信息是在本应用的边界之外被控制的。EIF 的主要目的是容纳一组在本应用中由一个或者一组基本处理引用到的数据。这就意味着在本应用中的 EIF 必须是另外一个应用中的 ILF。

◆ 控制信息

控制信息指的是**影响到被分析的 ứng dụng的基本处理的信息**。它规定了什么，何时和怎样对信息进行处理。例如，在一个工资系统中规定了每一个场地上员工的发薪日，这个日期就包含了对处理发薪信息的时间要求。

◆ 用户可以识别的

用户可以识别的指的是对数据组的需求的状态是经过相互认可，被用户和软件开发人员共同理解的。

◆ 维护

维护的含义是通过基本处理对数据进行修改的能力。

◆ 基本处理

基本处理指的是对用户来说有意义的最小的功能活动单元。

3.4.2. 计数流程概述

本节描述的是计数流程的大略，主要目的是为了说明计数规则的环境

步骤	行动
1	识别 ILF 和 EIF；
2	确定 ILF 或者 EIF 的复杂度以及判断它们对未经调整的功能点的贡献。

在每一个行动中都需要应用 ILF 和 EIF 的计数规则。计数规则包括如下两种：

-
- ◆ 识别规则
 - ◆ 复杂度和贡献规则

具体的规则大略如下：

- ◆ ILF 识别规则
- ◆ EIF 识别规则
- ◆ 数据元素类型（DET）
- ◆ 记录元素类型（RET）

3.4.3. ILF 识别规则

要寻找 ILF, 就必须找到符合 ILF 定义的数据集合或者控制信息的集合。这样的集合必须符合如下的规则：

- ◆ 数据的集合必须是逻辑相关的并且是用户可以识别的；
- ◆ 这些数据或者控制信息必须是在本应用的边界内被维护的。

3.4.4. EIF 识别规则

要寻找 EIF, 就必须找到符合 EIF 定义的数据集合或者控制信息的集合。这样的集合必须符合如下的规则：

- ◆ 数据的集合必须是逻辑相关的并且是用户可以识别的。
- ◆ 这一组数据或者控制信息必须是在被应用内被引用, 但并非是在本应用边界范围内的。
- ◆ 这一组数据或者控制信息的维护工作不是在本应用内进行的
- ◆ 这一组数据或者控制信息是另一个应用的 ILF

3.4.5. 复杂度和贡献的定义和规则

一个 ILF 或者 EIF 的复杂度直接决定了该功能对未经调整的功能点数的贡献状况。每一个 ILF 或者 EIF 都必须有一个复杂度与它相关联。该复杂度是由这个 ILF 或者 EIF 的数据元素类型（DET）数和记录元素类型（RET）数决定的。

DET 的定义：

一个数据元素类型（DET）指的是一个用户可以识别的，非重复的域。

DET 计数规则

-
- ◆ 每一个通过执行基本处理从 ILF 或者 EIF 进行维护或者访问的用户可以识别的以及非重复的域都作为一个 DET

例如：一个账号，就算是在多个字段中进行保存的，也作为一个 DET；又例如，在客户订单处理中计算出来的营业税字段也要算做一个 DET

- ◆ 每一个在 ILF 或者 EIF 中作为对其他数据功能的联系而存在的字段都应该作为一个 DET 而被计数

例如：一个关系型数据库的表中的外键在它所联系的两张表中都应该作为一个 DET 进行计数

RET 的定义：

一个记录类型元素(RET)指的是一个 EIF 或者 ILF 中用户可以识别的数据的子集。

RET 有两种类型：可选的或者必须的

可选的子集是用户可以选择在一个基本处理中使用或者不使用它来创建或者增加这些数据的实例的子集。

必须的子集是用户必须使用其中至少一个的数据子集。

例如：在一个人力资源管理应用中，要增加一个雇员的信息，必须输入一些基本数据，一个雇员可以是按月支取薪水的也可以是按小时支取薪水的。用户已经决定：**某个雇员必须是月薪或者时薪类型的。而每一种类型都可以有附加的“子女”的信息。**在这个例子中，一共有三种类型的 RET：

- ◆ 支取月薪的雇员：（必须的）包括基本信息
- ◆ 支取时薪的雇员：（必须的）包括基本信息
- ◆ 子女：（可选的）

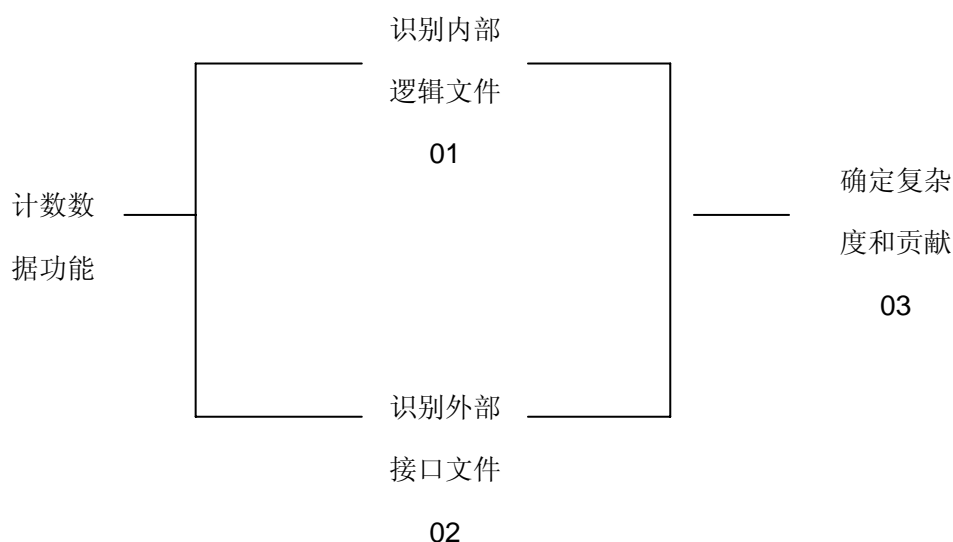
RET 计数规则：

以下两个原则之一必须得到应用：

- ◆ 在一个 ILF 或者 EIF 中的任何一个必须的或者可选的数据子集必须被计数为一个 RET
或者
- ◆ 如果一个 ILF 或者 EIF 中不存在数据子集，则该 ILF 或者 EIF 被计数为一个 RET

3.4.6. ILF/EIF 计数流程

下图表示了计数 ILF 和 EIF 的概要流程：



识别流程：

步骤	行动	相关的标准
01	识别内部逻辑文件	ILF识别规则
02	识别外部接口文件	EIF识别规则
03	确定复杂度和贡献	复杂度和贡献确定流程

3.4.7. 复杂度和贡献确定流程

遵循以下的步骤来计算 ILF 和 EIF 的复杂度和对未经调整的功能点（UFPC）的贡献：

步骤 1：根据[复杂度和贡献的定义和规则](#)中的要求识别和计数DET以及RET；

步骤 2：使用下面的矩阵来决定功能的复杂度：

	1~19 个 DET	20~50 个 DET	51 个以上 DET
1 个 RET	低	低	中
2~5 个 RET	低	中	高
6 个以上 RET	中	高	高

步骤 3：参照下面的表格，为每一个 ILF 和 EIF 决定它们未经调整的功能点数

ILF 参照表：

复杂度	功能点数
低	7
中	10
高	15

EIF 参照表:

复杂度	功能点数
低	5
中	7
高	10

步骤 4: 计算各 ILF 和 EIF 对 UFPC 的贡献, 例如:

功能类型	功能复杂度	小计	总计
ILF	低: <u>1</u> X7=	7	22
	中: <u>0</u> X10=	0	
	高: <u>1</u> X15=	15	
EIF	低: <u>0</u> X5=	0	64
	中: <u>2</u> X7=	14	
	高: <u>5</u> X10=	50	

3.4.8. 数据功能计数技巧

下面的技巧能够帮助我们应用 ILF 和 EIF 的计数规则, 但是这些技巧本身并不是规则

该数据组是否是一个支持特定用户需求的逻辑的集合?

- ◆ 一个应用可以在不同的处理中使用同样的 ILF 和 EIF, 但是它们只被计数一次;
- ◆ 一个逻辑文件在一个应用中不可能同时既是 ILF 又是 EIF;
- ◆ 如果一组数据没有单独被看作一个 ILF 或者 EIF, 那么这组数据中的 DET 应该记入它所属的 ILF 或者 EIF 中;
- ◆ 不要想当然的认为一个物理文件或者数据库中的一个表就是一个 ILF, 虽然它们是紧密相关的, 但是这种一对一的对应关系并非总是成立;

-
- ◆ 并非所有的物理文件都是一个逻辑文件或者逻辑文件的一部分。

数据是在哪里得到维护的？是在应用边界之内还是之外？

- ◆ 考察工作流程；
- ◆ 在功能拆分的时候，识别出与用户以及与其他应用的接口；
- ◆ 完整考察处理过程图表以获得提示。

如果一个 ILF 被多个应用维护，那么对于每一个维护它的应用都要计算一个 ILF。这些 ILF 的复杂度由它在不同应用中的 DET 所决定。

3.5. 计数交易功能

简介：交易功能代表应用所提供给用户的用来处理数据的功能。交易功能包括外部输入、外部输出和外部查询。本节定义了 EI, EO 以及 EQ，包括了它们的计数规则以及流程。

3.5.1. 定义

3.5.1.1. 基本定义

- ◆ 外部输入：

外部输入（External Input -- EI）是指一个处理来自本应用边界之外的一组数据或者控制信息的基本处理。外部输入的基本目的是为了维护一个内部逻辑文件（ILF）或者改变系统的行为。

- ◆ 外部输出：

外部输出（External Output -- EO）是指一个向应用边界之外发送数据或者控制信息的基本处理。外部输出的基本目的是为了向用户展示一组经过了除了提取之外的其他逻辑处理的数据或者控制信息。这里得其他处理包括至少一个数学演算或者对衍生数据的生成。外部输出也可能包括对内部逻辑文件（ILF）的维护或者对系统行为的改变。

- ◆ 外部查询：

外部查询（External Query -- EQ）是指一个向应用边界之外发送数据或者控制信息的基本处理。外部查询的基本目的是为了向用户展示提取的数据或者控制信息。外部查询的逻辑处理里面不包含数学公式或者计算以及对衍生数据的生成。外部查询不维护内部逻辑文件（ILF），也不会引起系统行为的改变。

3.5.1.2. 交易功能的总结

交易功能（EI,EO,EQ）之间的主要区别在于他们的目的。下表中列出了这三种功能在

目的上的主要区别。外部输出和外部查询的主要区别在于外部输出除了对数据进行表示之外，还可能完成对数据进行修改、维护 ILF 或者改变系统行为的功能。

功能	交易功能		
	EI	EO	EQ
改变系统行为	PI	F	N/A
维护 ILF	PI	F	N/A
向用户展示数据	F	PI	PI

示例：

PI: Primary Intent 该功能的主要目的

F: Function 该交易功能具有这项功能，但不是主要目的，有的时候适用

N/A: Not Applicable 不适用

3.5.1.3. 相关术语的定义

◆ 基本处理

基本处理是指对用户来说有意义的最小的功能活动单元。

◆ 控制信息

控制信息是指影响到被分析的应用的基本处理的信息。它规定了什么，何时和怎样对信息进行处理。例如，在一个工资系统中规定了每一个场地上员工的发薪日，这个日期就包含了对处理发薪信息的时间要求。

◆ 维护

维护的含义是通过基本处理对数据进行修改的能力。

◆ 用户

用户是指发起用户规格说明书的人，也是与应用进行交互的任何人或者对象。例如，在一个人力资源管理系统中，人力资源部的操作员是用户，同时，一个处理福利应用，因为要从人力资源管理系统中提取数据，也是该系统的用户。

◆ 逻辑处理

逻辑处理指的是用户请求进行的完成一个基本处理的需求，逻辑处理的例子包括：

- 数据验证

例如在增加一个雇员的信息的时候，系统需要对加入的信息进行验证

- 数学公式和计算

例如，在对雇员信息进行报告的时候，需要计算领取周薪的雇员数量和领取时薪的雇员数量以及雇员总数

- 等效数据的换算

例如一个货币换算基本处理引用了一个货币汇率，但是这个汇率是由一个表中取出的，并不需要进行计算。

- 数据的过滤和选择

例如，为了生成一个按照角色选择的雇员列表，一个基本处理将雇员的信息与工作角色进行比较和选择以生成这个列表。

- 分析适用的条件

例如，在添加一个雇员的时候，首先判断他是支取月薪还是支取时薪的雇员，然后决定采取何种逻辑处理。

- 更新一个或者多个 ILF

例如：当添加一个雇员的信息的时候，雇员这个 ILF 就被更新了。

- 引用一个或者多个 ILF 或者 EIF

例如，当添加一个雇员的时候，货币这个 EIF 被引用以决定该雇员的薪水折算成美元的数量。

- 提取数据或者控制信息。

例如，当用户察看薪酬级别的时候，薪酬级别信息被提取出来。

- 运用现有的数据生成衍生数据

例如，当生成一个病历代码的时候，例如（SMIJO01）这个代码由下列的数据衍生而来：病人的名字的前三个字母+病人的姓的前两个字母+唯一的两位数字代码。

- 改变系统的行为

例如，当把发薪日由每月的 15 日和最后一天改为每周的周五的时候，发薪这个基本处理的行为就被改变了。

-
- 对应用范围之外准备和展示数据

例如，向用户显示雇员列表。

- 接受进入系统边界的数据或者控制信息的

例如，用户输入一些信息来添加一个客户订单。

- 恢复和重新整理数据

例如，用户要求按照字母顺序列出雇员名单。

3.5.1.4. 交易功能执行的逻辑处理总结

下表列出了每一种交易功能可能进行的逻辑处理，对于每一种交易功能，有些逻辑处理是必须的，而有些则是可选的。具体见下表：

逻辑处理	EI	EO	EQ
1. 数据验证	C	C	C
2. 数学公式和计算	C	M*	N
3. 等效数据的换算	C	C	C
4. 数据的过滤和选择	C	C	C
5. 分析适用的条件	C	C	C
6. 更新一个或多个ILF	M*	M*	N
7. 引用一个或多个ILF或者EIF	C	C	M
8. 提取数据或者控制信息	C	C	M
9. 运用现有数据生成衍生数据	C	M*	N
10. 改变系统的行为	M*	M*	N
11. 对应用范围之外准备和展示数据	C	M	M
12. 接受进入系统边界的数据或者控制信息	M	C	C
13. 恢复和重新整理数据	C	C	C

说明：

C = can	该功能可以执行此项逻辑处理，但并非必须
M = mandatory	该功能必须执行此项逻辑处理
M*=mandatory(*)	该功能必须执行列表中所有带 M*标记的逻辑处理中的至少一项
N=cannot	该功能一定 不能 执行该逻辑处理

3.5.2. EI,EO,EQ 计数规则

本节定义计数 EI，EQ 和 EQ 时所适用的规则。

3.5.2.1. 交易功能计数的概要流程

本节的目的是提供对交易功能进行计数的环境概述，具体的流程请参见：[3.5.4 EI,EO,EQ的计数流程](#)

计数的概要流程包含如下的步骤：

步骤	行动
1	识别基本处理
2	决定识别出来的基本处理的主要目的，并把他们界定为 EI,EO 或者 EQ
3	应用识别规则，对 EI,EO 和 EQ 进行验证
4	决定交易功能的复杂度
5	确定它们对未经调整的功能点的贡献

具体规则将在以后的内容中进行定义。

3.5.2.2. 基本处理的识别规则

为了识别基本处理，必须在应用中寻找用户的活动。一个处理必须符合以下的所有规则才可以被计数为一个基本处理。

- ◆ 该处理是对用户来说有意义的最小的活动单位。
- ◆ **该处理是一个自含的活动，应用的业务状态在执行过该处理后保持稳定。**

3.5.2.3. 交易功能计数规则

要对基本处理进行分类，首先查找该基本处理是否满足以下对 EI，EO 和 EQ 的主要目的的定义，然后应用它们各自的计数规则来判断是否真的属于该类别。

3.5.2.3.1. EI 的主要目的描述：

该基本处理的主要目的应该是对 ILF 进行维护或者改变系统行为。

3.5.2.3.2. EI 的计数规则：

对于具有上述主要目的的基本操作，它还必须满足以下所有的规则。

- ◆ 数据或者控制信息是从应用边界以外接收到的；
- ◆ 如果接收的信息不是控制信息，那么至少需要有一个 ILF 被改变了；
- ◆ 该基本操作必须符合以下三条描述之一：
 - 使用的处理逻辑是独特的，与应用中其他外部输入的处理逻辑是不同的；
 - 在该处理中识别出来的数据元素是与应用中其他外部输入的数据元素不同的；
 - 在该处理中引用的 ILF 和 EIF 是与应用中其他外部输入所引用的不同的。

3.5.2.3.3. EO 和 EQ 的共同主要目的描述：

该基本处理的主要目的应该是向用户展示信息。

3.5.2.3.4. EO/EQ 共享的计数规则：

对于具有上述主要目的的基本操作，它还必须满足以下所有的规则才能被看作一个 EO 或者 EQ。

- ◆ 该操作向应用边界之外发送数据；
- ◆ 该基本操作必须符合以下三条描述之一：
 - 使用的处理逻辑是独特的，与应用中其他 EO/EQ 的处理逻辑是不同的；
 - 在该处理中识别出来的数据元素是与应用中其他 EO/EQ 的数据元素不同的；
 - 在该处理中引用的 ILF 和 EIF 是与应用中其他 EO/EQ 所引用的不同的。

3.5.2.3.5. EO 计数的补充规则：

以下规则是 EO 所特有的，一个基本操作出了要满足以上的共享规则之外，还必须满足以下补充规则之一才可以被认为是一个 EO：

- ◆ 该基本处理中的处理逻辑中包含至少一个数学公式；
- ◆ 该基本处理中的处理逻辑创建衍生数据；
- ◆ 该基本处理中的处理逻辑维护至少一个 ILF；

-
- ◆ 该基本处理中的处理逻辑改变系统行为。

3.5.2.3.6. EQ 计数的补充规则:

以下规则是 EQ 所特有的，一个基本操作出了要满足以上的共享规则之外，还必须满足以下所有的补充规则才可以被认为是一个 EQ:

- ◆ 该基本处理中的处理逻辑从 ILF 或者 EIF 中提取信息或者控制信息;
- ◆ 该基本处理中的处理逻辑不包含数学公式;
- ◆ 该基本处理中的处理逻辑不生成衍生数据;
- ◆ 该基本处理中的处理逻辑不维护 ILF;
- ◆ 该基本处理中的处理逻辑不改变系统行为;

3.5.3. 复杂度和贡献的定义和规则

一个交易功能的复杂度直接决定了该功能对未经调整的功能点数的贡献状况。每一个交易功能都必须有一个复杂度与它相关联。该复杂度是由这个 EI, EO 或者 EQ 的数据元素类型 (DET) 数和引用文件类型 (FTR) 数决定的。

FTR 的定义:

FTR = File Type Referenced = 引用文件类型

一个 FTR 的含义是:

- ◆ 一个由交易功能所维护的 ILF;
- 或者
- ◆ 一个被交易功能所读取的 EIF。

3.5.3.1. EI 的复杂度和贡献规则

本节定义了为了确定 EI 的复杂度而对 FTR 和 DET 进行计数的规则。

3.5.3.1.1. EI 的引用文件类型 (FTR) 计数规则

- ◆ 每一个被维护的 ILF 计为一个 FTR;
- ◆ 为在处理这个 EI 过程中所读取得每一个 ILF 或者 EIF 计数一个 FTR;
- ◆ 对于**既被读取又被维护的 ILF**，只计一个 FTR。

3.5.3.1.2. EI 的数据元素类型（DET）计数规则

- ◆ 对完成这个输入操作的每一个满足如下条件的数据项计一个 DET：
 - 非重复的
 - 用户可以识别的
 - 进入或者输出该应用的边界的
- ◆ 对于由该处理所提取的或者衍生的一个数据，如果它没有穿越该应用的边界，则不能计为一个 DET；
- ◆ 发布到应用之外的每一个信息（例如一个出错信息，或者确认操作成功的信息）计一个 DET。

3.5.3.2. EO/EQ 的复杂度和贡献规则

本节定义了为了确定 EO/EQ 的复杂度而对 FTR 和 DET 进行计数的规则。

3.5.3.2.1. EO/EQ 共享的引用文件类型（FTR）计数规则

- ◆ 为在处理这个 EI 过程中所读取得每一个 ILF 或者 EIF 计数一个 FTR。

3.5.3.2.2. EO 特定的引用文件类型（FTR）计数规则

- ◆ 每一个被维护的 ILF 计为一个 FTR；
- ◆ 对于既被读取又被维护的 ILF，只计一个 FTR。

3.5.3.2.3. EO/EQ 共享的数据元素类型（DET）计数规则

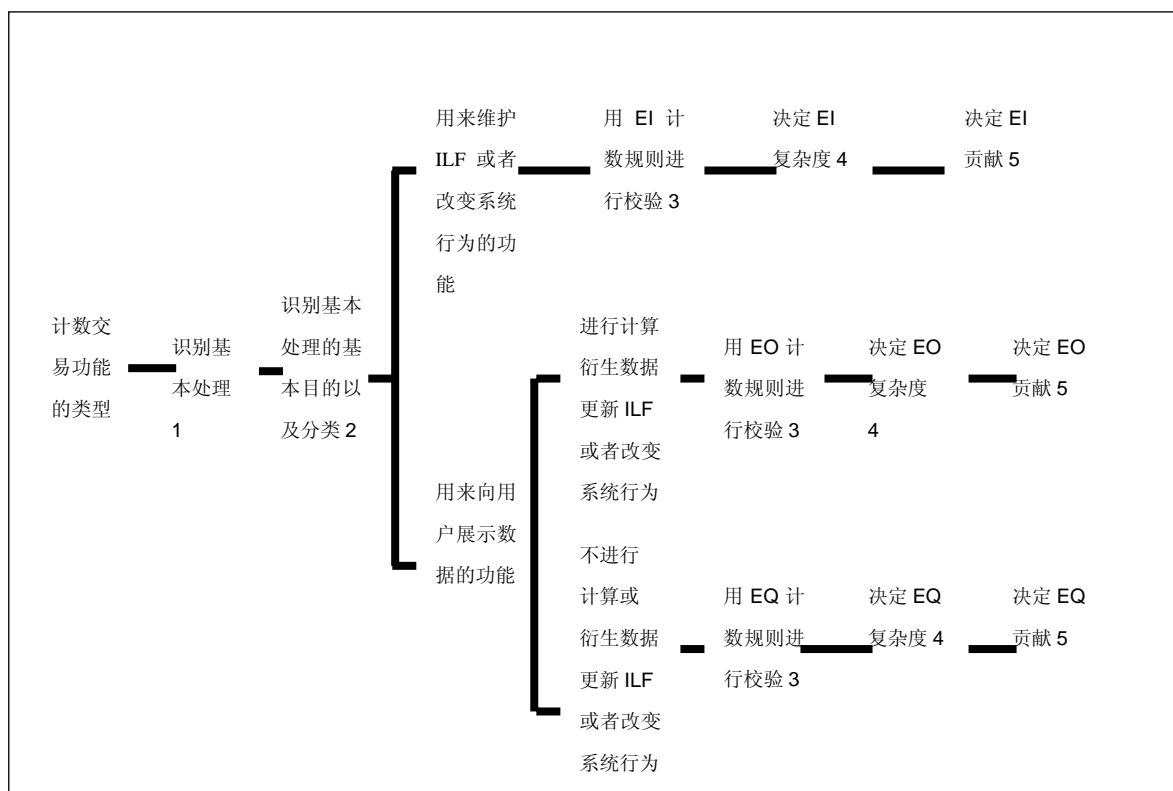
- ◆ 为每一个进入应用边界的，界定什么数据，将何时，和怎样被提取和生成的用户可以识别的以及不重复的数据计为一个 DET；
- ◆ 为每一个用户可以识别的，不重复的退出该应用边界的数据计一个 DET；
- ◆ 如果一个 DET 既进入又退出该应用的边界，则只计一个 DET；
- ◆ 如果要引发一个基本处理有多种方法，那么这多种方法也只计为一个 DET；
- ◆ 发布到应用之外的每一个信息（例如一个出错信息，或者确认操作成功的信息）计一个 DET ；
- ◆ 对于由该处理所提取的或者衍生的一个数据，如果它没有穿越该应用的边界，则不能计为一个 DET；
- ◆ 不要将格式化文本计为 DET（例如报表标题，栏标题等）；

- ◆ 不要将页码信息和系统标签（例如时间标记）计为 DET。

3.5.4. EI,EO,EQ 的计数流程

本节包括了对 EI,EO 和 EQ 的详细记述流程的描述。

流程图：下图显示了计数流程的概要



下面的段落描述了这个流程：

步骤	活动	适用规则
1	识别基本处理	3.5.2.2 基本处理的识别规则
2	识别基本处理的主要目的，并且划分为 EI, EO 和 EQ	3.5.2.3.1 EI的主要目的描述： 3.5.2.3.3 EO和EQ的共同主要目的描述：
对于主要目的是维护 ILF 或者改变系统行为的基本处理		
3	用 EI 识别规则进行校验	3.5.2.3.2 EI的计数规则：

4	决定 EI 复杂度	3.5.5 复杂度和贡献确定流程
5	决定 EI 的贡献	3.5.5 复杂度和贡献确定流程
对于主要目的是向用户展示数据，并且进行计算，或者维护 ILF,或者改变系统行为的基本处理		
3	用 EO 识别规则进行校验	3.5.2.3.4 EO/EQ共享的计数规则: 3.5.2.3.5 EO计数的补充规则:
4	决定 EO 复杂度	3.5.5 复杂度和贡献确定流程
5	决定 EO 的贡献	3.5.5 复杂度和贡献确定流程
对于主要目的是向用户展示数据，并且不进行计算，或者维护 ILF,或者改变系统行为的基本处理		
3	用 EQ 识别规则进行校验	3.5.2.3.4 EO/EQ共享的计数规则: 3.5.2.3.6 EQ计数的补充规则:
4	决定 EQ 复杂度	3.5.5 复杂度和贡献确定流程
5	决定 EQ 的贡献	3.5.5 复杂度和贡献确定流程

3.5.5. 复杂度和贡献确定流程

本节规定的是确定 EI，EO 和 EQ 复杂度以及它们对未经调整的功能点数的流程。

步骤 4：使用[3.5.3.1 EI的复杂度和贡献规则](#)决定各EI的FTR和DET

用下表决定各 EI 的复杂度：

FTR \ DET	DET	1~4	4~15	16 以上
0~1 个	低	低	中	
2 个	低	中	高	
3 个以上	中	高	高	

步骤 4： 使用[3.5.3.2 EO/EQ的复杂度和贡献规则](#)决定各EO和EQ的FTR和DET

用下表决定各 EO 和 EQ 的复杂度：

FTR \ DET	DET	1~5	6~19	20 以上
0~1 个	低	低	中	
2~3 个	低	中	高	
4 个以上	中	高	高	

步骤 5： 参照下面的表格决定它们对未经调整功能点的贡献，为每一个交易功能（EI、EO、EQ）决定它们未经调整的功能点数。

EI、EQ 参照表：

复杂度	未经调整的功能点数
低	3
中	4
高	6

EO 参照表：

复杂度	未经调整的功能点数
低	4
中	5
高	7

3.5.6. 交易功能计数技巧

下面的技巧能够帮助我们应用交易功能的计数规则，但是这些技巧本身并不是规则。

- ◆ 数据是否是从应用边界之外获得的？
 - 检查工作流；
 - 在对处理功能进行拆分的时候识别用户或其他应用与该应用的接口。
- ◆ 处理是否从用户角度看是最小的逻辑活动单元？
 - 从纸张或者在线的报表中判断；
 - 检查 ILF 以判断用户是如何对数据进行分组的；
 - 在对处理功能进行拆分的时候识别用户或其他应用与该应用的接口；
 - 看看在原始的手动系统中的运行情况；
 - 注意：有时候一个物理的输入，一个交易文件或者一个屏幕可能对应着多个 EI、EO 或者 EQ。
- ◆ 该处理是否是一个自含的操作，它是否将该应用的业务形态保持在一个稳定的状态？
 - 看看其他的 EI、EO、EQ,以理解用户是怎样和这些数据互动的；
 - 察看流程图来获得提示；
 - 看看在原始的手动系统中的运行情况；
 - 检查与其他决定之间的一致性。
- ◆ 该处理逻辑是独特的么？
 - 识别出批处理的输入和输出；

-
- 排序和重组数据不是独特的处理。
 - ◆ 这个交易功能的数据元素与其他的交易功能有所不同么？
 - 如果一个交易功能的数据元素是另一个交易功能数据元素的子集，那么必须确认用户确实要求两个不同的处理，一个是关于全集的，一个是关于子集的。
 - ◆ 在把一个基本处理界定为 EI、EO 或者 EQ 之前，需要先明确它的主要目的；
 - ◆ 对基本处理的识别是基于用户和开发人员的共同理解的；
 - ◆ 功能拆分的元素并非总能够映射到一个基本处理；
 - ◆ 识别基本处理的时候需要对用户需求进行解释；
 - ◆ 就算一个 ILF/EIF 具有多个 RET，在引用到它的时候也只计一个 FTR。

3.6. 决定调整系数

本节解释了在功能点计数中的调整系数

3.6.1. 调整系数的决定

调整系数—value adjustment factor (VAF)是建立在 14 个用来评价被分析的应用的功能的通用系统特性（General System Characteristics—GSC）的基础之上的。每一个特性都有一些规则来进行评分，以判断该特性对这个应用的影响程度。评分的范围是从 0 到 5，分别代表没有影响到影响很大。

对这 14 个通用系统特性进行总结，然后计算出调整系数 VAF，该系数会在正负 35%的幅度上调整功能点数。

3.6.2. 确定 VAF 的流程

以下是确定 VAF 的概要流程：

步骤	活动
1	评估每一个通用系统特性，并且为它们确定影响程度（Degree of influence – DI）
2	将所有的 GSC 的 EI 相加，得到整体影响程度（Total Degree of Influence — TDI）
3	将 TDI 代入下列公式进行计算，算出 VAF $VAF = (TDI * 0.01) + 0.65$

3.6.3. 通用系统特性及其影响程度的评定

通用系统特性包括如下 14 个：

1. 数据通讯 （Data Communications）
2. 分布式数据处理 （Distributed Data Processing）
3. 性能 （Performance）
4. 使用强度高的配置 （Heavily Used Configuration）
5. 交易速度 （Transaction Rate）
6. 在线数据输入 （Online Data Entry）
7. 最终用户的效率 （End-User Efficiency）
8. 在线更新 （Online Update）
9. 复杂的处理 （Complex Processing）
10. 可重用性 （Reusability）
11. 安装的简易性 （Installation Ease）
12. 运行的简易性 （Operational Ease）
13. 多场地 （Multiple Sites）
14. 允许变更 （Facilitate Change）

影响程度分成如下 6 级：

影响程度级别	影响状况
0	不存在或者没有影响
1	偶尔的影响
2	轻微的影响
3	中等的影响
4	显著的影响
5	强烈的影响

各 GSC 的 DI 分级详述。

以下是各个 GSC 的 DI 分级的详细解释，如果在表中找不到严格的对应状况，则需要按照专业判断自行定义 DI 级别。

3.6.3.1. 数据通讯

数据通讯指的是应用直接与处理器通讯的程度。

在一个应用中所使用的数据或者控制信息一定是通过某种通讯手段来实现的。连接到本地控制器上的终端被认为是使用通讯设施，而协议指的是两个系统或者两个设备之间进行通讯时所使用的一种约定。所有的数据通讯链路都需要某种协议。

影响程度级别	影响程度的描述
0	应用是单纯的批处理或者 PC stand-alone
1	应用是批处理，但是有远程的数据输入或者打印
2	应用是批处理，但是有远程的数据输入以及打印
3	应用包括了在线数据收集或者一个批处理或者查询系统的远程处理前端
4	应用不仅包括前端，但是仅支持一种远程处理通讯协议
5	应用不仅包括前端，而且仅支持不止一种远程处理通讯协议

3.6.3.2. 分布式数据处理

分布式数据处理指的是应用在内部的组件之间传递信息的程度。这个特性是在应用边界内体现的一个特性。

影响程度级别	影响程度的描述
0	应用不支持组件之间的数据传输和处理功能
1	应用为用户可能进行的处理准备数据（例如用户使用电子表格或者数据库应用进行的处理）
2	应用准备了用于传输的数据，数据被传输并且被系统的其他组件所处理（不同于用户的处理）
3	分布式的数据传输和处理是在线的，但只是单向的

4	分布式的数据传输和处理是在线的，而且是双向的
5	处理功能是由系统中最恰当的组件动态地执行的

3.6.3.3. 性能

性能指的是处理时间，吞吐量等指标对开发的影响。

用户所提出的关于处理时间，吞吐量的要求直接影响到系统的设计，实施，安装和支持。

影响程度级别	影响程度的描述
0	用户没有提出特定的性能要求
1	提出了性能和设计方面的要求，但不需要采取特定行动
2	响应时间和吞吐量在系统峰值时是关键的，但是不需要采取相应的 CPU 使用方面的特殊设计。交易的完成期限是下一个营业日
3	响应时间和吞吐量在所有时间都是关键的，但是不需要采取相应的 CPU 使用方面的特殊设计。交易的完成期限比较严格
4	除了 3 中的要求之外，因为对需求的要求比较严格，在设计阶段需要进行性能分析
5	除了 4 中的要求之外，在设计和实施阶段需要使用性能分析工具来判断性能要求的完成情况

3.6.3.4. 使用强度高的配置

使用强度高的配置指的是计算机资源的限制对应用开发的影响程度。

使用强度高的运行配置是需要在设计时加以考虑的一个系统特性。例如，用户需要在现有的系统上运行这个应用，或者这个应用会在一个使用强度很高的专用配置上运行。

影响程度级别	影响程度的描述
0	没有明确的提出运行方面的限制
1	存在运行方面的限制，但是限制不如典型的应用中那样严格，不需要进行特殊的努力以满足运行限制
2	考虑了一些安全和时间方面的限制

3	应用中的一个部分包含了对处理器的特定需求
4	提出的运行限制对应用的中央处理器或者专用处理器提出了特殊的要求
5	除了 4 之外，还对应用的分布式的组件提出了限制

3.6.3.5. 交易速度

交易速度指的是业务交易处理速度的要求对应用开发的影响。

用户所提出的关于交易速度的要求直接影响到系统的设计，实施，安装和支持。

影响程度级别	影响程度的描述
0	没有预期的业务交易的高峰期
1	有预期的交易高峰期（例如，每月，每季度，每年）
2	每周都有预期的交易高峰期
3	每天都有预期的交易高峰期
4	用户在需求或者协议中提出了较高的交易速度要求，需要在设计阶段进行性能分析
5	用户在需求或者协议中提出了较高的交易速度要求，需要在设计阶段进行性能分析。此外，在设计，实施以及安装阶段还需要使用性能分析的工具

3.6.3.6. 在线数据输入

在线数据输入指的是数据通过交互的方式输入应用的程度。

应用提供在线数据输入和控制功能。

影响程度级别	影响程度的描述
0	所有的交易都是用批处理的方式完成
1	交互式的数据输入占总交易的 1%~7%

2	交互式的数据输入占总交易的 8%~15%
3	交互式的数据输入占总交易的 16%~23%
4	交互式的数据输入占总交易的 24%~30%
5	交互式的数据输入占总交易的 30%以上

3.6.3.7. 最终用户的效率

最终用户效率指的是对应用的人文因素以及使用简便性的考虑程度。

下列的功能设计是强调最终用户效率的：

- ◆ 导航帮助（例如功能键，跳转，动态菜单等）
- ◆ 菜单
- ◆ 在线帮助和文档
- ◆ 光标自动移动
- ◆ 滚屏
- ◆ 在线远程打印
- ◆ 预定义的功能键
- ◆ 在线方式提交的批处理任务
- ◆ 屏幕数据的光标选取
- ◆ 大量使用反白，高亮，颜色表示以及其他的标示方法
- ◆ 用户手册
- ◆ 鼠标界面
- ◆ 弹出窗口
- ◆ 尽可能少的屏幕
- ◆ 双语言支持（当 4 项）
- ◆ 多语言支持（当 6 项）

影响程度级别	影响程度的描述
--------	---------

0	以上一项都没有
1	1~3 项
2	4~5 项
3	6 项以上，但是没有用户对于效率的要求
4	6 项以上，用户对效率的要求足以使设计时必须考虑人文要素
5	6 项以上，用户对效率的要求使得开发人员必须使用特定的工具和流程以判定用户对效率的要求已经被达成

3.6.3.8. 在线更新

在线更新指的是内部逻辑文件 ILF 被在线更新的程度。

应用提供内部逻辑文件在线更新的功能。

影响程度级别	影响程度的描述
0	没有
1	包括对 1~3 个控制文件的在线更新。更新的流量低，恢复容易
2	包括对 4 个以上控制文件的在线更新。更新的流量低，恢复容易
3	包括对主要 ILF 的更新
4	除了 3 之外，对数据丢失的防范非常重要，在设计和实施中已经进行了这方面的特殊考虑。
5	除了 4 之外，大量的数据恢复工作存在成本考虑，包含了高度自动化的恢复流程。

3.6.3.9. 复杂的处理

复杂的处理指的是逻辑处理对应用设计和开发的影响。

复杂处理包括以下这些要素：

- ◆ 敏感的控制（例如特别的审计处理）或者应用的安全处理
- ◆ 大量的逻辑

- ◆ 大量的数学处理
- ◆ 因为例外处理造成的需要重新处理的情况
- ◆ 多种可能的输入/输出造成的复杂处理

影响程度级别	影响程度的描述
0	没有
1	有 1 种
2	有 2 种
3	有 3 种
4	有 4 种
5	有 5 种

3.6.3.10. 可重用性

可重用性指的是应用中以及其中的代码被进行特殊的设计,实施以及支持以便在其他的
应用中再次使用的程度。

影响程度级别	影响程度的描述
0	没有可重用的代码
1	代码在应用之内重用
2	应用中需要被其他用户重用的部分不足 10%
3	应用中需要被不止一个用户使用的部分超过 10%
4	应用被按照一种易于重用的方式被打包和文档化。用户在源代码级客户化该应用
5	应用被按照一种易于重用的方式被打包和文档化。用户使用用户参数来对该应用进行客户化

3.6.3.11. 安装的简易性

安装的简易性指的是从以前的环境转化该应用对开发的影响程度。

影响程度级别	影响程度的描述
0	用户对安装没有特定的要求，对安装没有特定的设置要求
1	用户对安装没有特定的要求，但是有特定的安装环境要求
2	用户提出了转化和安装的要求，转化/安装指南被提供给用户并且经过测试。但是转化的影响对该应用来说并非重要
3	用户提出了转化和安装的要求，转化/安装指南被提供给用户并且经过测试。转化的影响对该应用来说是重要的
4	除了 2 的要求之外，需要提供自动化的安装和转化工具并经过测试。
5	除了 3 的要求之外，需要提供自动化的安装和转化工具并经过测试。

3.6.3.12. 运行的简易性

运行的简易性指的是应用对运行的影响程度，例如启动，备份以及恢复流程的影响。

运行的简易性是应用提供的一种特性，它最小化了手工操作的要求，例如磁带的 mount，硬拷贝处理，以及现场的手工干预。

影响程度级别	影响程度的描述
0	除了备份之外，用户没有提出运行方面的特定要求
1-4	<p>以下四个特性，满足一个就在影响度上加 1</p> <ul style="list-style-type: none"> ◆ 提供了有效的启动，备份和恢复流程，但是需要操作员的手工干预 ◆ 提供了有效的启动，备份和恢复流程，但是不需要操作员的手工干预（计为 2 个特性） ◆ 应用最小化对于磁带 mount 的要求 ◆ 应用最小化对于硬拷贝处理的要求
5	应用的实际支持无人干预的运行。无人干预的运行意味着出了启动和关闭之外，系统不需要操作员的手工干预。错误的自动处理是应用的特性之一

3.6.3.13. 多场地

多场地指的是应用被设计用于多个场地以及多个用户组织的程度。

应用经过特殊的设计开发和支持，可以被安装在多个用户场地以及多个用户组织。

影响程度级别	影响程度的描述
0	用户需求不包括多场地和组织的要求
1	设计中考虑了多场地的要求，但是设计要求应用在不同的场地使用相同的软硬件环境
2	设计中考虑了多场地的要求，但是设计要求应用在不同的场地使用类似的软硬件环境
3	设计中考虑了多场地的要求，设计支持应用在不同的场地使用不同的软硬件环境
4	在 1 或者 2 的要求之上，提供了多场地的文档和支持计划并进行了测试
5	在 3 的要求之上，提供了多场地的文档和支持计划并进行了测试

3.6.3.14. 允许变更

允许变更指的是应用在设计上考虑允许处理逻辑和数据结构变化的程度。

应用可以具有如下的特性：

- 提供弹性的查询/报告功能以支持简单的请求。例如：对一个 ILF 进行的与/或逻辑（计为 1 项）
- 提供弹性的查询/报告功能以支持中等复杂度的请求。例如：对多个 ILF 进行的与/或逻辑（计为 2 项）
- 提供弹性的查询/报告功能以支持复杂的请求。例如：对多个 ILF 进行的与/或逻辑的组合（计为 3 项）
- 用表来保存业务控制数据，用户使用在线的互动手段对这些数据进行维护，但是维护在下一个交易日才能生效（计为 1 项）
- 用表来保存业务控制数据，用户使用在线的互动手段对这些数据进行维护，维护的结果即时生效（计为 2 项）

影响程度级别	影响程度的描述
0	没有以上的项
1	有 1 项
2	有 2 项
3	有 3 项
4	有 4 项
5	有 5 项

3.7. 计算调整功能点

本章介绍了完成功能点分析的最后一个步骤所需要的公式，它包含了为三种功能点分析的最后一个步骤（开发项目，升级项目以及应用）计算调整后的功能点的公式。

回顾功能点分析的步骤：

下表中列出了我们在进行功能点分析的时候所需要的所有步骤

步骤	活动
1	决定功能点分析的类型
2	识别应用边界
3	确定未经调整的功能点数： <ul style="list-style-type: none"> ● 计数数据功能 ● 计数交易功能
4	确定调整系数
5	计算调整后的功能点

3.7.1. 开发项目功能点的计算

开发项目的功能点包括如下这三个方面的内容。

-
- ◆ 项目的用户需求中包括的应用功能
 - ◆ 项目的用户需求中包括的转换功能
 - ◆ 应用的调整系数

应用功能：

软件被安装后所提供的满足用户业务需求的功能

转换功能：

安装时用来转换数据的功能或者满足用户其他特定转换要求的功能

应用的调整系数：

调整系数的定义和计算请参见[3.6 决定调整系数](#)

功能点公式： **$DFP = (UFP + CFP) * VAF$**

其中：

DFP = 开发项目的功能点

UFP = 应用在安装以后向用户提供的未经调整的功能点

CFP = 额外的转换功能的未经调整的功能点

3.7.2. 升级项目功能点的计算

升级项目的功能点包括如下这三个方面的内容。

- ◆ 项目的用户需求中包括的应用功能
- ◆ 项目的用户需求中包括的转换功能
- ◆ 应用的调整系数

应用功能：

应用功能包括：

- ◆ 升级项目所增加的功能
- ◆ 升级项目所改变的功能
- ◆ 升级项目所删除的功能

转换功能：

用户所要求的安装时用来转换数据的功能或者满足用户其他特定转换要求的功能的功

能点

应用的调整系数：

调整系数的定义和计算请参见[3.6 决定调整系数](#)

在升级项目中，有两个调整系数：

- ◆ 升级前的应用的调整系数
- ◆ 升级后的应用的调整系数

功能点公式：

$$EFP = (ADD + CHGA + CFP) * VAFA + DEL * VAFB$$

其中：

EFP = 升级项目的功能点

ADD = 升级项目中增加的未经调整的功能点

CHGA = 升级项目中改变的功能在改变后所具有的未经调整的功能点

CFP = 额外的转换功能的未经调整的功能点

VAFA = 升级后的应用的调整系数

DEL = 被删除的功能的未经调整的功能点

VAFB = 升级前的应用的调整系数

3.7.3. 应用功能点的计算

应用的功能点有两种计算方法，分别满足如下两个目的：

- ◆ 建立一个应用的初始功能点基线
- ◆ 在应用被升级后重新计算应用的功能点

因为转换功能与应用直接提供的功能无关，因此在应用的功能点计算中被全部省略。

- ◆ 建立功能点基线的计算公式：**AFP = ADD * VAF**

AFP = 应用的功能点

ADD = 安装的功能的 UFPC

VAF = 调整系数

注意该公式中不包含转换功能

◆ 应用升级后功能点的计算公式：

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA$$

其中：

AFP = 应用的功能点

UFPB = 应用升级前的未经调整功能点

ADD = 新增功能的 UFPC

CHGA = 升级后的修改功能的 UFPC

CHGB = 升级前的修改功能的 UFPC

VAFA = 升级后的调整系数

注意该公式中不包含转换功能