

IFPUG 功能点估算

任甲林 高级咨询顾问
麦哲思科技（北京）有限公司

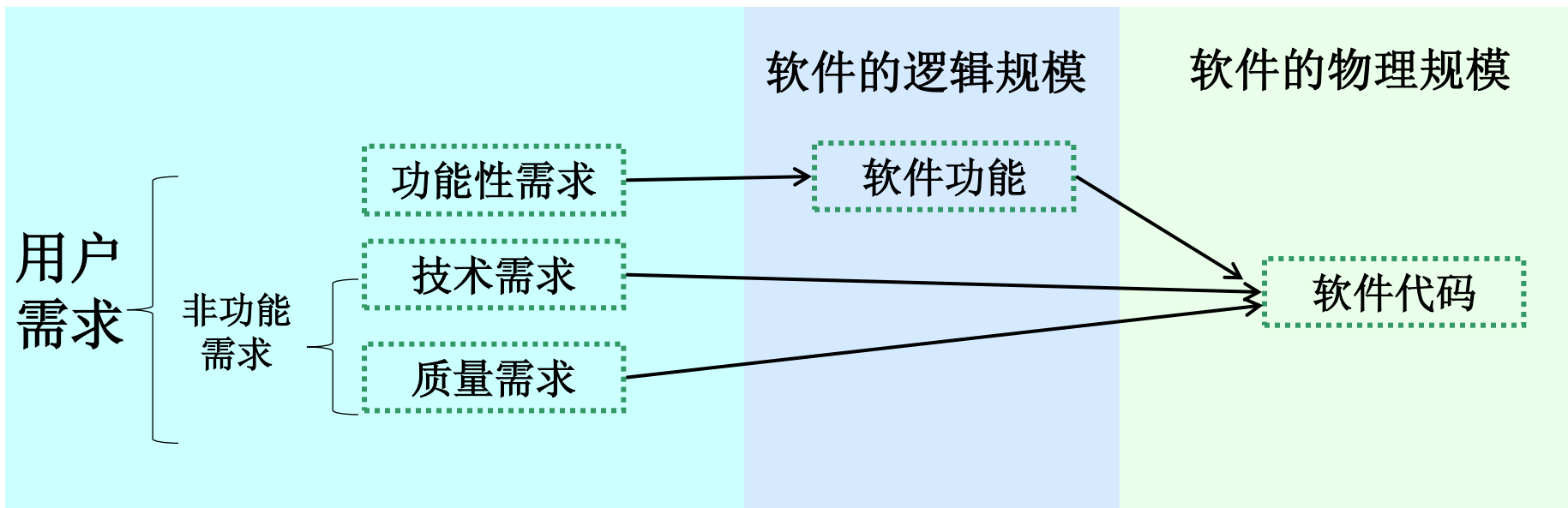
- 功能点估算概述
- 基本思想
- 估算过程
- 数据功能计数
- 事务功能计数
- 简单案例
- 计算调整后的功能点
- 三种类型的规模估算
- IFPUG适用场景
- 小结

为什么要进行规模估计？

- 如果要你估计某人的体重，你会如何估计？
- 如果要你估计某个项目的工作量，你会如何估计？
- 如果有2个项目，项目A测试出了200个bugs，项目B测试出了100个bugs，哪个项目质量更差呢？

- 软件规模的表达形式：
 - 源代码行：SLOC（站在乙方立场，侧重“如何做”）
 - 历史数据或专家意见法(PERT)
 - 单位：KSLOC
 - 用于度量程序开发中的智能工作量
 - 规模与开发语言联系密切，技术的视角反映软件规模
 - 已逐渐失去吸引力，多用于乙方内部核算用
 - 功能点：FP（站在甲方立场，侧重“做什么”）
 - 基于软件的功能数和一组独立的项目因子
 - 基于需求，在项目早期即可得到的信息
 - 与SLOC存在一定的对应关系
 - 优势在于独立于软件开发语言，用业务的视角反映软件规模
 - 国际主流测算方法，适合甲乙双方谈判沟通

软件需求到软件规模的转换



- 软件规模派生于量化的用户功能需求与非功能需求
- 软件的物理规模-代码量不仅取决于软件功能（用户的功能性需求），还取决于用户的技术需求和质量需求（隐含）。

代码行

- 优点
 - 人员不需要专门的培训，简单易行
 - 结果直观
- 缺点
 - 估计结果依赖于个人的估算水平
 - 同一个系统不同的技术平台下估算出规模不同
 - 不同的人员实现同一个系统，实际规模可能相差很大
 - 不同的系统之间无法比较规模的大小
 - 需要明确定义代码行的含义
 - 是否含有注释行
 - 是否含有空行
 - 是逻辑行还是物理行
 - 是否包含开发平台自动生成的代码

功能点

- 优点
 - 与技术平台无关
 - 与实现的人员无关
 - 系统之间的规模具有可比性
 - 客观度量，不依赖于参与估算的人员
 - 通过认证的功能点分析师一般能产生彼此相差不超过10%的计数值
- 缺点
 - 估算与度量人员需要经过专门的培训
 - 估算结果不够直观

- IFPUG
 - IFPUG (International Function Points Users' Group)即国际功能点用户组是一个致力于功能点分析研究的非营利性组织。
 - 1986年IFPUG组织成立，后续的FP指南都是由IFPUG组织所发布的Function Point Counting Practices Manual 系列版本
 - <http://www.ifpug.org>

IFPUG功能点的发展历程

- 1979 IBM的Allan Albrecht提出FP方法
- 1984 正式的FP使用指南发布
- 1988 FP CPM release 2.0
- 1990 FP CPM release 3.0
- 1994 FP CPM release 4.0
- 1999 FP CPM release 4.1
- 2003 加入ISO/IEC标准
- 2004 FP CPM release 4.2
- 2011 FP CPM release 4.3

- 国际功能点用户协会 (International Function Point Users' Group, IFPUG) (www.ifpug.org) 提出的IFPUG功能点分析方法, 相应的国际标准编号是ISO/IEC 20926:2003。
- 荷兰软件度量协会 (Netherlands Software Metrics Association, NESMA) (www.nesma.nl) 提出的荷兰软件功能点分析方法, 相应的国际标准编号是ISO/IEC 24570。
- 英国软件度量协会 (UK Software Metrics Association, UKSMA) (www.ukσμα.co.uk) 提出的Mk II功能点分析方法, 相应的国际标准编号是 ISO/IEC 20968:2002 。
- 通用软件度量国际协会 (COmmon Software Measurement International Consortium , COSMIC) (www.cosmicon.com) 提出的全功能点分析方法, 相应的国际标准编号是ISO/IEC 19761:2003。

- 把用户的业务功能需求分为数据功能需求和处理数据的事务功能需求
- 数据分为应用内部逻辑数据和应用外部的接口数据，事务分为对数据的外部输入、输出和查询
- 分别分析这些组成部分的功能点

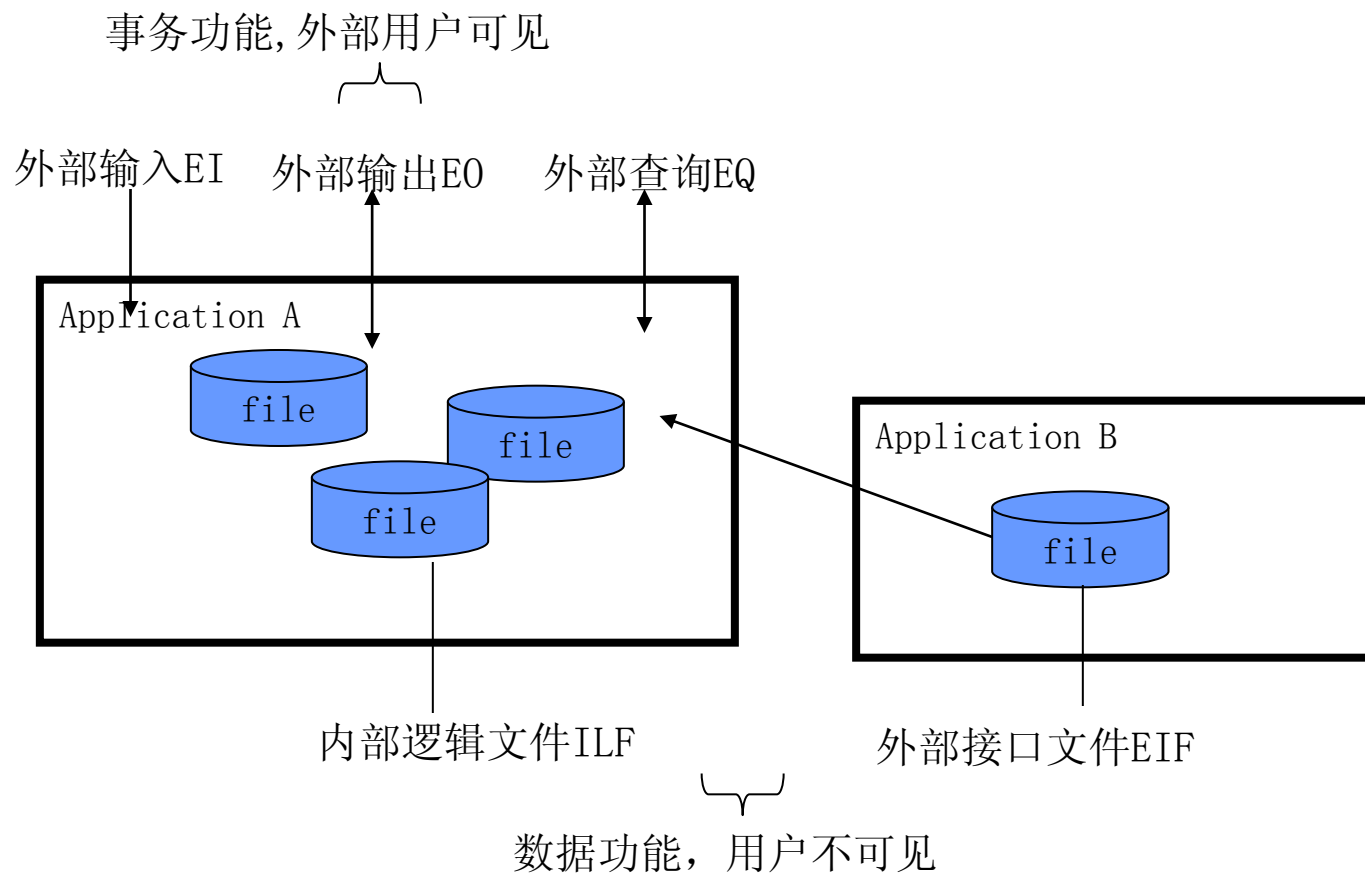
数据功能（Data Function）

- 指提供给用户的以满足应用内部和外部数据需求的功能性
- 分为内部逻辑文件（ILF）和外部接口文件（EIF）
- 复杂性由数据元素类型（DET）和记录元素类型（RET）决定

事务功能（Transaction Function）

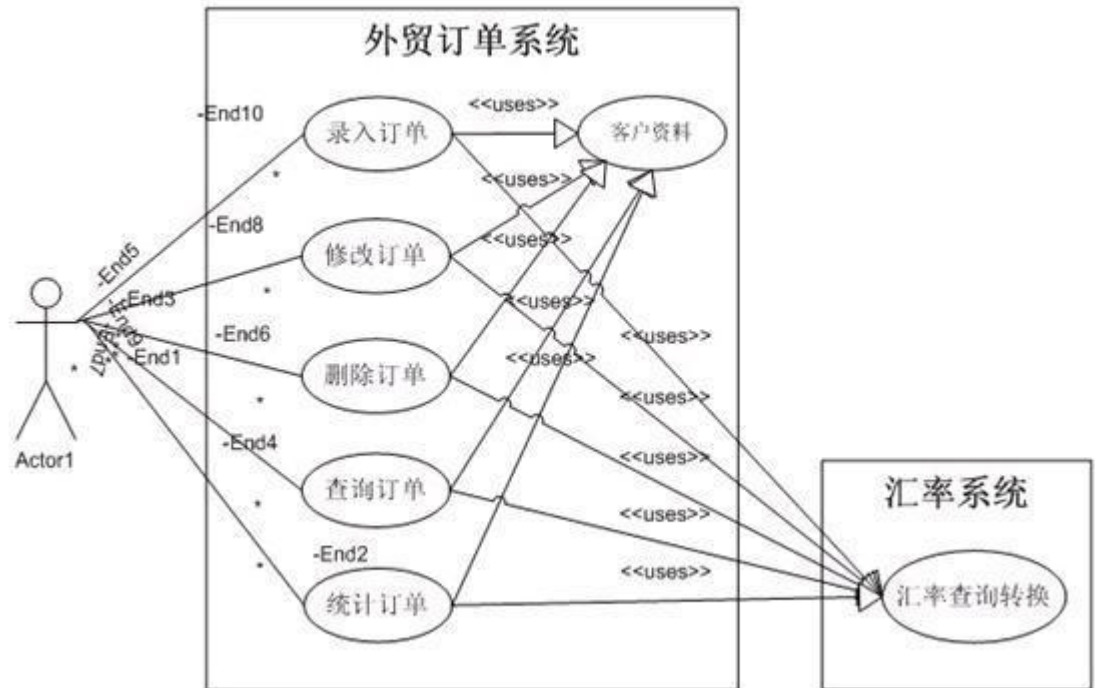
- 指提供给用户的以满足应用数据处理需求的功能性
- 分为外部输入（EI）、外部输出（EO）、外部查询（EQ）
- 复杂性由数据元素类型（DET）和文件引用类型（FTR）决定(file type referenced)

- EI/EO/EQ反映的是运动中的数据(data in motion)，对EI、EO、EQ复杂度的计算可理解为对程序开发复杂度的计算。
- ILF/EIF反映的是存储中的静态数据 (data at rest)，对ILF和EIF复杂度的计算可简单理解为对数据库复杂度的计算。

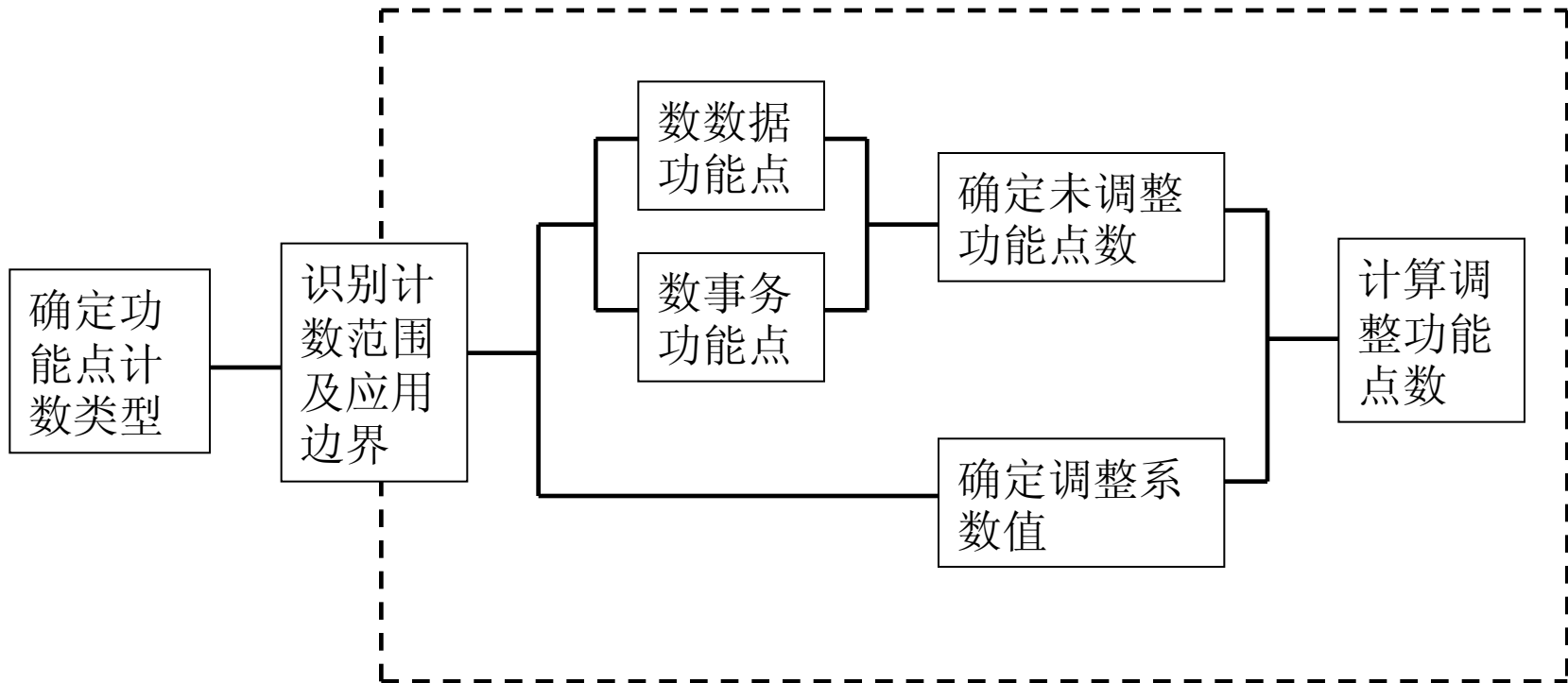


功能点估算五种元素举例

- 以外贸订单系统项目为例：
 - 录入订单、修改订单、删除订单是EI；
 - 查询订单是EQ
 - 统计订单是E0
 - 汇率查询转换系统为EIF
 - 订单和客户是ILF

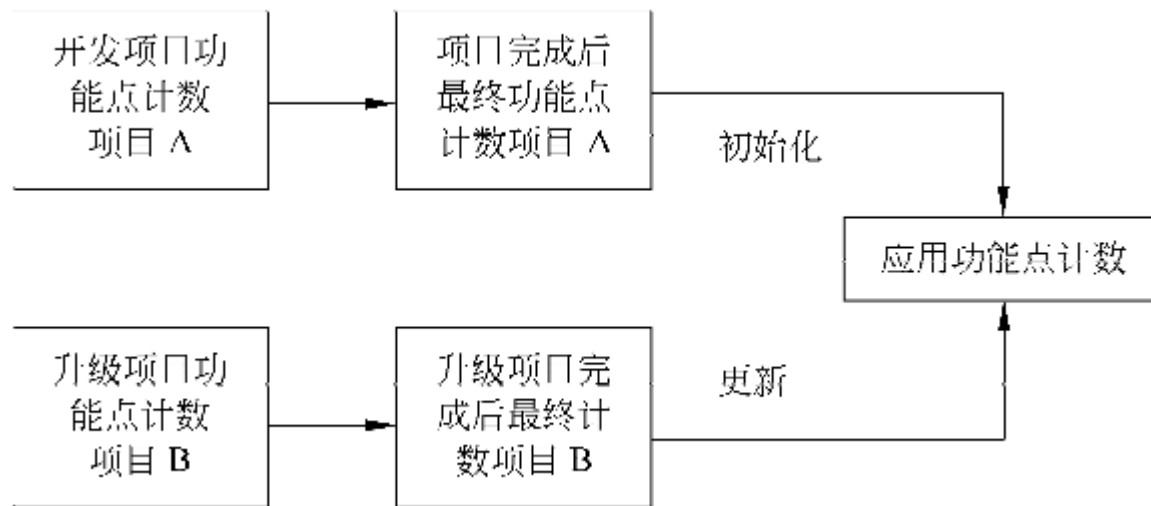


IFPUG功能点计数过程



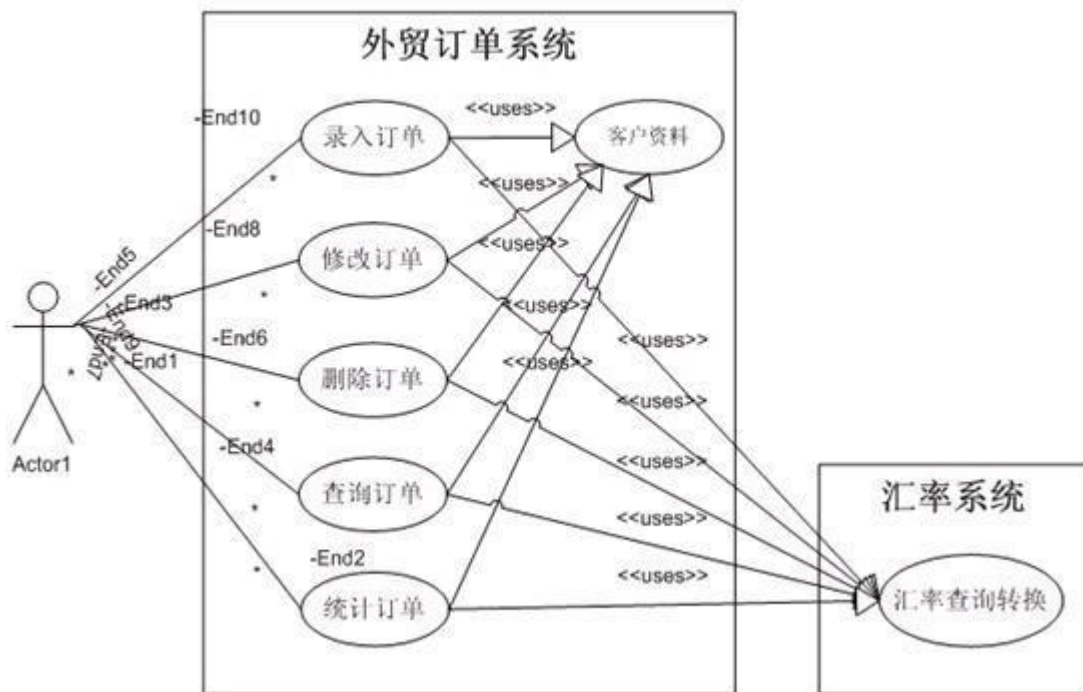
确定功能点计数类型

- 开发项目
- 升级项目
- 应用



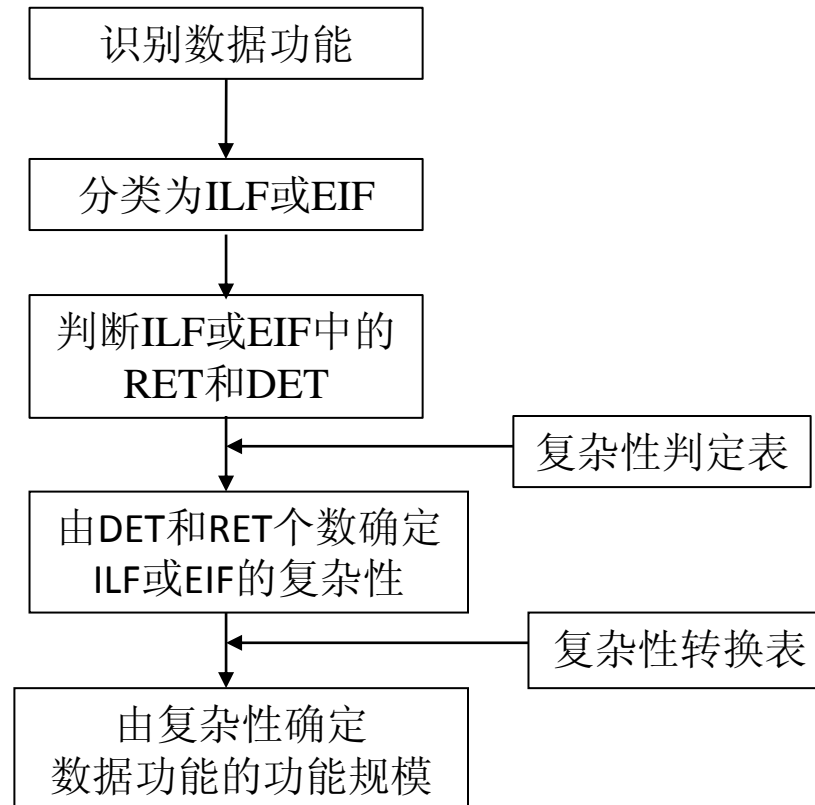
识别项目的范围和边界

- 用例图是以用户角度进行识别项目范围和边界的最好方法，在画用例图时必须明确系统的边界。通过系统的边界，我们可以知道哪些功能要计算功能点，哪些功能点是外部系统负责计算的。
- 应用程序边界的识别规则：不能从技术角度去思考，必须从用户角度来定义；如果项目牵扯到多个系统，那么必须将这多个系统的边界全部描述清楚。



度量数据功能的规模

度量数据功能的步骤



什么样的数据被度量规模？

- 业务数据：
 - 核心用户数据或业务对象。用户可以对其增删改，也是生成报表的来源，包含有关键字段与多个属性，可以由零到无限多个记录。
 - 例如：商品信息、人员信息、部门信息等等
- 引用数据：
 - 是用来维护业务数据的业务规则而存储的数据。一般为管理员用户维护，通常在应用第一次安装时建立，中间间断维护，动态性弱，只有业务规则发生变化时才需要修改，包含关键字段和数量很少的属性，通常至少有一条记录或有限数量的记录。
 - 例如：信用等级类型列表（信用等级、信用额度）

什么样的数据不被度量规模？

- 代码数据：
 - 由开发人员识别出来，用来满足一项或多项非功能用户需求的数据。代码数据的典型属性是代码和描述，代码和它相关的描述具有相同的含义，如果没有描述，代码将不可理解。代码数据是为了使数据容易输入、提升数据的一致性、保证数据的完整性而设计的。代码数据一般由支持人员维护，由关键字段和少数几个属性组成，通常有固定数量的记录。
 - 例如：
 - 国家代码列表（国家编码，国家名称）
 - 地区代码列表（地区编码，地区名称）
 - 代码数据可以互相替代，引用数据不可互相替代。比如机场代码和机场名称可以互相替代，税率代码和税率不能替代。

- 代码数据不能当做数据功能对待，维护代码数据的事务也不能当做事务功能对待。

	分 类	表 现 形 式	说 明	举 例
代 码 数 据	替代数据	代码+描述	包含代码和解释性描述	国家：国家代码，国家名称 颜色：颜色代码，颜色名称
	静态或常量 数据	单一事件	此类代码数据只有一行记录	一个表示特定组织的数据实体：名称， 地址
		静态数据	包含基本静态的数据	化学元素表 功能点复杂性判定表
		缺省值	包含业务对象的缺省值	送票城市的缺省值是出发城市
	有效值数据	有效值	只包含一个属性，提供属性的 有效值列表	城市名称：包含所有此属性的有效值 供选择
		有效值范围	提供属性的有效值范围	允许的电话号码范围 加热温度范围

- 区分一下数据的类型：
 - 人员信息
 - 人员姓名、性别、出生年月、公司名称、联系电话、邮件地址、人员类别1、人员类别2、人员类别3、人员类别4
 - 人员类别
 - 类别编号、分类维度、类别名称
 - 省份
 - 省份编号、省份名称
 - 电话区号
 - 区号、地区

- ① 识别计数范围内所有逻辑相关且用户可识别的数据或控制信息
- ② 排除不被任何应用维护的实体
- ③ 分组实体依赖的相关实体
- ④ 排除代码数据实体
- ⑤ 排除不包含用户要求的属性的实体（如索引文件）
- ⑥ 去掉包含非用户要求的附加属性的关联实体以及仅包含外键的关联实体；把外键属性分组给主实体

何谓控制信息？

- 影响基本处理的数据，它定义了哪些数据、什么时间、如何被处理。



- 如果几个实体总是被同时创建、同时删除，则说明它们应该作为一个数据功能对待。修改数据的事务一般只针对实体组中的一个实体，因此修改事务不能像创建和删除事务那样作为判断数据功能的有效事务。
- 实体独立：即使没有其他实体的支持，实体本身对用户也有意义。如维护员工和公司收养孩子的信息，员工离职了，可以把孩子分配给其他的员工。
- 实体依赖：如果没有其他实体的支持，实体本身对用户业务无意义。如一个员工有多个孩子，当员工离开公司时，公司不再保留孩子的信息。

- 内部逻辑文件是指一组用户可识别的、在应用程序边界内维护的、逻辑相关的数据或控制信息。ILF的主要目的是保存由应用程序的一个或多个基本处理来维护的数据。
- 外部接口文件是指一组在被度量程序所引用，但在其他应用程序中被维护的、以用户角度来识别的、逻辑上相关的数据或控制信息。一个应用程序中的EIF必然是其他应用程序中的ILF。EIF的主要目的是为边界内的应用程序提供一个或多个通过基本处理来引用的一组数据。

- 用户可识别：是指数据或事务需求是被用户和软件开发人员双方共同认可并理解的。
- 维护：通过基本处理增加、修改或删除数据的能力。如对数据的创建、添加、修改、删除、导入和更新等。
- 控制信息：是影响基本处理的数据，它定义了哪些数据、什么时间、如何被处理。如报表定义信息中包含了报表名称、报表显示的字段、报表用到的公式、数据排序方式等。
- 基本处理：是对用户有意义的最小活动单元。每个事务功能都是一个基本处理，功能规模度量时需要把用户功能需求分解或组合为最小的活动单元，使其满足如下的条件：对用户有意义、构成一个完整的事务、自包含、使业务保持持续状态。

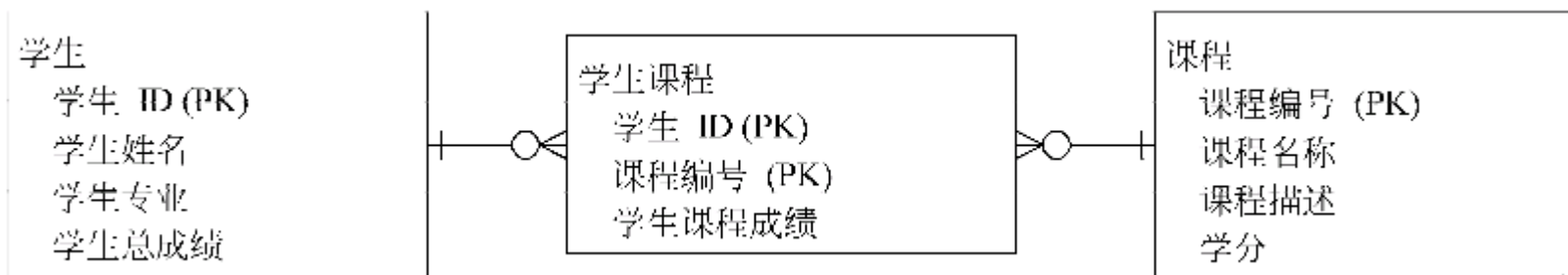
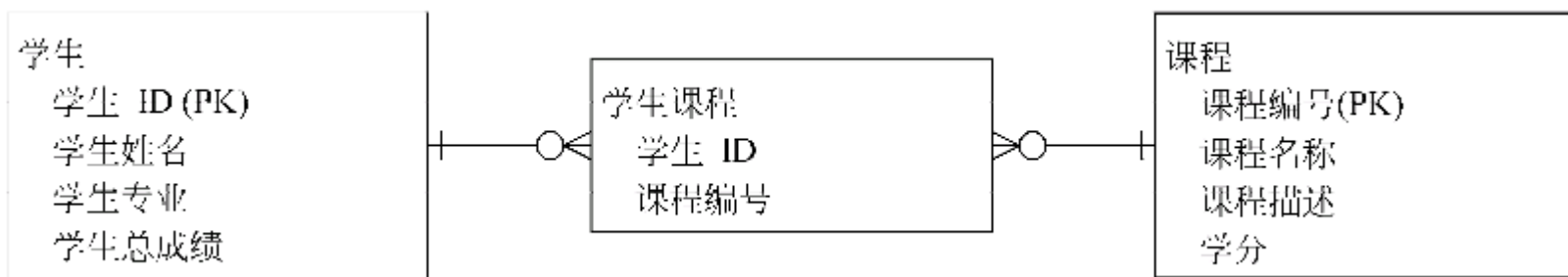
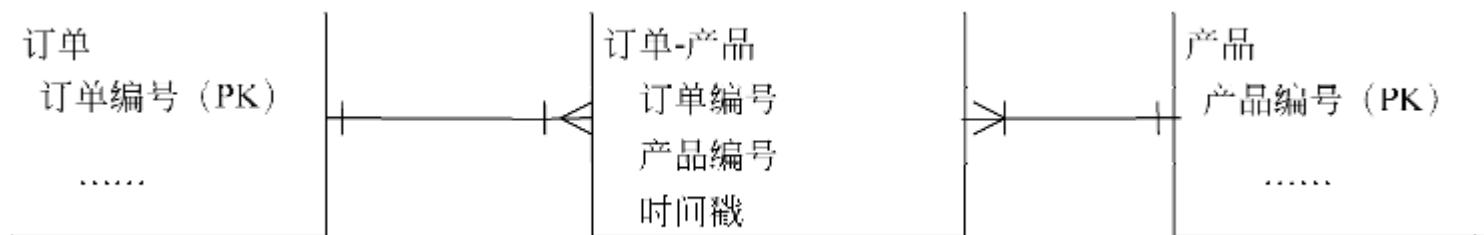
- ILF和EIF的复杂性是取决于RET (Record element type) 和DET (Data element type) 的数量。
- DET是一个以用户角度识别的、唯一的、非重复的属性 (字段) 。
- RET是指一个EIF/ILF中用户可以识别的数据元素类型的子集。

- 把每一个通过基本处理在数据功能中维护或者从数据功能中提取的、用户可识别的、非重复属性作为一个DET。
 - 例如：添加一个外贸订单时需要保存“订单号码、订单日期、地址、邮编”，那么对于ILF订单来说它的DET就是4个。
 - 再如：保存订单时还会保存订单的明细。订单的明细往往作为一个子表进行保存，那么“订单号码”在主表和子表中都同时存在（主外键）。但以用户角度来识别时，存盘操作是一个最小的单位，那么订单号码只能算做一个DET。
- 当两个或多个应用维护或引用相同的数据功能，只计算由被度量应用所使用的那些DET。
- 把每个用来建立和其他数据功能之间关系的属性当做一个DET
- 考察相关属性来确定它们是否组成一个DET或者算作多个DET。组合依赖于应用中基本处理如何使用这些属性。
 - 例如，一个应用程序的两个“Elementary Process”基本处理都需要使用到“地址”的信息，地址信息又可以细分为“国家、城市、街道、邮编”。那么对于其中一个基本处理来说，它将整个地址信息作为一个整体进行处理，只算一个DET；另外一个基本处理使用每个地址的详细信息，那么DET就是4个。

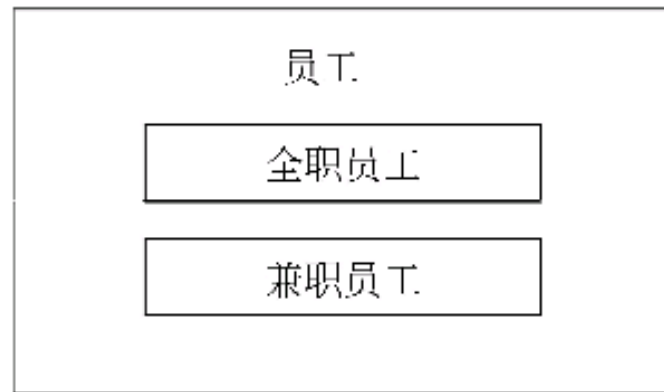
- 每个数据功能当做一个RET。
- 把数据功能中包含一个以上DET的每个逻辑DET子集当做一个RET。逻辑DET有如下情况：
 - 包含非键值属性的关联实体
 - 实体子类型
 - 非强制一对一关系的属性实体
- 例如：在外贸订单系统中添加一个订单时会保存“订单信息、客户的ID、部门的ID”。那么订单系统ILF中的RET为：
 - ① 订单信息（必选的）
 - ② 客户信息（必选的）
 - ③ 部门信息（可选的）因此ILF中RET的个数为3个。

- 关联实体：进一步描述两个其他实体类型之间多对多关系的，包含属性的实体类型
- 实体子类型：一个实体类型的派生实体，它继承了其父实体类型的所有属性和关系，也可以由附加的、独有的属性和关系
- 属性实体：进一步描述另一个实体的一个或多个特征的实体类型。

三种关联实体

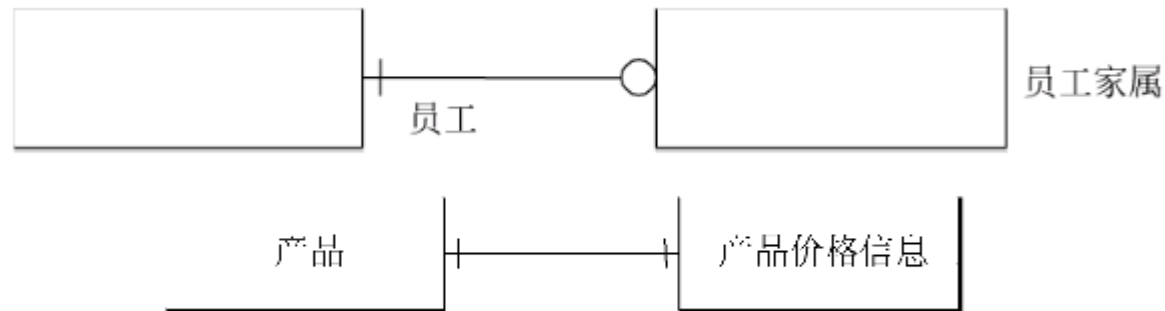


- 实体子类型继承了其父实体类型的所有属性和关系，也有自己独有的属性和关系，每个子类型只有一个父实体，一个父实体可以由多个子类型。子类型不能当做一个独立的数据功能，它们和父实体共同构成一个数据功能。包含多个独有属性的子类型是该数据功能的一个逻辑DET子集。



可选属性实体与强制属性实体

- 属性实体是另一个实体的逻辑扩充，有两种类型的属性实体：可选的和强制的。可选是在添加或创建数据的基本过程中，用户可以选择使用或不使用属性实体；强制是在添加或创建数据的基本过程中，用户必须使用属性实体。在功能规模度量中，属性实体不能脱离主实体而存在，所以主实体及其属性实体共同组成一个数据功能，可选属性实体应作为主实体的一个RET，而强制属性实体不做能作为一个RET对待。



数据复杂性判定表

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

规模换算表

	低	中等	高
ILF	7	10	15
EIF	5	7	10

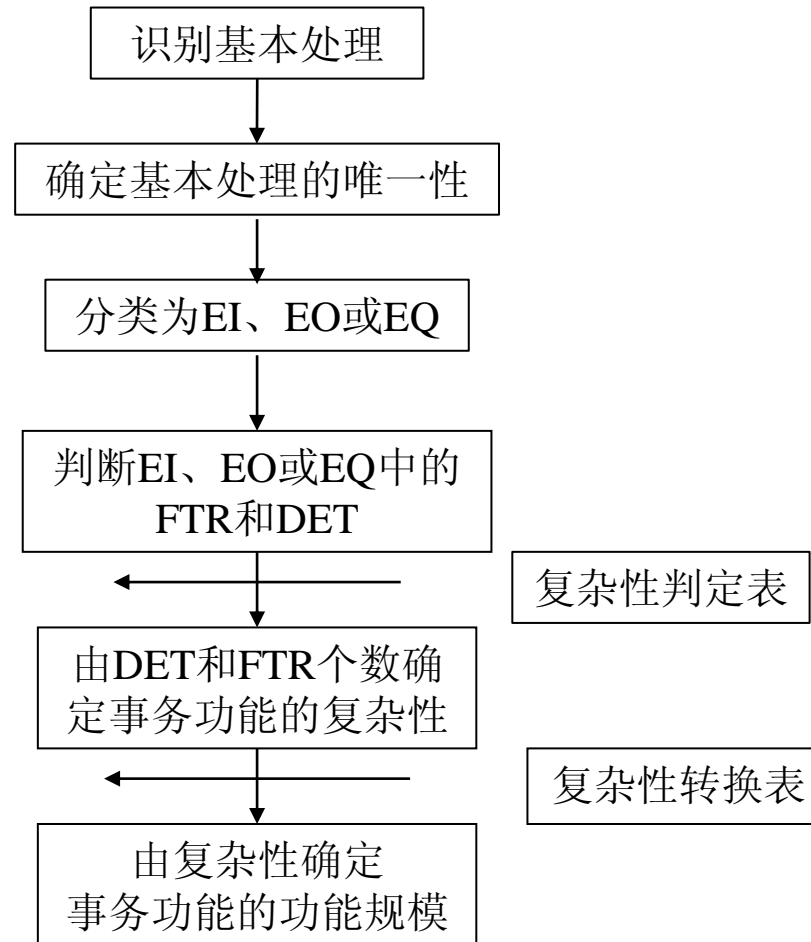
练习:计算下述需求的数据功能的规模

- 某公司图书管理系统有如下需求：
 - (1) 公司员工可以注册成为图书管理系统的用户，并且可以查看或修改自己的信息。
 - (2) 用户注册流程为：用户在注册界面上输入员工编号，系统按照员工编号在人力资源管理系统中查询有没有此员工。若有，则会在注册界面上显示该员工的姓名、性别和所属部门；若无，则会提示用户“该员工编号不存在”。员工编号正确时，用户需要再填写密码设置、重新输入密码、学历、阅读爱好，然后单击“确定按钮”，注册成功。
 - (3) 注册时查询的“员工信息”在人力资源管理系统中包含的属性有员工编号、姓名、性别、出生年月、所属部门、住址和入职时间。

数据功能	分类	DET	RET	DET数	RET数	复杂性	功能规模
用户信息	ILF	员工编号、姓名、性别、所属部门、密码、学历、阅读爱好	用户信息	7	1	低	7
员工信息	EIF	员工编号、姓名、性别、所属部门	员工信息	4	1	低	5
合计							12

度量事务功能的规模

度量事务功能的过程



什么是基本处理？

- 基本处理是功能规模度量过程中一个非常重要的概念，因为只有满足了基本处理的特点，一项软件功能才会成为事务功能。基本处理是由功能用户需求组合或者拆分而来的最小活动单元，它应该同时满足以下4个条件：
 - (1) 对用户有意义。
 - (2) 构成一个完整的事务。
 - (3) 自包含。
 - (4) 使应用的业务保持持续状态。
- 在识别软件功能的基本处理时，经常会遇到非常相似的功能，这些功能是相同的基本处理还是当作不同的基本处理，应该参照以下的判定方式，如果以下的三个条件都满足，则两个功能视为同一基本处理，否则视为两个单独的基本处理。判定基本处理唯一性的三个依据是：
 - (1) **DET**相同。
 - (2) **FTR**相同。
 - (3) 处理逻辑相同。

- 1. 执行验证
 - 验证是常用的一种处理逻辑，如各种登陆界面都需要验证用户信息。
- 2. 执行数学公式和计算
 - 一些应用中需要用到数学公式或进行计算，如要在报表中显示合计和总计数据，那么就需要进行简单的计算。
- 3. 等价值转换
 - 例如，员工年龄用一个表转换为一个年龄范围组。
- 4. 通过特定规则对多组数据进行过滤和选择
 - 查询功能中通常要执行这样的处理逻辑，如要在人力资源管理系统中查出年龄在20~30岁，做研发工作的员工信息列表，则需要根据要求在员工数据组和工作数据组中进行选择。
- 5. 分析条件以判断适用情况
 - 有条件分支的时候需要分析条件以判断后续的操作。如用户登录的时候可能分析用户的安全等级来确定用户可操作的功能，这个基本处理就用到了该处理逻辑。

- 6. 更新一个或多个ILF
 - 该处理逻辑是一种常见的处理逻辑，通常一个应用中的增加、删除、修改功能都会执行这样的处理逻辑。
- 7. 引用一个或多个ILF/EIF
 - 完成一个基本处理时，有时需要引用ILF或EIF，如果显示商品的人民币价格及美元价格，则需要引用汇率表中的汇率。
- 8. 提取数据或控制信息
 - 该处理逻辑很常见，在查询、报表等功能中都会用到。如要查询用户信息，则需要从用户信息数据表中提取数据。
- 9. 通过转换现有数据来创建衍生数据
 - 在E0和EQ的定义中都提到了衍生数据，衍生数据是指数据不仅仅是从数据功能中直接提取的，还是对现有数据进行处理的结果。如用户的注册号是系统自动把用户的姓名和手机号联结成一个数据来显示的，这个注册号就是一个衍生数据。

- 10. 改变应用行为
 - 通常要通过控制信息来改变应用行为。如人力资源管理系统中每月底自动通知人事经理合同即将到期的员工，改为每月15号和月底个通知一次，那么自动通知基本处理的行为就发生了改变，每月通知一次改为每月通知两次。
- 11. 向边界外部呈现信息
 - 向边界外部呈现信息是EQ和EO的主要目的，所以任何EQ和EO都会执行这个处理逻辑。如查询和报表都会向边界外部呈现信息。
- 12. 接受进入应用边界内的数据或控制信息
 - 从EI的定义可以看出，每个EI都会执行这样的处理逻辑。如用户注册时，应用会接收用户输入的用户信息。
- 13. 对数据进行分类和排序
 - 这种形式的处理逻辑不会影响基本处理的识别及唯一性。如商品信息可以按照价格，也可以按照销量来排序显示，但都是同一个基本处理。

练习：识别基本处理

- A网站有如下需求：
 - (1) 用户可以通过注册功能注册成为网站会员，注册时需要添加用户的用户名、密码、真实姓名、学历、电话、邮箱地址和家庭住址，用户名和邮箱都是唯一的，如果和其他用户重复，则应会做出相应提示。
 - (2) 用户有两种登录方式，通过用户名和密码或者邮箱地址和密码登录网站，登录后网页上会显示该用户的用户名。登录错误会有相应提示。
 - (3) 用户可以通过查看个人信息功能来查看自己的注册信息。
 - (4) 用户可以通过修改个人信息功能来修改自己的相关信息，修改时页面显示用户已注册信息，用户可选择需要修改的字段进行修改，修改完成后，提示用户个人信息已修改。
 - (5) 如果用户有没有查看的消息，则网站会弹出提示框来提示，显示“您有n条未读消息”，n为消息数。

需求编号	基本处理	唯一性	备注
1	用户注册	唯一	
2	用户名登录	唯一	两种登录方式包含不同的DET
	邮箱地址登录	唯一	
3	查看个人信息	唯一	
4	个人信息显示	不唯一	和查看个人信息包含相同的DET、FTR和处理逻辑
	修改个人信息	唯一	
5	未读消息提示	唯一	

- 是处理来自边界外的数据或控制信息的基本处理，它的主要目的是维护一个或多个ILF或者改变应用行为。
 - 数据可能来自于数据输入屏幕、电子输入或其它应用程序。
 - 数据可以是控制信息或业务信息。如果数据是业务信息，它用于维护一个或多个内部逻辑文件。如果数据是控制信息，它不必更新内部逻辑文件。
- 边界：是软件和其用户之间概念上的分界。

- 把数据或者控制信息发送到边界外部的基本处理，它的主要目的是通过数据或者控制信息的提取而把信息呈现给用户。其处理逻辑：
 - 不包含数据公式或计算
 - 不会创建衍生数据
 - 不能维护ILF
 - 不能改变应用行为

- 是发送数据或控制信息到边界外的基本处理。其主要目的是通过处理逻辑呈现信息给用户，不仅仅是在应用中提取数据或控制信息。E0的处理逻辑必须满足以下4项中的一项：
 - 包含数据公式或计算
 - 创建衍生数据
 - 维护一个或多个ILF
 - 改变应用行为

EI、E0、EQ的比较

- EI、E0、EQ的比较

- EI是处理来自应用程序边界外部的一组数据输入，它的主要目的是维护一个或多个ILF，以及/或者更改系统的行为。
- EQ是向应用程序边界外发送数据基本处理的过程。其主要目的是从ILF或EIF中通过恢复数据信息来向用户呈现。该处理逻辑**不包括任何数学公式或计算方法**，也不会生成任何派生数据。EQ不会维护任何一个ILF，也不会改变应用程序的系统行为。
- E0是输送数据到应用程序边界外部的过程。它的主要目的是通过逻辑处理过程向用户呈现信息。该处理过程**必须包含至少一个数学公式或计算方法**，或生成派生数据。一个E0也可以维护一个或多个ILF，并/或改变系统行为。

E0和EQ的共同点是，其主要目的都是通过基本操作过程展现数据给用户。

目的	EI	E0	EQ
改变应用程序的属性或行为	主要目的	次要目的	不允许
维护一个或多个ILF	主要目的	次要目的	不允许
显示信息给用户	次要目的	主要目的	主要目的

事务功能执行的处理逻辑

处理逻辑	可执行情况(可√; 否×)		
	EI	EO	EQ
1. 执行验证	√	√	√
2. 执行数学公式和计算	√	√	×
3. 等价值转换	√	√	√
4. 通过特定规则对多组数据进行过滤和选择	√	√	√
5. 分析条件以判断适用情况	√	√	√
6. 更新一个或多个ILF	√	√	×
7. 引用一个或多个ILF/EIF	√	√	√
8. 提取数据或控制信息	√	√	√
9. 通过转换现有数据来创建衍生数据	√	√	×
10. 改变应用行为	√	√	×
11. 向边界外部呈现信息	√	√	√
12. 接收进入应用边界内的数据或控制信息	√	√	√
13. 对数据进行分类和排序	√	√	√

练习：区分EQ与EO

- 列举出所有的省份
- 列举出所有的人员
- 统计出各个省份的人员
- 统计出部门的工资
- 打印出每个人的家庭信息
- 打印出每个人年度的合计工资
- 打印出每个月在手机话费并统计出全年总的话费

- (1) 把事务功能中穿过（进入或离开）边界的每一个用户可识别的、非重复属性都当作一个DET。
- (2) 把事务功能发送应用响应消息的能力仅当做一个DET，无论功能中有多少个消息。
- (3) 把事务功能激发动作的能力当作一个DET，即使有多种方式激发同一个动作。
- (4) 以下情形不能当作DET：
 - ① 文字信息，如报表标题、界面或面板标识、列名以及属性名称。
 - ② 应用生成的标记，如日期和时间等。
 - ③ 页面变量、页码和位置信息，如“从37到54行，共211行”。
 - ④ 导航助手，如在列表中“上一个”、“下一个”、“第一个”、“最后一个”以及和它们功能相同的图形来导航的能力。
 - ⑤ 在边界内由事务功能生成并保存到ILF中，但没有穿过边界的属性。
 - ⑥ 事务处理中从ILF或EIF中提取或引用的，但没有穿过边界的属性。

- 外贸订单系统中，订单的金额是被单价和数量自动计算的，那么金额是没有通过系统边界输入的，因此在EI操作中就不应该算做一个DET。
- 在网站注册用户信息时，由于输入错误系统会显示提示信息，那么这些提示信息应该被逐个计算为一个DET。
- 当EI操作完成时系统提示并显示出来的信息，应该被计算为一个DET。
- 报表中的每个字段都是一个DET。
- 在报表中起到解释或备注作用的文字信息，不管是一个字、一个词或一段话，都当作一个DET。
- 某种编号或日期，即使它被物理存储在不同字段中，但从用户角度看是一个整体的信息，因此被算作一个DET。
- 在饼图中百分比和分类算作不同的DET。
- 在报表查询时，输入的字段在报表上也有显示，那么将算作同一个DET。
- 用户查询一个列表时被拒绝，那么拒绝的提示信息就算为一个DET。
- 在公司发工资的时候，员工对应的状态信息被更新，但这个状态信息的更新是没有通过系统边界输入的，因此也不能算做一个DET。

- 引用文件类型（FTR）是由事务功能读取或维护的数据功能。数据功能包含ILF和EIF，ILF能被事务功能读取或维护，EIF只能被事务功能读取，所以一个事务功能的FTR包含其读取或维护的ILF，还包含其读取的EIF。
- 引用文件类型识别规则如下：
 - 把事务功能读取或维护的每个数据功能当作一个FTR。
 - 根据FTR的识别规则，对于EQ来说，因为它不可以维护ILF，所以EQ的FTR是其读取的ILF或EIF。EI和EO的FTR是其读取的ILF或EIF，及其维护的ILF，对于既维护又读取的ILF，只能当作一个FTR。

案例：识别FTR

- 在图书管理系统中添加用户信息时，需要访问人力资源管理系统中的“员工信息”来判断用户是否是本单位员工，非本单位员工不允许使用图书管理系统
- EI：添加用户信息
- FTR：
 - EIF:员工信息
 - ILF:用户信息

案例：识别FTR

- 在图书管理系统中查询借阅单时，显示图书ID、图书名称、作者、借阅用户ID、借阅用户名、用户电话、借阅日期和到期日期
- EQ: 查询借阅单
- FTR:
 - ILF:
 - 图书信息
 - 用户信息
 - 借阅信息

事务功能的复杂性判断

EI复杂度计算矩阵

	1~4个 DET	5~15个 DET	多于15个 DET
0~1个FTR	低	低	中等
2个FTR	低	中等	高
大于2个FTR	中等	高	高

EO和EQ复杂度计算矩阵

	1~5个 DET	6~19个 DET	多于19个 DET
0~1个FTR	低	低	中等
2~3个FTR	低	中等	高
多于4个FTR	中等	高	高

事务功能复杂性转换表

- EI、EO、EQ、ILF和EIF技术复杂度对应的功能点如下表所示：

	低	一般	高
EI	3	4	6
EO	4	5	7
EQ	3	4	6

案例：计算事务功能的规模

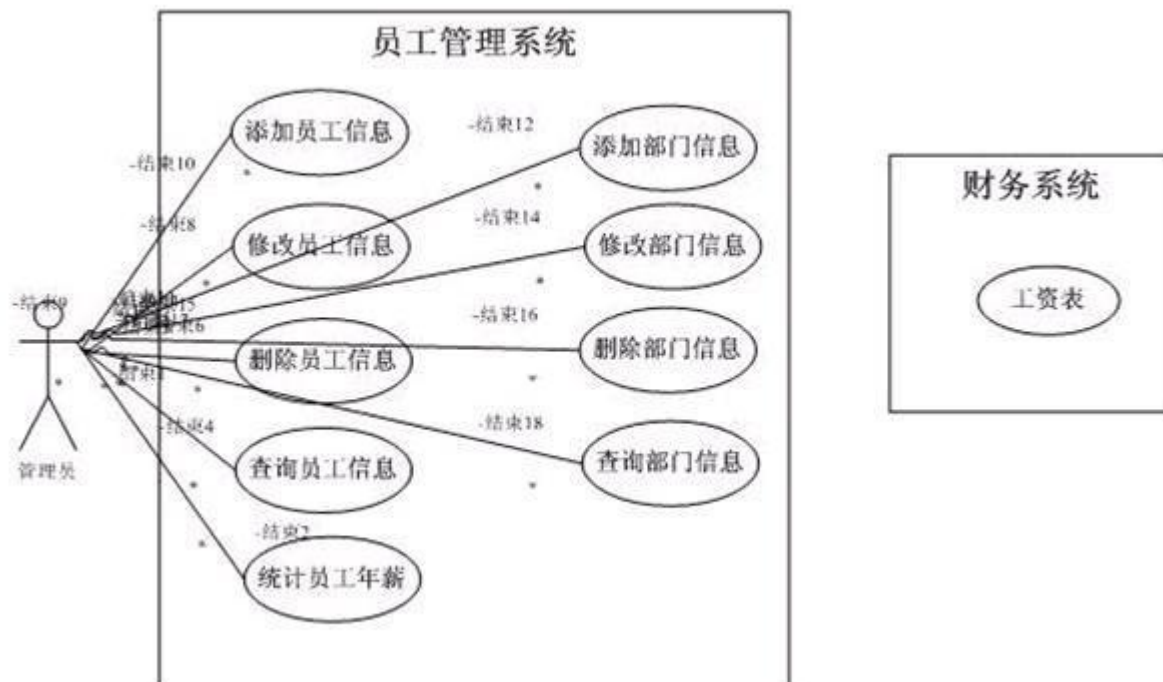
- A网站有如下需求：
 - (1) 用户可以通过注册功能注册成为网站会员，注册时需要添加用户的用户名、密码、真实姓名、学历、电话、邮箱地址和家庭住址，用户名和邮箱都是唯一的，如果和其他用户重复，则应会做出相应提示。
 - (2) 用户有两种登录方式，通过用户名和密码或者邮箱地址和密码登录网站，登录后网页上会显示该用户的用户名。登录错误会有相应提示。
 - (3) 用户可以通过查看个人信息功能来查看自己的注册信息。
 - (4) 用户可以通过修改个人信息功能来修改自己的相关信息，修改时页面显示用户已注册信息，用户可选择需要修改的字段进行修改，修改完成后，提示用户个人信息已修改。
 - (5) 如果用户有没有查看的消息，则网站会弹出提示框来提示，显示“您有n条未读消息”，n为消息数。

需求编号	基本处理	分类	DET数	FTR数	复杂性	功能规模
1	用户注册	EI	9	1	低	3
2	用户名登录	EQ	4	1	低	3
	邮箱地址登录	EQ	5	1	低	3
3	查看个人信息	EQ	8	1	低	3
4	修改个人信息	EI	9	1	低	3
5	未读消息提示	EO	1	1	低	4
合计						19

IFPUG估算案例

举例

- 在员工管理系统中添加一个员工资料，会使用到员工的一般信息、教育情况、工作经历和家属信息。员工隶属于某个部门，在本系统中会有一个对部门进行维护的功能。员工的工资则由另外一个财务系统提供。其用例图如图：



数据模型如下

- 假设员工基本信息如下所示：

- 员工ID
- 员工名称
- 性别
- 生日
- 婚否
- 所属部门ID（
标签控件）
-

- 受教育的时间
- 学校名称
- 所学专业
- 工作时间
- 工作单位
- 工作部门
- 工作职务
- 亲属的姓名
- 之间关系
- 亲属年龄
- 工作单位

- 假设部门信息如下所示：

- 部门ID
- 部门名称

- 假设工资表信息如下所示：

- 员工ID
- 员工姓名
- 金额
- 单位

ILF和EIF的功能点数

ILF内部逻辑文件	RET	DET个数	复杂度	未调整的FP个数
员工信息	员工基本信息、受教育情况、工作经历、亲属信息，共4个。	18	低	7
部门信息	部门基本信息，共1个。	2	低	7

EIF外部接口文件	RET	DET个数	复杂度	未调整的FP个数
工资表	员工基本信息、工资信息，共2个。	4	低	5

合计：19个

EI、EQ和EO的功能点数

EI	FTR	DET个数	复杂度	未调整的FP个数
添加员工信息	员工、部门、工资表	18个 员工信息的两个标签控件内容不是用户输入的，因此不算，共16个；部门信息与员工信息中的部门字段重复，因此一个都不算；工资表中的员工ID和名称不能重复，因此只能算金额和单位，所以共2个。	高	6
修改员工信息	员工、部门、工资表	18个 同上	高	6
删除员工信息	员工、部门、工资表	1个 员工ID	中等	4
添加部门信息	部门	1个 一个标签控件的内容不是用户输入的，因此不算	低	3
修改部门信息	部门	1个 一个标签控件的内容不是用户输入的，因此不算	低	3
删除部门信息	部门	1个 部门ID	低	3

合计：⁶⁵25个

EQ, E0的功能点数

EQ	FTR	DET个数	复杂度	未调整的FP个数
查询员工信息	员工、部门、工资表	20	高	6
查询部门信息	部门	2	低	3

合计：9个

E0	FTR	DET个数	复杂度	未调整的FP个数
统计员工年薪	员工、工资表	员工ID、员工名称、年份、年薪、单位共5个	低	4

合计：4个

总计：19+25+9+4=57

规模如何转化为工作量

- 将规模转化为工作量时，要考虑生产率。生产率与具体项目的类型（如C/S、基于Web等）有关。
- 假定开发C/S项目的生产率是0.9 FP/人天，FP是155，则

$$\text{Effort} = 155 / 0.9 = 172 \text{人天}$$

计算调整因子

- 用功能点估算法计算软件项目功能点时会用到调整因子（或称调整系数）。功能点的调整系数是通过通用系统特性及其影响程度来评定的，对每个常规系统特性的评估由其影响程度（DI）而定，分为0－5级：
 - 0 毫无影响
 - 1 偶然影响
 - 2 适度影响
 - 3 一般影响
 - 4 重要影响
 - 5 强烈影响
- 然后依次对以下14个系统常规特性进行打分，并带入以下计算公式算出功能点的调整因子。
$$\text{Value Adjustment Factor} = (\text{sum of (DI)} * 0.01) + 0.65$$

1. 数据通讯

- 数据通讯指的是应用程序直接与处理器通讯的程度。通常我们都是通过某种通讯手段来实现在一个应用中所使用的数据或者控制信息。
 - 连接到本地控制器上的终端被认为是通讯设施；
 - 协议则指两个系统或设备之间进行通讯时使用的一种约定。所有的数据通讯链接都需要某种协议。

0	应用程序是单纯的批处理或者PC stand-alone
1	应用程序是一种批处理过程，但是包含远程数据的录入或远程打印
2	应用程序是一种批处理过程，但是包含远程数据的录入和远程打印
3	应用程序包括在线数据收集或者包括批处理或查询系统的远程处理的前端应用
4	应用程序不单只是前端应用，但是仅支持一种远程处理通讯协议
5	应用程序不单只是前端应用，还支持多于一种的远程处理通讯协议

2. 分布式数据处理

- 分布式数据处理是应用在内部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。

0	应用程序不支持组件之间的数据传输和处理功能
1	应用程序为用户可能进行的处理准备数据（例如使用电子表格或者数据库等）
2	应用程序所准备的数据是为了在系统另外一个组件上传输和处理，并非为终端用户所处理。
3	分布式处理和数据传输是在线的，并且是单向的
4	分布式处理和数据传输是在线的，并且是双向的
5	由系统中最恰当的组件动态地执行处理功能

3. 性能

- 性能是吞吐量、处理时间等指标对开发的影响。用户所提出的性能要求将直接影响到系统的设计、实施、安装和支持。

0	用户没有提出性能方面的要求
1	用户提出了性能和设计方面的要求，但不需要采取特定措施
2	响应时间和吞吐量在系统峰值时是关键，但是不需要采取相应的CPU使用方面的特殊设计。处理的最后期限是在下一个工作日。
3	在任何时候响应时间和吞吐量都是关键的，但是不需要采取相应的CPU使用方面的特殊设计。处理的完成期限比较严格。
4	除了上面一项的要求外，由于对需求的要求比较严格，在设计阶段就要进行性能分析。
5	除了上面一项的要求之外，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。

4. 大业务量配置

- 大业务量配置是指计算机资源对应用开发的影响程度。大业务量的运行配置对设计有特殊要求，是必须考虑的一个系统特性。

0	没有提出明确的运行方面的限制
1	有运行方面的限制，但是不需要采取特别的措施以满足运行限制
2	提出了一些安全和时间方面的限制
3	应用程序的某些部分对处理器有特定的要求
4	提出的运行限制对应用的中央处理器或者专用处理器有特殊的要求
5	除上面一项之外，还对应用的分布式组件提出了限制

5. 事务处理率

- 事务处理率是业务交易处理速度对系统的设计、实施、安装和支持等的影响。

0	预计不会出现周期性的高峰事务处理期
1	预计会有周期性的高峰事务处理期 (例如: 每月、每季、每年)
2	预计每周都会出现高峰事务处理期
3	预计每天都会出现高峰事务处理期
4	用户在应用程序需求或者服务级别协议中对事务率要求很高, 因此必须在设计阶段进行性能分析。
5	用户在应用程序需求或者服务级别协议中对事务率要求很高, 因此必须进行性能分析并在设计、开发和安装阶段中使用到性能分析工具。

6. 在线数据输入

- 在线数据输入是指数据通过交互的方式输入系统的程度。系统中包括在线数据输入和控制信息功能。

0	所有事务都是批处理的
1	1%~7%的事务是以交互式的方式进行数据录入
2	8%~15%的事务是以交互式的方式进行数据录入
3	16%~23%的事务是以交互式的方式进行数据录入
4	24%~30%的事务是以交互式的方式进行数据录入
5	30%以上的事务是以交互式的方式进行数据录入

7. 最终用户效率

- 最终用户效率是指对应用的人文因素及使用的便捷程度等的考虑程度。
- 如下功能设计是针对最终用户效率的：
 - 页面导航
 - 菜单
 - 在线帮助或文档
 - 光标自动跳转
 - 可以滚动
 - 在线远程打印
 - 预定义的功能键
 - 在线做批量提交任务
 - 光标可以选取界面上的数据
 - 用户使用大量反白显示、重点显示、下划线或其他标识
 - 在线copy用户文档
 - 鼠标拖动功能
 - 弹出窗体
 - 使用最少的界面完成某种商业功能
 - 双语言支持（如果选择了这个就算4项）
 - 语言支持（如果选择了这个就算6项）

0	以上的一个都不包括
1	包括以上的1~3个
2	包括以上的4~5个
3	包括以上的6个或以上，但是没有用户对于效率的要求
4	包括以上的6个或以上，对用户使用效率有较高要求，因而必须考虑用户方面的设计（例如，最少击键次数、尽可能提供默认值、模版的使用）
5	包括以上的6个或以上，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

8. 在线更新

- 在线更新是指内部逻辑文件ILF被在线更新的程度。应用系统提供在线更新内部逻辑文件的功能。

0	没有在线更新
1	包含1~3 个控制文件的在线更新。更新的流量低，恢复容易。
2	包含对4 个以上控制文件的在线更新。更新的流量低，恢复容易。
3	包含对主要ILF 的更新。
4	除了3 之外，在设计和实施中要考虑对数据丢失的防范。
5	除了4 之外，大量的数据恢复工作要考虑成本因素，同时包含了高度自动化的恢复流程。

9. 复杂处理

- 复杂处理描述了逻辑处理对应用开发的影响程度。它包含以下要素：
 - 敏感控制（例如特殊的审核过程）和/或程序特定的安全处理
 - 大量的逻辑处理
 - 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（例如，由TP中断、数据值缺少和验证失败导致的ATM事务）
 - 多种可能的输入/输出造成的复杂处理

0	上面一个都不满足
1	只满足一个
2	只满足两个
3	满足三个
4	满足四个
5	都满足

10. 可复用性

- 应用系统中的应用和代码经过特殊设计、开发和支持，可以在其他应用系统中复用。

0	没有可复用的代码
1	代码在应用之内复用
2	应用中被其他用户复用的部分不足10%
3	应用中被不止一个用户使用的部分超过10%
4	应用遵从一种易于复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	应用按照一种易于复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。

11. 易安装性

- 易安装性指应用系统的转换和安装容易度对开发的影响程度。系统测试阶段提供了转换和安装计划/转换工具。

0	用户对安装没有特定的要求
1	用户对安装没有特定的要求，但有特定的安装环境要求
2	用户提出了安装和转化的要求，转化/安装指南被经过测试提供给用户。但是转化的影响对该应用不重要。
3	用户提出了安装和转化的要求，转化/安装指南被经过测试提供给用户。转化的影响对该应用来说是重要的。
4	除了2的要求之外，需要提供经过测试的自动化的安装和转化工具。
5	除了3的要求之外，需要提供经过测试的自动化的安装和转化工具。

12. 易操作性

- 易操作性指的是应用对运行的影响程度，如有效启动、备份和恢复规程的影响。
- 易操作性是应用提供的一种特性，它最小化了手工操作的要求。

0	用户没有指定除正常备份程序外的其它特定操作
1	提供高效的启动、备份和恢复进程，但需要人手操作
2	提供高效的启动、备份和恢复进程，不需要人手操作（当作两项计算）
3	应用程序对磁带的需求最小化
4	应用程序对硬拷贝处理的需求最小化
5	程序设计成无人操作模式。无人操作模式的意思是除了启动和关闭之外，不需要对系统进行操作。程序的其中一个功能就是错误自动恢复。

13. 多场地

- 多场地指应用系统经特殊设计、开发可以在多个组织、多个地点应用的程度。

0	用户需求不含多场地和组织的要求
1	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	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	考虑了多场地的要求，同时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，提供了经过测试的多场地的文档和支持计划
5	在3 的要求之上，提供了经过测试的多场地的文档和支持计划

14. 支持变更

- 支持变更是指应用在设计上考虑支持处理逻辑和数据结构变化的程度。
- 可以具有如下的特性：
 - 提供可以处理简单要求的弹性查询和报告功能，如对于一个ILF进行与（或）逻辑
 - 提供可以处理一般复杂度要求的弹性查询和报告功能，如对于多于一个的ILF进行与（或）逻辑（当作两项计算）
 - 提供可以处理复杂要求的弹性查询和报告功能，如对于一个或多个ILF进行与（或）逻辑的组合（当作三项计算）
 - 业务控制数据被保存到用户通过在线交互进程维护的表中，但变更只会在第二个工作日生效
 - 业务控制数据被保存到用户通过在线交互进程维护的表中，且变更即时生效

0	一个都不满足
1	合计满足一个
2	合计满足二个
3	合计满足三个
4	合计满足四个
5	合计满足五个

计算调整后的功能点个数

- 功能点的原始计算公式: $FP\ Count = UFP * VAF$

三类项目的规模估算

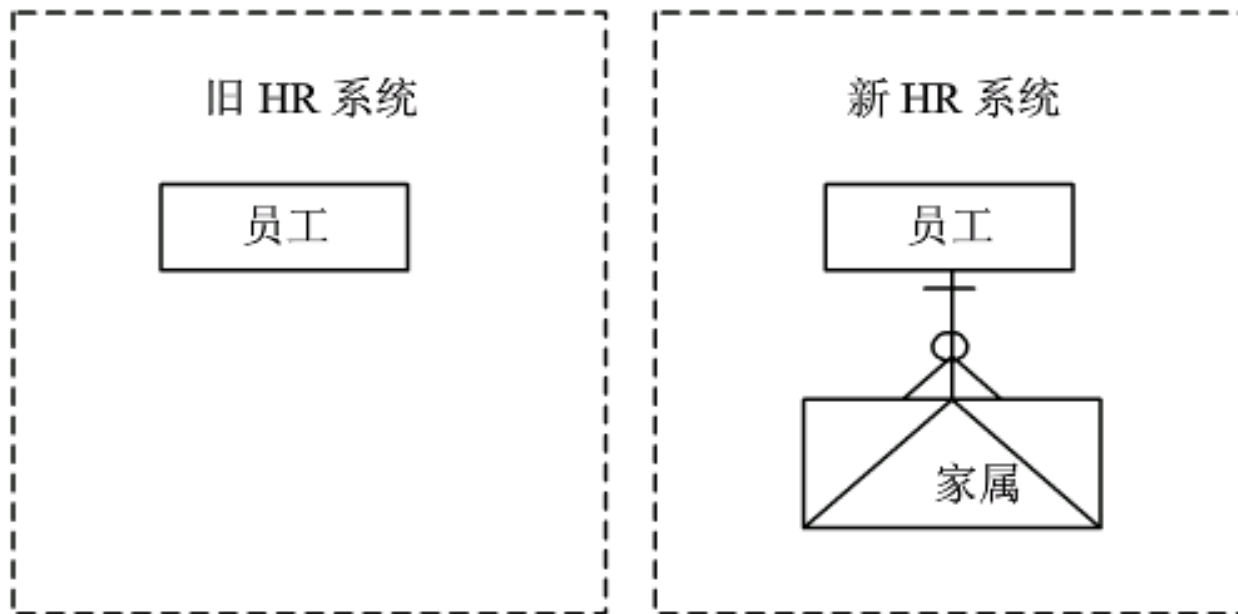
- 开发项目功能点计数是对第一个发布版本提供给用户的功能进行的度量，得到开发项目的功能规模。
- 开发项目功能规模=开发项目交付给用户的功能规模+转换功能的功能规模
- $DFP = ADD + CFP$

- 升级项目功能点计数是对已安装应用的用户功能的添加、修改和删除的度量，度量结果为升级项目规模。
- 升级项目功能规模=升级项目中新增功能的功能规模+升级项目中修改功能的功能规模+转换功能的功能规模+升级项目删除功能的功能规模
- $EFP = ADD + CHGA + CFP + DEL$

- 应用功能点计数是对一个应用提供给用户的当前功能的度量，度量结果为应用的功能规模。
- 应用功能规模=应用被度量时提供给用户的功能规模
- 应用的功能规模不包含转换功能的功能规模，转换功能是一次性的功能，只有在开发项目和升级项目中可能会有，在应用中不包含转换功能。

- 转换功能是用于完成用户指定的数据转换需求的事务功能或数据功能。数据转换需求是在开发项目或升级项目中需要把原始应用中的一些数据转换到新开发的项目或升级项目中去，以便维护新开发项目或升级项目中的ILF。
- 转换功能的度量方法和其他用户功能的度量方法相同，依据的规则也相同。。
- 举例：
 - HR系统进行了升级，用户要求把旧系统中的员工信息（包括员工编号、姓名、性别、身份证号、所属部门、学历、手机号码和工资标准）转换到新系统中。新系统中的员工信息包括员工基本信息（员工编号、姓名、性别、身份证号、所属部门、学历、手机号码、工资标准和家属数目）及员工家属信息（家属姓名、家属性别、与员工关系和生日）。旧系统中没有提供维护员工家属信息的功能，员工信息转换到新系统后，可以创建家属信息。

HR系统数据转换的数据模型



实体类型



属性实体类型



可选一对多关系

HR系统转换部分的功能度量

数据功能	分类	DET数	RET数	复杂性	功能规模 (FP)
新员工信息	ILF	13	2	低	7

基本处理	分类	DET数	FTR数	复杂性	功能规模 (FP)
数据转换	EI	8	1	低	3

- 首先收集与升级有关的文档，包括升级的需求等；
- 然后确定计数的范围，包括增加、修改和删除的所有功能，也包括作为升级项目一部分的转换功能，不做任何修改的功能不应包含在内。
- 识别升级的功能用户需求，将其划分为数据功能和事务功能，分别进行度量。
- 汇总计算结果，形成度量报告。

练习：度量升级项目的规模

- 一个现有人力资源系统已进行过功能规模度量，其功能规模为112FP。
- 现有如下升级要求：
 - （1）人力资源系统需要发送给福利系统关于员工家属的信息。信息以文件形式发送，文件每天创建，文件中包含新添加的家属信息。文件格式如下：
 - 头记录
 - 文件名
 - 创建日期
 - 福利记录
 - 员工身份证号
 - 员工家属身份证号
 - 家属姓名
 - 家属生日
 - 尾记录
 - 文件结束标记
 - （2）不再需要以饼图显示的每月员工报表功能（现有系统中每月提供以饼图形式显示的按工资类型分布的员工数量百分比）。
 - （3）查询工作分配详细信息中显示工作分配状态。
 - （4）用户要求按工作分配期倒序排列的员工工作分配报表。报表内容包括员工身份证号、员工姓名、工作编号、工作名称、分配期、分配期超过24个月的员工总数及分配期超过12个月的员工总数。
- 针对以上4条升级要求度量其功能规模。

需求ID	功能名称	升级类型	功能类型	复杂性	功能规模
1	发送员工家属信息	新增	EQ	低	3
2	员工饼图报表	删除	EO	低	4
3	查询工作分配项目信息	修改	EQ	平均	4
4	工作分配报表	新增	EO	平均	5
合计					16

功能点估算方法的应用场景

1、项目前期的可行性分析

- 关注技术可行性之外的内容
- 采用快速功能点方法判断项目所需完成的工期和初步预算，从而决定组织是否能够支撑或接受该项目

示例：某组织希望启动一个网上书城的应用项目，涉及的业务实体大概为30个左右，可以快速得到项目的功能点数为1200（ $30 \times 10 \times 4$ ）个。所以对应的工作量为40到80人月，需要的预算为120万到240万左右，对应的开发时间为8到17个月

2、甲方确立项目范围与标的

- 甲方在招标的过程中首先需要在内部立项、申请预算
- 功能点方法有助于给出明确的预算申请依据，使得预算过程更加透明

示例：1000个功能点的项目，甲方内部申请的预算为 $1000 \times 2K = 2000K$ ，其中1000FP是根据功能点标准得到，而每个功能点的费用为2K则可以依据行业数据得到（假定一个功能点的开发成本为1.5K左右，考虑到乙方的利润为20%，以及甲方10%左右的管理成本）

3、为乙方确立项目范围与标的

- 甲方要求乙方在投标过程中采用功能点报价或评估工期，便于甲方从第三方的角度审查核实乙方的报价是否过高或偏低

4、甲乙双方合同谈判的依据

- 与传统的基于模块的报价方法相比较，基于功能点的谈判过程对双方更为透明。
- 甲方不必再单独依赖于所谓的“最低价中标”或“中间价中标”等评判方式
- 甲方可以依据乙方所提供的软件功能点数量进行验收并支付合同款项

问题： 如何确保质量标准？

5、乙方项目立项的依据

- 基于功能点方法，乙方内部的人员配备、费用安排以及工期设定等都可以更透明

- 乙方比较不同项目的预算

- 可以更好地将市场因素与技术因素区分开来，从而给定相对客观的立项依据

示例：项目的合同价分别是50万、100万、150万，那么乙方内部立项的预算是否会有所不同？

6、项目计划与跟踪的基础

- 对于增量型项目，采用功能点可以直接衡量项目的产出，甚至可以作为依据向客户收取与功能点数量对应的费用

- 对于瀑布型项目则可以考虑引入转换机制，使得每阶段产出都可用功能点衡量

问题：如何减小双向转换所带来的误差？

7、评审质量目标的确定

■每次评审会议的出口约束条件如何设定？基于过程约束的相对标准往往效果有限

例如：评审会议要关闭所有已经发现的问题？那些没有发现的问题如何处理？

■例如需求评审的质量目标为0.1-0.3缺陷/FP，则规模为1000FP的需求应该在需求评审之后发现100到300个缺陷，否则应该进行原因分析

8、测试阶段的质量目标设定

■软件测试除了关注测试用例是否充分外，还要考虑所发现的问题是否已经足够

■测试用例的充分性可以用功能点去衡量，测试发现的问题也用功能点去衡量，再结合缺陷趋势分析、缺陷正交分析，最后可以判断测试后质量状况

9、基于功能点的验收方式

- 与需求的功能点一一对应进行验收
考虑完整的需求变更流程，否则无法使用功能点严格对应方法
- 基于功能点评价质量标准
- 基于功能点预估应用系统的运营质量

10、项目需求变更的测量

- 软件项目往往都面临需求变更的可能，采用功能点方法有助于将变更的程度和粒度描述清楚
- 开发过程中新增加了2个需求，修改了1个需求，则有可能是50功能点，也有可能是80个或者100个功能点，所以使用功能点方法可以统一对需求变更程度的衡量。
- 从过程度量的角度，需求稳定度指标才更有意义

11、软件需求详细程度的约束条件

- 甲乙双方往往无法界定需求描述的详细程度，从而为后续需求变更“埋下伏笔”。
- 使用功能点的方法可以有效地指导需求描述的详细程度，将需求评审的标准由相对标准转变为绝对标准
- 最大程度避免因为前期需求描述不清晰而导致需求频繁变更

12、大项目与组合项目管理

- 大项目与组合项目管理要解决的关键问题之一就是在不同的项目间进行比较
- 而采用功能点方法可以对不同的项目进行归一化处理，从而比较单位生产率、单位成本或者单位质量状况

13、甲方内部资产管理

■对许多甲方组织的IT部门来说，描述现有的IT系统规模非常有意义，对于硬件设备相对容易，对于软件系统则可以采用功能点的方法。

■例如计费系统为40K功能点，客服系统为32K功能点、内部工作管理系统为13K功能点。这样便于从资产管理的角度知道“自己的家底有多少”

14、维护项目人员的工作量考核

■维护人员的工作量往往难以确定，因为他们工作性质的复杂性，例如日常维护、简单开发、外包项目管理等。

■采用功能点有助于核定维护人员的工作量，例如对于关键业务系统每个维护人员每年对应的工作量为3K左右的功能点

15、公司范围内项目间的数据比对

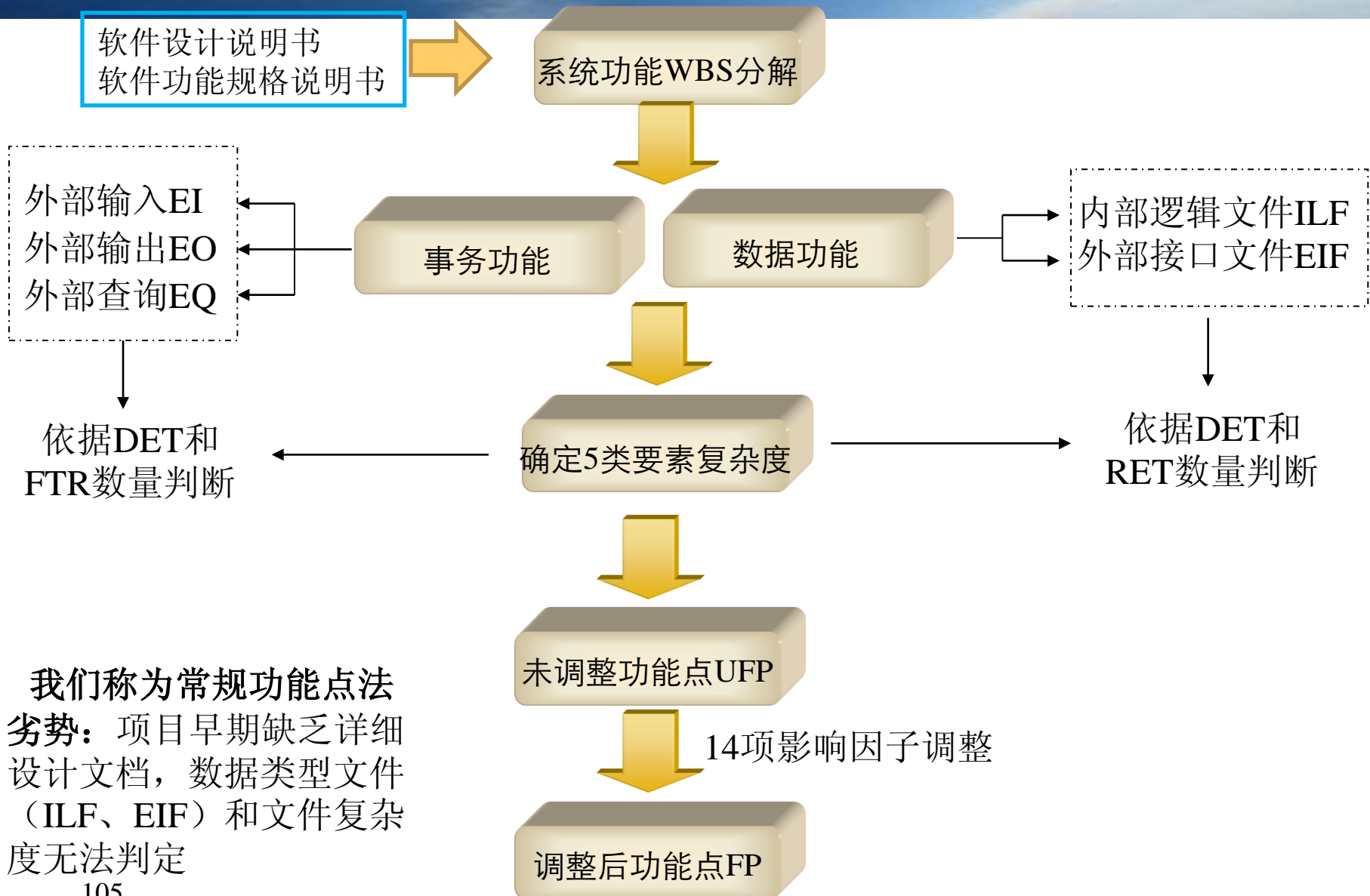
- 建立了功能点度量与估算的框架之后，组织内部可以基于过程改进的方式对项目的现状进行分析
- 比对组织内项目的生产率、缺陷率、需求变更频繁程度等，在组织范围内推广项目的最佳实践等
- 根据统计控制过程原理，从数据分析的角度进行过程改进

16、行业或地区间项目数据比对

- 基于功能点的数据分析亦可以扩展到行业分析或地区分析
- 例如嵌入式行业与商业应用相比较；北京、上海、广州、深圳等地区与杭州、西安、武汉、长沙等地相比较；
- 中国的软件行业与美国、欧洲、印度、日本等比较，而这些比较对于软件公司的市场定位、政府部门的决策制定都有着重要的参考意义

方法小结

功能点分析法小结（一）



我们称为常规功能点法
劣势：项目早期缺乏详细设计文档，数据类型文件（ILF、EIF）和文件复杂度无法判定

测算条件

系统功能需求类推

测算内容

假设1：同类应用功能需求具相似性

计数EI、EO、EQ，类推EIF、ILF

系统5类文件确定

假设2：相同业务处理对应的数据
实体复杂度相似

借助历史数据，计算5类文件复杂度期望值 $E_i = P_i * W_i$

未调整功能点UFP

假设3：系统特征具相似性

14项环境影响因素分五种情形简单
赋权

调整后功能点FP

我们称为：快速功能点法，适用于项目早期计划用

劣势：有参照系的系统测算精度取决于历史数据积累的多寡，无历史参照的新系统测算误差较大，且一般低于完工时实测数

功能点与代码行之间的转换关系

代码行和功能点数之间的关系依赖于程序设计语言和设计质量。下表给出了不同程序设计语言中，建造一个功能点所需的平均代码行数。

程序设计语言	LOC/FP (平均值)	程序设计语言	LOC/FP(平均值)
汇编语言	320	面向对象语言	30
C	128	第四代语言	20
Cobol	105	(4GLs)	15
Fortran	105	代码生成器	6
Pascal	90	电子表格	4
Ada	70	图形语言(图标)	

- 功能点数/150=软件开发人员数量
- 功能点数/3500=维护程序员数量
- 功能点数*1000美元=美国的软件开发成本
- 功能点数*2%=每月的需求蔓延率
- $(\text{功能点数})^{1.2}$ =软件中潜在的缺陷总数
- $(\text{功能点数})^{1.25}$ =需要的测试用例总数
- $(\text{功能点数})^{1.15}$ =全部文本文档的页数

Q&A

谢谢!