软件项目管理高级研讨班：补充材料

廖彬山
美国CMU/SEI授权的CMMI主任评估师
Email：liaobs@263.net
手机：13910021839

功能点方法 -1

- **最初的功能点分析方法是由**Allan Albrecht**开发的，现在直接延续了该方法的是国际功能点用户协会（**IFPUG**）**
- Albrecht/IFPUG**功能点分析方法的基本历史过程如下：**
  - 1979**年，**IBM**的**Albrecht**发表**"Measuring Application Development Productivity"**，这是公认的所有功能点分析方法的最早文献**
  - IBM 1984**年发表了第一个功能点规范**"IBM CIS & A Guidelines 313AD/M Productivity Measurement and Estimate Validation"
  - 1988**年，国际功能点协会发布了**"Function Point Counting Practices Manual"2.0**版本，即《功能点实践手册》**
  - 1990**年，国际功能点协会发布了该功能点实践手册的**3.0**版本，主要是增加了一些数据和解释**

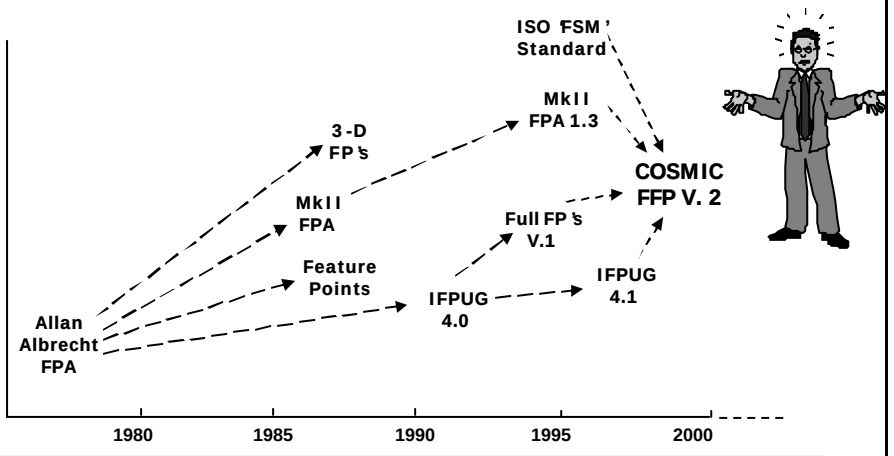
功能点方法 -2

- 1994年，国际功能点协会发布了该功能点实践手册的4.0版本，主要增加了图形用户界面的计算以及有关一致性的问题
- 1999年，国际功能点协会发布了该功能点实践手册的4.1版本，主要是澄清一些已存在的概念和规则，并且符合ISO/IEC 14143标准
- 2003年10月6日国际标准化组织正式接纳认可《国际功能点协会（IFPUG）4.1版本未调整功能点计算手册》，正式的国际标准编号是ISO/IEC 20926:2003
- IFPUG成立了专门的认证管理部门推广CFPS认证，以培养更多的功能点分析方法方面的专业人才

概述 -3

- 其他功能点方法
  - 荷兰软件功能点分析方法（NESMA方法）
  - Mk II功能点分析方法
  - 全功能点分析方法
  - SPR功能点分析方法（或称特征点分析方法）
  - 3D功能点分析方法
  - 对象点（Object Points）分析方法

发展历史



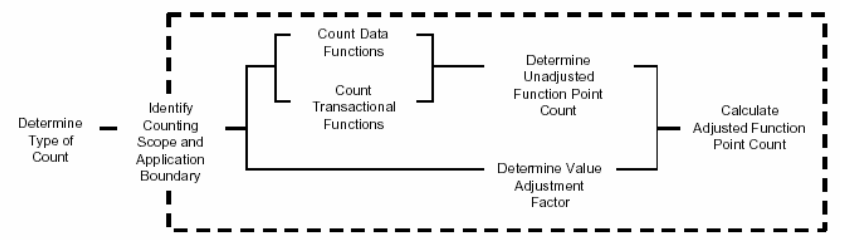
Objectives of Function Point Analysis

- Function point analysis is a standard method for measuring software development from the user's point of view
- Function point analysis measures software by quantifying the functionality the software provides to the user based primarily on logical design
- Objectives of function point analysis are to:
  - Measure functionality that the user requests and receives
  - Measure software development and maintenance independently of technology used for implementation
- In addition to meeting the above objectives, the process of counting function points should be:
  - Simple enough to minimize the overhead of the measurement process
  - A consistent measure among various projects and organizations

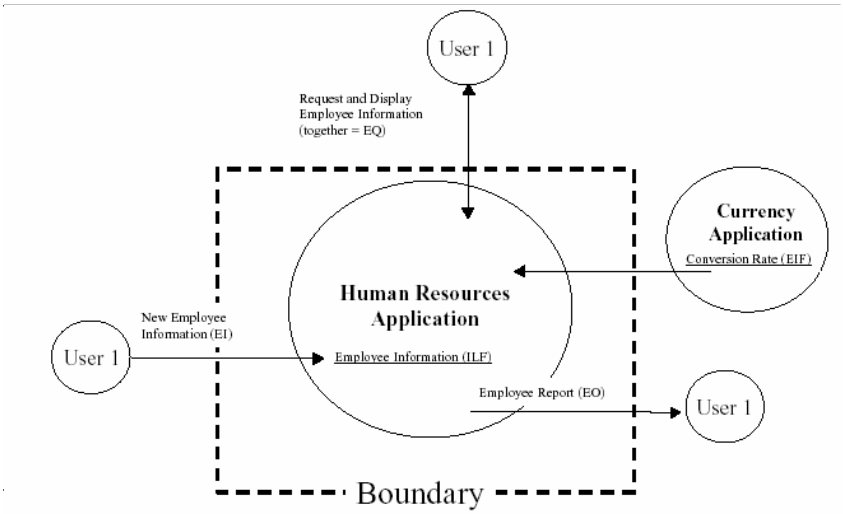
Benefits of Function Point Analysis

- Organizations can apply function point analysis as:
  - A tool to determine the size of a purchased application package by counting all the functions included in the package
  - A tool to help users determine the benefit of an application package to their organization by counting functions that specifically match their requirements
  - A tool to measure the units of a software product to support quality and productivity analysis
  - A vehicle to estimate cost and resources required for software development and maintenance
  - A normalization factor for software comparison

Function Point Counting Procedure



示例



Determine the Type of Function Point Count

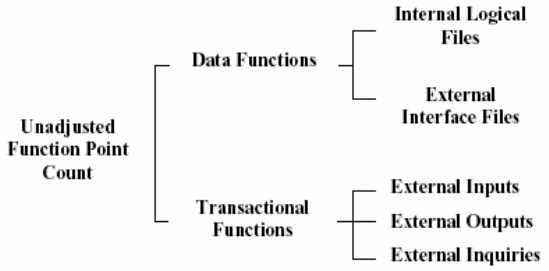
- Function point counts can be associated with either projects or applications. There are three types of function point counts:
  - Development project function point count
    - The development project function point count measures the functions provided to the users with the first installation of the software delivered when the project is complete.
  - Enhancement project function point count
    - The enhancement project function point count measures the modifications to the existing application that add, change, or delete user functions delivered when the project is complete.
    - When the functionality from an enhancement project is installed, the application function point count must be updated to reflect changes in the application's functionality.
  - Application function point count
    - The application function point count and project count are associated with an installed application. It is also referred to as the *baseline* or *installed* function point count. This count provides a measure of the current functions the application provides the user. This number is initialized when the development project function point count is completed. It is updated every time completion of an enhancement project alters the application's functions.
- The above example is for a project function point count, which will also evolve into an application function point count.

Identify the Counting Scope and Application Boundary

- The counting scope defines the functionality which will be included in a particular function point count.
- The application boundary indicates the border between the software being measured and the user.
- The above example shows the application boundary between the Human Resources Application being measured and the external Currency Application.
- It also shows the application boundary between the Human Resources Application and the user.

Determine the Unadjusted Function Point Count

- The unadjusted function point count (UFPC) reflects the specific countable functionality provided to the user by the project or application.
- The application's specific user functionality is evaluated in terms of *what* is delivered by the application, not *how* it is delivered. *Only* user-requested and defined components are counted.
- The unadjusted function point count has two function types—data and transactional. These function types are further defined as shown in the following diagram.



UAFP

- **外部输入**（EI：External Inputs）
- **外部输出**（EO：External Outputs）
- **外部查询**（EQ：External Inquiries）
- **内部逻辑文件**（ILF：Internal Logical Files)
- **外部接口文件**（EIF：External Interface Files)

内部逻辑文件（ILF）-1

- **ILF是用户确认的、在应用程序内部维护的、逻辑上相关的数据块或控制信息**
- **用户确认指的是用户和软件开发人员针对过程共同定义的需求，或已经过共同批准和理解的数据块**
- **逻辑上相关是指每组数据的描述在逻辑上应该相适应。为了维护自己的存在，一个ILF不应该依赖于或归结于其他ILF。ILF通常由实体类型的第2范式（要求实体唯一）或第3范式（要求字段没有冗余，即任何字段不能由其他字段派生出来）表示**
- **控制信息指被度量应用程序用来影响基本处理的数据。它会指定何时、如何处理哪些数据。做为ILF，这些数据、规则或者参数保存在应用程序内部，由应用程序维护**

内部逻辑文件（ILF）-2

- 维护是指通过应用程序的基本处理来修改数据
- 基本处理是指对用户有意义的一组最小的活动单位。如在股市抛出股票可以分解为很多工序，包括创建、读取、更新和显示，这样就可以创建到期金额、读取文件确认用户是否有信用，并且更新持股数量
- 识别为ILF：
  - 应用程序中的业务数据，如：库存记录，职员培训记录，工资记录，信用卡交易数据、产品销售信息，客户电话数据
  - 应用程序维护的安全数据和口令数据
  - 应用程序维护的帮助数据
  - 应用程序维护的参数数据
  - 应用程序维护的出错信息文件及其描述

内部逻辑文件（ILF）-3

- 不应该识别为ILF的例子：
  - 临时文件
  - 摘录文件或视图文件
  - 因为技术原因而引入的文件
  - 同一个文件的复本
  - 索引文件
  - 历史数据或备份的数据
  - 由其他程序维护，仅为本程序读取的文件
  - 包含不完整信息的中间文件

外部接口文件（EIF）-1

- 外部接口文件是由用户确认的、由被度量的应用程序引用，但是在其他应用程序内部维护、逻辑上相关的数据块或控制信息。
- 常见的EIF的例子：
  - 在应用程序外部维护的口令数据或权限数据
  - 在应用程序外部维护的帮助数据，参数数据，出错信息等

外部接口文件（EIF）-2

- 不应该识别为EIF的例子：
  - 从另一个应用程序接收的数据，这些数据用于维护被测程序中的一个或多个ILF；这种情况通常被看做业务数据，它应该算EI
  - 临时文件和不同迭代阶段的同一个文件
  - 因为技术原因而引入的文件
  - 历史数据，他们应该和应用程序业务数据一起计算

外部输入（EI）-1

- 数据由外向内跨越边界的基本处理过程
  - 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
- 识别为EI的例子：
  - 维护ILF的输入
  - 提供某控制信息的输入
  - 来自其他应用程序、需要被处理的消息
  - 项目数据的升级转换

外部输入（EI）-2

- 不应该识别为EI的例子
  - 被度量的程序读取其他应用程序存放的参考数据，但这些数据并不用来维护被度量程序的ILF
  - 包含在查询或者输出中的输入请求
  - 帮助用户进入应用程序的登录窗口
  - 刷新或取消窗口中的数据
  - 需要用户确认删除或者其他事务信息的反应
  - 激活同一个逻辑的不同方法
  - 在同一个应用程序中，客户端和服务器间的数据传递

外部输出（EO）-1

- 外部输出是应用程序向其边界之外提供数据或控制信息的基本处理。EO的主要意图是向用户提供经过处理逻辑加工的、除了检索数据或者控制信息之外的信息或附加信息。处理逻辑必须至少包含一个数学公式或计算、创建导出数据、维护一个或多个ILF，并且/或者改变系统的行为
- EO的例子：
  - 需要使用算法或计算公式的报表
  - 经过基本处理的计算、推导或者更新后，被传递、归档给其他应用程序的数据以及（或者消息）
  - 在屏幕上显示或者在文件中传输的，经过推导或者计算而得出来的信息
  - 需要经过计算的图示，如饼图，柱图等
  - 针对某中保险政策的保费计算值

外部输出-2

- 不应该识别为EO的例子：
  - 详细报告中包含的总计字段（详细报告是EO）
  - 发送给其他不包含公司、计算值或推导数据的应用程序的文件，并且这些应用程序不在发送数据的应用程序中维护一个ILF
  - 处理逻辑相同的多种媒体
  - 更新或取消屏幕上的数据（不计算）
  - 不包含其他处理逻辑的一组数据的重排序或重新安排
  - 帮助（多数情况下算EQ）
  - 表明数据已被处理的确认消息
  - 请求用户确认删除或其他操作的消息
  - 用户在SQL等语言的控制或指导下随便产生的报表
  - 在同一个程序内部，客户端和服务器之间传递的数据，它并没有穿过应用程序的边界

外部查询（EQ）-1

- 外部查询是应用程序向其边界之外提供数据或控制信息查询的基本处理
- EQ的主要意图是通过查询ILF或EIF中的数据或控制信息来为用户提供信息
- 处理逻辑中既不包含数学公式或者计算也不产生导出数据
- 处理过程不维护ILF，系统行为不受影响
- 外部查询的例子:
  - 通过查询1个或多个ILF/EIF而得出并显示事务数据
  - 用户功能,例如视图、查找、显示、浏览和打印（集注，包含相同处理逻辑的打印和视图功能算一个EQ而不是2个）
  - 应用程序中出现的间接查询（即在变更或删除功能之前，对数据的检索）；此查询应该可用作独立的处理，而且与已计算过的EQ不重复
  - 各种级别的帮助
  - 传送给其他应用程序的文件；文件不包含公式，计算或导出数据，而且也不维护发送应用程序中的ILF
  - 从邮箱中检索邮件

外部查询（EQ）-2

- 不应该识别为EQ的例子：
  - 激活同一个逻辑的多种方式----例如，执行相同功能的2个功能键，或者在多个屏幕中的相同事务（只能被计算一次）
  - 可以被应用程序的多个区域或屏幕所访问的查询（被计算1次）
  - 用于导航或选择，而不用于数据检索的菜单屏幕
  - 与数据检索相对的数据推导或计算
  - 不包含其他处理逻辑的、数据集的重新排序或重新整理
  - 用户对确认数据消息的反应
  - 错误消息和/或确认信息
  - 在同一个应用程序中，客户端和服务器间的数据传递

计算UAFP

- 将带有权重的外部输入、外部输出、外部查询、内部逻辑文件和外部接口文件的数目加在一起，其结果即为未调整的功能点数

计算UAFP

	低	平均	高	合计
外部输入	__×3	__×4	__×6	
外部输出	__×4	__×5	__×7	
外部查询	__×3	__×4	__×6	
内部逻辑文件	__×7	__×10	__×15	
外部接口文件	__×5	__×7	__×10	
UAFP				

讨论与练习

ABC公司大约有3000名员工，要开发一个工资系统，需求大致如下：

- 要产生一个工资单，显示所有的工资计算。要打印收入和纳税扣除额。还要求将工资单显示在屏幕上。工资单的功能复杂度是“高”。另外，要产生7个报表，每个报表的复杂度是“低”
- 系统的输入是：
  - 员工定期信息（员工ID，基本工资，等级，部门等）
  - 考勤细节及月信息（如专门的纳税扣除额和收入等）

  以上两个输入都是屏幕输入，复杂度为“高”

  另外，还要有一个所得税信息的输入，该输入的复杂度是“平均”

讨论与练习

- 总共有20个查询，每个查询的复杂度是“低”
- 系统内要维护一个Employee Master File，该文件的复杂度是“高”
- 系统引用了三个目录/表：
  - Employee Name and static details file
  - Department
  - Grade

  第一个目录的复杂度是“平均”，其他两个的复杂度是“低”

EI表

FTR's	数据元素（DET）		
	1-4	5-15	>15
0-1	低	低	平均
2	低	平均	高
3或多个	平均	高	高

FTR：File type reference（文件类型引用）

数据元素：每个唯一的、用户可识别的、不重复的字段，算一个数据元素

EO和EQ表

FTR's	数据元素		
	1-5	6-19	>19
0-1	低	低	平均
2-3	低	平均	高
>3	平均	高	高

FTR：File type reference（文件类型引用）

数据元素：唯一的用户可识别的不重复的字段

ILF和EIF表

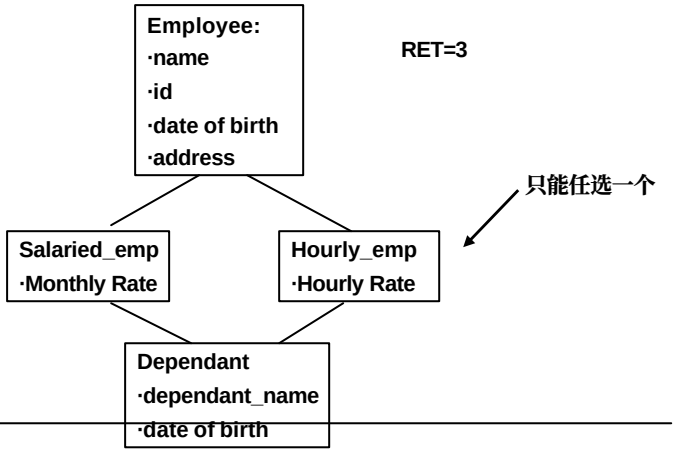
RET's	数据元素		
	1-19	20-50	>50
1	低	低	平均
2-5	低	平均	高
>5	平均	高	高

RET：Record element types（记录元素类型）

数据元素：唯一的用户可识别的不重复的字段

记录元素类型（RET）

- **在一个ILF或EIF内用户可识别的数据元素的分组**



讨论与练习

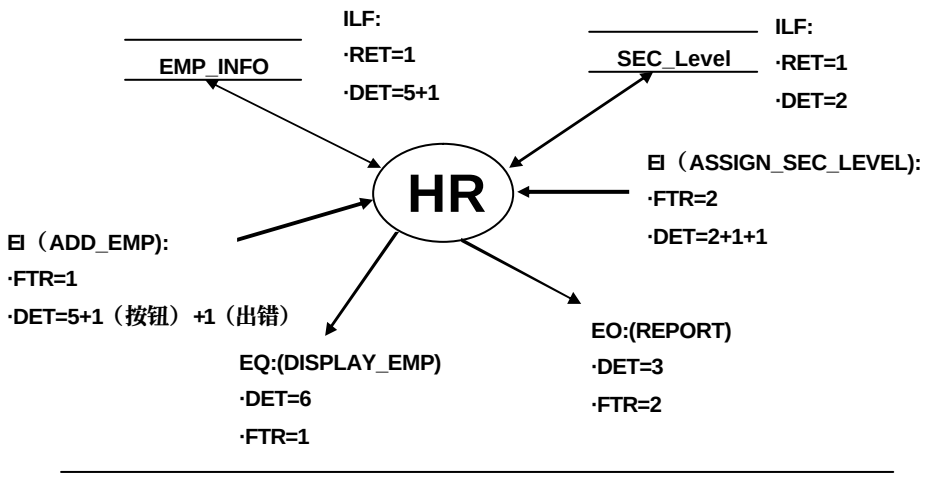
HR（Human Resource)用于维护每名新员工的有关信息。

- 由HR用户维护的信息包括
  - Employee ID
  - Employee Name
  - Employee Mailing Address
  - Employee Pay Grade
  - Employee Job Title
- 创建新的员工记录后，员工预期的Pension Eligibility Date（养老金有效日期）应该自动加以计算，并与其他的员工信息一起保存

讨论与练习

- 雇用一名新员工后，安全部门的用户要为每名新员工赋一个适当的安全等级。由安全用户维护的信息包括：
  - Employee ID
  - Employee Security Level
- 安全用户还要求一个报表，该报表包含以下信息：
  - Count of Employee IDs
  - Employee Name
  - Employee Security Level

讨论与练习



计算调整因子

- 调整因子基于14个基本系统特征

因子	值	因子	值
数据通信		联机更新	
分布式处理		复杂处理	
性能		可重用性	
配置负载		易安装	
事务率		易操作	
联机数据登录		多个场所	
最终用户效率		设施变更	

调整因子的含义

- 数据通信:应用程序与处理器之间直接通信的程度
- 分布式处理:应用程序部件间数据传输的程度(跨CPU)
- 性能:需要考虑的响应时间和吞吐量对应用程序开发的影响程度
- 配置负载:计算机资源限制对应用程序开发的影响程度
- 事务率:事务率对应用程序开发的影响程度
- 联机数据输入:数据通过交互处理进入系统的程度
- 最终用户使用效率:描述了被测应用程序中对人性化因素和易用性的考虑程度
- 联机更新:内部逻辑文件的在线更新程度
- 复杂处理:处理逻辑对应用程序开发的影响程度
- 可重用性:应用程序和应用程序代码被专门设计、开发和支持，以便在其他应用程序中加以利用的程度
- 易安装性：从以前环境转换到新环境对应用程序开发的影响程度
- 易操作性：应用程序对操作方面的关注程度
- 多场所：描述了为多个地点和用户机构开发应用程序的一种程度
- 支持变更：所开发的应用程序修改处理逻辑或数据结构的容易程度

计算调整因子

- 每个调整因子的取值在0到5之间：
  - 0：没有影响（No influence)
  - 1：偶然的（Incidental)
  - 2：适度的（Moderate)
  - 3：平均的（Average)
  - 4：重大的（Significant)
  - 5：严重的（Essential)
- VAF = Σ Fi

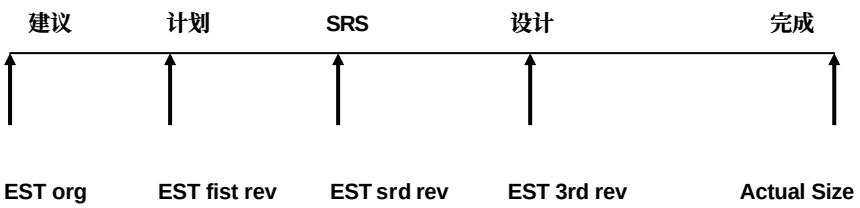
计算调整后的功能点

- 调整后的功能点（AFP）由以下公式计算：

  AFP = UAFP * [0.65 + 0.01 * **VAF**]

- AFP = 150 * (0.65 + 0.01 * VAF)
  = 150 * (0.65 + 0.01 *27)
  = 138

SDLC（软件开发生命周期）中的估算



- 建议项目的开始点就要做估算，称为原始估算，并根据原始估算制订建议书
- 计划时，要进行第一次修订
- 在SRS时，要进行第二次修订
- 在设计时，要进行第三次修订
- 每次修订都是对前次估算的细化
- 完成后，要给出实际的FP数

规模偏离

- Size Variance org = (Actual Size – EST org) / EST org
- Size Variance first rev = (Actual Size – EST first rev) / EST first rev
- Size Variance sec rev = (Actual Size – EST sec rev) / EST sec rev
- **假如历史数据的偏离度是**12%，**原始估算是** 138FP，**则写建议书时，应该考虑的**FP**数是：**

  138 *（1+0.12）=155 FP

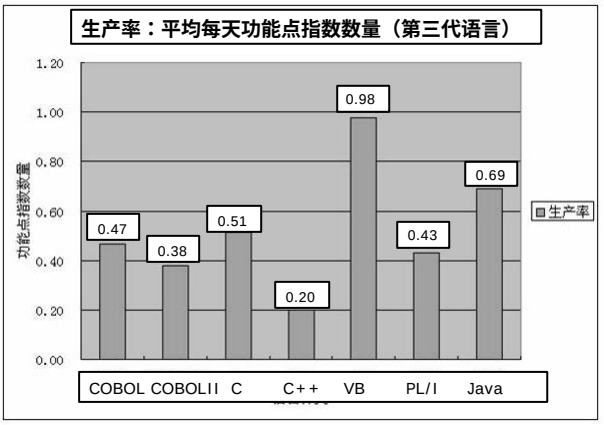
规模如何转化为工作量

- **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**

  Effort=155/0.9=172**人天**

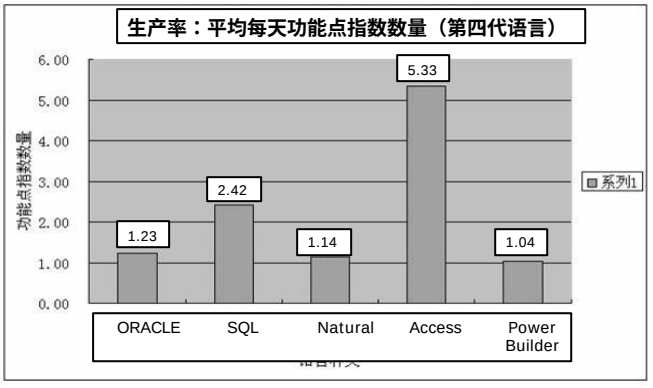
平均生产率（功能点）--1

- 对于三层结构的应用软件，平均生产率大约是每人天0.9功能点



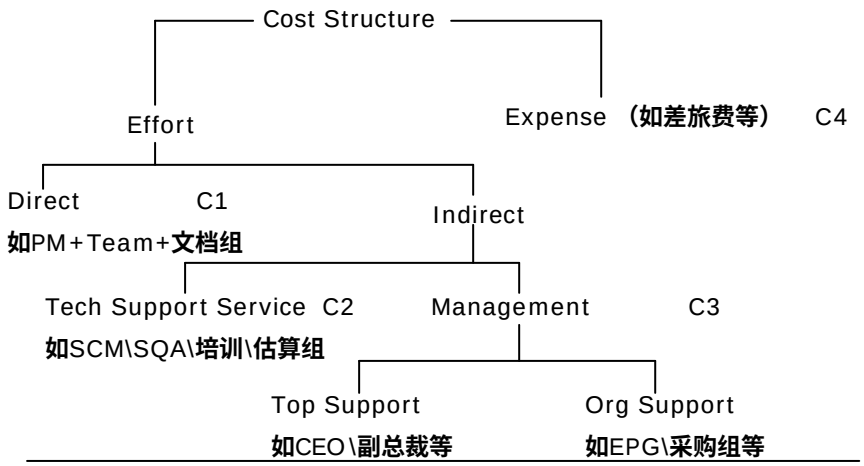
•数据来源：ISBSG v8.0。主要是国外的数据，国内尚无标准

平均生产率（功能点）--2



•数据来源：ISBSG v8.0。主要是国外的数据，国内尚无标准

工作量如何转化为成本？



工作量如何转化为成本

- 指定生产率时，必须指明是基于什么的生产率，假若生产率基于直接工作量（C1）
- C1+C2=100%
- 若C1占80%，172人天，C2占20%，那么 C2=172/80*20 =43天
- 总的人天=172+43 =215天
- 根据总的人天，再乘以平均成本/人天，即得到C1+C2
- 总的成本=C1+C2+Allocation of C3 to the project + C4

GUI的计算规则

- 在一个框架（frame)中的所有Radio Buttons，算一个DE
- 在一个框架中的每个Check Box，算一个DE
- 每个Command Button(如Add，Delete，Modify等），算一个DE
- 每个类似于Next这样的Command Button，实际上算作一个查询或另一个事务(transaction）的输入边（input side)
- 每个图形图像的显示、每个声音（无论多大），算一个DE
- 每个Pick List（Drop Down/Look Up表）是一个EQ（外部查询），但查询的结果是一个EI（外部输入）的一个DE

GUI的计算规则

- 单个的GUI屏幕可能包含几个事务处理类型
- 每个Error消息或Confirmation消息，算一个DE
- 每个Notification消息，算一个EO（外部输出）
- 每个字段/状态信号（Status Signal)/Flag/变量的状态/实时和嵌入式系统的Stimulus Signal，算一个DE

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度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 (4GLs)	20
Cobol	105	代码生成器	15
Fortran	105	电子表格	6
Pascal	90	图形语言(图标)	4
Ada	70		

Development Project Function Point Calculation

- The development project function point calculation consists of three components of functionality:
  - Application functionality included in the user requirements for the project
  - Conversion functionality included in the user requirements for the project
  - Application value adjustment factor

Application Functionality

- Application functionality consists of functions used after software installation to satisfy the ongoing business needs of the user.

Data Functions	RETs	DETs	Functional Complexity
Internal Logical Files			
• Job information	2	5	Low
• Suspended jobs	2	6	Low
• Report definition	1	4	Low
• Employee information	1	6	Low
External Interface Files			
• Location information	1	6	Low
• Conversion information	1	2	Low
• Window help information	1	2	Low
• Field help information	1	5	Low

Transactional Functions	FTRs	DETs	Functional Complexity
External Inputs			
Assignment report definition	1	5	Low
Add job information (screen input)	1	7	Low
Add job information (batch input)	2	6	Average
Correct suspended jobs	1	7	Low
Employee job assignment	3	7	High
EI with screen output –1	2	11	Average
EI with screen output –2	1	6	Low
External Outputs			
Jobs with employees report	4	5	Average
Employees by assignment duration report	3	7	Average
Performance review notification	3	4	Low
Weekly employees report	1	3	Low
Printed check	1	3	Low
Check transaction file	1	4	Low
External Inquiries			
List of retrieved data	1	4	Low
Drop-down list box	1	2	Low
Field level help	1	6	Low
Weekly membership report	1	3	Low
Daily check file	1	2	Low

Application Contribution to the Unadjusted Function Point Count

Function Type	Functional Complexity			Complexity Totals	Function Type Totals
ILFs	4	Low	X 7 =	28	
	0	Average	X 10 =	0	
	0	High	X 15 =	0	
					28
EIFs	4	Low	X 5 =	20	
	0	Average	X 7 =	0	
	0	High	X 10 =	0	
					20
EIs	4	Low	X 3 =	12	
	2	Average	X 4 =	8	
	1	High	X 6 =	6	
					26
EOs	4	Low	X 4 =	16	
	2	Average	X 5 =	10	
	0	High	X 7 =	0	
					26
EQs	5	Low	X 3 =	15	
	0	Average	X 4 =	0	
	0	High	X 6 =	0	
					15
			Unadjusted Function Point Count		115

Conversion Functionality

- Conversion functionality consists of functions provided only at installation to convert data and/or provide other user-specified conversion requirements, such as special conversion reports.
- For example, if a Human Resources (HR) software application was in use and a new HR application is installed, the users may require that information about employees be converted and loaded into the new application. The user-specified conversion requirement is to transfer the current employee data into the new HR system.

Transactional Function	FTRs	DETs	Functional Complexity
External Input			
Employee migration	1	11	Low

Function Type	Functional Complexity			Complexity Totals	Function Type Totals
EIs	1	Low	X 3 =	3	
	0	Average	X 4 =	0	
	0	High	X 6 =	0	
			Unadjusted Function Point Count		3

Application Value Adjustment Factor

- The value adjustment factor is determined by using the 14 general system characteristics to rate the application functional complexity.

Enhancement Project Function Point Calculation

- The enhancement project function point calculation consists of three components of functionality:
  - Application functionality included in the user requirements for the project
  - Conversion functionality included in the user requirements for the project
  - Application value adjustment factor

Application Functionality

- Application functionality consists of:
  - Function points identified from the functionality that is added by the enhancements
  - Function points counted because existing functionality is changed during the enhancement project
  - Function points counted for functionality deleted during the enhancement project

Conversion Functionality

- The conversion functionality consists of function points delivered because of any conversion functionality required by the user.

Value Adjustment Factor

- The two value adjustment factors are the:
  - Application value adjustment factor before the enhancement project begins
  - Application value adjustment factor after the enhancement project is complete

Function Point Formula

- Use the following formula to calculate the enhancement project function point
- count.. **Note:** Data conversion requirements *are included* in this count.
- EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL* VAFB)
- Where:
  - EFP is the enhancement project function point count.
  - ADD is the unadjusted function point count of those functions that were or will be added by the enhancement project.
  - CHGA is the unadjusted function point count of those functions that were or will be modified by the enhancement project. This number reflects the size of the functions *after* the modifications.
  - CFP is the function point count of those functions added by theconversion
  - VAFA is the value adjustment factor of the application *after* the enhancement project is complete.
  - DEL is the unadjusted function point count of those functions that were or will be deleted by the enhancement project.
  - VAFB is the value adjustment factor of the application *before* the enhancement project begins.
- **Note:** When an enhancement project is installed, the application function point count must be updated to reflect changes in the application's functionality. See Application Function Point Calculation presented earlier in this chapter.

Example: Enhancement Project Count

- This section shows an example for a sample enhancement project. The requirements for the enhancement project include the following changes:
  - The user no longer needs to add a job online, therefore, that functionality is to be or was removed.
  - The user needs to receive an additional report about jobs that includes totals.
  - Additional DETs are required to add jobs in batch and correct suspended transactions. A reference to security is also added for the add job transaction.

Added Functionality

- The following table shows the functional complexity for the added functionality counted when the project was completed.
- **Note:** Providing a new report was an additional external output.

Transactional Functions	FTRs	DETs	Functional Complexity
External Output			
Job report	1	15	Low

Function Type	Functional Complexity		Complexity Totals	Function Type Totals
EOs	1	Low	X 4 = 4	
	0	Average	X 5 = 0	
	0	High	X 7 = 0	
				4

Changed Functionality

- The following table shows the functional complexity for the changed functionality, as the functions will exist after the enhancement project is completed.
- **Note:** The complexity for adding a job was increased because of the additional file type referenced. The complexity for correcting suspended transactions remained low.

Transactional Functions	FTRs	DETs	Functional Complexity
External Input			
Add job information (batch input)	3	8	High
Correct suspended transaction	1	8	Low

Function Type	Functional Complexity		Complexity Totals	Function Type Totals
EIs	1	Low	X 3 = 3	
	0	Average	X 4 = 0	
	1	High	X 6 = 6	
				9

Deleted Functionality

- The following table shows the functional complexity for deleted functionality identified at the end of the project.

Transactional Functions	FTRs	DETs	Functional Complexity
External Inputs			
Add job information (screen input)	1	7	Low

Function Type	Functional Complexity		Complexity Totals	Function Type Totals
EIs	1	Low	X 3 = 3	
	0	Average	X 4 = 0	
	0	High	X 6 = 0	
				3

Final Calculation

- The application value adjustment factor was 1.05 before the project began. The value adjustment factor remained the same after the project was completed.
- Using the complexity and contribution counts for this example, the enhancement project function point count is shown below. (The formula was explained on page 9-11.)
- EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL* VAFB)
- EFP = [(4 + 9 + 0) * 1.05] + (3 * 1.05)
- EFP = 16.8 or 17

Application Function Point Calculation

- This section provides the formulas to calculate the application function point count. There are two variations of this formula:
  - Formula to establish the initial function point count for an application
  - Formula to re-establish the function point count for an application after an enhancement project has changed the application functionality

Formula to Establish the Initial Count

- Use the formula in this section to establish the initial function point count for an application. Initially, the user is receiving new functionality. There are no changes to the existing functionality or deletions of obsolete or unneeded functionality. The application function point count *does not* include conversion requirements.
- AFP = ADD * VAF

- Where:
  - AFP is the initial application function point count.
  - ADD is the unadjusted function point count of those functions that were installed by the development project.
  - VAF is the value adjustment factor of the application.

Formula to Reflect Enhancement Projects -1

- When an enhancement project is installed, the existing application function point count must be updated to reflect modifications to the application. The functionality for the application can be altered in one or more ways:
- Added (new) functionality increases the size of the application
- Changed functionality increases, decreases, or has no effect on the size of the application
- Deleted functionality decreases the application size
- Changes to the value adjustment factor adds, subtracts, or has no effect on the function point count but does affect the adjusted function point count
- **Note:** Because conversion functionality does not affect the application function point count, any conversion functionality associated with an enhancement project is omitted entirely from the application function point calculation.

Formula to Reflect Enhancement Projects -2

- Use the following formula to calculate the application function point count after an enhancement project:
- AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA
- Where:
  - AFP is the application's adjusted function point count.
  - UFPB is the application's unadjusted function point count *before* the enhancement project begins.
    - **Note:** If this count is unavailable, it can be calculated using the formula UFBP = AFPB/VAFB; where AFPB is the adjusted application function point count before the enhancement project. VAFB is the value adjustment factor of the application before the enhancement project.
  - ADD is the unadjusted function point count of those functions that were added by the enhancement project.
  - CHGA is the unadjusted function point count of those functions that were changed by the enhancement project. This number reflects the size of the functions *after* the changes.
  - CHGB is the unadjusted function point count of those functions that were changed by the enhancement project. This number reflects the size of the functions *before* the changes were made.
  - DEL is the unadjusted function point count of those functions that were deleted by the enhancement project.
  - VAFA is the value adjustment factor of the application *after* the enhancement project is complete.

Example: Application Count

- This section shows an example for the initial count and the count that reflects an enhancement project. Numbers for these counts are from the application count on page 9-8 and the enhancement count on page 9-11.

Initial Count

- The initial application project count is shown below. The value adjustment factor is 1.05. (The formula was explained on page 9-17.)
  - AFP = ADD * VAF
  - AFP = 115 * 1.05
  - AFP = 120.75 or 121
- **Note:** Only the size of the application functionality installed for the user is included in the initial count.

Count After Enhancement

- The application project function point count to reflect enhancements is shown below. The value adjustment factor is 1.05. (The formula was explained on page 9-17.)
  - AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)]* VAFA
  - AFP = [(115 + 4 + 9) - (9 + 3)]* 1.05
  - AFP = 121.8 or 122

谢谢！！！