FUNCTION POINT ANALYSIS:
Sizing The Software Deliverable

Fundamentals of Counting Function Points

2008

Course Objectives

The primary course objectives are to:

- transfer function point knowledge and skills to the participants
- demonstrate Function Point Analysis through class exercises

Basis For Course

- The David Consulting Group's Course: “Function Point Analysis, Sizing The Software Deliverable” contains material which has been extracted from the International Function Point Users Group (IFPUG) Counting Practices Manual (CPM), Version 4.2
- CPM is reproduced in this document with permission of IFPUG

Student Aids

- Helpful Hints in Counting Function Point Components
- Who Should Count
- Useful Project/Application Documentation
- Function Point Counting Documentation
- Common Industry Definitions
- Work Breakdown Structure of Activities Common in Phases
- Blank Counting Sheets and Function Point Worksheets

Course Agenda

Module 1: Function Point Analysis, Uses and Benefits

Module 2: Function Point Counting Rules

- Application Function Point Count
- Enhancement Function Point Count

Module 1: Function Point Overview

- Function Point/IFPUG History
- Objectives and Benefits of Function Point Analysis
- Using Function Points
- Counting -- A High Level View

International Function Point Users Group

- IFPUG chartered in 1986
- IFPUG Purpose
  - To promote and encourage use of Function Points
  - To develop consistent and accurate counting guidelines
  - Maintains Public Standard for Sizing Software (CPM)
- Benefits:
  - Networking with other counters
  - Management Reporting
  - Research projects
  - Web site
  - “Metric Views”, “Voice”
  - Certifications for Function Point Methodology and Measurement

Function Point History

- Function Points were introduced in 1979 by A. Albrecht, IBM, at joint Share/ Guide Conference
- Now used world-wide in North America, Central America, South America, Australia, Europe, and Asia
- Recognized Functional Size Measure for ISO Standards
- IFPUG White Papers published on a variety of measurement-related topics
- IFPUG CPM Version 4.2 issued in 2005
- IFPUG - Certification in Software Measurement in 2004
- ISMA - Fall Conferences on the Function Point Methodology and Measurement Topics

Module 1: Function Point Overview

- Function Point/IFPUG History
- Objectives and Benefits of Function Point Analysis
- Using Function Points
- Counting -- A High Level View

What is (are) Function Points?

- IFPUG Definition

  Standard method for **measuring** software development from the **user's point of view**

A standardized, structured method of measuring the size of an application or project based upon systematic classification of user requirements (processes) into system components (functions). The unit that is used to measure the software functions is *function points*.

Function points are a unit of measure for software size based upon the functionality of the system.

Objectives of Function Point Analysis

- The purpose of FPA is to.....
  - Measure functionality that the user requests and receives
  - Measure software development and maintenance activities independently of technology used for implementation
  - Provide a consistent, normalized measure across various projects, organizations and technology platforms
  - Provide a method of sizing that is simple enough to minimize the overhead of the measurement process

Benefits of Function Point Analysis

- A tool to determine the benefit of an application to an organization by counting functions that specifically match requirements
- A tool to measure the units of software product to support quality and productivity analysis
- A vehicle to estimate cost and resources required for software development and maintenance (Must also consider project attributes, work breakdown structure, etc.)
- A tool to size purchased application package
- A normalization factor for software comparison
- Function point count is independent of platform, language or technology

Module 1: Function Point Overview

- Function Point/IFPUG History
- Objectives and Benefits of Function Point Analysis
- Using Function Points
- Counting -- A High Level View

Using Function Points (FPs) - What can we count?

Any system that has inputs or outputs, and has and/or uses data or control information

- *Installed application: Baseline (or application) count*
- *Development project: New system or subsystem*
- *Enhancement project: Add, change or delete to present system*
- *Maintenance or Support projects: Minor enhancements*
- *Technical changes are not countable nor some types of maintenance*

What can we count? Enhancement versus Maintenance

- **Adaptive Maintenance – Software maintenance performed to make a computer program usable in a changed environment**
  - Includes modifications to meet new or changing business or technical requirements
  - Initiated by business requests
  - ***"Enhancement" requests are a subset of this group***

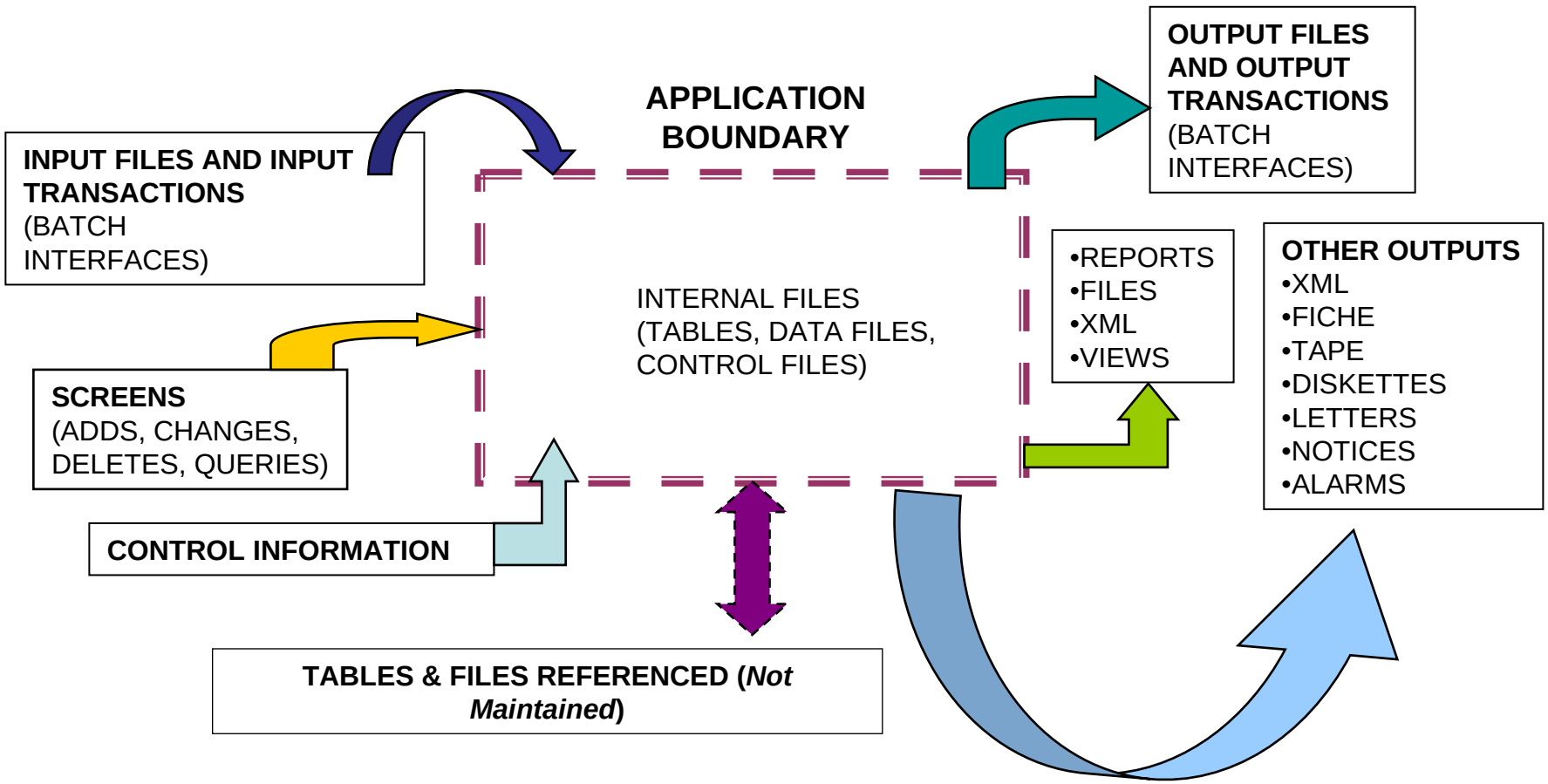
- **Corrective Maintenance – Software maintenance performed to correct faults in hardware or software**
  - Includes defect repair
  - Ensures that previously delivered functionality performs as required
  - Effort should be attributed to the original project that introduced the defect

- **Perfective Maintenance – Software maintenance performed to improve the performance, maintainability, or other attributes of a computer program**
  - Includes system upgrades, performance optimization, platform service agreement maintenance
  - No business functionality

What can we count? Enhancement versus Maintenance

- Function Point Analysis *is* applicable to a subset of Adaptive Maintenance
- Function Point Analysis *should not be used* to size Perfective or Corrective Maintenance work.
- Mixing enhancement work with maintenance work can be expedient or necessary, even efficiency, but the hours must be tracked separately
- For accurate metrics, separation of work effort is necessary
- Preventative and/or corrective maintenance have zero function points
- Packing into releases and apportioning releases is usually the most manageable approach

Using Function Points (FPs) - What do we count?



Using Function Points (FPs) - How do we count?

1. Identify the processes that are identifiable to the user - files, reports, inputs, outputs, transactions (*System functions).*

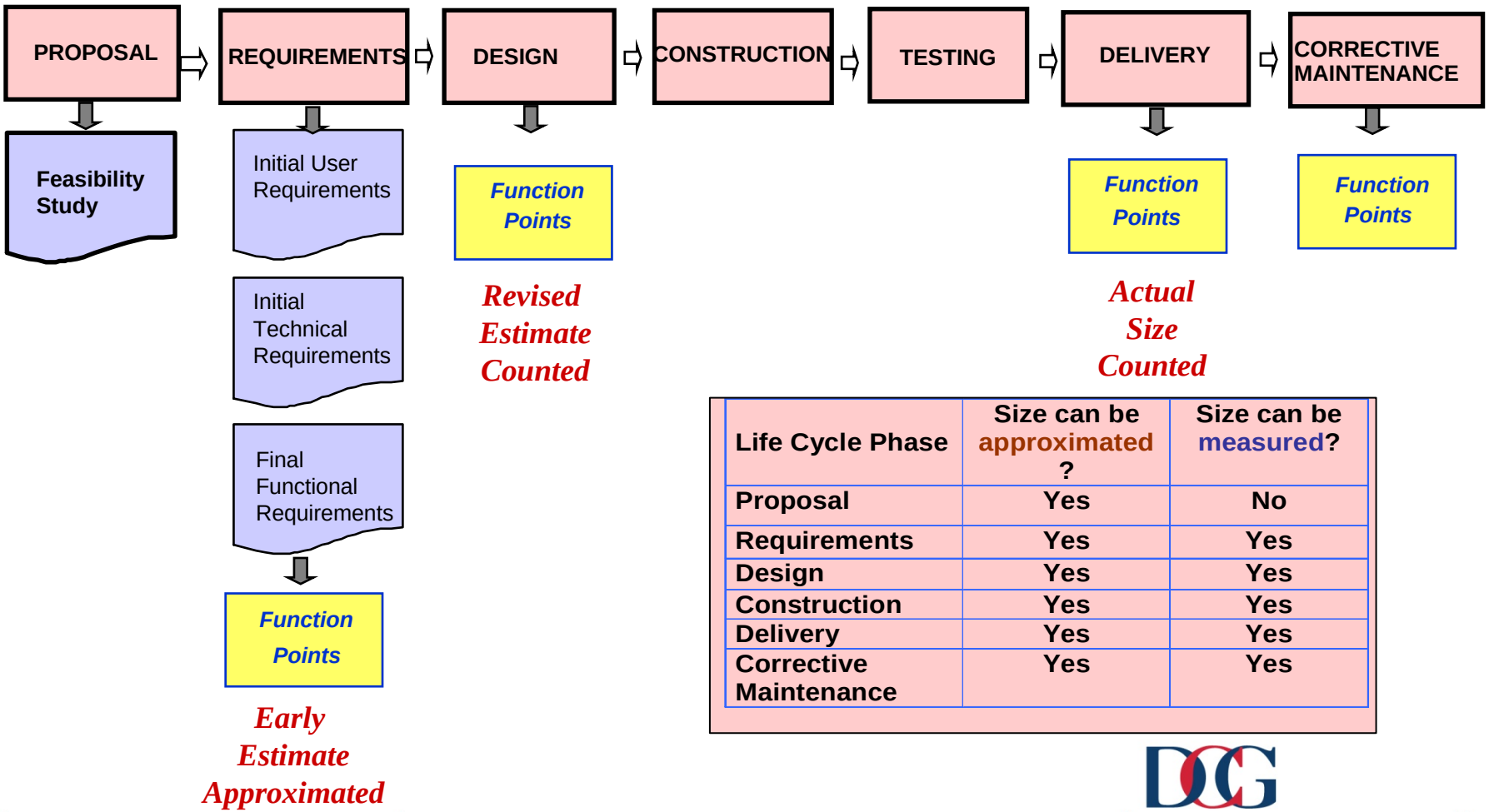
2. Classify system components into function point categories *(Function Point entities). {Report could be an External Output}*

3. Assign each entity a complexity rating (*low, average, high*) and a weight (*in function points*). {*High = 7 fps*}

4. Add all the weights together and apply a system complexity factor.

5. Size can be estimated or actual but is always based upon user - recognizable functionality.

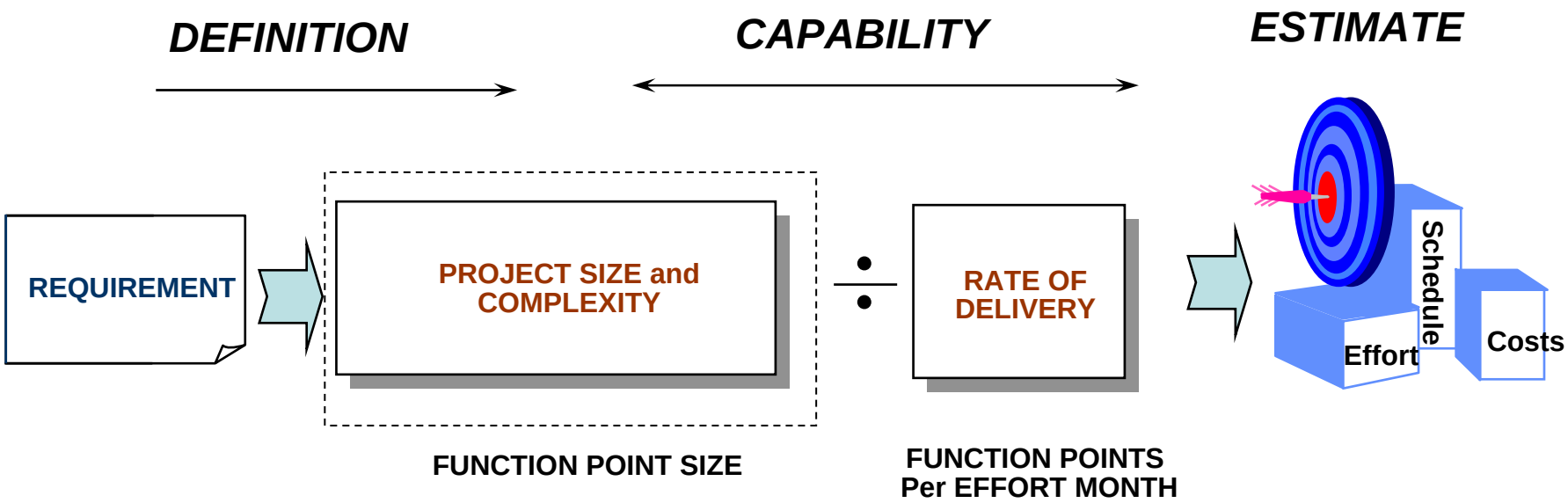
Using Function Points (FPs) - When can we count?



Life Cycle Phase	Size can be approximated ?	Size can be measured?
Proposal	Yes	No
Requirements	Yes	Yes
Design	Yes	Yes
Construction	Yes	Yes
Delivery	Yes	Yes
Corrective Maintenance	Yes	Yes

Using Function Points – Why do we count?

To determine delivery rates for better estimation and process capability, in support of internal or outsourcing activities



Using Function Points to Support Organizational Goals

***Productivity Rate or Delivery Rate** = staff hour/fp*

100 fp 100 sh = 1 sh/fp
250 fp 1000 sh= 4 sh/fp
10 fp 2 sh = .2 sh/fp
500 fp 25 sh = .5 sh/fp

860 fp 1127 sh = 1.31 sh/fp

Cost per function pt

Costs/fps = $67620/860 = $78.62/fp

Defects per function pt

Defects/fps = 4/860 = 1 defect/215 fps

Estimated Project Cost
Project/SCR = 1000 fps

*Estimated staff hrs = Proj fps * prod rate = 1000*1.3 = 1310 sh*

*1310 * rate/hr = 1310 * $60 = $78,600*

***Support rate =** portfolio size/ftes*
***Cost per fp - support** = Support $/fps*

300,000 fps/300 ftes = 1000 fps/fte
$12,000,000/300,000 fps = $40/fp

New system estimate = 20,000 fps
Support staff estimate = 20,000/1000 = 20
*Support costs = 20,000 * $40 = $800,000*

Project and Application Counts

↓

Sizing Basis for Measures

↓

Organizational *Goal-Based* Metrics Program

↓

Executive Level Strategic Goals

Common Metrics Formulas

- **Productivity Rate = Effort in Hours / Size in Function Points**
- Rework Percentage = Rework Hours / Total Project Hours
- Defect Removal Efficiency = Total Number of Defects Found Prior to Delivery / Total Number of Defects Discovered (before and after delivery)
- **Production Defect Rate = Total Production Defects / Function Points Installed**
- **Warranty Defect Rate = Defects during Warranty Period/Function Points Delivered**
- **Scope Creep = Add + Del + Chg Function Points / Original Function Point Count**
- **Cost Efficiency = Actual Cost / Function Points Installed**
- Project Duration = Project End Date – Project Start Date
- Estimated Project Duration = Estimated Project End Date – Estimated Project Start Date
- **Development cost = Effort*Hourly Rate (or Effort Costs)/Function Points Delivered**
- Cost Variance = Actual Cost - Estimated Cost / Estimated Cost
- Effort Variance = Actual Effort - Estimated Effort / Estimated Effort (in hours)
- Duration Variance = Actual Duration – Est. Duration / Est. Duration (in months)
- **Production Support Rate = Full time equivalent (FTEs)/Application Function Points Supported**

Common Metrics Formulas

Development Scorecard

- Productivity Rate = Effort in Hours / Size in Function Points
- Scope Creep = Added + Deleted + Changed Function Points / Original Function Point Count
- Warranty Defect Rate = Defects during Warranty Period/Function Points Delivered
- Delivery costs = Project Costs (Effort)/Function Points Delivered

Application Support Scorecard

- Production Support Rate = Full time equivalent (FTEs)/Function Points (Installed) Supported
- Production Defect Rate = Total Production Defects / Function Points Installed
- Cost Efficiency = Actual Cost / Function Points Installed
- Application Portfolio Ratio = Total Application Counts (Installed) Before/Total Application Counts (Installed) After

Identifying a core set of performance metrics is a critical best practice.

Best Practices in Benchmarking and Metrics

Organizations use application baselines for maintenance service levels and establish performance baseline levels for enhancement and new development activities.

- Focus on performance levels for currently active, business critical applications for baselining and expected performance levels in accordance with industry standards.
- Create an application portfolio profile categorizing applications based on platform, application type, volatility, etc.
- Apply industry accepted measurement tools and techniques to size the defined application portfolio.
- Identify a core set of measures to baseline current performance levels.
- Measure performance levels for enhancement and new development activities.
- Compare supplier performance levels to internal and external benchmarks.

Organizational Baseline and Performance Examples

PRODUCTIVITY			
FP/EM	Client	Industry Average	Industry Best Practices
Overall	3 - 12	6 - 18	42 - 98
New	6 - 12	6 - 14	42 - 77
Enhancement	3 - 10	10 - 18	56 - 98
Mainframe	3 - 8	6 - 12	42 - 86
Client Server	6 - 12	9 - 18	51 - 98

DURATION			
Months	Client	Industry Average	Industry Best Practices
Overall	5 - 12	8 - 17	3.0 - 7.8
New	5 - 10	8 - 14	4.0 - 7.8
Enhancement	7 - 12	10 - 17	3.0 - 6.2
Mainframe	9 - 12	8 - 17	3.8 - 7.8
Client Server	5 - 10	9 - 14	3.0 - 7.5

COST			
$$/FP	Client	Industry Average	Industry Best Practices
Overall	$535 - $2345	$629 - $1692	$158 - $473
New	$712 - $2345	$823 - $1692	$305 - $473
Enhancement	$535 - $1660	$629 - $1300	$158 - $289
Mainframe	$650 - $2345	$930 - $1692	$216 - $473
Client Server	$535 - $1245	$629 - $1154	$158 - $420

QUALITY			
Defects/FP	Client	Industry Average	Industry Best Practices
Overall	.0478 - .7060	.0333 - .0556	.0000 - .0175
New	.0478 - .6664	.0333 - .0556	.0095 - .0175
Enhancement	.0873 - .7060	.0400 - .0556	.0000 - .0098
Mainframe	.2568 - .7060	.0357 - .0556	.0095 - .0175
Client Server	.0478 - .5566	.0333 - .0526	.0000 - .0098

Note: Examples not actual data

Effective Use of Baseline Data

Using the data from the portfolio and performance baselines a comprehensive set of core measures can be established that will provide management with an increased ability to more effectively manage resources, plan budgets and make more informed decisions.

Portfolio Management

- Bring "maintenance" costs, effort and defects in-line with Industry Average benchmarks – driving to Industry top-quartile performance
- Identify costly problem areas within the application portfolio

Project Performance Levels

- Drive cost reductions and increase performance
- Deliver projects on time and within budget

Business Value Management

- Monitor contribution to the business

Core Measures
Portfolio Management

- Focus: Recognizing and managing the cost for corrective maintenance (repairing defects). Reduce maintenance costs.
- Portfolio Management measures:
  - **Assignment Scope - number of resources required to support application functionality**
  - **Growth Rate – increased number of Function Points per application**
  - **Stability Ratio – number of changes per application Function Points**
- Application Maintenance Support measures:
  - **Maintainability - Maintenance Cost per Function Point**
  - **Reliability – Number of production failures per total application Function Points**

Core Measures - Performance Levels

- **Focus: "Enhancement" counts by project/release to monitor and manage performance – effort, cost, and quality**
- Application enhancement measures:
  - Delivery Rate (FP/SM)
  - Time to Market
  - Cost Ratio ($/FP)
  - Defect Density (Defects/FP)
- Project management measures:
  - Delivery Rate (Hours per Function Point)
    - By Platform
    - By Type of Development
    - By Team
    - By Application
  - Speed of Delivery or Function Points per Elapsed Month
    - By Platform
    - By Type of Development
    - By Team
    - By Application

Module 1: Function Point Overview

- Function Point/IFPUG History
- Objectives and Benefits of Function Point Analysis
- Using Function Points
- Counting -- A High Level View

FP Terms

- User -- Any person that specifies Functional User Requirements and/or any person or thing that communicates or interacts with the software at any time
  - User can be another application
- User identifiable --Defined requirements for processes and/or groups of data that are agreed upon, and understood by, both the user(s) and software developer(s)
- Elementary Process - the smallest unit of activity that is meaningful to the user(s), which is self-contained and leaves the business of the application being counted in a consistent state
- Control Information - is data that influences an elementary process of the application; it specifies what, when or how data is to be processed
- Maintained - the ability to modify data through an elementary process

Role of User in Requirements

- User View
  - Represents a formal description of the user's business needs in the user's language; developers translate the user information into information technology language in order to provide a solution
  - Function Point count accomplished using the information in a language common to both users and developers
- User View
  - is a description of the business function
  - is approved by the user
  - can be used to count function points
  - can vary in physical form (documentation formats)

Types of FP Counts

- Development Project
  - New application build
- Enhancement Project
  - Changes to an existing production application
- Application
  - Baseline profile of the system available in production

FPs Logical View

Physical

- Lines of code or programs/modules
- Physical database and/ or files
- Physical transactions (screens)

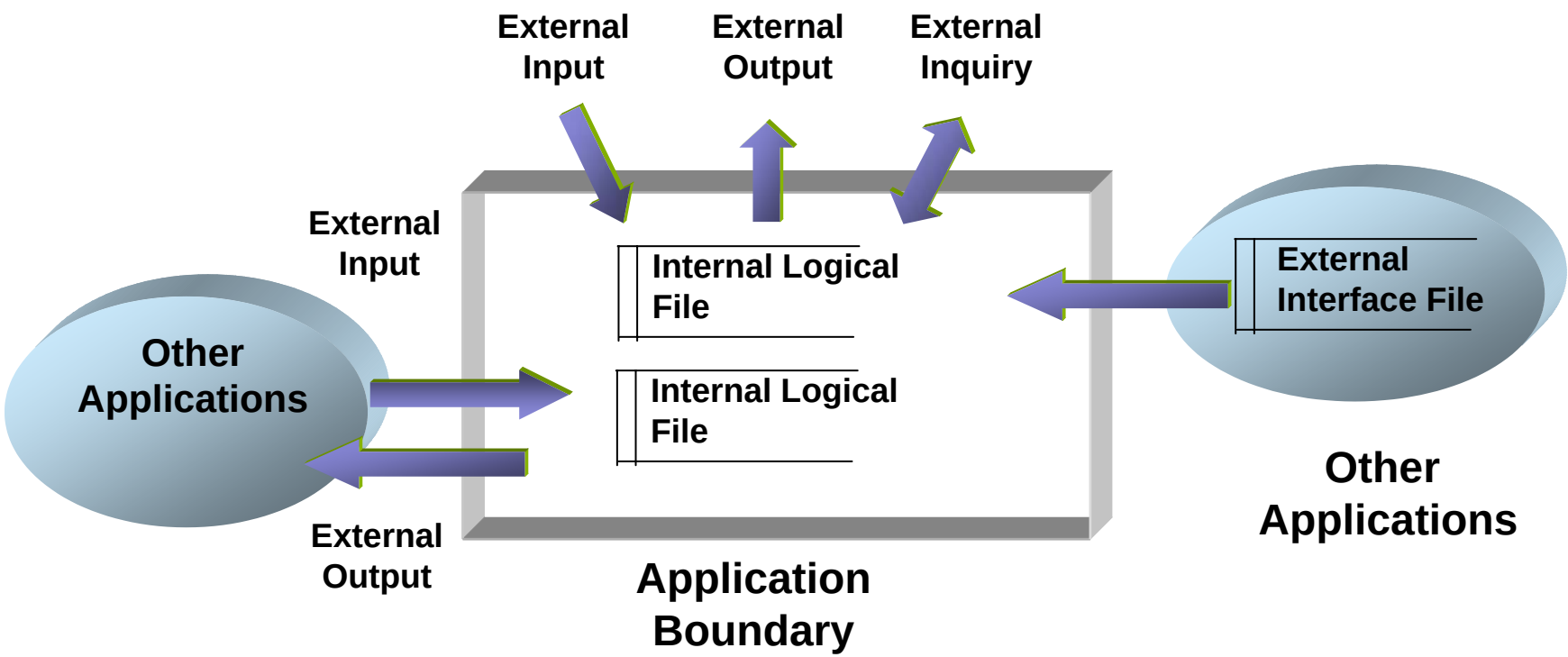
Logical

- Functionality
- Logically grouped stores of user data
- Elementary processes which leave the business in a consistent state

FP Evaluates these Components - Logically

- Data Functions
  - Internal Groupings of data called Internal Logical Files (ILF)
  - External Groupings of data or External Interface Files (EIF)
  - The term file does not refer to files in the physical sense, rather refers to logical data stores, entities, objects or super-classes
- Transaction Functions
  - External Inputs (EI)
  - External Outputs (EO)
  - External Inquires (EQ)
- General System Characteristics

The Logical View



Component Complexity

- FP Counting is based on *Identification Rules* and *Complexity Rules* (CPM)
- Components are assessed based upon complexity:
  - Data Element Types (Number of user variables or fields)
  - File Types Referenced (Number of data groups , either internal or external, that are referenced, read, created or updated)
  - Record Element Types (Number of data sub-groupings or unique record formats)

Complexity

Components:	**Low**	**Avg.**	**High**	**Total**
Internal Logical File (ILF)	__ x 7	__ x 10	__ x 15	___
External Interface File (EIF)	__ x 5	__ x 7	__ x 10	___
External Input (EI)	__ x 3	__ x 4	__ x 6	___
External Output (EO)	__ x 4	__ x 5	__ x 7	___
External Inquiry (EQ)	__ x 3	__ x 4	__ x 6	___

FP Terms

Unadjusted Function Points (UFPs)

- The sum of the complexity weighted functions

Value Adjustment Factor (VAF)

- Reflects the functionality provided to the user by the 14 General System Characteristics

Adjusted Function Points (AFPs)

- The total Function Point count based on the Unadjusted Function Point count multiplied by the Value Adjustment Factor

Module 2: The Counting Rules

- ➡ The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Function Point Process

1. Determine the type of Function Point count
2. Identify the counting scope and application boundary
3. Count the data functions to determine their contribution to the unadjusted function point count
4. Count the transactional functions to determine their contribution to the unadjusted function point count
5. Determine the value adjustment factor - 14 General System Characteristics
6. Calculate adjusted function point count

Module 2: The Counting Rules

- The Process
- ➡ Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Types of Function Point Count

- **Development Project Function Point Count**
  - Measures the functions provided to users with first install of the software delivered when project is complete
- **Enhancement Project Function Point Count**
  - Measures the modifications to existing application add, change or delete user functions delivered when project is complete
  - When functionality is installed, the application count must be updated to reflect changes in the application functionality

Types of Function Point Count

Application Function Point Count

- Measures an installed application
- Also referred to as the baseline or installed count
- Provides count of the current functions provided to users
- 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

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- ➡ Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Application Boundary Counting Tasks

Step	Action
1	Establish the purpose of the count
2	Identify the counting scope
3	Identify the application boundary
4	Determine the following items • The purpose of the count • The counting scope • The application boundary • Any assumptions related to the above

Examples of Purpose

- To provide function point count as input to estimation process to determine effort
- To provide function point count of installed base of applications
- To provide function point count to enable comparison of functionality delivered by two different suppliers' packages

Scope and Boundary

- Definition:
  - A function point count provides an answer to a business problem
- Purpose:
  - Determines the type of function point count and scope of required count
  - Influences the positioning of the boundary between the software under review and the surrounding software

Identify the Counting Scope

- Definition:
  - Scope defines the functionality which will be included in a particular function point count
- The Scope:
  - Defines a (sub) set of software being sized
  - Is determined by the purpose for performing the function point count
  - Identifies which functions will be included in the function point count, to provide answers relevant to the purpose of counting
  - Could include more than one application

Counting Scope Examples

Scope of enhancement count

- Includes all functions being added, changed and deleted
- Boundary of impacted application(s) remains the same
- Functionality of the application reflects the impact of the functions added, changed or deleted
- Includes all functions impacted (built or customized) by the project activities

Scope of a development count

- Includes all functions impacted (built or customized) by the project activities

Scope of an application count may include

- Only the functions being used by the user (if purpose is to provide package as software solution), or all the functions delivered
- Re-hosting or re-engineering of an application may include
  - Only the functions being added/changed
  - COTS customization functions (added/changed)

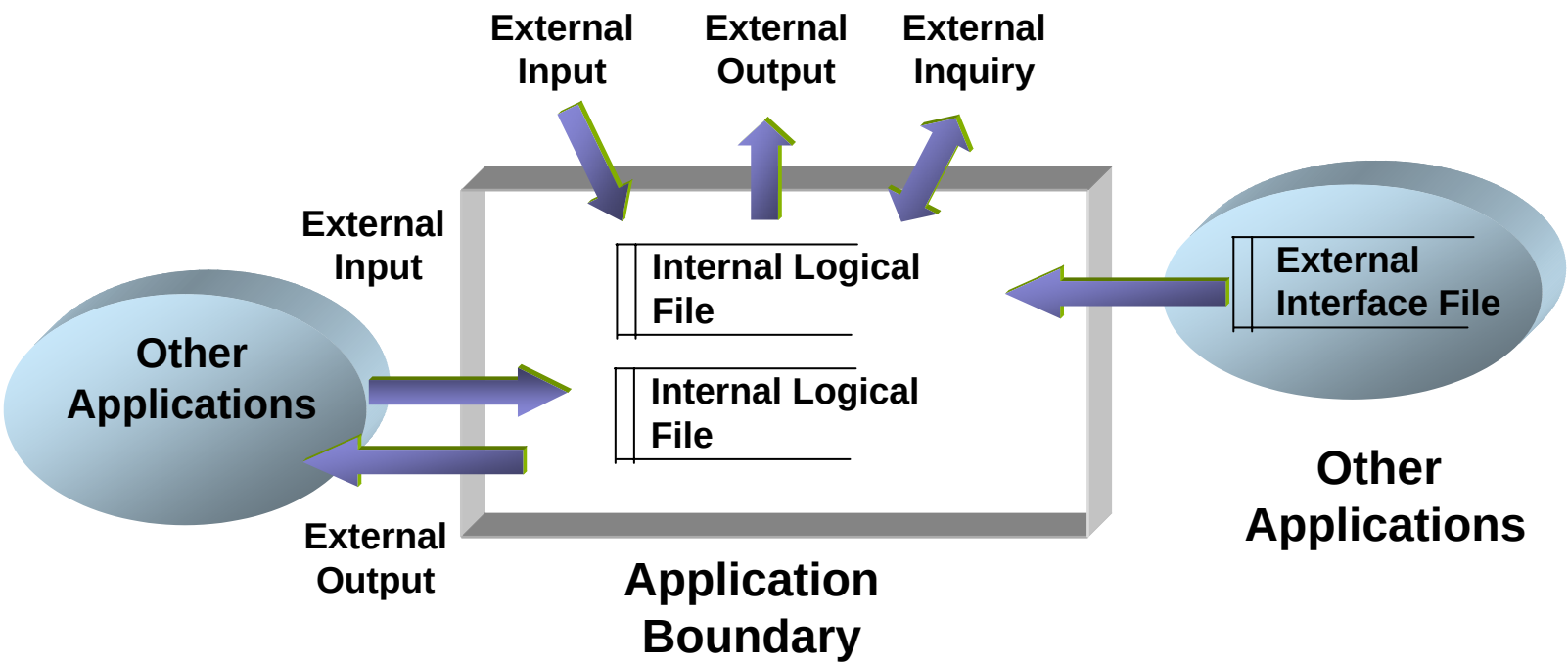
Identify the Boundary

- Definition
  - The boundary indicates the border between the project or application being measured and the user
- Application Boundary
  - Defines what is external to the application
  - Is the conceptual interface between the “internal” application and the “external” user world
  - Acts as a “membrane” through which data processed by transactions pass into and out of the application
  - Encloses the logical data maintained by the application
  - Assists in identifying the logical data referenced by but not maintained within an application
  - Is dependent on the user’s external business view of the application
  - Is independent of technology and implementation

Application Boundary Rules

- Is based on user's view
  - The focus is on what the user can understand and describe
- Between related applications is based on separate functional areas as seen by the user, not on technical considerations
- Initial boundary already established is not influenced by the counting scope

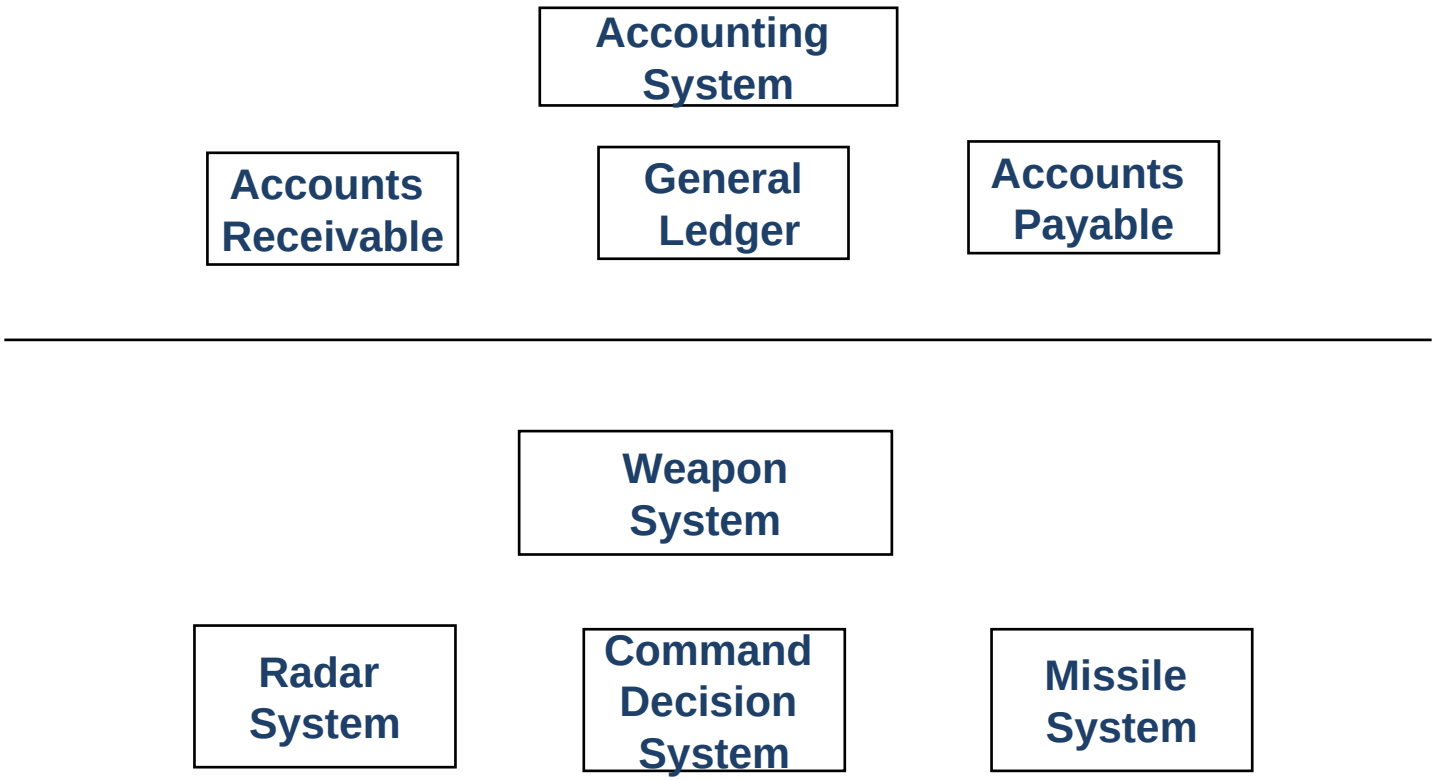
Diagramming an Application Boundary



Application Boundary Hints

- Use the system external specifications or get a system flow chart to draw a boundary around the functions
- Look at how the groups of data are maintained
- Identify functional areas by elementary processes
- Look at the associated measurement data
  - Effort, Cost, Defects
- The boundaries for function point counting and other measurement data should be the same

Application Boundary Examples



New Development Project Boundaries

- May include one or more new application(s) / subsystem(s)
- Count each application separately

Enhancement Project Boundary

- May include one or more:
  - New application(s) / subsystem(s)
  - Current application(s) / subsystem(s) being changed
  - Current application(s) / subsystem(s) not being changed
- Count only new or revised applications
- Count each application separately

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- ➡Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Data Counting Procedures

Step	Action
1	Identify the ILFs and EIFs
2	Determine the ILF or EIF complexity and their contribution to the unadjusted function point count

Data Functions

- Data functions represent the functionality provided to the user to meet internal and external data requirements
  - Internal Logical Files (ILFs)
  - External Interface Files (EIFs)

- The term *file* here does not mean file in the traditional data processing sense; in this case, file refers to a logically related group of data and not the physical implementation of those groups of data

Internal Logical Files

- An internal logical file (ILF) is a user identifiable group of logically related data or control information maintained within the boundary of the application
- The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted

ILF Identification Rules

- To identify ILFs, look for groups of data or control information that satisfy the definition of an ILF
- All of the following counting rules must apply for the information to be counted as an ILF:
  - The group of data or control information is logical and user identifiable
  - The group of data is maintained through an elementary process within the application being counted

External Interface Files

- An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application
- The primary intent of an EIF is to hold data referenced through one or more elementary processes within the application of the application counted; this means an EIF counted for the application must be an ILF in another application

EIF Identification Rules

- To identify EIFs, look for groups of data or control information that satisfy the definition of an EIF
- All of the following counting rules must apply for the information to be counted as an EIF:
  - The group of data or control information is logical and user identifiable
  - The group of data is referenced by, and external to, the application being counted
  - The group of data is not maintained by the application being counted
  - The group of data is maintained in an ILF of another application

Data Function Identification Process

- Determine the type of data – Business, Reference, or Code Data
- Identify Logical Files & Classify as Internal and External
  - Internal Logical File
    - Logical and user identifiable
    - Maintained through one or more elementary processes within the application being counted
  - External Interface File
    - Logical and user identifiable
    - Referenced by, and external to, the application being counted
    - Not maintained by the application being counted
    - Maintained in an ILF of another application

Business Data

- Business Data - Reflects the information needed to be stored and retrieved by the functional area addressed by the application.
  - Mandatory for the operation of the user's functional area
  - User identifiable (usually by business user)
  - User maintainable (usually by business user)
  - Stores the user's Core User Data to support business transactions
  - Very dynamic – normal business operations cause it to be regularly referenced and routinely added to, changed or deleted
  - Reported on
  - Typically has key field and many attributes
  - May have 0 to infinity records
- Examples: Customer Data

Reference Data

- Reference Data - Stored to support the business rules for the maintenance of the business data
  - Mandatory for the operation of the user's functional area
  - User identifiable and user maintainable
  - Stores the data to support core user activities
  - Maintained intermittently /Less dynamic – occasionally changes in response to changes in the functional areas' environment, external functional processes and/or business rules
  - Transactions processing business data often use reference data
  - Typically has key fields and few attributes
  - Usually at least one record or a limited number of records
- Examples: Policy Information (Types and Terms of Policies)

Code Data

- CODE Tables - Code data provides a list of valid value values that a descriptive attribute might have.

- Mandatory to the functional area, but optionally stored as a data file to standardize and facilitate business activities and business transactions
- Not usually identified as part of the functional requirements; it is usually identified as part of design to meet technical requirements
- Sometimes user maintainable, but essentially static – only changes in response to changes in the way that the business operates
- Business transactions access Code Data to improve ease of data entry, improve data consistency, ensure data integrity, etc.
- Generally consists of a key field and one or two attributes only
- Typically has a stable number of records

Code Data - Continued

- Typically the attributes of code data are code, description and/or other 'standard' attributes describing the code, i.e. standard abbreviation, effective date, termination date, audit trail data, etc.
- In code tables, the data can be substituted. For example, TX – Texas, OH – Ohio.
- Code (data) tables are considered part of the technical implementation, designed for efficiency and ease of application use.
- *Code tables are not countable by IFPUG standards even if the data is user recognizable and/or being maintained through an application by a user*.
- Since code data can not be an ILF or EIF, then it can not be a file type referenced in transactions.

ILF - What to Count

Count as Internal Logical Files --

- Business or Reference data maintained by the application
  - Data maintained by application required to produce transactions (inputs, outputs, queries)
- Application maintained Help
- Application maintained Security Files

ILF –What Not to Count!

- Files introduced because of technology or technical requirements
- Index files, Join/Merge Tables
- Development of prototypes, or files built but not used in application
- Code data/Valid Value tables
- Temporary data files (including those passed internally from job step to job)
- Data in copybooks
- Work files
- Sort files
- Files maintained by other applications and read only
- Separate physical entities if they are logically one data group

EIF – What to Count

- Count as External Interface Files
  - Business, reference data or edit criteria obtained or read from ILFs maintained by other applications
  - Application information not maintained within counted application, but required to produce outputs/inquiries
  - Help, Security File (if externally maintained as ILFs in another application)

EIF – What Not to Count!

- Data received from another application that adds, changes or deletes – these are transactions, not EIFs!
- File already counted as an ILF within counted application
- Data maintained by the application being counted
- Data formatted and sent to another application
- Code data provided by another application

*Note: The primary difference between an internal logical file (ILF) and an external interface file (EIF) is that an EIF is **not maintained** by the application being counted, while an ILF is.*

Complexity Rules for ILF/EIF - RET

- Definition
  - A record element type (RET) is a user recognizable subgroup (optional or mandatory) of data elements within an ILF or EIF
  - Optional subgroups are those that the user has the option of using one or none of the subgroups during an elementary process that adds or creates an instance of the data
  - Mandatory subgroups are those of which the user must use at least one
- Rules
  - Count an RET for each optional or mandatory subgroup of the ILF or EIF
  - If there are no subgroups, count the ILF or EIF as one RET

Complexity Rules for ILF/EIF - DET

- Definition
  - A data element type (DET) is a unique user recognizable, non-repeated field
- Rules
  - Count a DET for each unique user recognizable, non-repeated field maintained in or retrieved from the ILF or EIF through the execution of an elementary process
  - When two applications maintain and/or reference the same ILF/EIF, but each maintains/references separate DETs, count only the DETs being used by each application to size the ILF/EIF
  - Count a DET for each piece of data required by the user to establish a relationship with another ILF or EIF

Examples of DETS

- Data described as
  - Address, city, state, zip
- Counting impact:
  - Application 1 uses the information as a single block of data to create an address label; count as 1 DET
  - Application 2 uses the information as a customer's address, and considers each piece of data as a unique part; count as 4 DETs

ILF/EIF Complexity Matrix

- Assign each identified ILF and EIF a functional complexity based on the number of data element types (DETs) and record element types (RETs) associated with the ILF or EIF, within that application

(Sum of all fields of all subgroups of the logical data grouping)

INTERNAL LOGICAL OR EXTERNAL INTERFACE FILES:

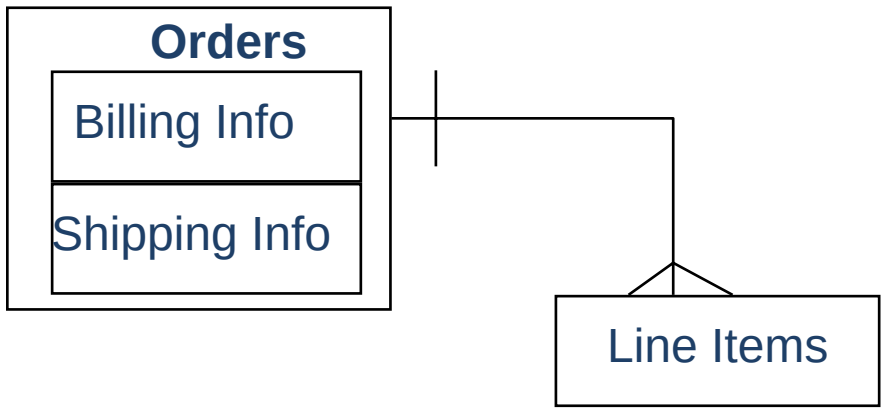
(Data Subgroups)

Record Element Types	Data Elements		
	1-19	20-50	51+
<2	Low	Low	Average
2-5	Low	Average	High
>5	Average	High	High

ERD and FPA

- Using the logical data model, circle those entities which are not dependent on other entities for their existence
  - Hint – entities having multiple keys from other entities are probably dependent on the other entity
- Entities not circled (those dependent on other entities for their existence) should be grouped with the entity to which it is dependent and included as part of its circle. Each dependent entity corresponds to Record Element Type in a Logical File.
- Consider foreign keys as a guide to better determine dependence/independence

Internal Logical File Example



User requires that an order includes billing information, shipping information and line item information before it is complete.

Count as 1 ILF with 3 Record Element Types:

- **Orders - Billing Information**
- **Orders - Shipping Information**
- **Line Items**

Internal Logical File Example

Entity	"Instructor Data" Element	Length
INSTRUCTOR	Name	30
	Last Name	
	First Name	
	Middle Init	
	Address	50
	City	30
	State	20
	Zip	9
	Telephone	10
COURSES TAUGHT		
	Instructor Name	30
	Last Name	
	First Name	
	Middle Init	
	Course Name	20
	Course Number	8

Courses Taught Dependent on Instructor

Sample Counting Form

Solution

Form used by DCG in examples

Description	ILF/EIF EI/EO/ EQ	RET	DET	Complexity
INSTRUCTOR DATA	**ILF**	**2**	**8**	**Low**

Internal Logical Files - Exercise

User requirements:

The ability to maintain CUSTOMER and ORDER information.

Marketing department needs to retrieve all past ORDER information, even if a CUSTOMER is deleted in order to develop marketing strategies for the following year.

The ability to view CUSTOMER in sorted order.

Internal Logical Files - Exercise

Count and rate the following Sales Order System files (tables):

CUSTOMER: Contains company name, street address, city, state, zip code, customer code, region, sales agent, billing street address, city, state and zip code.

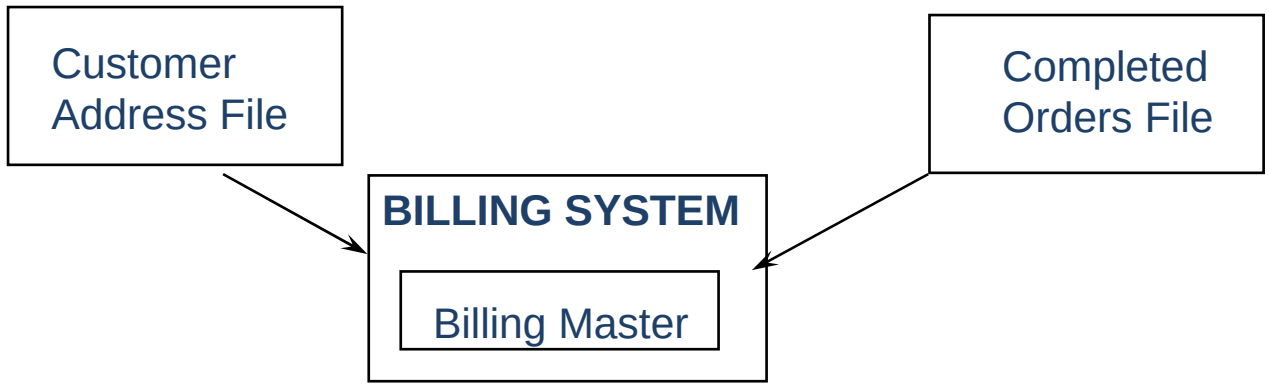
ORDERS: Record One (Ship to) contains customer code, record ID, "attention to" name (First Name, MI, Last Name), street address, city, state, zip code, carrier code, date to be shipped.
Record Two (Details) contains customer code, record ID, item code, item quantity, item dollar amount, bulk order discount code, total order dollar amount, tax code, tax amount, date order placed.
Record Three (Billing) contains customer code, record ID, company code, billing street address, city, state, zip code, orderer's name (First Name, MI, Last Name).

CUSTID: Contains same information as CUSTOMER sorted by customer code and is used in processing outputs.

Solution

Description	ILF/EIF EI/EO/ EQ	RET	DET	Complexity

External Interface File Example



The Customer Address File is used to print the addresses on the bills. No Billing System files are updated with this data.

Completed Orders File updates the Billing Master and cause the bills to be processed.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

An Example in Counting

1. Identify ILFs and EIFs from these High Level Requirements
2. Assume Average Complexity

- The user requirements for this example include:
  - The ability to enter, inquire and report employee information, including the location of a specific employee
  - The ability to inquire and report on information about the different locations within the company, including a list of employees at a particular location; the location data is for reference only, and is maintained by another application
  - The ability to enter, inquire and report on information about jobs
  - The ability to enter, inquire and report on job assignments for employees
  - The ability to send an employee job application for an open position to the department manager, and simultaneously update job assignments

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

Example in Counting – Design Phase

- Using the previous FP exercise:
  - Review the following Data Descriptions
  - Perform a detailed FP count (count the RETs and DETs for each Data Component)
  - Compute the FP count

ILF/EIF Counting Examples

The attributes contained in each of the entity types are listed below:

EMPLOYEE entity type

- **Name**
- **Social Security Number (SSN)**
- **Number-Dependents**
- **Type Code (salaried/hourly)**
- **SALARIED-EMPLOYEE entity type**
  - Supervisory-level
- **HOURLY-EMPLOYEE entity type**
  - Standard-Hourly-Rate
  - Collective-Bargaining-Unit-Number
- **Location-Name (Foreign Key)**

DEPENDENT entity type

- **Name**
- **Birth-Date**
- **Employee-SSN (Foreign Key)**

JOB entity type

- **Name**
- **Number**
- **Pay-Grade**
- **Description**

JOB-ASSIGNMENT entity type

- **Effective-Date**
- **Job Submittal Status (submitted, accepted)**
- **Salary**
- **Performance-Rating**
- **Job- Number (Foreign Key)**
- **Employee SSN (Foreign Key)**

LOCATION entity type

- **Name**
- **Address**
- **Building Number**
- **Site Description**

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- ➡ Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Transaction Function Process

Step	Action
1	Identify the elementary process
2	Determine the primary intent of the identified elementary processes, and classify as an EI, EO or EQ
3	Validate against the transaction (EI, EO, EQ) identification rules
4	Determine the transaction (EI, EO, EQ) complexity
5	Determine the transaction (EI, EO, EQ) contribution to the unadjusted function point count

Transaction Functions

- Transactional functions represent the functionality provided to the user for the processing of data by an application
- Transactional functions are defined by their elementary process and primary intent as follows:
  - For the Elementary Processes where the Primary Intent is to maintain an ILF or to alter the behavior of the system:
    - **External Inputs (EIs)**
  - For the Elementary Processes where the Primary Intent is to present information to the user and that perform calculations, derive data, update an ILF or alter the behavior of the system:
    - **External Outputs (EOs**)
  - For the Elementary Processes where the Primary Intent is to present information to the user and that do not perform calculations, derive data, update an ILF or alter the behavior of the system:
    - **External Inquiries (EQs)**

Definition of New Term

Processing Logic

- Requirements to complete an elementary process (not to determine separate EIs, EOs or EQs); an elementary process may include the following actions:
- Validations are performed
- Mathematical formulae or calculations are performed
- Equivalent values are retrieved
- Data is filtered and selected using specified criteria to compare multiple sets of data
- Conditions are analyzed to determine which are applicable
- One or more ILFs are updated
- One or more ILFs or EIFs are referenced
- Data or control information is retrieved
- Derived data is created
- Behavior of the system is altered
- Information is prepared and presented outside the boundary
- Data or control information that enters the application boundary is accepted
- Data is resorted or rearranged

External Inputs

- An external input is an elementary process that processes data or control information that comes from outside the application's boundary
- The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system

EI Identification Rules

All of the following counting rules must apply for data being processed to be counted as an external input:

- The data or control information is received from outside the application boundary
- At least one ILF is maintained if the data entering the boundary is not control information that alters the behavior of the system
- For the identified process, one of the three statements must apply:
  - Processing logic is unique from processing logic performed by other EIs for the application
  - The set of data elements identified are different from the sets identified for other EIs in the application
  - The ILFs or EIFs referenced are different from the files referenced by the other EIs in the application

EI Examples

Count as External Input

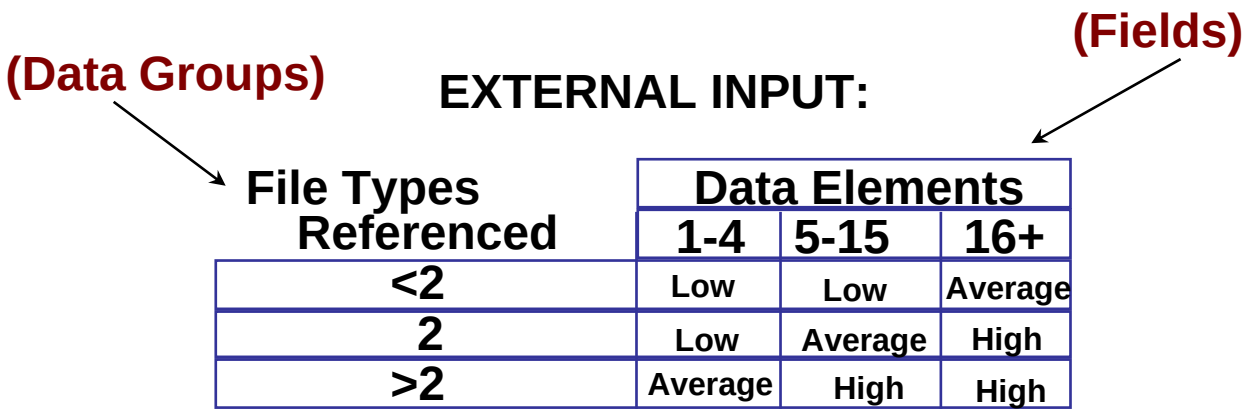
- Transactional data
- Screen or batch input
- Messages from other applications
- Control Information
- User functions
- Physical data which initiates processing

Don’t count as External Input

- Reference data (EIFs)
- Input side of EQ or EO
- Menu screens
- Logon screens
- Multiple methods of invoking similar logic
- Navigation

EI Complexity Matrix

Complexity for External Inputs is based on the number of logical groups (FTRs) and fields (DETs) for the EI



File Types Referenced	Data Elements		
	1-4	5-15	16+
<2	Low	Low	Average
2	Low	Average	High
>2	Average	High	High

Guideline: When estimating a FP count, use average unless High is obvious

Complexity Rules for EIs - FTR

- Definition
  - File Types Referenced (FTR)
    - Number of files referenced, read, created or updated
- Rules
  - Count an FTR for each ILF maintained
  - Count an FTR for each ILF or EIF read during the processing of the EI
  - Count only one FTR for each ILF that is both maintained and read by the EI

Complexity Rules for EIs - DET

- Count only one DET for each user recognizable, non-repeated field that enters or exits the application boundary and is required to complete the EI
- Do not count fields that are retrieved or derived by the system and stored on an ILF during the elementary process if the fields did not cross the application boundary
- Count one DET for the capability to send a system response message outside the application boundary to indicate an error occurred during the processing, confirm the processing is complete or verify that processing should continue
- Count one DET for the ability to specify an action to be taken even if there are multiple methods for invoking the same logic

An Example in Counting

1. Identify EIs from these High Level Requirements
2. Assume Average Complexity

- The user requirements for this example include:
  - The ability to enter, inquire and report employee information, including the location of a specific employee
  - The ability to inquire and report on information about the different locations within the company, including a list of employees at a particular location; the location data is for reference only, and is maintained by another application
  - The ability to enter, inquire and report on information about jobs
  - The ability to enter, inquire and report on job assignments for employees
  - The ability to send an employee job application for an open position to the department manager, and simultaneously update job assignments

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Input Examples

On-line

Maintain Education Information

Employee Number (PK)____
Course Date (PK)___
Course Name (PK)____

Notes: _____
_____

Action ____ (A, C OR D)

(Error/Confirm Messages are returned)

STUDENT ILF is updated.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Input Example

Course Registration Screen

NAME _____ START DATE (PK)_____

SSN (PK)_____

COURSE # (PK)_____

COURSE TITLE _____

LOCATION _____ ACTION (A, C OR D)____

(Both Error and Confirmation Messages Occur)

STUDENT and COURSE are updated/referenced.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Input Exercises

Identify the following in the Sales Order System:

1. Once a year the sales manager submits a request for processing a year-end sales report, which contains start and end dates and region numbers for the report; no files are updated during this process.

2. Bulk orders are entered into the system via the Bulk Order File from the Regional System. The transaction contains 29 fields and updates the ORDERS file. One of the fields contains an action field with the values A (add) or C (change).

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Outputs

- An external output (EO) is an elementary process that sends data or control information outside the application's boundary
- The primary intent of an external output is to present information to a user through the processing logic other than or in addition to the retrieval of data or control information
- The processing logic must contain at least one mathematical formula or calculation, create derived data, maintain one or more ILFs and/or alter the behavior of the system

EO Identification Rules

All of the following counting rules must apply for the elementary process to be counted as an external output:

- Sends data or control information external to the application's boundary
- For the identified process, one of the three statements must apply:
  - Processing logic is unique from the processing logic performed by other EOs for the application
  - The set of data elements identified are different from other EOs in the application
  - The ILFs or EIFs referenced are different from files referenced by other EOs in the application

EO Identification Rules

In addition, one of the following rules must apply

- The processing logic contains at least one mathematical formula or calculation
- Derived data is created
- The processing logic maintains at least one ILF
- The processing logic alters the behavior of the system

External Output Examples

- Count as External Output
  - Data transfers to other applications with calculations
  - Reports with calculations
  - On-line reports with calculated information
  - Derived data displayed on a screen
  - Message/file sent to another system, that includes an update (send a check for payment, and update an ILF that the check has been processed)
  - Note: Summary fields on a report are not counted as a separate EO; these computed fields are DETs within the same EO
- Don't count as External Output
  - Reports without other processing (calculations, update, etc.)
  - Identical reports/unique data values
  - Resorting or rearrangement of a set of data without other processing logic

EO Complexity Matrix

Complexity is based on the number of FTRs/DETs for the EO

EXTERNAL OUTPUT:

(Data Groups)

(Fields)

File Types Referenced	Data Elements		
	1-5	6-19	20+
<2	Low	Low	Average
2 or 3	Low	Average	High
>3	Average	High	High

Complexity Rules for EOs - FTR

- Definition
  - File Types Referenced (FTR)
    - The number of logical groups of data (internal or external) that are referenced, read, created or updated
- Rules
  - Count an FTR for each ILF or EIF read during the processing of the elementary process
  - Count an FTR for each ILF maintained during the processing of the elementary process
  - Count only one FTR for an ILF that is both maintained and read by the elementary process

Complexity Rules for EOs - DET

The number of DETs is equal to that total number of fields identified from the following rules:

- Count one DET for each user recognizable, non-repeated field that enters that application boundary and is required to specify when, what and/or how the data is to be retrieved or generated by the elementary process
- Count one DET for each user recognizable field that exits the boundary
- If a DET both enters and exits the boundary, count that DET only once for the elementary process

Complexity Rules for Eos - DET

- Count one DET for the capability to send a system response message outside the boundary to indicate an error occurred during the processing, confirm that the processing is complete or verify that processing should continue
- Count one DET for the ability to specify an action to be taken even if there are multiple methods for invoking the same logical process
- Do not count fields that are saved, retrieved or derived if they did not cross the boundary
- Do not count literals, paging or system-generated stamps

An Example in Counting

1. Identify EOs from these High Level Requirements
2. Assume Average Complexity

- The user requirements for this example include:
  - The ability to enter, inquire and report employee information, including the location of a specific employee
  - The ability to inquire and report on information about the different locations within the company, including a list of employees at a particular location; the location data is for reference only, and is maintained by another application
  - The ability to enter, inquire and report on information about jobs
  - The ability to enter, inquire and report on job assignments for employees
  - The ability to send an employee job application for an open position to the department manager, and simultaneously update job assignments

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Output Example

College Credits Report

Student	Course Number	Course Date	Credits
nnnnnnn, nnnnn n.	xxxxxxx	xx/xx/xxxx	xx
nnnnnnn, nnnnn n.	xxxxxxx	xx/xx/xxxx	xx
nnnnnnn, nnnnn n.	xxxxxxx	xx/xx/xxxx	xx
Total Students for Course Number xxxxx: 15			
nnnnnnn, nnnnn n.	xxxxxxx	xx/xx/xxxx	xx
Total Students for Course Number xxxxx: 15			

STUDENT ILF is referenced.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Output Example

Course Roster Report

Course Name	xxxx	Course Number	xxx
Course Location	xxxxx	Instructor Name	xxxxxxx

Student Name	Address	SSN
xxxxxxx, xxxx x.	xxxxxxxxxxxx xxxxxxxxxxxx	nnn-nn-nnnn
xxxxxxx, xxxx x.	xxxxxxxxxxxx xxxxxxxxxxxx	nnn-nn-nnnn

TOTAL NUMBER OF STUDENTS nnnn

STUDENT and COURSE are referenced.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Output Exercise

Identify the following in the Sales Order System:

1. There are 20 Sales Reports issued every month, one per region, with totals per region. The reports are identical and contain 13 fields using the CUSTOMER and ORDER files.

2. Once a day the Sales Order System sends a file to the Inventory System. This file contains information about the items ordered and has 8 fields from the ORDER file. The Inventory System uses this data to change their inventory amounts.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Inquiries

- An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary
- The primary intent of an external inquiry is to present information to the user through the retrieval of data or control information
- The processing logic contains no mathematical formula or calculation, and creates no derived data
- No ILF is maintained during the processing, nor is the behavior of the system altered

EQ Identification Rules

- All of the following must apply
  - Sends data or control information external to the application's boundary
  - For the identified process, one of the three statements must apply:
    - Processing logic is different from the processing logic performed by other EQs for the application
    - The set of data elements identified are different from other EQs in the application
    - The ILFs or EIFs referenced are different from files referenced by other EQs in the application

EQ Identification Rules

- In addition, all of the following rules must apply:
  - The processing logic retrieves data or control information from one or more ILFs or EIFs
  - The processing logic does not contain mathematical formula or calculation
  - The processing logic does not create derived data
  - The processing logic does not alter the behavior of the system
  - The processing logic does not maintain an ILF

External Inquiry Examples

- Count as an External Inquiry
  - User functions such as view, lookup, display, browse, drop down
  - Logon screens that provide application specific security
  - Help - each level
  - Output that contains no calculated, derived data
- Don’t count as External Inquiry
  - Multiple methods of invoking the same inquiry logic
  - Inquiries that can be accessed from multiple areas/screens of an application (count once)
  - Logon screens that do not provide security
  - Derived data vs. retrieval of data
  - Resorting or rearrangement of the same set of data without other processing logic

EQ Complexity Matrix

Complexity is based on the number of FTRs/DETs for the EQ

EXTERNAL INQUIRY:

(Data Groups) → File Types Referenced

(Fields) → Data Elements

File Types Referenced	Data Elements 1-5	6-19	20+
1	Low	Low	Average
2 or 3	Low	Average	High
>3	Average	High	High

Complexity Rules for EQs - FTR

- Definition
  - File Types Referenced (FTR)
    - Number of logical groups of data (internal or external) that are referenced or read
- Rules
  - Count an FTR for each ILF or EIF read during the processing of the elementary process

Complexity Rules for EQs - DET

The number of DETs is equal to that total number of fields identified from the following rules:

- Count one DET for each user recognizable, non-repeated field that enters that application boundary and is required to specify when, what and/or how the data is to be retrieved or generated by the elementary process
- Count one DET for each user recognizable field that exits the boundary
- If a DET both enters and exits the boundary, count that DET only once for the elementary process

Complexity Rules for EQs - DET

- Count one DET for the capability to send a system response message outside the boundary to indicate
- An error occurred during the processing, confirm that the processing is complete or verify that processing should continue
- Count one DET for the ability to specify an action to be taken even if there are multiple methods for invoking the same logical process
- Do not count literals, paging or system-generated stamps

An Example in Counting

1. Identify EQs from these High Level Requirements
2. Assume Average Complexity

- The user requirements for this example include:
  - The ability to enter, inquire and report employee information, including the location of a specific employee
  - The ability to inquire and report on information about the different locations within the company, including a list of employees at a particular location; the location data is for reference only, and is maintained by another application
  - The ability to enter, inquire and report on information about jobs
  - The ability to enter, inquire and report on job assignments for employees
  - The ability to send an employee job application for an open position to the department manager, and simultaneously update job assignments

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ FTR	DET	Complexity

External Inquiry Examples

Screen 1 - input

Restaurant Selection

S	Type
	Italian
x	Mexican
	Bistro

Screen 2 - output

Restaurant Selection

Name:	TACO BELL
Location:	EVERY CORNER
Golden Spoon Rating	**-2**
Price Range	CHEAP
Reservations	Never Needed

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ DET	FTR	Complexity

External Inquiry Examples

College Credits Report

Student	Course Number	Course Date	Credits
xxxxxxxxxxx, xxx	nnnnnnn	nn/nn/nnnn	nnnn
xxxxxxxxxxx, xxx	nnnnnnn	nn/nn/nnnn	nnnn
xxxxxxxxxxx, xxx	nnnnnnn	nn/nn/nnnn	nnnn
xxxxxxxxxxx, xxx	nnnnnnn	nn/nn/nnnn	nnnn
xxxxxxxxxxx, xxx	nnnnnnn	nn/nn/nnnn	nnnn

(Info is retrieved from Course ILF)

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ DET	FTR	Complexity

External Inquiry Exercise

Identify the following in the Sales Order System:

1. Using a selection of orders from the main menu, the user can request a display list of orders that were shipped last week. The list contains date ordered, date shipped, company name, customer code and total order dollar amount. All of the information is retrieved from the ORDER file.

2. From the list of shipped orders, the user can select one of the orders. This selection displays the details of the shipped order. This detail display has 22 additional fields and uses the ORDER file.

Solution

Description	ILF/EIF EI/EO/ EQ	RET/ DET	FTR	Complexity

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- ➡ Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Summary of Functions (EI, EO, EQ)

Function	EI	EO	EQ
Alter the behavior of the system	PI	F	N/A
Maintain one or more ILF	PI	F	N/A
Present information to the user	F	PI	PI

Legend:

PI: primary intent of the transaction function

F: function of the transaction function, but is not the primary intent and is sometimes present

N/A: the function is not allowed by the transaction function

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- ➡ Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- Calculations

Summary Of Processing Logic (EI, EO, EQ)

PROCESSING LOGIC	EI	EO	EQ
Validations are performed	C	C	C
Mathematical formulae or calculations are performed	C	M*	N
Equivalent values are converted	C	C	C
Data is filtered and selected using specified criteria to compare multiple sets of data	C	C	C
Conditions are analyzed to determine which are applicable	C	C	C
At least one ILF is updated	M*	M*	N
At least one ILF or EIF is referenced	C	C	M
Data or control information is retrieved	C	C	M
Derived data is created	C	M*	N
Behavior of the system is altered	M*	M*	N
Prepare and present information outside the boundary	C	M	M
Capability to accept data or control information that enters the application boundary	M	C	C
Resorting or rearranging of data	C	C	C

Summary Of Processing Logic (EI, EO, EQ)

Legend:

M	it is mandatory that the function type perform the form of processing logic
M*	it is mandatory that the function perform at least one of these (M*) forms of processing logic
C	The function can perform the form of processing logic, but it is not mandatory
N	the function can not perform the form of processing logic

Transaction Guidelines

- Online Screens
  - One online screen may perform multiple logical transactions and have multiple functions (EI - Add, EI - Change, EI - Delete, EQ, EO). Multiple online screens can make up one logical function. The key is to identify each logical function.
- Batch/Internal Processing
  - Batch processing is counted by assessing the primary intent of the process. If it is to produce an output, an output (EO) is counted. If it is to update internal databases, external inputs are counted according to the functionality involved (EI – Add, EI – Change, EI – Delete). One batch/internal job can perform multiple processes and multiple batch/internal jobs can perform one process. The key is to identify each logical process.

Transaction Guidelines

- Drop Down List Boxes
  - Each drop down list box that requires unique processing for population should be counted as a low query (EQ). The drop down list box must reference an ILF or EIF to be counted.
- Security Functions
  - Usually 1 low Query to check a user's validity.
  - A security screen which allows the user to change their password should be counted as an EI.
- Interface Files
  - Interface files coming into the application will be counted as one EI for each function performed (Add, Change, Delete). Interface files being sent out of the application will be counted as an EO or EQ.
  - Interface files that perform transaction functions are not EIFs.

Transaction Guidelines

- Reports generated by the application in accordance with user requirements are countable.
  - Evaluate as an EO or an EQ.
- Same Business Function, Different Delivery Mechanism
  - Where the same input/output data is provided by delivery methods requiring different formatting, each input/output cannot be counted separately unless there is *unique processing logic* used for each delivery mechanism.
    - Examples of different delivery mechanisms are:
      - formatted output to a printer versus print to a text file; and,
      - output to a character terminal versus output to a Web browser
    - Examples of unique processing logic are:
      - additional calculations;
      - additional derived data; and
      - additional validations on the data

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- ➡ General System Characteristics
- Calculations

General System Characteristics

- Data Communication
- Distributed Data or Processing
- Performance Objectives
- Heavily Used Configuration
- Transaction Rate
- On-Line Data Entry
- End-User Efficiency
- On-Line Update
- Complex Processing
- Reusability
- Conversion & Install Ease
- Operational Ease
- Multiple-Site Use
- Facilitate Change

Value Adjustment Factor (VAF)

- The value adjustment factor looks at 14 general system characteristics.
- Each characteristic is given a value
- Once all 14 GSCs have been evaluated, the VAF is computed:

$$\Sigma(DI) = TDI$$

$$(TDI * .01) + .65 = VAF$$

Determining Degree of Influence

- Based on the stated user requirements, each general systems characteristic (GSC) must be evaluated in terms of its degree of influence (DI) on a scale of zero to five.

Score As	System Influence
0	Not Present or no influence
1	Incidental influence
2	Moderate influence
3	Average influence
4	Significant influence
5	Strong influence throughout

General System Characteristic Guidelines

- Each of the following general system characteristic descriptions includes guidelines to determine the degree of influence.
- Each guideline contains:
  - A definition of the GSC
  - Rules for determining the score
  - If the rule needs further clarification, hints have been provided to help apply the rules consistently across all platforms.
  - Hints are not intended to cover all situations but are meant to provide additional guidance in determining the appropriate score

GSC1: Data Communications

Definition

- Data Communications describes the degree to which the application communicates directly with the processor.
- The *data* and *control* information used in the application are sent or received over communication facilities.
- Devices connected locally to the control unit are considered to use communication facilities.
- Protocol is a set of conventions that permit the transfer or exchange of information between two systems or devices.
- All data communication links require some type of protocol.

GSC 1: Data Communications

Score

Score As	Descriptions to Determine Degree of Influence
0	Application is pure batch processing **or** a stand-alone application.
1	Application is batch but has remote data entry ***or*** remote printing.
2	Application is batch but has remote data entry ***and*** remote printing.
3	Application includes on-line data collection or TP (teleprocessing) front end to a batch process or query system.
4	Application is more than a front-end, but supports ***only one*** type of TP communications.
5	Application is more than a front-end, and supports ***more than one*** type of TP communications protocol.

Hints

Protocol examples include FTP, dial-up, Token Ring, Ethernet, SNA, TCP/IP, IPX/SPX, HTTP, XML, WAP, NTP, ICQ and NETBEUI. This list should *not* be considered exhaustive.

Hints for Rules 1 & 2

Remote devices might include 3270 terminal connected to a mainframe computer that allows only simple edits (numeric vs. alpha), or printers connected via parallel port (the user can specify where to direct the output).

The entry of data does not involve reading or writing directly to an ILF.

Data are entered on-line, but the transactions are stored in a temporary file for batch update of ILF(s) at a later time.

The entry of data does not involve reading or writing directly to an ILF.

GSC 1: Data Communications

Hints for Rule 3

Simple business rules and minimal edits (e.g., alpha/numeric, range check, required data, etc.) may be performed.

When this data is eventually processed by the application, additional edits are performed.

The entry of data does not involve reading or writing directly to an ILF.

Data are entered on-line, but the transactions are stored in a temporary file for batch update of ILF(s) at a later time.

Hints for Rule 4

Data for the application is collected and may directly update ILF(s) or be stored for future processing using an input device, which performs edits based on business rules. Only one communication protocol is used.

Typically, when this data is processed by the application, no further edits are required.

The entry of data involves reading or writing to an ILF.

For example, client-server data entry or Internet data entry, but not both.

GSC 1: Data Communications

Hints for Rule 5

Same as 4; however,
Data collection is performed using multiple telecommunication protocols.

For example, client-server data entry and Internet data entry of the same transaction.

Typically for GSC 1

- Batch applications receive a score of 0 to 3
- On-line applications receive a score of 4
- Web-based applications receive a score of 4 or 5
- Real-time, telecommunication, or process control systems receive a score of 4 or 5

GSC2: Distributed Data Processing

Definition

- Distributed Data Processing describes the degree to which the application transfers data among physical components of the application.
- Distributed data or processing functions are a characteristic of the application within the application boundary.

GSC 2: Distributed Data Processing

Score

Score As	Descriptions to Determine Degree of Influence
0	Data is not transferred or processed on another component of the system.
1	Data is prepared for transfer, then is transferred and processed on another component of the system, for user processing.
2	Data is prepared for transfer, then is transferred and processed on another component of the system, ***not*** for user processing.
3	Distributed processing and data transfer are on-line and in **one** direction only.
4	Distributed processing and data transfer are on-line and in **both** directions.
5	Distributed processing and data transfer are on-line and are dynamically performed on the most appropriate component of the system.

GSC 2: Distributed Data Processing

Hints

Distributed data processing by definition is not an application that is contained on a central processor, which sends data to other applications.

In a distributed environment, the application is viewed as requiring multiple components (hardware) on which certain processing or data resides.

A knowledgeable user would usually recognize this configuration.

For example, in the typical 3-tier model, user interface processing is done in the PC at the user's location, business processing is done in a remote computer, and database access and processing is done in another computer that provides centralized access for many business processes

GSC 2: Distributed Data Processing

Hints for Rule 0

Presentation, processing and I/O components are all in the same place (i.e., stand-alone applications).

Hints for Rule 1

Application downloads data to a user's client machine, so the user can use Excel or other reporting tools to prepare graphs and perform other analysis.

Process that transfers data from mainframe to an external component for user processing. This transfer is performed using a simple protocol such as FTP.

Transferred to a user for processing

GSC 2: Distributed Data Processing

Hints for Rule 2

Process that transfers data from mainframe to mid-tier. For example, processing with SAS-PC.

Application sends data to client or server. This data is then processed or used to produce reports, etc. No data or confirmation is sent back to the client or server.

Transferred to a component for processing.

Hints for Rule 3

Data is sent between client and server in **one** direction only.

This data is then processed or used to produce reports, etc. by the receiving application.

This data typically includes transactions that update an ILF on the client or server.

For example - client-server or web-enabled applications.

GSC 2: Distributed Data Processing

Hints for Rule 4

Data is sent between client and server in **either** direction. This data is then processed or used to produce reports, etc. by the receiving application.

This data typically includes transactions that update an ILF on the client or server. For example - client-server or web-enabled applications.

The application runs under an operating system that automatically handles the allocation between components, **however,** the use of the operating system did not influence the design and implementation of the application

Hints for Rule 5

The developer must consider special application software that looks at multiple processors and runs the application on a specific type of processor.

This is invisible to the user.

The application runs under an operating system that automatically handles the dynamic allocation between components, and the use of the operating system specifically influenced the design and implementation of the application

GSC 2: Distributed Data Processing

Typically for GSC 2:

- Most applications, including legacy applications, receive a score of 0
- Primitive distributed applications that include batch applications in which data is not transferred on-line receive a score of 1 to 2
- Client-server or web-based applications receive a score of 3 to 4
- It is uncommon to score 5 - There must be multiple servers or processors, each of which would be selected dynamically on the basis of its real-time availability to score 5

GSC 3: Performance

Definition

- Performance describes the degree to which response time and throughput performance considerations influenced the application development.

- Application performance objectives, stated or approved (or implied) by the user, in either response or throughput, influence (or will influence) the design, development, installation and support of the application.

GSC 3: Performance

Score

Score As	Descriptions to Determine Degree of Influence
0	No special performance requirements were stated by the user.
1	Performance and design requirements were stated and reviewed but no special actions were required.
2	Response time or throughput is critical during ***peak*** hours. No special design for CPU utilization was required. Processing deadline is for the next business cycle.
3	Response time or throughput is critical during ***all business*** hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
4	In addition, stated user performance requirements are stringent enough to require performance analysis tasks in the design phase.
5	In addition, performance analysis tools were used in the design, development, and/or implementation phases to meet the stated user performance requirements.

Hints

GSCs 3, 4 and 5 are somewhat related.

For this GSC, think in terms of "How fast can we make the application go and how much did/does that impact the design, development and/or implementation?“

The users may require real time access to their data, stating or implying standards for response time and throughput capacity.

Response time typically relates to interactive processing; throughput relates to batch processing.

GSC 3: Performance

Typically for GSC 3:

- Batch applications receive a score of 0 to 4
- On-line (including interactive client-server or web-enabled)
- applications receive a score of 0 to 4
- Most MIS on-line systems receive a score of 2
- Real-time, telecommunication, or process control systems receive
- a score of 0 to 5
- A score of 5 requires the use of performance analysis tools

GSC 4: Heavily Used Configuration

Definition

Heavily Used Configuration describes the degree to which computer resource restrictions influenced the development of the application.

A heavily used operational configuration may require special considerations when designing the application. For example, the user wants to run the application on existing or committed equipment that will be heavily used

GSC 4: Heavily Used Configuration

Score

Score As	Descriptions to Determine Degree of Influence
0	No explicit or implicit operational restrictions are included.
1	Operational restrictions do exist, but are less restrictive than a typical application. No special effort is needed to meet the restrictions.
2	Operational restrictions do exists, but are typically for an application. Special effort through controllers or control programs is needed to meet the restrictions.
3	Stated operational restrictions require special constraints on ***one*** piece of the application in the central processor or a dedicated processor.
4	Stated operational restrictions require special constraints on the ***entire*** application in the central processor or a dedicated processor.
5	In addition, there are special constraints on the application in the distributed components of the system.

GSC 4: Heavily Used Configuration

Hints

- GSCs 3, 4 and 5 are somewhat related. For this GSC think in terms of "How much does the infrastructure influence the design?“
- Examples of operational restrictions may include the following (not an exhaustive list):
  - 1. This question indicates that the application must run on a computer that is under- powered and can not adequately handle the new or changed functionality and that somehow the developers can overcome this by developing the application differently.
  - More than one application accessing the same data can create operational restrictions.
  - 3. Application competing for the same resource and technologies with the potential deadlocks must be tuned and constrained to avoid performance degradation.

GSC 4: Heavily Used Configuration

Typically for GSC 4:

- Most applications receive a score of 2
- Client-server, web-enabled, real-time, telecommunication or process control systems receive a score of 3 to 5, but then you would need either a dedicated processor or multiple processors processing the same transactions and searching for the most expeditious means of processing.

Definition

Transaction Rate describes the degree to which the rate of business transactions influenced the development of the application.

The transaction rate is high, and it influences the design, development, installation and support of the application. Users may require what they regard as normal response time even during times of peak volume.

GSC 5: Transaction Rate

Score

Score As	Descriptions to Determine Degree of Influence
0	No peak transaction period is anticipated.
1	Low transaction rates have minimal effect on the design, development and installation phases.
2	Average transaction rates affect the design, development, and/or installation phases.
3	High transaction rates affect the design, development, and/or installation phases.
4	High transaction rate(s) stated by the user in the application requirements or service level agreements re high enough to require performance analysis tasks in the design, development, and/or installation phases.
5	High transaction rate(s) stated by the user in the application requirements or service level agreements re high enough to require performance analysis tasks; and in addition require the use of performance analysis tools in the design, development, and/or installation phases.

GSC 5: Transaction Rate

Hints

GSCs 3, 4 and 5 are somewhat related. For this GSC think in terms of "How many transactions can be processed by the application in a given period of time?“

Often this score is the same as the score for GSC 3 because transaction rates often influence performance requirements

GSC 5: Transaction Rate

Typically for GSC 5:

- Batch applications receive a score of 0 to 3
- On-line (including interactive client-server or web-
- enabled) applications receive a score of 0 to 4
- Real-time, telecommunication, or process control systems
- receive a score of 0 to 5
- A score of 5 requires the use of performance analysis tools

Definition

On-line User Interface describes the degree to which data is entered or retrieved through interactive transactions

On-line User Interface for data entry, control functions, reports and queries are provided in the application

GSC 6: On-Line Data Entry

Score

Score As	Descriptions to Determine Degree of Influence
0	All transactions are processed on a batch mode.
1	1% to 7% of transactions are interactive.
2	8% to 15% of transactions are interactive.
3	16% to 23% of transactions are interactive.
4	24% to 30% of transactions are interactive.
5	More then 30% of transactions are interactive.

Hints

This refers to types of transactions *not* volumes.

For example, if an application has 45 EIs, EOs and EQs, what percent of the EIs, EOs and EQs are accomplished via on-line transactions

GSC 6: On-Line Data Entry

Typically for GSC 6:

- Batch applications receive a score of 0 to 1
- On-line, real-time, telecommunication, or process control systems receive a score of 5
- Most contemporary on-line (including interactive client-server or web-enabled) applications receive a score of 5
- Batch systems with on-line features may have a lot of batch transactions, but it must be at least 71 percent batch to receive a score of less than 5

Definition

User Efficiency describes the degree of consideration for human factors and ease of use for the user of the application measured.

The on-line functions provided emphasize a design for user efficiency (human factor/user friendliness). The design includes:

- Menus
- Navigational Aids (e.g., function keys, jumps, dynamically generated menus, hyper-links)
- On-line Help and Documents
- Automated Cursor Movement (continued)

GSC 7: End User Efficiency

- Scrolling
- Remote Printing (via Online Transactions)
- Pre-Assigned Function Keys (e.g., clear screen, request help, clone screen)
- Batch jobs submitted from on-line transactions
- Drop down List box
- Heavy use of reverse video, highlighting, colors, underlining and other indicators
- Hard-copy documentation of on-line transactions
- Mouse interface
- Pop-up windows
- Templates and/or defaults
- Bilingual Support (supports two languages: count as four items)
- Multi-lingual Support (supports more than two languages: count as six items)

GSC 7:End User Efficiency

Score

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	One to three of the above.
2	Four to five of the above.
3	Six or more of the above, but there are no specific user requirements related to efficiency.
4	Six or more of the above, and stated requirements for user efficiency are strong enough to require **design tasks** for human factors to be included.
5	Six or more of the above, and stated requirements for user efficiency are strong enough to require **use of special tools and processes** in order to demonstrate that the objectives have been achieved.

GSC 7:End User Efficiency

Hints

Use a convention of a score of 4 whenever the application is deployed in a GUI environment (unless it scores 5).

Usually only software environments that prepare applications for mass-market or non-technical users score 5, and only if they have ergonomics specialists and/or usability studies as part of their process.

GSC 7: End User Efficiency

Typically for GSC 7:

- Pure batch applications receive a score of 0
- Character mode user interface receive a score of 1 or possibly a 2
- GUI user interface to be used for low volume transactions receive a score of 3
- GUI user interface to be used for high volume transactions and most Web *Intranet* user interfaces receive a score of 4 (requires design tasks for human factors)
- Web *Internet* user interfaces receive a score of 5 (requires special tools and processes to demonstrate that the objectives have been achieved)

GSC 8: On Line Update

Definition

On-line Update describes the degree to which internal logical files are updated on-line.

The application provides on-line update for the internal logical files.

GSC 8: On Line Update

Score

Score As	Descriptions to Determine Degree of Influence
0	None.
1	On-line update of one to three control files is included. Volume of updating is low and recovery is easy.
2	On-line update of four or more control files is included. Volume of updating is low and recovery is easy.
3	On-line update of major internal logical files is included.
4	In addition, protection against data loss is essential and has been specially designed and programmed in the system.
5	In addition, high volumes bring cost considerations into the recovery process. Highly automated recovery procedures with minimum human intervention are included.

GSC 8: On Line Update

Hints

On-line update usually requires a keyed file or database. Automatic recovery provided by the operating system counts if it impacts the application.

GSC 8: On Line Update

Typically for GCS 8:

- Pure batch applications receive a score of 0.
- On-line updates of files that modify the way an application processes or validates data receive a score of 1 or 2.
- On-line updates of user persistent data receive a score of 3.
- MIS applications receive a score of 3 or less.
- Most GUI type applications receive a score of 3 or above.
- Applications which use programmed recovery such as SQL roll back and commit receive a score of 4. Operational/routine backup is not considered protection against data loss.
- Applications required to recover data, reboot, or perform other self-contained functions in the event of a system error receive a score of 5. Recovery may require a human to press enter or perform some other minimal function to initiate this process

GSC 9: Complex Process

Definition

Complex processing describes the degree to which processing logic influenced the development of the application. The following components are present:

- Sensitive control and/or application-specific security processing.
- Extensive logical processing
- Extensive mathematical processing
- Much exception processing, resulting in incomplete transactions that must be processed again.
- Complex processing to handle multiple input/output possibilities

GSC 9: Complex Processing

Score

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	Any one of the above.
2	Any two of the above.
3	Any three of the above.
4	Any four of the above.
5	Any five of the above.

GSC 9: Complex Processing

Hints

Sensitive control or security process (e.g., individual users would have different access authority to screens where they could view and/or change data) may include special audit processing (audit data would be captured whenever data was viewed and/or changed and reported).

Application-specific security processing may include internally developed security processing or use of purchased security packages.

Extensive logical processing is Boolean logic (use of 'AND', 'OR') of greater than average difficulty or a minimum of 4 nested conditional (IF, CASE) statements. Extensive logical processing does not occur in most MIS applications.

GSC 9: Complex Processing

Hints

Extensive mathematical processing is arithmetic that is beyond the capability of a 4-function calculator (add, subtract, multiply, divide). This is usually not present in most MIS applications. However, an engineering application may qualify.

Exception processing includes incomplete ATM transactions caused by TP interruption, missing data values, failed validations, or cycle redundancy checks which can be used to recreate lost pieces of data

Multiple input/output possibilities include multi-media, device independence, voice, OCR reading, barcode reading, retinal scanning and Breathalyzer analysis.

Typically for GSC 9:

Scoring is not platform dependent

GSC 10: Reusability

Definition

Reusability describes the degree to which the application and the code in the application have been specifically designed, developed and supported to be usable in *other* applications.

GSC 10: Reusability

Score

Score As	Descriptions to Determine Degree of Influence
0	No reusable code.
1	Reusable code is within the application.
2	Less than 10% of the application code developed is intended for use in more than one application.
3	Ten percent (10%) or more of the application code developed is intended for use in more than one application.
4	The application was specifically packaged and/or documented to ease reuse, and the application is customized at the source code level.
5	The application was specifically packaged and/or documented to ease reuse, and the application is customized for use by means of user parameters maintenance.

Hints

Examples of reuse include objects or other static code maintained in an object/code library.

Hints for Rule 1

A score of 1 is awarded for reusing code regardless of where it was developed. Code developed specifically for reuse within the application and used more than once within the application counts as well as code retrieved from a central library and available for general use

Hints for Rule 2

To score 2 or more, the code must be developed for use in more then one application, stored and managed in a central library and be available for general use. Code from one application that is cut and pasted into another application is not considered reuse. The reusable code would be supported by documentation that enables and eases the reuse

GSC 10: Reusability

Hints for Rule 5

- Examples of applications customized through use of parameters include PeopleSoft and SAP and would generally receive a score of 5.
- Reused code may be *slightly* modified in the receiving application

Typically

Scoring is not platform dependent

GSC 11: Installation Ease

Definition

Installation Ease describes the degree to which conversion from previous environments influenced the development of the application.

Conversion and installation ease are characteristics of the application. A conversion and installation plan and/or conversion tools were provided and tested during the system test phase.

GSC 11: Installation Ease

Score

Score As	Descriptions to Determine Degree of Influence
0	No special considerations were stated by the user, ***and no*** special setup is required for installation.
1	No special considerations were stated by the user, ***but*** special setup is required for installation.
2	Conversion and installation requirements were stated by the user, and conversion and installation guides were provided and tested. The impact of the conversion on the project ***is not*** considered to be important.
3	Conversion and installation requirements were stated by the user, and conversion and installation guides were provided and tested. The impact of the conversion on the project ***is*** considered to be important.
4	In addition to 2 above, automated conversion and installation tools were provided and tested.
5	In addition to 3 above, automated conversion and installation tools were provided and tested.

GSC 11: Installation Ease

Hints

Conversion and installation includes converting pre-existing data into new data files, loading files with actual data, or developing special installation software, such as porting. Purchased or developed software must be used in order to take credit for installation and conversion

Hints for Rule 1

Most business applications require some special setup to install the application and receive a score of 1

Hints for Rule 2

If the application has conversion and installation requirements and installation guides were provided, and providing these functions and guides ***were not*** on the critical path of the project, score a 2

GSC 11: Installation Ease

Hints for Rule 3

If the application has conversion and installation requirements and installation guides were provided, and providing these functions and guides ***were*** on the critical path of the project, score a 3

Hints for Rules 4 and 5

If the application has conversion and installation requirements and can be installed with no external intervention, score a 4 or 5 depending on the other requirements for the scoring of 2 and 3.

Typically

Scoring is not platform dependent

GSC 12: Operational Ease

Definition

Operational Ease describes the degree to which the application attends to operational aspects, such as start-up, back-up and recovery processes.

Operational ease is a characteristic of the application. The application minimizes the need for manual activities, such as tape mounts, paper handling and direct on-location manual intervention.

GSC 12: Operational Ease

Score

Score As	Descriptions to Determine Degree of Influence
0	No special considerations other than the normal back-up procedures were stated by the user.
1-4	One some or all of the following items apply to the application. Select all that apply. Each item has a point value of one, except as noted otherwise. a. Start-up, back-up, and recovery processes were provided, but human intervention is required. b. Start-up, back-up, and recovery processes were provided, but human intervention is required (count as two items). c. The application minimizes the need for tape mounts and /or remote data access requiring human intervention. d. The application minimizes the need for paper handling.
5	The application is designed for unattended operation. Unattended operation means ***no human intervention*** is required to operate the system other than to start up or shut down the application. Automatic error recovery is a feature of the application.

GSC 12: Operational Ease

Hints for Rule 1-4 a

Application has the ability to perform start-up, back-up and recovery, however, human response is required to initiate the function.

Hints for 1-4b

Application has the ability to perform start-up, back-up and recovery; and no human response is required to initiate the function.

Hints for Rule 1-4 c

The application minimizes the need to access data that is not immediately available. This may include importing data from a distributed processor to the local processor prior to execution to eliminate access delays.

Hints for Rule 1-4 d

The application has been designed to provide the user with data in a condensed format or via a media other than paper. This could include elimination of detailed printed information or access to on-line reports, inquiries, microfiche, CD, or other such media.

GSC 12: Operational

Hints for Rule 5

A score of 5 is assigned to an application that runs and recovers automatically from errors, on its own – an unattended operation.

Unattended operation may include unmanned satellite, nuclear reactor or air traffic control.

Typically for GSC 12:

Scoring is not platform dependent

GSC 13: Multiple Sites

Definition

Multiple Sites describes the degree to which the application has been developed for different hardware and software environments

GSC 13: Multiple Sites

Score

Score As	Descriptions to Determine Degree of Influence
0	The needs of ***only one*** installation site were considered in the design.
1	The needs of more than one installation were considered in the design, and the application is designed to operate only under ***identical*** hardware and software environments.
2	The needs of more than one installation were considered in the design, and the application is designed to operate only under ***similar*** hardware and/or software environments.
3	The needs of more than one installation were considered in the design, and the application is designed to operate only under ***different*** hardware and/or software environments.
4	Documentation and support plan are provided and testing to support the application at multiple installation sites and the application is as described by 2.
5	Documentation and support plan are provided and testing to support the application at multiple installation sites and the application is as described by 3.

GSC 13: Multiple Sites

Hints

The term multiple sites is a logical term and is not necessarily physical. There can be multiple sites within the same physical location. The determining factor is based upon the needs of the various installations.

Hints for Rule 0

Most mainframe applications would probably score 0. However, if an application is installed on multiple mainframe computers with significantly different configurations or different operating systems, it would receive a score of greater than 0

Hints for Rule 1

For example, Windows NT on hardware with exactly the same configuration.

GSC 13: Multiple Sites

Hints for Rule 2

For example, Windows 95, 98 and NT on hardware with a similar configuration. Variations could include different memory sizes, various storage capability, different processor speeds, different printer types

Hints for Rule 3

For example, Windows, OS X, UNIX, Linux and VOS3 on different types of hardware. Differences could include Intel based PC, MAC, Tandem, Sun and AS400

Typically for GSC 13:

Scoring is dependent on the number of different platforms

GSC 14: Facilitate Change

Definition

Facilitate Change describes the degree to which the application has been developed for easy modification of processing logic or data structure.

Definition

The following characteristics can apply for the application:

Flexible Query and Report Facility

a. Flexible query and report facility is provided that can handle ***simple*** requests. (count as 1 item)
b. Flexible query and report facility is provided that can handle requests of ***average*** complexity. (count as 2 items)
c. Flexible query and report facility is provided that can handle ***complex*** requests. (count as 3 items)

Business Control Data

d. Business control data is kept in tables that are maintained by the user with on-line interactive processes, but changes take effect only on the ***next*** business cycle. (count as 1 item)
e. Business control data is kept in tables that are maintained by the user with on-line interactive processes, and the changes take effect ***immediately***. (count as 2 items)

GSC 14: Facilitate Change

Score

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	A total of one item from above.
2	A total of two items from above.
3	A total of three items from above.
4	A total of four items from above.
5	A total of five items from above.

GSC 14: Facilitate Change

Hints

- A flexible query and reporting facility means more than a list of choices in a 'canned' query or report. It is the ability of the user to control the data, data source, sequence and format of their query or report request. It means freedom to design screen layout, horizontal and vertical sorting, data item display formats, selection criteria for both files and data items. It includes true user programming for inquiries and is sometimes referred to as ad hoc query or reporting.
- Using filters which control the amount of data viewed or printed in a fixed format is not considered as flexible query and report facility.
- Query and/or report writer capability is often provided by languages such as SQL or Focus or by some of the more dynamic ad hoc reporting tools (e.g., Crystal Reports)

GSC 14: Facilitate Change

Hints for Rule 1

Characteristic a.
Simple requests may include and/or logic applied to only **one** internal logical file.

Hints for Rule 2

Characteristic b.
Requests of average complexity may include and/or logic applied to **more than one** internal logical file.

Hints for Rule 3

Complex requests may include and/or logic combinations on one or more internal logic files

Typically for GSC 14:

Scoring is not platform dependent

Module 2: The Counting Rules

- The Process
- Types of Function Point Counts
- Scope, Purpose and Application Boundaries
- Data Functions
- Transaction Functions
- Summary of Functions Performed by EI,EO,EQ
- Summary of Processing Logic used by EI,EO,EQ
- General System Characteristics
- ➡ Calculations

Formulas

- Total Degree of Influence (TDI) = Sum of DIs from all 14 GSCs
- Value Adjustment Factor (VAF) = (.65 + (.01 x TDI))
- Adjusted Function Points = Unadjusted FP x VAF
- NOTE: There are different algorithms for:
  - Development
  - Enhancement
  - Original Application and
  - Revised Application Counts

Development Project Calculation

The calculation consists of three components of functionality:

- Application functionality
  - functions used after software installation to satisfy the ongoing business needs of the user
- Conversion functionality
  - functions provided to convert data and/or provide other user-specified conversion requirements, such as special conversion reports
- Value adjustment factor
  - determined by using the 14 general system characteristics to rate the application complexity

Development Count

The following formula is used to calculate the development project function point count:

DFP = (UFP + CFP) * VAF

Where

DFP - development project function point count
UFP - unadjusted function point count
CFP - function points added by the conversion unadjusted function point count
VAF - value adjustment factor

After the software is installed, the application function point count is calculated using components of the development project function point count

Application Count – Initial Count

There are two different formulas to calculate the application count:

1. The following formula is used to calculate initial function point counts for an application; the user is receiving functionality; there are no changes to existing functionality or deletions of unneeded functionality; the application function point count does not include conversion requirements:

   AFP = ADD * VAF

   Where

   AFP - initial application function point count

   ADD - unadjusted function point count of functions installed

   VAF - value adjustment factor of the application

Counting Enhancements

- Calculation
- Hints
- Examples
- Impact on Application Count
- Revised Application Calculation
- Case Study

Enhancement Project Calculation

The calculation consists of three components of functionality:

- Application functionality - consists of:
  - Function points identified from functionality added by enhancements
  - Function points counted because existing functionality is changed during the project
  - Function points counted for functionality deleted during the project
- Conversion functionality - consists of function points delivered because of any conversion requirements by the user
- Value adjustment factor - two adjustment factors include:
  - adjustment factor before enhancement project begins
  - adjustment factor after enhancement project completes

Enhancement Count

The following formula is used to calculate the enhancement project function point count:

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

Where

EFP - enhancement project function point count
ADD - unadjusted function point count of added functions
CHGA - unadjusted count of functions modified; reflects functions after modification
CFP - function point count added by conversion
VAFA - value adjustment factor of the application after the enhancement
DEL - unadjusted count of functions deleted
VAFB - value adjustment factor of application before the enhancement

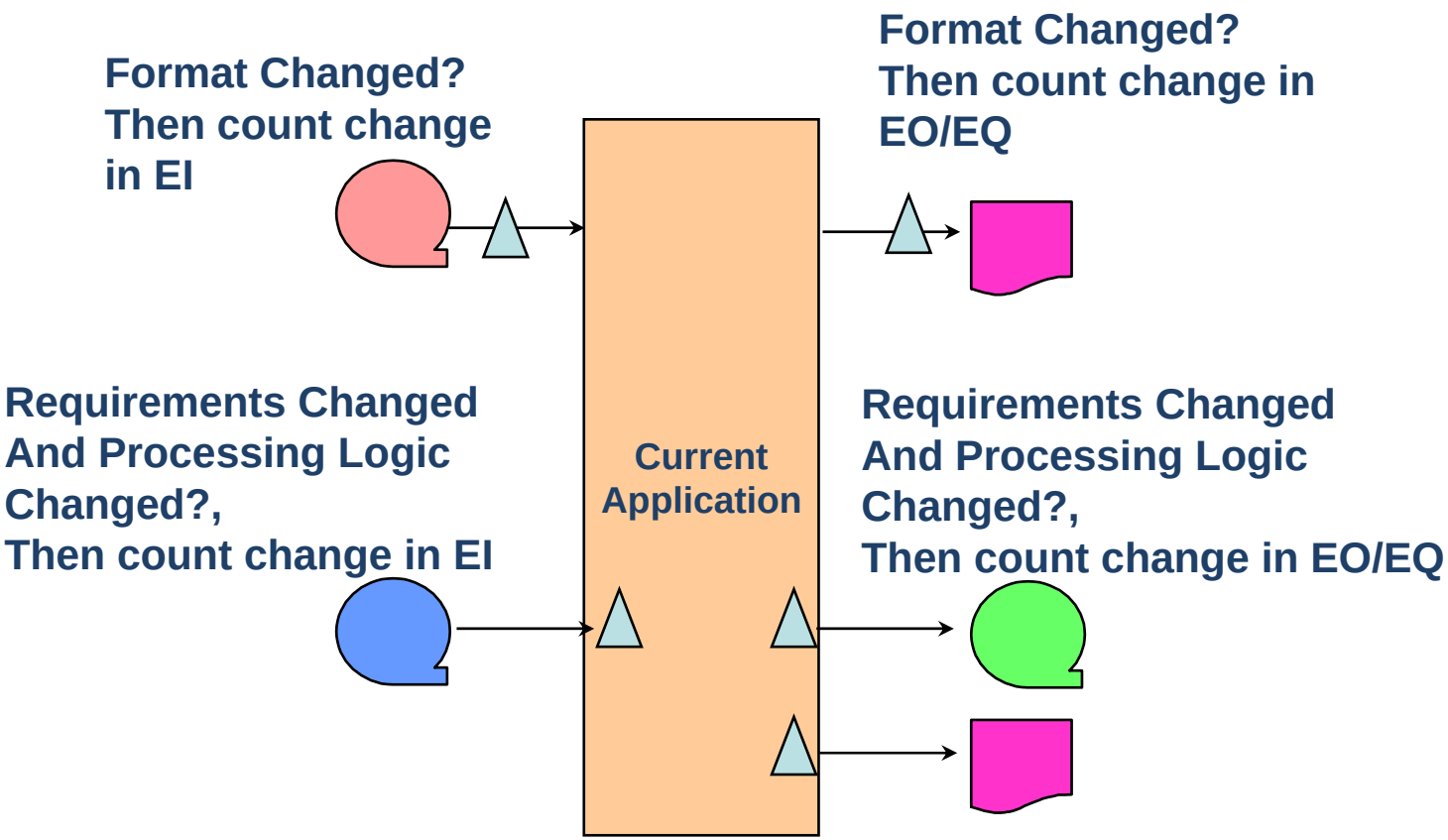
Enhancement FP Count Includes

- Application functionality - consists of:
  - Function points identified from functionality added by enhancements
  - Function points counted because existing functionality is changed during the project
  - Function points counted for functionality deleted during the project
- Conversion functionality - consists of function points delivered because of any conversion requirements by the user
- Value adjustment factor

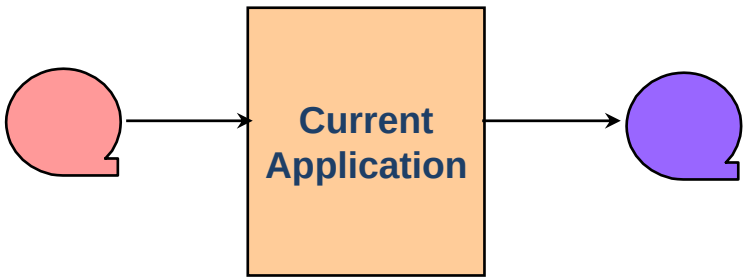
Enhancement Counting Hints

- If an existing application count exists, review existing boundary
- Identify all data transactions that were modified, added, or deleted
- Identify all new function transactions
- Identify all changed function transactions
  - Changes due to fields added or removed, changes due to new FTRs, changes in processing logic to satisfy the business requirements for the enhancements
- Identify all deleted function transactions
- Identify conversion function points for new or changed ILFs

Enhancement Counting Hints Changes



Enhancement Counting Hints Deletes



If the requirements state that Input/Output or Query will no longer be received/Sent/Used

Then:

Identify the component
Assess the Complexity and
Include in FP count As “Deleted”

Enhancement Counting Hints Conversion

- Consider one-time conversion programs that have to be written for this project due to data changes:
  - Did you have to convert data (in one table/database) from one release to another?
  - Did you add fields and have to pre-load them with values?
  - Did you have to meet other implementation requirements; e.g., conversion reports?

Enhancement Example

Project Count		Application Count
	Unadjusted Count Before -	1,000
	New	
+ 4	A EI (12 DETS/2FTRS)	+ 4
+ 3	L EI (3 DETS/2FTRS)	+ 3
+ 4	A EQ (14 DETS/2 FTRs)	+ 4
	Changed	
+ 5	A EO (From: 18 DETS/2FTRS A)	0
	(To: 19 DETS/3FTRS A)	
+ 10	A ILF (From: 19 DETS/2RETS L	+ 3
	(To: 20 DETS/2RETS A)	
	Deleted	
+ 5	A EO (21 DETS/1FTR)	- 5
31		1009

Enhancement Project = 31
Current Application Value = 1009 Unadjusted Function Points

Application Count - Revised

- After the enhancement is installed, the application function point count must be updated to reflect changes in the application's functionality

Measuring Enhancement Projects

Type of Project	Development Function Points	Support or Maintenance Function Points
	Function Points (FP)	Installed Function Pts. (IFP)
New Development	Added Function Points	Added Function Points
Enhancement to Existing Application	Deleted FP + Added FP + Changed FP = Development FP	Original FP - Deleted FP + Added FP +/- ▲ Changed FP = Support FP

Application Calculation

- When an enhancement project is installed, the existing count must be updated to reflect modifications; 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
  - Deleted functionality decreases the size
  - Changes to the value adjustment factor adds, subtracts, or has no effect on the count
  - Conversion functionality does not affect the count; any conversion functionality associated with an enhancement project is not included in the application count

Application Calculation

The following formula is used to calculate the application function point count after an enhancement project

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

Where:

AFP - application adjusted function point count
UFPB - unadjusted function point count before the enhancement project
ADD - unadjusted count of functions added by the enhancement project
CHGA - unadjusted count of functions changed by the enhancement; reflects functions *after* the change
CHGB - unadjusted count of functions changed by the enhancement; reflects functions *before* the change
DEL - unadjusted count of functions deleted
VAFA - value adjustment factor of application after the enhancement

References

- For more information:
  - The David Consulting Group
    - www.davidconsultinggroup.com
  - International Software Benchmarking Standards Group
    - www.isbsg.org.au
  - International Function Point Users Group
    - www.ifpug.org
  - Software Engineering Institute, Project Sizing & Estimating
    - www.cmu.sei.edu
-