

Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual

ICS 35.080

National foreword

This British Standard reproduces verbatim ISO/IEC 20926:2003 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software engineering, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international publications referred to in this document may be found in the *BSI Catalogue* under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the ISO/IEC title page, pages ii to xii, pages 1-1 to 1-10, pages 2-1 to 2-10, pages 3-1 to 3-8, pages 4-1 to 4-4, pages 5-1 to 5-6, pages 6-1 to 6-84, pages 7-1 to 7-156, pages 8-1 to 8-18, pages 9-1 to 9-20, pages A-1 to A-4, pages B-1 to B-10, the Index (8 pages), the Glossary (5 pages) and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2003

© BSI 31 October 2003

INTERNATIONAL STANDARD

ISO/IEC 20926

First edition
2003-10-01

Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual

*Ingénierie du logiciel — Méthode de mesure de la taille fonctionnelle
non ajustée de IFPUG 4.1 — Manuel des pratiques de comptage*



Table of Contents

	Foreword.....	vii
	Scope	viii
	IFPUG Foreword.....	ix
	IFPUG Preface.....	xi
Chapter 1	Introduction	1-1
	Objectives of this International Standard	1-2
	Guidelines for ISO/IEC 20926	1-2
	Intended Audience.....	1-2
	Organization of this International Standard.....	1-3
	Preface and Introduction	1-3
	Overview of Function Point Analysis	1-3
	Explanation of the Counting Practices	1-4
	Manual Revision Process	1-5
	Frequency of Changes	1-5
	Change Process	1-5
	Related IFPUG Documentation.....	1-8
	Training Requirements.....	1-9
Chapter 2	Overview of Function Point Analysis	2-1
	Objectives and Benefits of Function Point Analysis	2-2
	Objectives of Function Point Analysis	2-2
	Benefits of Function Point Analysis	2-2
	Function Point Counting Procedures	2-3
	Procedure Diagram	2-3
	Procedure by Chapter	2-3
	Summary Counting Example.....	2-4
	Summary Diagram.....	2-4
	Determine the Type of Function Point Count	2-5
	Identify the Counting Scope and Application Boundary	2-5
	Determine the Unadjusted Function Point Count.....	2-6
	Determine the Value Adjustment Factor	2-9
	Calculate the Adjusted Function Point Count.....	2-9
Chapter 3	User View.....	3-1
	Definition of User View.....	3-2
	Sizing During the Life Cycle of an Application.....	3-3
	Phase: Initial User Requirements.....	3-4
	Phase: Initial Technical Requirements	3-5
	Phase: Final Functional Requirements	3-6
	Life Cycle Phase Comparisons	3-7
	Hints to Help with Counting.....	3-8
Chapter 4	Determine Type of Count	4-1
	Definitions: Types of Function Point Counts	4-2
	Development Project	4-2
	Enhancement Project	4-2
	Application	4-2
	Diagram of Types of Counts.....	4-3

	Estimated and Final Counts.....	4-3
Chapter 5	Identify Counting Scope and Application Boundary.....	5-1
	Definition of Counting Scope and Application Boundary	5-2
	Definition of the Purpose of the Count.....	5-2
	Definition of the Counting Scope	5-2
	Definition of the Application Boundary.....	5-3
	Counting Scope and Application Boundary Rules and Procedures	5-5
	Boundary Rules	5-5
	Counting Scope and Application Boundary Procedures.....	5-5
	Hints to Help to Identify the Counting Scope and the Application Boundary ...	5-6
Chapter 6	Count Data Functions.....	6-1
	Definitions: ILFs and EIFs	6-3
	Internal Logical Files	6-3
	External Interface Files	6-3
	Difference between ILFs and EIFs	6-3
	Definitions for Embedded Terms	6-3
	ILF/EIF Counting Rules.....	6-5
	Summary of Counting Procedures.....	6-5
	ILF Identification Rules	6-6
	EIF Identification Rules	6-6
	Complexity and Contribution Definitions and Rules	6-7
	DET Definition	6-7
	DET Rules.....	6-7
	RET Definition	6-9
	RET Rules.....	6-9
	ILF/EIF Counting Procedures.....	6-10
	Procedure Diagram.....	6-10
	Identification Procedures	6-10
	Complexity and Contribution Procedures	6-11
	Hints to Help with Counting	6-13
	ILF/EIF Counting Examples.....	6-14
	ILF Counting Examples.....	6-18
	EIF Counting Examples	6-58
Chapter 7	Count Transactional Functions	7-1
	Definitions: EIs, EOs and EQs	7-3
	External Inputs	7-3
	External Outputs	7-3
	External Inquiry	7-3
	Summary of the Functions Performed by EIs, EOs and EQs.....	7-4
	Definitions for Embedded Terms	7-5
	Summary of Processing Logic Used by EIs, EOs and EQs	7-8
	EI/EO/EQ Counting Rules.....	7-9
	Summary of Counting Procedures.....	7-9
	Elementary Process Identification Rules	7-10
	Transactional Functions Counting Rules	7-11
	Primary Intent Description for EIs	7-11
	External Input Counting Rules	7-11
	Primary Intent Description for EOs and EQs	7-12
	Shared EO and EQ Counting Rules	7-12
	Additional External Output Counting Rules	7-12
	Additional External Inquiry Counting Rules	7-13
	Complexity and Contribution Definitions and Rules	7-13
	FTR Definition	7-13
	DET Definition	7-13
	EI Complexity and Contribution Rules	7-14
	FTR Rules for an EI	7-14

	DET Rules for an EI.....	7-14
	EO/EQ Complexity and Contribution Rules.....	7-16
	Shared FTR Rules for EOs and EQs	7-16
	Additional FTR Rules for an EO	7-16
	Shared DET Rules for EOs and EQs	7-16
	EI, EO and EQ Counting Procedures	7-18
	Procedure Diagram	7-18
	Identification Procedures	7-19
	Complexity and Contribution Procedures	7-21
	Hints to Help with Counting EIs, EOs and EQs	7-24
	Additional Hints to Help Counting EOs and EQs	7-26
	Elementary Process Identification Examples	7-27
	EI/EO/EQ Counting Examples	7-45
	EI Counting Examples.....	7-52
	EO Counting Examples	7-89
	EQ Counting Examples	7-123
Chapter 8	Determine Value Adjustment Factor (Optional)	8-1
	Value Adjustment Factor Determination	8-3
	Procedures to Determine the VAF	8-3
	General System Characteristics	8-4
	Degrees of Influence	8-5
	Guidelines to Determine Degree of Influence.....	8-6
	1. Data Communications	8-6
	2. Distributed Data Processing	8-7
	3. Performance	8-8
	4. Heavily Used Configuration	8-9
	5. Transaction Rate	8-10
	6. Online Data Entry	8-10
	7. End-User Efficiency	8-11
	8. Online Update.....	8-12
	9. Complex Processing.....	8-13
	10. Reusability	8-14
	11. Installation Ease	8-15
	12. Operational Ease.....	8-16
	13. Multiple Sites	8-17
	14. Facilitate Change.....	8-18
Chapter 9	Calculate Adjusted Function Point Count.....	9-1
	Review of Steps for Function Point Analysis	9-3
	Development Project Function Point Calculation	9-4
	Application Functionality	9-4
	Conversion Functionality	9-4
	Application Value Adjustment Factor	9-4
	Function Point Formula	9-5
	Example: Development Project Function Point Count	9-6
	Application Functionality	9-6
	Conversion Functionality	9-8
	Application Contribution to the Unadjusted Function Point Count	9-9
	Conversion Contribution to the Unadjusted Function Point Count.....	9-10
	Final Calculation	9-10
	Enhancement Project Function Point Calculation	9-11
	Application Functionality	9-11
	Conversion Functionality	9-11
	Value Adjustment Factor	9-11
	Function Point Formula	9-12
	Example: Enhancement Project Count	9-13
	Application Functionality	9-13

	Application Contribution to the Unadjusted Function Point Count.....	9-14
	Final Calculation	9-16
	Application Function Point Calculation	9-17
	Formula to Establish the Initial Count.....	9-17
	Formula to Reflect Enhancement Projects	9-18
	Example: Application Count	9-19
	Initial Count	9-19
	Count After Enhancement	9-19
Appendix A	Calculation Tables	A-1
	Unadjusted Function Point Count Calculation Table	A-2
	Value Adjustment Factor Calculation Table	A-3
Appendix B	The Change from CPM 4.0 to 4.1	B-1
	Introduction	B-2
	Major Functional Change Areas in CPM 4.1	B-2
	Version Control	B-3
	Overview of Changes.....	B-3
	Background	B-8
	The Impact Study	B-8
	Conversion from CPM 4.0 to 4.1	B-9
	Impact on 4.0 Users Changing to 4.1	B-10
	Recommendations	B-10
Index		
Glossary		

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 20926 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and system engineering*.

Scope

This International Standard specifies the International Function Point Users Group (IFPUG) Release 4.1 unadjusted Functional Size Measurement Method. It provides:

- clear and detailed description of function point counting
- A foundation to ensure that counts are consistent
- Guidance to allow function point counting of Functional User Requirements from the deliverables of popular software development methodologies and techniques
- A framework to enable automated support for function point counting

The provisions of this International Standard can be applied by anyone using function point analysis for software measurement. It was designed for use by persons new to function point counting as well as those with intermediate and advanced experience.

IFPUG Foreword

Function points are the leading metric of the software world. Although function points originated as a sizing mechanism for software projects, the power and utility of function points have expanded into new uses far beyond that basic purpose. As the twenty-first century approaches, function points are now being applied to all of these tasks:

- Benchmark studies
- Development cost estimating
- Litigation involving software contracts
- Litigation involving software taxation
- Maintenance cost estimating
- Outsource contracts
- Process improvement analysis
- Quality estimating
- Quality measurements
- Sizing all software deliverables (documents, source code, test materials)
- Year 2000 software cost estimating

As usage of function point metrics expands throughout the software world, more and more companies and government agencies are starting function point programs. This implies that the need for certified function point analysts is rising even faster than the demand for other software professionals. Certification would not be possible without a complete and stable set of counting rules for function point analysis.

A great deal of the credit for the rapid expansion of function point metrics should go to the International Function Point Users Group (IFPUG) and its officers, committees, and members. One of the committees that merits commendation is the Counting Practices Committee.

Although the basic principles of function point analysis are simple and straightforward, the real-life application of these principles across thousands of software projects is not simple at all.

If function point counts fluctuated by more than 150% when counted by different individuals (as do lines of code counts) then function points would have no claim to be considered a useful business metric. But thanks to the work of the Counting Practices Committee, the reliability of function point analysis is good enough to allow function points to serve as the basis for contracts, for carrying out scholarly research, for cost estimating, and for creating reliable benchmarks. So far as can be determined, the accuracy of function points is equal or superior to many other business metrics such as internal rate of return, net present value, or return on investment.

The move to version 4.0 of the IFPUG counting practices in January of 1994 was somewhat contentious and controversial. This is because the version 4.0 rules had the affect of reducing function point totals for some applications, by fairly significant amounts.

The move to the version 4.1 rules should be much smoother and less controversial. The reason that 4.1 was selected rather than 5.0 as the name of this release is because the numeric results of the new version are close enough to the version 4.0 rules that recounting will not be necessary.

The major changes in the version 4.1 rules are in the examples, the clarification of some complex counting situations, and improvements in the overall exposition of function point counting principles. Those learning to

use function points should find the version 4.1 rules to be easier to understand and apply than the prior versions.

As software itself expands and changes, the rules for counting function points must also be expanded. When Allan Albrecht first introduced function points in October of 1979, many of the kinds of software projects being created in 1999 did not exist. For example, in 1979 software such as multi-tier client-server applications, web applets, and massive enterprise resource planning (ERP) systems were still in the future.

It is a tribute to Allan Albrecht's vision that function point metrics are as useful today as they were in 1979. But without the work of the IFPUG organization and the Counting Practices Committee, function point metrics would not be expanding in utility at the beginning of the twenty-first century. In fact, function points are now used for more business purposes than any other metric in the history of software.

T. Capers Jones
Chief Scientist
Artemis Management Systems

IFPUG Preface

- Introduction** The use of function points, as a measure of the functional size of software, has grown in the past decade from a few interested organizations to an impressive list of companies worldwide. The IFPUG method is applicable to measuring all software
- IBM CIS & A Guidelines 313** In the late 1970s, Allan Albrecht of IBM defined the concepts that enabled measuring the output of software development projects. These definitions were extended in *IBM CIS & A Guideline 313, AD/M Productivity Measurement and Estimate Validation*, dated November 1, 1984.
- Release 2.0** With the growth in the use of function points, there was wider and wider application of the measure. This broadening of the application tested the original description of the measure and made it necessary to create guidelines to interpret the original rules in new environments. This was reflected in Release 2.0 of the *International Function Point Users Group (IFPUG) Function Point Counting Practices Manual*.
- Release 3.0** Release 3.0 of the *IFPUG Function Point Counting Practices Manual* was a major milestone in the evolution of functional size measurement. For the first time, the IFPUG Counting Practices Committee made an effort to change the document from a collection of many interpretations of the rules to a truly coherent document that represented a consensus view of the rules of function point counting. In this sense, it was the first step to truly establishing standards for function point measurement which could be applied across organizations.
- Release 4.0** Release 4.0 (January 1994) was the next milestone in the evolution of functional size measurement. This release reflected the use of function points early in project development to estimate project size using information engineering disciplines. The rapidly increasing number of graphical user interface (GUI) windows applications mandated that we include GUI counting in the release. Because more counting was occurring across a wider variety of situations, the release placed an emphasis on interpreting and practicing using the counting rules. Examples were included throughout the documentation and case studies supplemented the material. Finally, release 4.0 continued to clarify and increase the consistency of function point counting.
- Release 4.1** Release 4.1 (January 1999) provides clarifications to existing rules, new or amended rules which address previously undocumented situations and new hints and examples to aid understanding. The IFPUG Counting Practices Committee has reviewed and processed requests from members, following the Manual Revision Process contained in Chapter 1 of this manual.

The revisions included in 4.1 clarify:

- the identification of a user, an elementary process, and control information
- the differentiation between External Outputs (EOs) and External Inquiries (EQs)
- the identification of Data Element Types (DETs) and Record Element Types (RETs) for data functions
- the identification of Data Element Types (DETs) for transactional functions

Release 4.1 continues the process of clarifying and improving the consistency of function point counting.

Finally, with the exception of the 14 General Systems Characteristics, it was designed to be compliant with existing ISO standards if and when any compliance guide becomes a standard.

Future Releases

This document is meant to be a living one. We must recognize how to count new environments as they are introduced. We need to be able to do this in the context of maintaining the validity of the counts we have already made. This will not be an easy task, yet it is an essential one if we are to be able to measure the progress we are making in delivering value to the users and to the organizations they represent.

The Counting Practices Committee wishes to thank all those who have helped us in our research and in the production of this manual.

Mary S. Bradley

Chairperson, Counting Practices Committee

1 Introduction

Introduction This chapter defines the objectives of this International Standard and this International Standard revision process. It also describes publications that are related to this International Standard.

The IFPUG method is applicable to measuring all software.

Contents This chapter includes the following sections:

Topic	See Page
Objectives of this International Standard	1-2
Guidelines for ISO/IEC 20926	1-2
Intended Audience	1-2
Organization of this International Standard	1-3
Preface and Introduction	1-3
Overview of Function Point Analysis	1-3
Explanation of the Counting Practices	1-4
Manual Revision Process	1-5
Frequency of Changes	1-5
Change Process	1-5
Related IFPUG Documentation	1-7
Training Requirements	1-9

Objectives of this International Standard

The primary objectives of this International Standard ISO/IEC 20926 are to

- Provide a clear and detailed description of function point counting
- Ensure that counts are consistent with the counting practices of IFPUG affiliate members
- Provide guidance to allow function point counting from the deliverables of popular methodologies and techniques
- Provide a common understanding to allow tool vendors to provide automated support for function point counting

Guidelines for ISO/IEC 20926

The following guidelines were used to develop this release:

- This International Standard is based primarily on the *IFPUG Function Point Counting Practices Manual, Release 4.0*.
- Secondly, this International Standard is based on *IBM CIS & A Guideline 313, AD/M Productivity Measurement and Estimate Validation*, dated November 1, 1984. The function point counting methodology described in 313 is generally referred to as Albrecht 1984.
- Finally, issues not sufficiently covered in the sources listed above were decided by the IFPUG Counting Practices Committee and validated through impact studies.

With its release, this International Standard should be considered the IFPUG standard for function point counting. It is imperative that each IFPUG member takes an active role to ensure counting consistency. IFPUG member adherence to this standard will contribute greatly to counting consistency.

Intended Audience

The standards in this International Standard should be applied by anyone using function point analysis for software measurement. This International Standard was designed for use by persons new to function point counting as well as those with intermediate and advanced experience.

Organization of this International Standard

There are three major parts in this International Standard:

- Preface and introduction
- Overview of function point analysis
- Explanation of the counting practices

Examples are used extensively throughout this International Standard to explain counting practices concepts, rules, and procedures. Detailed examples conclude chapters 6 and 7.

Note: A separate IFPUG Glossary includes definitions of terms used across IFPUG publications.

Preface and Introduction

The Preface and Introduction provide an overview of this International Standard and function point counting.

Overview of Function Point Analysis

The Overview introduces the function point counting procedures and includes a summary example of the procedures.

Explanation of the Counting Practices

Chapter 3 explains the concept of user view.

Chapters 4 through 9 present details about each of the procedure steps introduced in the Overview.

For example, Chapter 4, Determine Type of Count, is the first step in the function point counting procedure. Chapter 9, Calculate Adjusted Function Point Count, is the last step.

Information within chapters 5 through 7 is presented in the following sequence:

- Definitions
- Rules
- Procedures
- Counting Hints
- Examples

Manual Revision Process

This section explains the frequency of changes to this International Standard and defines the change process.

Frequency of Changes

During January of each year, a new version of this International Standard *may* become effective. It will include any new or changed definitions, rules, or counting practices that have been finalized by the Counting Practices Committee (CPC) since the previous January.

Change Process

The following activities outline the process for adding or changing information in this International Standard. Explanations of each activity follow the table.

Step	Action
1	The issue is submitted to the CPC.
2	The issue is assigned for research.
3	The CPC reviews and discusses the issue.
4	The CPC presents a proposed solution to the IFPUG membership.
5	An impact study is initiated if the proposed change would have any impact on existing counts.
6	The final decision is made.
7	The IFPUG membership is informed of the decision.
8	Changes become effective with, and are reflected in, the next release of this International Standard.

Issue Submitted	The reader submits ideas, changes, or issues to the Counting Practices Committee using the Reader's Request Form at the end of this International Standard. If the page is not available, send comments to the address in the front of this International Standard and mark it, "ATTN: Counting Practices Committee."
Research Assigned	A member of the CPC is assigned the responsibility for identifying all alternatives, the rationale, and the potential impact of each alternative if it is implemented. Thorough examination of existing counting standards and historical papers is completed while compiling alternatives. In addition, an effort is made to determine what is thought to be <i>common practice</i> .

CPC Review	The CPC reviews and discusses the rationale for each alternative, and its potential impact. The review and discussion may result in a proposal for change or the review may lead the committee to reject the change request.
Solution Proposed	A proposed solution is made to the IFPUG membership and written comments are solicited. A copy of the proposed changes is mailed to IFPUG contacts at member organizations. The proposal also may be announced and distributed during an IFPUG conference. The latter depends on the timing of the committee meeting rather than the conference schedule.
Impact Study Initiated	The CPC has adopted a conservative stance on initiating impact studies. If it is possible that <i>common practice</i> must change, or several organizations or types of applications will be impacted by the change, an impact study is initiated. The success of the impact study is the responsibility of every IFPUG member. If the CPC receives written feedback indicating there is little or no impact, the study is discontinued.
Final Decision Made	The committee makes a final decision using results from research, written comments from members, and the impact study. The committee can complete more than one iteration of Steps 2 through 5 (research through impact study) before making a final decision. The final decision can result in a change or the committee may decide that a change is not warranted.
Decision Communicated	The final decision is communicated in writing to IFPUG members via the IFPUG contact at the various organizations. If any impact study results contributed to making a decision, the results and a recommendation on how to minimize the impact of the change will also be communicated.
Decision Effective Date	This International Standard is updated to reflect the decisions. The effective date of the decisions is the date of the next January release of this International Standard.

Related IFPUG Documentation

This Counting Practices Manual is one module in the IFPUG documentation. All documents complement each other.

The following table describes the other publications.

Document	Description
IFPUG Brochure (Available)	This publication is an introduction to the International Function Point Users Group. It includes a brief history of the organization, introduces function point analysis, and defines the purpose of IFPUG. The brochure also includes a membership application. Audience: This publication is for anyone who wants an overview of IFPUG or an application for membership.
IFPUG: Organizational Structure and Services (Available)	This publication describes IFPUG services, and lists the board of directors, committees, and affiliate members worldwide. Audience: This publication is for anyone who wants background information about IFPUG.
Guidelines for Software Measurement (Release Date: April 1994)	This International Standard provides an overview of software metrics for organizations working to create or improve software measurement programs. This International Standard addresses both system and customer management, provides high-level justifications for software measurement, and examines the components of effective measurement programs. Audience: This International Standard is intended for IFPUG members, Function Point Coordinators, persons who prepare the reports to management, and other persons knowledgeable about and working directly with function points.
Application of Measurement Information (Current release is available as <i>Function Points as an Asset</i> Update Release: September 1994)	This International Standard explains how function points are an asset and provides information to assist in implementing the use of function points. Audience: This International Standard is intended for IFPUG members, Function Point Coordinators, persons who prepare the reports to management, and other persons knowledgeable about and working directly with function points.
Quick Reference Counting Guide (Release Date: January 1999)	This quick reference guide is a summary of function point counting rules and procedures. Audience: This summary information is intended for anyone applying function point analysis.
Function Point Analysis Case Studies (Release Dates: Case Study 1: May 1994 Case Study 2: September 1994 Case Study 3: September 1996 Case Study 4: September 1998)	The case studies illustrate the major counting techniques that comprise the Function Point Counting Practices Manual. The cases illustrate function point counts for a sample application. The cases include the counting that occurs at the end of the analysis phase of software development and after system construction. Audience: The case studies are intended for persons new to function point analysis as well as those with intermediate and advanced experience.

Document	Description
IFPUG Glossary (Available with <i>CPM</i> and <i>Function Points as an Asset</i>)	This is a comprehensive glossary that defines terms used across IFPUG publications. Audience: The glossary is recommended for anyone who receives any of the other IFPUG documents or anyone who needs definitions of IFPUG terms.

Training Requirements

Usability evaluations of this publication have verified that reading this International Standard alone is not sufficient training to apply function point counting at the optimum level. Training is recommended, particularly for those new to function point counting.

Note: For function point training, be sure you are trained using IFPUG certified materials. Call the IFPUG Executive Office at 614-895-7130 for a list of instructors with certified training courses.

In addition to the function point specific information, this International Standard includes the use of structured analysis and design terms, such as business systems and entity. The glossary includes definitions of these terms, but this International Standard does not include detailed explanations of structured analysis and design techniques. Therefore, all of the material will not apply or be helpful if you have not been trained in structured analysis and design techniques.

(Blank page)

2 Overview of Function Point Analysis

Introduction This chapter presents an overview of the function point counting process. It includes the objectives of function point counting and presents a summary and example of the function point counting procedures.

Contents This chapter includes the following sections:

Topic	See Page
Objectives and Benefits of Function Point Analysis	2-2
Objectives of Function Point Analysis	2-2
Benefits of Function Point Analysis	2-2
Function Point Counting Procedure	2-3
Procedure Diagram	2-3
Procedure by Chapter	2-3
Summary Counting Example	2-4
Summary Diagram	2-4
Determine the Type of Function Point Count	2-5
Identify the Counting Scope and Application Boundary	2-5
Determine the Unadjusted Function Point Count	2-6
Determine the Value Adjustment Factor	2-9
Calculate the Adjusted Function Point Count	2-9

Objectives and Benefits of Function Point Analysis

Function point analysis is a standard method for measuring software development from the user's point of view. The IFPUG method is applicable to measuring all software.

Objectives of Function Point Analysis

Function point analysis measures software by quantifying the functionality the software provides to the user based primarily on logical design. With this in mind, the objectives of function point analysis are to:

- Measure functionality that the user requests and receives
- Measure software development and maintenance independently of technology used for implementation

In addition to meeting the above objectives, the process of counting function points should be:

- Simple enough to minimize the overhead of the measurement process
- A consistent measure among various projects and organizations

Benefits of Function Point Analysis

Organizations can apply function point analysis as:

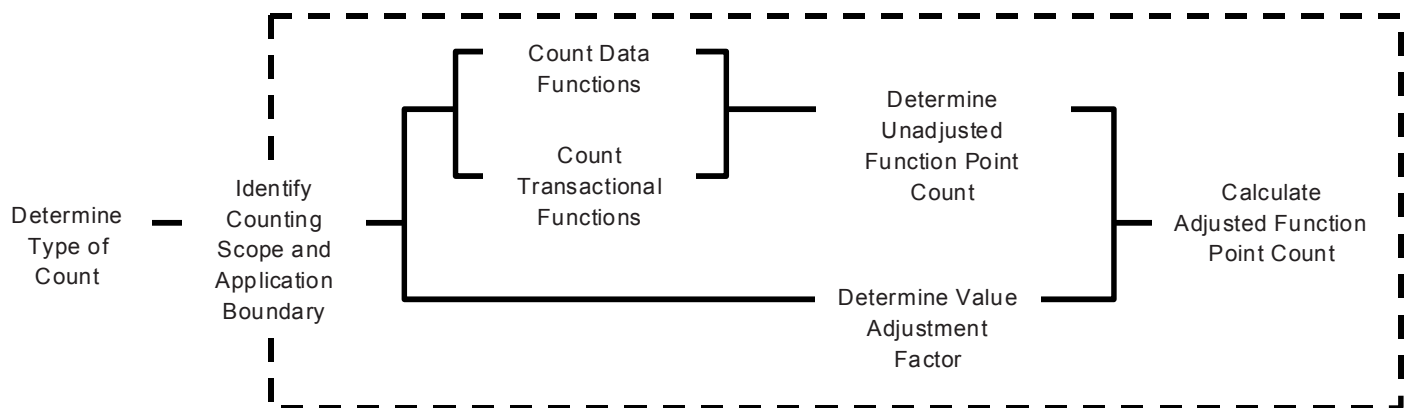
- A tool to determine the size of a purchased application package by counting all the functions included in the package
- A tool to help users determine the benefit of an application package to their organization by counting functions that specifically match their requirements
- A tool to measure the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

Refer to other IFPUG documents such as *Function Points as an Asset* for additional information about the benefits of function point analysis, or see the IFPUG web site at <http://www.ifpug.org> for additional information.

Function Point Counting Procedure

This section presents the high-level procedure for function point counting.

Procedure Diagram



Procedure by Chapter

The following table shows the function point counting procedures as they are explained in the remaining chapters of the manual.

Note: A summary example of the counting procedures is presented on the following pages in this chapter.

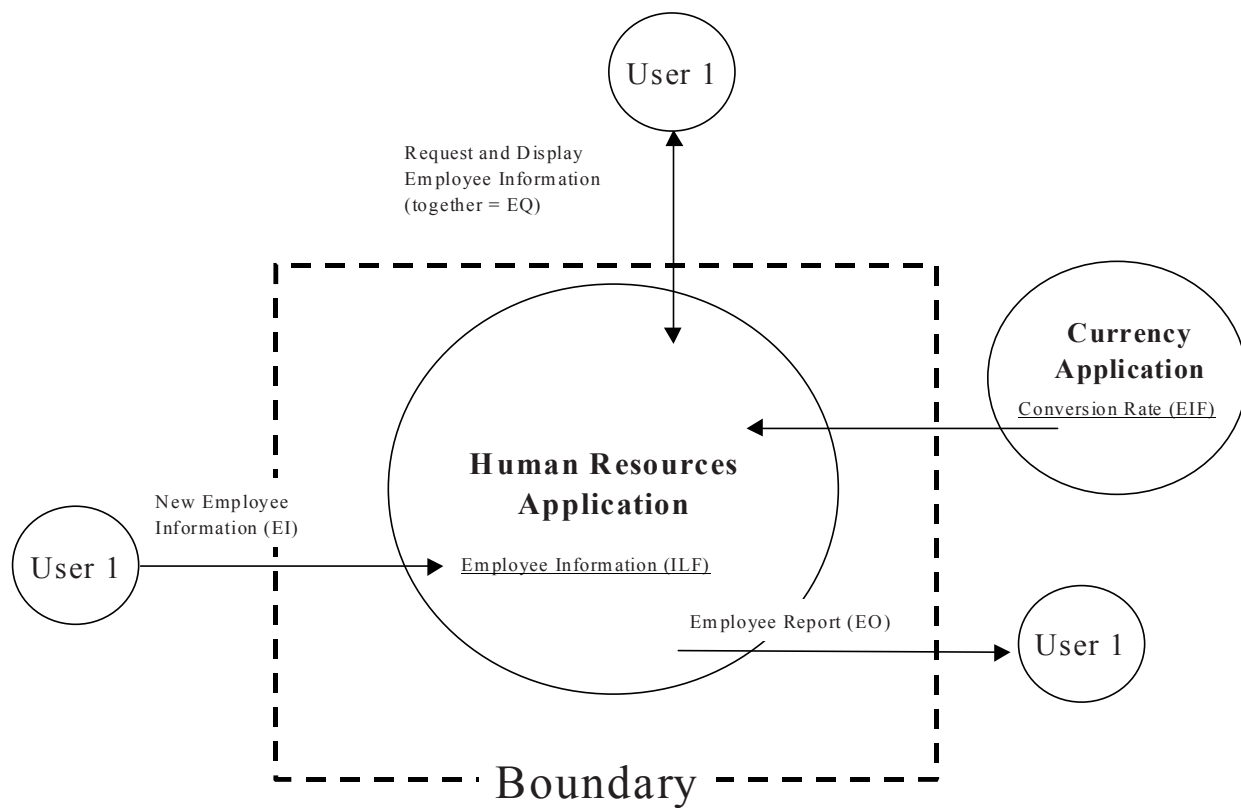
Chapter	Procedure
4	Determine the type of function point count.
5	Identify the counting scope and application boundary.
6	Count the data functions to determine their contribution to the unadjusted function point count.
7	Count the transactional functions to determine their contribution to the unadjusted function point count.
8	Determine the value adjustment factor.
9	Calculate the adjusted function point count.

Summary Counting Example

This section presents a summary example of the function point counting procedure and the components that comprise the count.

Summary Diagram

The following diagram shows the components for the example function point count for a Human Resources Application. Refer to the diagram while reading the remaining paragraphs in this chapter.



Determine the Type of Function Point Count

The first step in the function point counting procedure is to determine the type of function point count.

Function point counts can be associated with either projects or applications. There are three types of function point counts:

- Development project function point count
- Enhancement project function point count
- Application function point count

The example on page 2-4 is for a project function point count, which will also evolve into an application function point count.

Chapter 4 includes detailed definitions of each type of function point count. Chapter 9, the last chapter in this manual, explains the formulas to calculate the adjusted function point count for each of the three types of counts.

Identify the Counting Scope and Application Boundary

The counting scope defines the functionality that will be included in a particular function point count.

The application boundary indicates the border between the software being measured and the user.

The example on page 2-4 shows the application boundary between the Human Resources Application being measured and the external Currency Application. It also shows the application boundary between the Human Resources Application and the user.

Chapter 5 explains counting scope and application boundary.

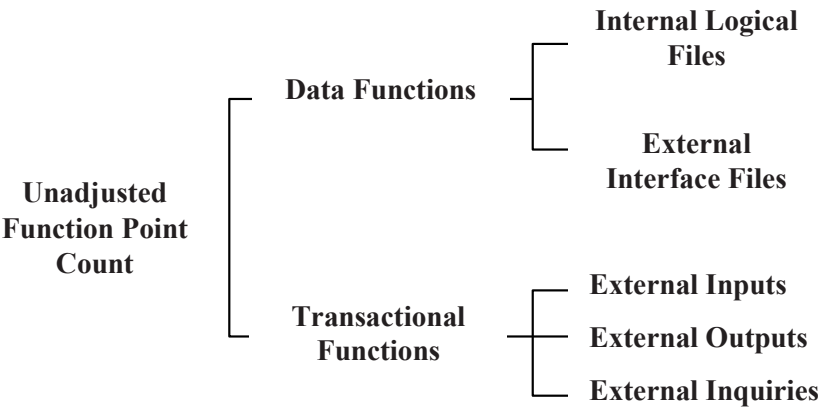
Determine the Unadjusted Function Point Count

The unadjusted function point count (UFPC) reflects the specific countable functionality provided to the user by the project or application.

The application's specific user functionality is evaluated in terms of *what* is delivered by the application, not *how* it is delivered. *Only* user-requested and defined components are counted.

The unadjusted function point count has two function types—data and transactional. These function types are further defined as shown in the following diagram.

The unadjusted functional size reported with ‘unadjusted function point units’ is equivalent to the functional size as defined within ISO/IEC 14143-1:1998.



Count Data Functions

Data functions represent the functionality provided to the user to meet internal and external data requirements. Data functions are either internal logical files or external interface 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.

The example on page 2-4 shows a group of related employee data maintained within the Human Resources Application.

- 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 boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application.

The example on page 2-4 shows conversion rate information maintained by the Currency Application and referenced by the Human Resources Application.

Chapter 6 explains the data functions.

**Count
Transactional
Functions**

Transactional functions represent the functionality provided to the user to process data. Transactional functions are either external inputs, external outputs, or external inquiries.

- An external input (EI) 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.

The example on page 2-4 shows the process of entering employee information into the Human Resources Application.

- 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 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, or create derived data. An external output may also maintain one or more ILFs and/or alter the behavior of the system.

The example on page 2-4 shows the process of producing a report that lists all employees stored in the Human Resources Application.

- 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 a 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.

The example on page 2-4 shows the process of inquiring on employee information (input request) and viewing an employee's information when it appears on a screen (output retrieval).

Chapter 7 explains the transactional functions.

Determine the Value Adjustment Factor

The value adjustment factor (VAF) indicates the general functionality provided to the user of the application.

The VAF is comprised of 14 general system characteristics (GSCs) that assess the general functionality of the application. Each characteristic has associated descriptions that help determine the degree of influence of the characteristic. The degrees of influence range on a scale of zero to five, from no influence to strong influence.

Chapter 8 explains how to determine the value adjustment factor. Note that this is an optional step in the function point counting process.

VAF is the value adjustment factor (if the value adjustment factor was not calculated, VAF is 1.00 then the result of the calculation is functional size, reported with units of 'unadjusted function points'. If the VAF is calculated then the result is reported with units of 'adjusted function points').

Calculate the Adjusted Function Point Count

The adjusted function point count is calculated using a specific formula for a development project, enhancement project, or application (system baseline) function point count.

Chapter 9 includes formulas and detailed explanations for each of the three types of function point counts, and for both unadjusted and adjusted function points.

(Blank page)

3 User View

Introduction This chapter presents the concept of the user's role in defining the functional requirements for a project or an application.

Contents This chapter includes the following sections:

Topic	See Page
Definition of User View	3-2
Sizing During the Life Cycle of an Application	3-3
Phase: Initial User Requirements	3-4
Phase: Initial Technical Requirements	3-5
Phase: Final Functional Requirements	3-6
Life Cycle Phase Comparisons	3-7
Hints to Help with Counting	3-8

Definition of User View

A *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.

A function point count is accomplished using the information in a language that is common to both user(s) and developers.

A user view:

- Is a description of the business functions
- Is approved by the user
- Can be used to count function points
- Can vary in physical form (e.g., catalog of transactions, proposals, requirements document, external specifications, detailed specifications, user handbook)

Sizing During the Life Cycle of an Application

User requirements evolve rapidly in the early phases of a project. Decisions must be agreed upon by the users and the developer on which functions will be included in an application. These decisions regarding the functions of the project may be influenced by:

- The needs of the organization
- The risk (business and technical) associated with the project
- The resources available (e.g. budget, staff) in the organization for the project
- The technology available in the organization
- The influence of either users or developers through comments and suggestions

At the beginning of a project, the feasibility study is produced. The feasibility study is the highest level of specification and is usually very short; for example:

- The organization needs an application to comply with a new tax law
- The organization needs an application to manage inventory more efficiently
- The organization needs an application to manage human resources more efficiently

After the feasibility study, the user develops requirements that become more precise over time. At some point, the user will consult with software developers to create the detailed requirements. Software developers can get an early start with their own development and implementation requirements based upon the feasibility study. The discussions between the user and the software developers lead to enhanced requirements. The development process varies among different organizations. This manual will consider, for illustration purposes, a model with three categories of requirements documents:

- Initial User Requirements
- Initial Technical Requirements
- Final Functional Requirements.

As with other development methodologies, the Final Functional Requirements Phase is the most accurate phase to count function points.

Phase: Initial User Requirements

This phase represents user requirements prior to the sessions between the users and the software developers. It may have one or more of the following characteristics:

- Incomplete
For example, Initial User Requirements may lack functions necessary for referential integrity.
- Lack "utility" functionality
For example, essential validation reports or inquiries may be missing.
- Impossible to implement or very difficult to use
For example, a user may ask for an on-line inquiry that requires an hour of CPU processing.
- Too general
For example, requirements may not include the number of fields.
- Varying functional needs, if more than one user is responsible for the project
For example, the requirements of a specific project may vary from one user to another if they do not have the same functional needs.
- Stated requirements without regard for application boundaries
For example, current and/or future application boundaries may not have been considered.
- Expressed in a different context or a terminology incompatible with function point analysis
For example, Initial User Requirements may refer to the physical or manual aspects of the system.

Example

In the Human Resources Department of a corporation, a user expresses his requirements as:

"Whenever I'm working with an employee, I want to be able to view the employee's information by entering his or her name."

This requirement implies the development of an inquiry screen and a group of data on employees. (To keep the example simple, assume that the employee group of data is maintained internally by other employee functions, such as create, update and delete employee, which are not described here).

Functions of the Initial User Requirements example:

EQ	inquiry on a specific employee
ILF	employee group of data

Phase: Initial Technical Requirements

This second phase represents the software developers' view of requirements created from the feasibility study. One task of the software developers, among others, is to organize the requirements into existing applications, if any. The Initial Technical Requirements may include elements which are necessary for the implementation, but which are not used in function point counting (e.g., temporary files, index, etc.). This phase may have one or more of the following characteristics:

- Technology dependence

For example, physical files vary based on the database environment.

- Incorrect identification of the functional needs of the users

For example, software developers may add functions not requested by the users.

- Terminology unfamiliar to the users

For example, software developers may refer to physical files rather than to logical groups of data.

- Functionality may be determined by placing too much emphasis on technical constraints

For example, some developers tend to limit the scope of the requirements by focusing on the computing capacity (CPU) currently available in the organization.

- Boundaries are determined according to the technical architecture of the other applications in the organization

For example, there may be separate technical requirements for client and server, but they would be contained in the same application boundary when counting function points.

Example

Continuing with the same example, the developer states:

"I recognize the need for an employee inquiry. An index is necessary to speed up the retrieval of specific employees."

Functions of the Initial Technical Requirements might be identified as:

EQ inquiry on a specific employee

ILF employee group of data

ILF* index on the employee file

*According to the IFPUG CPM, index files are not counted. In this example, the index file was incorrectly identified as an ILF to illustrate a potential counting error by software developers.

Phase: Final Functional Requirements

This third phase of requirements results from joint sessions between the user(s) and the software developer(s). The joint sessions are necessary to achieve consistent and complete functional requirements for the application. This phase is the final version of the functional requirements before the development phase begins and has the following characteristics:

- Contains terminology which can be understood by both users and software developers
- Provides integrated descriptions of all user requirements, including requirements from different users
- Is complete and consistent enough to accurately count function points
- Each process and group of data is approved by the user
- The feasibility and usability are approved by the software developers

Example

Continuing with the same example:

User: "Whenever I'm working with an employee, I want to be able to view the employee's information by entering his or her name."

Developer: "I recognize the need for an employee inquiry, but many employees may have the same name. It is not possible to specify an individual employee by typing his/her name; therefore, I suggest an on-line employee list (name, location and social security number) from which to select an employee. An index will be necessary to speed up the retrieval of a specific employee."

User: "I agree that the employee selection list is necessary in this case, and it may also be used for purposes other than selecting an employee."

Result of the discussions between the user and the developer:

- Add the on-line list of employees to the functional requirements and the function point count
- Exclude the employee index from the function point count since it is a technical solution

Functions of the Final Functional Requirements example:

EQ inquiry on a specific employee

EQ on-line list of employees

ILF employee group of data

The Final Functional Requirements document is the final version of the requirements before beginning the development phase. At this time, there should be agreement that the documented requirements are complete, formal and approved. The function point count, assuming no additional changes of scope, should be consistent with the count at the completion of development.

Life Cycle Phase Comparisons

Prior to beginning a function point count, determine the application's life cycle phase and whether you are approximating or measuring. Document any assumptions.

Approximating permits assumptions about unknown functions and/or their complexity to accomplish a function point analysis.

Measuring includes the identification of all functions and their complexity to accomplish a function point analysis.

At an early stage, Initial Users Requirements could be the only document available for function point analysis. Despite the disadvantages, this count can be very useful to produce an early estimate. Uses of function point analysis for approximating at the various life cycle phases is presented below:

Life Cycle Phase	Size can be approximated	Size can be measured
Proposal: users express needs and intentions	yes	no
Requirements: developers and users review and agree upon expression of user needs and intentions	yes	yes
Design: developers may include elements for implementation that are not used for function point analysis	yes	yes
Construction	yes	yes
Delivery	yes	yes
Corrective Maintenance	yes	yes

Note: No specific development life cycle is implied. If using an iterative approach, you may expect to approximate for some time into the application life cycle.

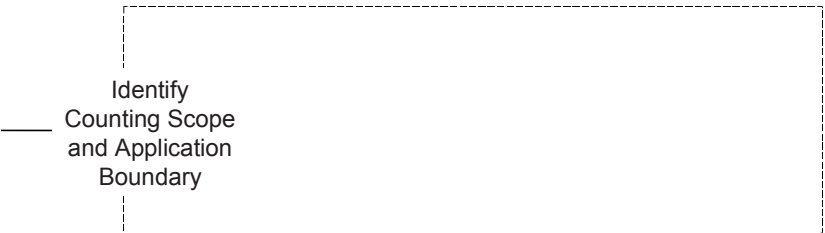
Be aware and measure only new or refined agreed upon user needs and intentions.

Hints to Help with Counting

The following hints may help identify the user view of an application and apply function point analysis.

- Do not assume that one physical file equates to one logical file when viewing data logically from the user perspective.
- Although some storage technologies, such as tables in a relational DBMS or a sequential flat file, relate closely to ILFs or EIFs, do not assume that this is always equal to a one-to-one physical-logical relationship.
- Do not assume all physical files must be counted or included as part of an ILF or EIF.
- Look at the different paper forms currently used by the user(s) when identifying transactional functions.
- A transaction which occurs in multiple physical inputs, transaction files or screens, but which has identical processing logic typically corresponds to one transactional function (EI, EO, EQ).
- Remember that one physical report, screen or batch output file can, when viewed logically, correspond to a number of EOs/EQs.
- Remember that two or more physical reports, screens or batch output files can correspond to one EO/EQ if the processing logic is identical.
- Remember that resorting or rearranging a set of data does not make processing logic unique.

4 Determine Type of Count



Introduction The first step of the function point counting procedure is to identify the type of function point count.

This chapter includes a detailed explanation of the types of function point counts: development project, enhancement project, and application.

Contents This chapter includes the following sections:

Topic	See Page
Definitions: Types of Function Point Counts	4-2
Development Project	4-2
Enhancement Project	4-2
Application	4-2
Diagram of Types of Counts	4-3
Estimated and Final Counts	4-3

Definitions: Types of Function Point Counts

Function point counts can be associated with either projects or applications. There are three types of function point counts:

- Development project
- Enhancement project
- Application

The following paragraphs define each type of function point count.

Note: Chapter 9 includes the formulas used to calculate the adjusted function point count for each of the three types of counts.

Development Project

The development project function point count measures the functions provided to the users with the first installation of the software delivered when the project is complete.

Enhancement Project

The enhancement project function point count measures the modifications to the existing application that add, change, or delete user functions delivered when the project is complete.

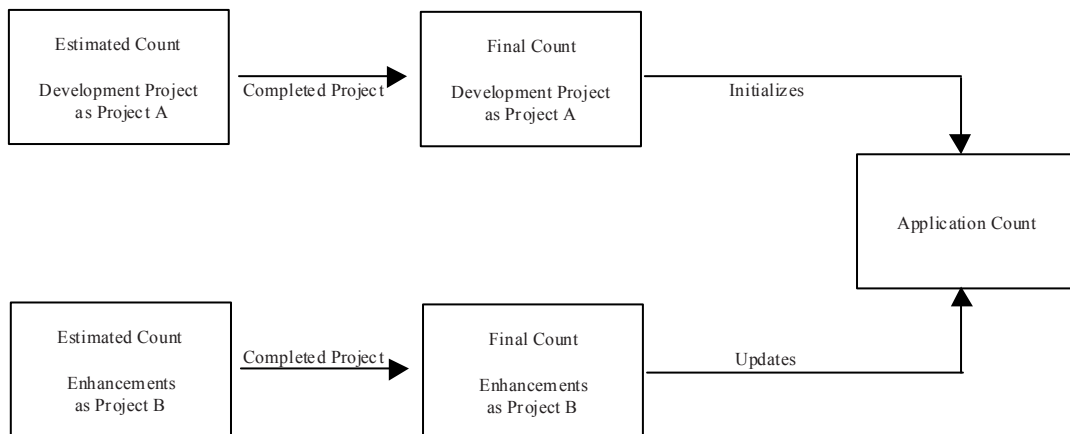
When the functionality from an enhancement project is installed, the application function point count must be updated to reflect changes in the application's functionality.

Application

The application function point count and project count are associated with an installed application. It is also referred to as the *baseline* or *installed* function point count. This count provides a measure of the current functions the application provides the user. This number is initialized when the development project function point count is completed. It is updated every time completion of an enhancement project alters the application's functions.

Diagram of Types of Counts

The following diagram illustrates the types of function point counts and their relationships. (Project A is completed first, followed by Project B.)

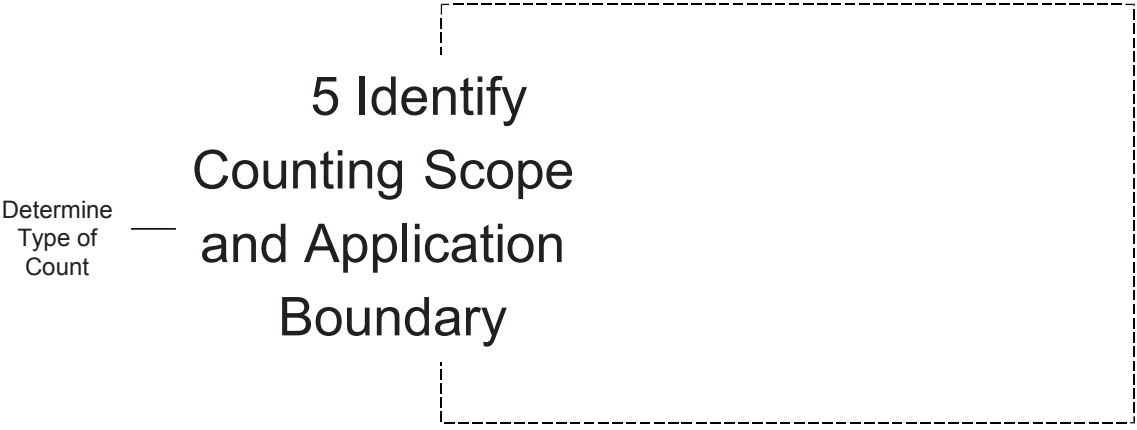


Estimated and Final Counts

It is important to realize that early function point counts are estimates of delivered functionality. In addition, as the scope is clarified and the functions developed, it is quite normal to identify additional functionality that was not specified in the original requirements. This phenomenon is sometimes called *scope creep*.

It is essential to update application counts upon completion of the project. If the functionality changes during development, the function point count at the end of the life cycle should accurately reflect the full functionality delivered to the user.

(Blank page)



Introduction This chapter defines the terms: purpose of the count, counting scope and application boundary. It includes rules, procedures, and hints to determine boundaries for applications and to establish the scope of the count.

Contents This chapter includes the following sections:

Topic	See Page
Definition of Counting Scope and Application Boundary	5-2
Definition of the Purpose of the Count	5-2
Definition of the Counting Scope	5-2
Definition of the Application Boundary	5-3
Counting Scope and Application Boundary Rules and Procedures	5-5
Boundary Rules	5-5
Counting Scope and Application Boundary Procedures	5-5
Hints to Help to Identify the Counting Scope and the Application Boundary	5-6

Definition of Counting Scope and Application Boundary

This section defines counting scope and application boundary and explains how they are influenced by the purpose of the count.

Definition of the Purpose of the Count

The purpose of a function point count is to provide an answer to a business problem.

The purpose:

- Determines the type of function point count and the scope of the required count to obtain the answer to the business problem under investigation
- Influences the positioning of the boundary between the software under review and the surrounding software; e.g., if the Personnel Module from the Human Resources System is to be replaced by a package, the users may decide to reposition the boundary and consider the Personnel Module as a separate application

Examples of purposes are:

- To provide a function point count as an input to the estimation process to determine the effort to develop the first release of an application
- To provide a function point count of the installed base of applications
- To provide a function point count to enable the comparison of functionality delivered by two different suppliers' packages

Definition of the Counting Scope

The counting scope defines the functionality which will be included in a particular function point count.

The scope:

- Defines a (sub) set of the 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 so as to provide answers relevant to the purpose for counting
- Could include more than one application

The *scope* of:

- An enhancement function point count includes all the functions being added, changed and deleted. The boundary of the application(s) impacted remains the same. The functionality of the application(s) reflects the impact of the functions being added, changed or deleted.
- A development function point count includes all functions impacted (built or customized) by the project activities.
- An application function point count may include, depending on the purpose (e.g., provide a package as the software solution):
 - only the functions being used by the user
 - all the functions delivered

The application boundary of the two counts is the same and is independent of the *scope*.

Definition of the Application Boundary

The application boundary indicates the border between the software being measured and the user.

The 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 (EIs, EOs and EQs) pass into and out from the application
- Encloses the logical data maintained by the application (ILFs)
- Assists in identifying the logical data referenced by but not maintained within this application (EIFs)
- Is dependent on the user’s external business view of the application. It is independent of technical and/or implementation considerations

For example, the following diagram shows boundaries between the Human Resources application and the external applications, Currency and Fixed Assets. The example also shows the boundary between the human user (User 1) and the Human Resources application.

