



MANUAL TRANSMITTAL

Department of the Treasury
Internal Revenue Service

2.5.10

DECEMBER 17, 2021

EFFECTIVE DATE

(12-17-2021)

PURPOSE

- (1) This transmits revised IRM 2.5.10, Systems Development, Function Point Standards.

MATERIAL CHANGES

- (1) Signature, Changed the title and name of the Associate, Chief Information Officer from Linda Gilpin to title Nancy A. Sieger Chief Information Officer
- (2) Effect on Other Documents, Removed the date, and included “supplements IRM 2.5.1, Systems Development”
- (3) Audience, Added the word “software ”to “developing” and included “systems analysis”
- (4) 2.5.10.1 (5) (6), Changed “work flow” to “workflow” and removed period after program
- (5) 2.5.10.1.3 (1) (c), Change Chief Technology Office to Chief Technology Officer
- (6) 2.5.10.1.3 (3) (a), changed “an ”unified to “a ”unified
- (7) 2.5.10.1.8 (13), Added Consortium for Information & Software Quality, <https://www.it-cisq.org/index.htm>
- (8) 2.5.10.1.8 (14), Added International Organization for Standards (ISO) /International Electrotechnical Committee (IEC) 5055:2021 Information Technology - Software Measurement - Software Quality Measurement - Automated Source Code Quality Measures, First edition 2021-03
- (9) 2.5.10.2 (1) (i), Corrected Counting Rules (EO):’ Acronym should be ‘CR
- (10) 2.5.10.4.1 (1) (2) Removed duplicate paragraph, and included new information
- (11) 2.5.10.5.3, Added Software Sizing Standards
- (12) 2.5.10.5.3 (1a), Added explanation of Software Sizing Standards
- (13) 2.5.10.5.3 (2), Added hyperlink for Consortium of IT Software Quality
- (14) 2.5.10.5.3.1(1), Added Automated Function Points (AFP) Standard
- (15) 2.5.10.5.3.1 (1), Added the purpose of Automated Function Points (AFP) Standard
- (16) 2.5.10.5.3.1 (2), Added the advantage of Automated Function Points (AFP) Standard
- (17) 2.5.10.5.3.1 (3), Added sentence how to use function points for the following actions
- (18) 2.5.10.5.3.1 (a-f), Added the use of function points for the following actions:
 - a. Perform analysis of software quality and productivity.
 - b. Determine expenses or resources for software development, enhancement, and maintenance.
 - c. Align estimation methods against the results of past estimates.
 - d. Normalize data used in software comparisons.

- e. Estimate the size of a procured application package, e.g., Commercial Off The Shelf (COTS), or customized system) by sizing all the functionality included in the package.
- f. Estimate the Return On Investment (ROI) of an application by sizing the functionality that matches the organization's requirements.

- (19) 2.5.10.5.3.2, Added Automated Enhancement Points Standards (AEP).
- (20) 2.5.10.5.3.2 (1), Added the purpose of Automated Enhancement Points Standards (AEP)
- (21) 2.5.10.5.3.3, Added Measuring for Code Quality and corrected table format
- (22) 2.5.10.5.3.4, Added Automated Technical Debt
- (23) 2.5.10.5.3.4 (1) (a-b) (2) (a-b), Added purpose and examples of Technical Debt.
- (24) Exhibit 2.5.10-1, Acronyms/Terms - Included (IEC - International Electrotechnical Committee, AEP - Automated Enhancement Points, AFP - Automated Function Point, ISO - International Standards Organization, CISQ - Consortium for IT Software Quality, and Commercial Off The Shelf, ROI - Return On Investment and corrected "IFP "from Internal Function Point to and ITM - Integrated Talent Management
- (25) Exhibit 2.5.10-2, Terms/Definitions - Included (Benchmark Testing, Quality Adjusted Productivity, and Return On Investment)
- (26) Added Figure 2.5.10-1, Automated Technical Debt Example

EFFECT ON OTHER DOCUMENTS

This IRM 2.5.10 dated 05-22-2020 is superseded, and supplements IRM 2.5.1, Systems Development.

AUDIENCE

The audience intended for (but not restricted to) this transmittal is for personnel responsible for engineering, software developing, systems analysis, or maintaining Agency software systems identified in the Enterprise Architecture. The audience includes employees, contractors, all other operating divisions, and functional employees.

Nancy A. Sieger
Chief Information Officer

2.5.10

Function Point Standards

Table of Contents

2.5.10.1 Program Scope and Objectives

2.5.10.1.1 Background

2.5.10.1.2 Authority

2.5.10.1.3 Roles and Responsibilities

2.5.10.1.4 Program Management and Review

2.5.10.1.5 Program Controls

2.5.10.1.6 Acronyms/Terms

2.5.10.1.7 Terms/Definitions

2.5.10.1.8 Related Resources

2.5.10.2 Function Point Analysis Fundamental Concepts

2.5.10.3 Estimation Techniques Overview

2.5.10.3.1 Estimation Techniques - Function Points (FP)

2.5.10.3.2 Application Function Point Count

2.5.10.3.3 Development Function Point Count

2.5.10.3.4 Enhancement Function Point Count

2.5.10.4 Progressive Function Point Analysis Fundamental Concepts

2.5.10.4.1 Progressive Function Point Analysis Overview

2.5.10.5 Additional Industry Standard Estimation Techniques

2.5.10.5.1 IRS Services - Statistical Analysis

2.5.10.5.2 Enterprise Life Cycle (ELC) Accepted Methods - Estimation Techniques

2.5.10.5.3 Software Sizing Standards

2.5.10.5.3.1 Automated Function Points (AFP) Standard

2.5.10.5.3.2 Automated Enhancement Points Standard (AEP)

2.5.10.5.3.3 Measuring for Code Quality

2.5.10.5.3.4 Automated Technical Debt

Exhibits

2.5.10-1 Acronyms/Terms

2.5.10-2 Terms/Definitions

2.5.10.1
(12-17-2021)
Program Scope and Objectives

- (1) **Purpose:** This Internal Revenue Manual (IRM) establishes standards and guidelines to provide essential aspects of the Function Point Analysis program.
- (2) **Audience:** The audience intended for this transmittal (but not restricted to) is personnel responsible for: project management, engineering, software developing, or maintaining Agency software systems identified in the Enterprise Architecture. The audience includes employees, contractors, and all other operating divisions and functional employees.
- (3) **Policy Owner:** The Information Technology, Associate Chief Information Officer, Application Development is the authority for this IRM.
- (4) **Program Owner:** The Information Technology, Application Development, Director of Technical Integration Organization (TIO) is responsible for the administration, procedures, and updates related to the program.
- (5) **Primary Stakeholders:** These can be other areas that are affected by these procedures or have input to the procedures. The affects may include a change in workflow, additional duties, change in established time frames, and similar issues.
- (6) **Program Goals:** The objective this program. is to provide a reliable measurement tool option for developers to be able to accurately quantify the functionality of the software product they are building or maintaining, and compare the factors that impact the project's cost.

2.5.10.1.1
(12-17-2021)
Background

- (1) The method Function Point Analysis (FPA) was first developed during 1979 by Allan Albrecht of IBM, and is supported by the International Function Point User Group. FPA is a structured technique of problem solving from the functional aspect. It is a method used to break the systems into smaller components, so they can be clearly understood and analyzed.
- (2) Function Point is an element of software development which helps to approximate the cost of development early in the process.
- (3) Traditionally, one of the most challenging aspects of system analysis is the accurate estimation of project sizing, required development time, and end-user value. Productivity evaluation methods, such as lines of code or specific coding constructs, require the program to be written first. Function Point Analysis sizes an application from an end-user rather than coding language perspective. In fact, it is totally independent of all language considerations. FPA provides essential background information pertinent to the program. For example, new legislation changes in work flow or program ownership, and reasons why the work is being done.
- (4) A function point is defined as one end-user business function. Therefore, a program rated as having 'x' function points delivers 'x' business functions to the user.

2.5.10.1.2
(12-17-2021)
Authority

- (1) IRM 2.5.1 System Development, is the overarching IRM for all System Development programs within the IRS
- (2) Clinger-Cohen Act of 1996
- (3) Federal Information Security Modernization Act of (FISMA) 2014

- (4) The Office of Management and Budget (OMB)
- (5) Government Performance Results Act, 1993
- (6) Treasury Inspector General Tax Administration (TIGTA)
- (7) International Function Point User Group (IFPUG)

2.5.10.1.3
(12-17-2021)

Roles and Responsibilities

- (1) Application Development's (AD) chain of command and responsibilities include:
 - a. **Commissioner:** Oversees and provides overall strategic direction for the IRS. The Commissioner's and Deputy Commissioner's main focus is for the IRS's services programs, enforcement, operations support, and organizations. Additionally, the Commissioner's vision is enhancing services to the nation's taxpayers, balancing appropriate enforcement of the nation's tax laws while respecting taxpayers' rights.
 - b. **Deputy Commissioner, Operation Support (DCOS):** Oversees the operations of Agency-Wide Shared Services: Chief Financial Officer, Human Capital Office, Information Technology, Planning Programming and Audit Oversight and Privacy, and Governmental Liaison and Disclosure.
 - c. **Chief Information Officer (CIO):** The CIO leads Information Technology, and advises the Commissioner on Information Technology matters, manages all IRS IT resources, and is responsible for delivering and maintaining modernized information systems throughout the IRS. Assisting the Chief Technology Officer (CTO) is the Deputy Chief Information Officer for Operations.
 - d. **Application Development (AD) Associate Chief Information Officer (ACIO):** The AD ACIO reports directly to the CIO; oversees and ensures the quality of: building, unit testing, delivering and maintaining integrated enterprise-wide applications systems to support modernized and legacy systems in the production environment to achieve the mission of the Service. AD works in partnership with customers to improve the quality of and deliver changes to IRS information systems products and services.
 - e. **Deputy AD Associate CIO (ACIO):** The Deputy AD ACIO reports directly to the AD ACIO, and is responsible for:
 - Leading all strategic priorities to enable the AD Vision, IT Technology Roadmap and the IRS future state
 - Executive planning, and management of the development organization which ensures all filing season programs are developed, tested, and delivered on-time and within budget
- (2) AD has the following Domains:
 - a. Compliance Domain
 - b. Corporate Data Domain
 - c. Customer Service Domain
 - d. Data Delivery Service (DDS) Domain
 - e. Delivery Management; Quality Assurance (DMQA) Domain
 - f. Identity & Access Management (IAM) Organization Domain
 - g. Internal Management Domain
 - h. Submission Processing Domain
 - i. Technical Integration Organization (TIO) Domain
- (3) **Director, Compliance:** Provides executive direction for a wide suite of Compliance focused applications and oversee the IT Software Development organization to ensure the quality of production ready applications.

- a. Directs and oversees a unified cross-divisional approach to compliance strategies needing collaboration pertaining for the following:
 - Abusive tax avoidance transactions needing a coordinated response
 - Cross-divisional technical issues
 - Emerging issues
 - Service-wide operational procedures
- (4) **Director, AD Corporate Data:** Directs and oversees the provisioning of authoritative databases, refund identification, notice generation, and reporting.
- (5) **Director, Customer Service:** Directs and oversees Customer Service Support for service and communication with internal and external customers and providing taxpayers with self-service online capabilities.
 - a. Customer Service Domain's applications and systems provide:
 - Tax law assistance
 - Taxpayer education
 - Access to taxpayer account data
 - Maintenance of modernized information systems that meet the customer's needs for researching, updating, analyzing, and managing taxpayer accounts
 - b. Services to internal and external customers are provided through five primary means:
 - Centralized Contact Centers (for telephone, written, and electronic inquiries)
 - Self-service applications (via the telephone and Internet)
 - Field Assistance (for walk-in assistance)
 - Web Services
 - Management of Taxpayer Accounts
- (6) **Director, Data Delivery Services:** Oversees and ensures the quality of data with repeatable processes in a scalable environment. The Enterprise Data Strategy is to transform DDS into a data centric organization dedicated to deliver Data as a Service (DaaS) through:
 - Innovation - new methods, discoveries
 - Renovation - streamline or automate
 - Motivate - incent and enable individuals
- (7) **Director, Delivery Management & Quality Assurance (DMQA):**
 - Executes the mission of DMQA by ensuring AD has a coordinated, cross-domain, and cross-organizational approach to delivering AD systems and software applications
 - Reports to the AD ACIO, and chairs the AD Risk Review Board.
- (8) **Director, Identity & Access Management (IAM) Organization:** Provides oversight and direction for continual secure online interaction by verification and establishing an individual's identity before providing access to taxpayer information "identity proofing" while staying compliant within federal security requirements.
- (9) **Director, Internal Management:** Provides oversight for the builds, tests, deliveries, refund identification, notice generation, and reporting.

- (10) **Director, Submission Processing:** Provides oversight to an organization of over 17,000 employees, comprised of: a headquarters staff responsible for developing program policies and procedures, five W&I processing centers, and seven commercially operated lockbox banks. Responsible for the processing of more than 202 million individual and business tax returns through both electronic and paper methods.
- (11) **Director, Technical Integration:** Provides strategic technical organization oversight ensuring applicable guidance, collaboration, and consolidation of technical integration issues and quality assurance for the Applications Development portfolio.
- (12) **IT Strategy and Planning, Financial Management Services, Estimation Program Office (EPO),** see IRM 2.5.10.1.5 for description of responsibilities.
- (13) **Chief, Research, Applied Analytics, and Statistics (RAAS):** The Chief of RAAS organization is responsible for the coordination of research within the IRS and research conducted within the RAAS organization.
- (14) **Director, RAAS, Management, and Engagement:** The Director of RAAS organization is responsible for coordinating research and analytical data across the enterprise, and Business Performance Reviews and measures pertaining to high-level organization performance information on a quarterly basis. The RAAS organization has five divisions:
 - a. **Data Exploration and Testing (DET):** Extracts value from analytics through exploration, testing and learning to improve tax compliance enterprise operations.
 - b. **Data Management Division (DMD):** Provides tools, services and processes to the IRS research community and operational customers for IRS programs.
 - c. **Knowledge Development and Application (KDA):** Provides domain expertise in tax administration, economics, behavioral research and statistical methodology for advising IRS senior leadership and agency stakeholders working in development, analysis, and implementation of policies and procedures.
 - d. **Statistics of Income (SOI):** Responsible for formulating and executing overall IRS statistical policies and programs.
 - e. **Strategy and Business Solutions (SBS):** Engage with stakeholders to frame and solve the major challenges facing the tax administration, and aligns with IRS enterprise-wide Research and Analytics talent, knowledge, and resources.

2.5.10.1.4
(12-17-2021)
**Program Management
and Review**

- (1) **Program Reports:** Analytical reports are implemented as listed below:
 - a. The Research, Applied Analytics, and Statistics (RAAS) organization has general program oversight of analytic research activities and reporting within the IRS.
 - b. Senior executives in each organization have the responsibility of conducting and managing research within their organizations, but can submit analytic data to the RAAS organization for additional research.
- (2) **Program Effectiveness:** Function points are often considered the cornerstone metric of most software measurement programs. Mainly because they can be used across the development life cycle, and their capability to display measured view of data with a technical and business orientation.

- a. The key ingredients to an effective and successful measurement program within the IRS include the following:
 - A defined set of measures that are properly documented
 - A metrics repository that is centrally located and secured
 - The ability of the organization to effectively report the results and appropriately analyze the metrics data

Note: The greatest value in utilizing function point measures is as an industry indicator of performance. For example, obtaining sources of data from each organization for the purpose of comparing rates of performance to external industry benchmarks.

2.5.10.1.5
(12-17-2021)
Program Controls

- (1) IRS Research and Analytics (R & A) repository is used to catalog IRS research project inventory, see IRM 1.7.1 Servicewide Research for Tax Administration.
- (2) The IT Strategy and Planning, Financial Management, Estimation Program Office (EPO) Support Services organization oversees finance operations for assigned IT Divisions to meet the IRS's business requirements through financial solutions in-line with organizational goals and strategies. Currently the subsection Support Services consist of four areas:
 - a. Team A - Provides support to ACIO Enterprise Operations
 - b. Team B - Provides support to Cybersecurity, Enterprise Services, Management, and Office of the Chief Information Officer (CIO)
 - c. Team C - Provides support to Application Development (AD)
 - d. Team D - Provides support to User Network Services (UNS)

2.5.10.1.6
(12-17-2021)
Acronyms/Terms

- (1) For Acronyms and Terms, see Exhibit 2.5.10-1

2.5.10.1.7
(12-17-2021)
Terms/Definitions

- (1) For Terms and Definitions see, Exhibit 2.5.10-2

2.5.10.1.8
(12-17-2021)
Related Resources

- (1) IRM 10.5.1 Security, Privacy and Assurance, Privacy and Information Protection, Privacy Policy
- (2) Federal Information Security Modernization Act (FISMA), 2014
- (3) International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 29881:2010
- (4) Information Technology, Systems and Software Engineering - FISMA 1.1 Functional Size Measurement Method, reviewed 2019
- (5) Object Management Group (OMG) <https://www.omg.org/> - This is a technology standards consortium which set the standards for:
 - Business Process Model and Notation (BPMN)
 - Common Object Request Broker Architecture (CORBA)
 - Common Warehouse Metamodel (CWM)
 - Data Distribution Service for Real-time Systems (DDS)
 - Model Driven Architecture (MDA)

- Meta Object Facility (MOF)
 - Systems Modeling Language (SML)
 - Unified Modeling Language (UML)
- (6) International Function Point User Group (IFPUG) - ISO/IEC 20926:2009 Software and System Engineering
 - (7) IRM 2.16.1.4.1.1 1 Considerations for ELC Path Selection
 - (8) Longstreet, David H. Function Point Manual., Publisher: Longstreet Consulting, Inc. April 2000. ISBN-10:0970243901, ISBN-13:0970243904
 - (9) Tutorial Point Estimation Techniques, https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_fp_counting_process.htm
- #

#
- (11) Ruben Gerad Mathew. Progressive Function Point Analysis: Advanced Estimation Techniques for IT Projects, 2014
 - (12) IRM 10.8.6 Application Security and Development
 - (13) Consortium for Information & Software Quality, <https://www.it-cisq.org/index.htm>.
 - (14) International Organization for Standards (ISO) / IEC 5055:2021 Information Technology - Software Measurement - Software Quality Measurement - Automated Source Code Quality Measures, First edition 2021-03

2.5.10.2 (12-17-2021)

Function Point Analysis Fundamental Concepts

- (1) Function Point Analysis (FPA) sizes an application from an end-user rather than coding language perspective, and is totally independent of all language considerations.
 - a. **Application Boundary:** To perform FPA, first the boundary of the application must be defined. A boundary is used to define what is internal to the application, and what components or interfaces are external to the application. To determine the boundaries of an application system, evaluate all functional and technical aspects as the following:
 - User interface design
 - Architecture (e.g., program language and program design)
 - Shared application system components (e.g., database tables, programs, and visual objects)
 - Identification of all interfaces or integration points with other application systems
 - b. **User:** Any person, device or process that interacts and communicates with the software
 - c. **User Recognizable:** Any directives, operation, or process or organization that is understood by to the developers and business users, and is commonly accepted as term user recognizable.
 - d. **User View:** This is a description of a business function, operation or process that is used to calculate function points.
 - e. **Control Information:** These are key fields that assist with determining how to process the data confined in it such as the marker for historical data mining, or an indicator to display the operation will be performed

while processing the data. To perform FPA the functional categories are divided into two main types:

- **Data Functions:** An action of storing and retrieving data in both local files or databases, and is also external to the application through remote interfaces, associated middleware or outside the boundary of the implicated application.

- **Transaction Functions:** Pertain to read / write operations performed on data. The transaction functions will read or write data from and to an Internal Logical File (ILF) or External Interface Files (EIF). There are three basic type of transaction functions: External Input, External Inquiry and External Output.

- f. **International Function Point User Group (IFPUG): External Input (EI):** Any data input screen or operation that is used to perform a Create, Read, Update, and Delete (CRUD) operation on the ILF. Control information may be passed as an indicator of the process to be performed.
- g. **IFPUG: External Inquiry (EQ):** This is a data read operation, where information is retrieved, or data is sent outside the boundary.
- h. **IFPUG: External Output (EO):** This is a data read operation, and the result may be displayed, printed, and transmitted to an external drive or application outside the application boundary. The main difference of an EO and EQ is that it may contain some mathematical equation, calculation of sum, average, count or manipulation of data, or may create additional fields e.g., totals, subtotals, calculation of final cost and may also update the ILF to reflect the computed sum.
- i. **Counting Rules (CR):** Sends data or control information external to the application's boundary. For the external process, one of these three conditions must apply:
 - The processing login is unique from the processing logic performed by other EOs for the application.
 - The set of data elements known are different from other EOs in the application.
 - The ILFs or EIFs referenced are different from files referenced by other EOs in the application.
- j. **IFPUG: Data Element Type (DET):** Is a unique user recognizable, non-recursive field, and is dynamic, not static. If a DET is recursive then only the first occurrence of the DET is considered not every occurrence. The dynamic field is read from a file or created from DETs contained in a File Type Referenced (FTR).
- k. **IFPUG: File Type Referenced (FTR):** File Type Referenced are logical groups either local storage or remote identified in a Transaction function. The logical groups of data must be identified as either an ILF or EIF.
 - **Counting Rules:** Count one FTR for each ILF or EIF in the transaction function. Identify all the ILFs and EIFs in an application, and follow the process steps in a transaction function by counting each unique reference to the EIF or ILF
 - **Identifiers:** Identify all the ILFs and EIFs in an application, and follow each unique reference to the EIF or ILF
- l. **Counting Function Points (Unadjusted Function Point (UFP)):** UFP is the total sum of the high, medium and low count of all the operations: EI, EO, EQ, ILF, and EIF.

- (2) **Reuse FP:** It is important to identify the functional reuse that may exist within a project. There are two types of Reuse implemented, the functional reuse is used with each data and transaction function to represent the percentage of

reuse in the specified function. This Reuse is only used in progressive function point estimation, and may be practical when estimating the size of modules for:

- Large complex libraries
- Commercial-Off-The-Shelf (COTS) products

Note: Reuse FP may not be available in some COTS when they are designed as a black box. For this scenario the weighted reuse percentage is based on user experience and analogy.

- (3) **Reuse Percentage:** A 0% reuse represents 100% new software build, any percentage assigned shows the total amount of reusable component available to the application for integration. See table below:

0% Reuse Example

First example of 0 % Reuse
Reuse % = Reusable FP Count / Total FP Count

- (4) **Reuse with Adaption Adjustment Factor (AAF):** This second method of reuse was presented at the IFPUG conference to calculate reuse with respect to integration of an entire application or component as a whole for any project. This reuse considers three factors: Design Modification (DM), Code Modification (CM), Integration & Testing (I&T). See table and steps for Calculation of Reuse count below:

Reuse Calculation with (AAF) Formula

Reuse % = Reusable FP Count / Total FP Count
$AAF = .4DM + .3CM + .3I\&T$

- Calculate FP count of Reusable component (100FP)
- Determine the percentage of the Development Effort (50%)
- Determine the percentage of Coding Effort (25%)
- Determine the Testing & Integration Effort (75%)
- Calculate Adaption Adjustment Factor

Example - Reuse using the Adaption Adjustment Factor(AAF)

Example - Reuse using Adaption Adjustment Factor
$AAF = (0.4 \times 0.25) + (0.3 \times 0.50) + (0.3 \times 0.50)$ $AAF = .1 + 0.15 + .15$ $AAF = .4$ Calculate Reuse Reuse = FP Count of Reusable component x AAF Reuse = 100 x .4 Reuse = 40FP

2.5.10.3
(12-17-2021)
**Estimation Techniques
Overview**

- (1) This IRM presents Function Point Analysis, and provides the appropriate IRS standards.
- (2) Traditionally, one of the most challenging aspects of system analysis is the accurate estimation of project sizing, required development time, and end-user value. Productivity evaluation methods, such as lines of code or specific coding constructs, require the program to be written first.
- (3) The Estimation Program Office owns the IT Estimation Procedure. EPO offers training courses with instruction for IT projects on how to comply with Estimation compliances, see <https://irsource.web.irs.gov/Pages/ITM.aspx> for IT Estimation Procedures.

2.5.10.3.1
(12-17-2021)
**Estimation Techniques -
Function Points (FP)**

- (1) A function point is defined as one end-user business function. Therefore, a program rated as having 'x' function points delivers 'x' business functions to the user.
- (2) FPA methodology validates the individual elements and the related groups with a complexity of high, medium, and low and assigns a function point count for each subset. There are five basic function types see table below:

Function Types Examples

Measurements Parameters	Example
1. Number of External Inputs (EI)	Input screen and tables
2. Number of External Output (EO)	Prompts and interrupts
3. Number of External Inquiries (EQ)	Prompts and interrupts
4. Number of Internal Logical Files (ILF)	Databases and directories
5. Number of External Interface Files (EIF)	Shared databases and shared routines

- (3) Set forth below is a list of each function point standard and its description:
 - a. **Acceptability:** Function Point Analysis is an acceptable software sizing methodology. It may be used for estimating the size of all modernized system new development projects and their enhancements, and all legacy production system software and their enhancements.
 - b. **FP Counting Process:** FP counts or estimates shall be conducted by IRS project developers who are experienced in the IRS function point counting practices. The FP Counting Process involves the following steps:
 - Step 1** - Determine the type of count: development, application, or enhancement.
 - Step 2** - Determine the boundary of the count - The boundary specifies the border between the application being measured, and the external application or the user domain.
 - Step 3** - Identify each Elementary Process (EP) required by the user.
 - Step 4** - Determine the unique EP.

Step 5 - Measure the data functions.

Step 6 - Measure the transactional functions.

Step 7 - Calculate functional size (unadjusted function point count).

- c. **Estimators:** Commercial-Off-The-Shelf (COTS) estimation tools may be used. The Function Point Team will conduct the resource estimates and forecasts. Project offices will review the templates to ensure that the profile for the project (as derived from the answers to the questionnaire) is accurate. This review will occur before the first resource forecast is finished. This in effect calibrates the tool to the IRS environment.
- d. **Forecasting:** Forecasting of resources needed for software projects within IS may be conducted based upon function point counts or estimates and appropriate associated software. These forecasts must be statistically tested for correctness and reliability.
- e. **Internal Logical File (ILF) Model:** User recognizable group of logically related data or control information within the boundary of the application being measured. The primary purpose of an ILF is to hold data maintained through one or more elementary processes of the application being counted.
- f. **IT Strategy and Planning, Financial Management, Estimation Program Management Office (EPMO):** Provides a staff of IRS employees who understand the project's requirements in sufficient detail to provide all necessary information for the required function point counts or estimates. This applies to software developed by either IRS project teams or contractor project teams.
- g. **Method:** Function point counts or estimates shall be conducted using either the long count, or fast count.
- h. **Repository:** All function point counts and estimates shall be recorded, stored, and maintained both in the electronic repository.
- i. **Process Modifications:** Definition and recommended modifications to the forecasting process shall be provided by a centralized Function Point Team.

(4) FPA requires the following two major steps:

- a. Identify business functions and then organize them into the following five groups: outputs, inquiries, inputs, files, and interfaces.
- b. Adjust the raw totals (derived during the identification step) according to production environment factors referred to as General Application Characteristics.

2.5.10.3.2
(12-17-2021)

Application Function Point Count

- (1) Application counts are calculated as the function points delivered, and exclude any conversion effort (prototypes or temporary solutions) and existing functionality that may have existed.

2.5.10.3.3
(12-17-2021)

Development Function Point Count

- (1) Development Function Point count is associated with new development work, and may include the prototypes which may have temporary solutions.

2.5.10.3.4
(12-17-2021)

Enhancement Function Point Count

- (1) Enhancement Function Point Count is used to size enhancement projects, and is counted as sum of all added, changed or deleted function points in the application. By tracking the size and associated costs, a historical database for your organization, division or domain can be built.

2.5.10.4

(12-17-2021)

Progressive Function Point Analysis Fundamental Concepts

- (1) **Progressive Function Point** - This measure uses coefficient index based on IFPUG standard function point reference values to create an accurate function point count. Progressive function point reference is useful for accurate sizing, incorporating workflows, functional reuse, and can associate to the actual effort involved since all variables in the function and process steps are taken into consideration.
- (2) **Progressive Function Point Analysis** - This estimation methodology was created to provide more enhanced accuracy in the estimation of Function Points using the same FPA principles used for size estimation of IT projects. The improvements are based on the actual Function Point count based on elementary inputs, outputs and integrated process flows with the corresponding functions. Progressive FPA is best suited for applications that have the following characteristics:
 - a. Projects that have complex business rules and multiple workflows.
 - b. Highly complex operations other than standard CRUD functionality.
 - c. Integration of Reuse within functions.
 - d. Greater accuracy of Cost and Schedule index.
 - e. Greater clarity and accountability into costing of applications.
 - f. When implementing FPA for agile projects.
 - g. Integration of new languages and programming tools.
- (3) **Progressive FPA: Integrated Process Flow (IPF)** - Consist of the following:
 - a. User Recognizable
 - b. Elementary non-repetitive process steps - both data and transaction functions.
 - c. Include one or more process steps within specific logical groups inside the boundary of the application.
 - d. Include External Component Interfaces (ECI) to reuse functionality in other application(s).
- (4) **Progressive FPA: External Component Interfaces (ECI)** - Integrated process flows of other applications. ECI are external functions referenced by applications to begin some needed functionality by reusing functions.
- (5) **Progressive FPA: Process Element Type (PET)** - User Recognizable elementary computational process steps to manage, process or present data and control information within each data or transaction function. The primary purpose is to complete a desired behavior through one or more activities within the application boundary that is unique to the application, and may include calls and external references to other ECIs outside the application boundary.

2.5.10.4.1

(12-17-2021)

Progressive Function Point Analysis Overview

- (1) Progressive function points was derived for greater accuracy in the estimation of Function Point Analysis, and is used for the size estimation of IT Projects. As projects are initiated in organizations, the first step is to complete the Analysis phase. Achieving a valid analysis methodology in a software project mitigate the risk of failure. It also contributes to better decision-making, management, and control of the resources assigned for the development of the project, as well as establishing the schedule and costs.
- (2) When providing Progressive FPA estimates, the project's cost and work efforts are justified, and can be validated against the actual work.

2.5.10.5
(12-17-2021)**Additional Industry
Standard Estimation
Techniques**

- (1) Estimation techniques are critical in the system development life cycle, where the time necessary to complete a task is estimated before a project begins. The estimation technique is a process of finding an approximation, which is a value to use even if input data may be uncertain. The following are additional frequently used estimation techniques:
- Analogous Estimation** - Uses similar historical project information to estimate the duration or cost of your current project. This estimation technique is used when there is limited information regarding your current project. If your current project has similar requirements and resources of a previous project, you may use the historical data.
 - Parametric Estimating** - Uses the relationship between variables to calculate the cost or duration when limited information is available for the current project. The estimate is determined by identifying the unit cost or duration and the number of units required for the project or activity. The measurement must be scalable in order to be accurate. For example, if it took you two hours to review and correct a 100 page technical document, and the next month you have a 400 page document to review and correct, then you would estimate that it will take eight hours to complete.
- (2) See table IRM 2.5.10.5 below for estimation techniques:

Example - Parametric Estimating

Level of Effort - Reviewing First Month	Level of Effort - Reviewing Second Month
2 hours per 100 page = 2 man-hours	2 hours per 100 page x 4 = 8 man-hours

- Scaling is used if some of the Level of Effort (LOE) is outside of the expected work i.e., one hour was used evaluating which documents were included for review/correction process, then the actual LOE would be seven hours instead of eight hours because evaluating the documents was not part of the scope.

2.5.10.5.1
(12-17-2021)**IRS Services - Statistical
Analysis**

- (1) The EPO delivers ELC resource estimates of effort, cost and schedule for the following:
- Estimates for major Development, Modernization and Enhancements (DME) proposals, pre-selected budget cycle proposals, and in flight projects.
 - Estimation Guidelines pertaining to actual results/historical data on completed projects that can be used for analogy-based estimations.
 - Economic Analysis Workbook - Used to support the development of a Very Rough Order of Magnitude (VROM) cost for pre-select proposals, and calculates Economic Analysis factors.

#

2.5.10.5.2
(12-17-2021)

**Enterprise Life Cycle
(ELC) Accepted Methods
- Estimation Techniques**

- (1) According to the ELC Office, to determine the percentage of system change compared to the existing system, use standard estimation practices that are used for all IRS investments. The four accepted ELC methods for creating software project estimates are:
 - a. **SEER for Software** - Uses program attributes Source Lines of Code (SLOC), function points or proxy) as the basis for estimating project effort, cost and schedule.
 - b. **AD Estimation Workbook** - Estimates are created by defining the low-level tasks needed to the finish the project.
 - c. **Simplified Estimation Workbook** - Mainly used to plan development and deployment activities (MS4b and MS5), and (MS4a) for detailed design activities.
 - d. **Legacy Systems** - If you do not have an estimate on the system in production, do the following:
 - Estimate the number of requirements for the original system.
 - Compare the original system's estimate to the number of requirements that will be affected by the changes being made, see IRM 2.5.10.3

2.5.10.5.3
(12-17-2021)

**Software Sizing
Standards**

- (1) Software sizing is used to access the size of a software application or element to support cost estimation, progress tracking, and other software project activities. This standard includes::
 - a. **Size of the Code Base:** Measuring for development productivity.
- (2) For more information see CISQ standards, <http://www.it-cisq.org/standards>.
- (3) The Consortium for Information and Software Quality (CISQ) is an IT leadership group that established two standards for automating measurement of software size from the source code as seen in sections IRM 2.5.10.5.3.1 and IRM 2.5.10.5.3.2.

2.5.10.5.3.1
(12-17-2021)

**Automated Function
Points (AFP) Standard**

- (1) As of October 2016, the Automated Function Points (AFP) specification became an official standard of the Object Management Group, and is used for automating functional sizing of transaction-oriented software applications.
- (2) The advantage of Automated Function Points measure encompasses speed, cost-effectiveness, repeatable and consistent counts that eliminate the $\pm 10\%$ variance among manual function point counters. AFP is also used to measure the size of a software product.
- (3) Use function points for the following actions:
 - a. Perform analysis of software quality and productivity.
 - b. Determine expenses or resources for software development, enhancement, and maintenance.
 - c. Align estimation methods against the results of past estimates.
 - d. Normalize data used in software comparisons.
 - e. Estimate the size of a procured application package, e.g., Commercial Off The Shelf (COTS), or customized system) by sizing all the functionality included in the package.
 - f. Estimate the Return On Investment (ROI) of an application by sizing the functionality that matches the organization's requirements.
- (4) For more information view, <https://www.it-cisq.org/>.

2.5.10.5.3.2
(12-17-2021)
**Automated
Enhancement Points
Standard (AEP)**

- (1) During October 2016, the Automated Enhancement Points (AEP) specification also became an official standard of the Object Management Group. Automated Enhancement Points improves the measurement of software size for use in productivity analysis by measuring both the functional and non-functional size of software. This is an advancement in automated software sizing that solves problems that functional size measures have experienced in analyzing productivity during maintenance and enhancement activities. See IRM 2.5.10.5.3.1 on how to use function points for actions.

2.5.10.5.3.3
(12-17-2021)
**Measuring for Code
Quality**

- (1) Software code structural flaws involve interactions among multiple components and application layers, and are undetectable by traditional testing. As an example see IRM 2.5.10.5.3.3

Structural Quality Flaw Example

Structural Flaw (Total Defects)	Structural Flaw (Total Repair Time)
% of Total Application Defects	% of Total Repair Effort
Code-level violations (92%) + Architecturally Complex Defects (8%)	Code-level violations (48%) + Architecturally Complex Defects (52%)
Contribute to 90% of System Down Time	

- (2) The steps to initiate measures and anti-patterns to be used for assessing multi-tier IT application software are:
- Establish a calculable software quality standard for IT applications with scoring guidelines.
 - Use measurable thresholds to be accessed against acceptable levels of quality and other attributes of business application software.
 - Create baselines for benchmarking: application quality, productivity, cost, and other attributes across application domains and industry segments
 - Conduct a case study research and validate application metrics, and any associated business value.
- (3) **Software Structural Quality and Analysis** - Consist of the following:
- Security** - Measures 22 violations in source code representing the most exploited security weaknesses in software related to: CWE, Sans, and OWASP Top 10.
 - Reliability** - Measures 29 violations in source code impacting the availability, fault tolerance, and recoverability of software.
 - Performance Efficiency** - Measures 15 violations in source code impacting response time and utilization of processor, memory, and other resources
 - Maintainability** - Measures 20 violations in source code impacting the comprehensibility, changeability, testability, and scalability of software. See table IRM 2.5.10.5.3.3

Example of Quality Measures

Consortium for Information and Software Quality (CISQ) Quality Characteristic Measures		Example Architectural and Coding Violations Composing the CISQ Measures
Security	22 violations (Top 25 CWEs)	<ul style="list-style-type: none"> • SQL injection • Cross-site scripting • Buffer overflow
Reliability	29 violations (Top 25 CWEs)	<ul style="list-style-type: none"> • Empty exception block • Unreleased resources • Circular dependency
Performance Efficiency	15 violations (Top 25 CWEs)	<ul style="list-style-type: none"> • Expensive loop operation • Un-indexed data access • Unreleased memory
Maintainability	20 violations (Top 25 CWEs)	<ul style="list-style-type: none"> • Excessive coupling • Dead code • Hard-coded literals

Note: CISQ complies to ISO 25010:2011 quality characteristic definitions.

2.5.10.5.3.4
(12-17-2021)
Automated Technical Debt

- (1) The Automated Technical Debt is to measure of corrective maintenance effort pertaining to violations (weaknesses) remaining in a software application, which encompassed a business risk consisting of the following characteristics:
 - a. Opportunity Cost - Benefits that could have been obtained if resources had been put on a new capability rather than technical debt.
 - b. Liability - Business cost related to outages, breaches, corrupted data, etc.
- (2) **Technical Debt:** Critical violations of good coding and architectural practice within the code i.e., “software coding rework and related corrective costs.” This also included the future cost of defects remaining in the code at release such as:
 - a. Interest on the debt - Continuing IT cost attributable to the violations that caused the technical debt.
 - b. Principal borrowed - Cost of fixing problems remaining in the code after release that must be remediated. .
- (3) See example Technical Debt model in table below:

Technical Debt Example

Technical Debt Example	
Business Risk	Technical Debt
Opportunity Cost	← ← ← ← Interest on the debt
Liability from debt	← ← ← ← Principal borrowed
	↑

Technical Debt Example	
Business Risk	Technical Debt
	Structural quality problems in production code

Figure 2.5.10-1

Exhibit 2.5.10-1 (12-17-2021)**Acronyms/Terms**

Acronym	Definition
AD	Application Development
AEP	Automated Enhancement Points
AFP	Automated Function Point
CISQ	Consortium for IT Software Quality
COTS	Commercial Off The Shelf
CRUD	Create, Read, Update, and Delete
DET	Data Element Type
EIF	External Interface Files
ELC	Enterprise Life Cycle
DMD	Data Management Division
EI	External Input
EO	External Output
EP	Elementary Process
EPMO	Estimation Program Management Office
FET	File Element Type
FPA	Functional Point Analysis
FTR	File Type Reference
IFPUG	International Function Point Users Group
FSM	Functional Size Measurement
IEC	International Electrotechnical Committee
IFP	Internal Function Point
ILF	Internal Logical File
IMF	Individual Master File
ISBSG	International Software Benchmarking Standards Group
ISO	International Standards Organization
ITM	Integrated Talent Management
KDA	Knowledge Development and Applications
LCS	Logical Collaborative Segments
OMG	Object Management Group
PET	Progressive Element Type

Exhibit 2.5.10-1 (Cont. 1) (12-17-2021)**Acronyms/Terms**

Acronym	Definition
RET	Record Element Type
ROI	Return On Investment
SBS	Strategy and Business Solutions

Exhibit 2.5.10-2 (12-17-2021)**Terms/Definitions**

Term	Definition
Automated Function Point	Specifies the definitions, rules and steps for applying the IFPUG
Application	Software program (or programs) that carries out a task or sequence of tasks to support a specific, well defined business process. An application is typically made up of several programs and software components (Framework, RDBMS) and includes different languages and technologies. Customized software packages and software products may be also considered as applications.
Application Boundary	A boundary used to define what is internal to the application and what component or interfaces are external to the application.
Application Model	The Application Model is produced by analyzing the source code of the application to be sized. It shall contain the static elements of the application that are used in the Automated Function Point counting process. The Application Model is defined as the list of all application's code entities and their dependencies.
Benchmark Testing	Measures a repeatable set of quantifiable results that serves as a point of reference against which products/services can be compared. The purpose of the benchmark testing is to compare the present and future software releases with their relevant benchmarks.
International Software Benchmarking Standards Group	An international data repository of software projects, submitted by leading global IT and metrics companies to assist software and IT customers with project estimation, risk analysis, productivity and benchmarking needs.
Object Management Group Special Interest Group	Develop standard, automatable measures for evaluating software size and quality with an ecosystem supporting policy and deployment.
Progressive Function Point Analysis	Uses the same FPA principles, but targets improvement by estimating the sized of project by considering the input and output
Source Code For Count List	A list of source code modules with programming language, scripts, data definition and manipulation languages for each module to be included in an AFP Count of one Application.
Return On Investment	A key financial ratio that measures the gain/loss from an investment in relation to the initial investment.
Quality Adjusted Productivity	A class of production function that connects the inputs and outputs accounted in effective measured quantities.

