

The IFPUG Counting Practices On-Going Effort in Sizing Functional Requirements

Janet Russac

2009

IFPUG's method for function point analysis is an ISO standard and must be conformant to ISO/IEC 14143-1:2007. Functional Size is a size of the software derived by quantifying the Functional User Requirements, which are a subset of the User Requirements. These are requirements that describe what the software shall do, in terms of tasks and services. IFPUG's Counting Practices Committee (CPC) continually works to adapt the Counting Practices Manual (CPM) to conform to the ISO standards. This article looks at CPM 4.3, scheduled for release in the last quarter of 2009, and the changes made to further evolve the methodology of sizing functional requirements.

History of Function Points

In the 1970s Allan Albrecht was the first to release publicly a method for functionally sizing software called function point analysis. The use of function points, as a measure of the functional size of software, has grown since that time from a few interested organizations to an impressive list of organizations worldwide. In 1986 the International Function Point Users Group (IFPUG) was formed and since then has continuously enhanced the original Albrecht method for functionally sizing software. The IFPUG functional size measurement method is known as function point analysis, and its units of functional size are called Function Points.

Function Point Analysis

Function point analysis measures software by quantifying the tasks and services (i.e., functionality) that the software provides to the user based primarily on logical design.

The objectives of function point analysis are to measure:

- functionality implemented in software, that the user requests and receives;
- functionality impacted by software development, enhancement and maintenance independently of technology used for implementation.

The process of function point analysis is:

- simple enough to minimize the overhead of the measurement process;
- a consistent measure among various projects and organizations.

Organizations can apply this International Standard to measure the size of a software product to:

- support quality and productivity analysis;
- estimate cost and resources required for software development, enhancement and maintenance;
- provide a normalization factor for software comparison;

- determine the size of a purchased application package by functionally sizing all the functions included in the package;
- assist users in determining the benefit of an application package to their organization by functionally sizing functions that specifically match their requirements.

Counting Practices Manual (CPM)

CPM 2.0 was released in 1987, and since then there have been several iterations. CPM 4.3 is an International Standard and is the latest release in the continually improving IFPUG method that promotes the consistent interpretation of functional size measurement in conformance with ISO/IEC 14143-1:2007. This International Standard specifies the set of definitions, rules and steps for applying the IFPUG functional size measurement (FSM) method. CPM 4.3 includes the Functional Size Measurement (FSM) document and the Implementation Guide. Together these make up CPM 4.3, a publication that is 100% ISO compliant. CPM 4.3, scheduled for publication in the last quarter of 2009, will become effective January 2010. The changes from 4.2 to 4.3 are enumerated later in this article.

The primary objectives of the IFPUG Counting Practices Manual 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

ISO/IEC 14143-1 - Definition of User Requirements

In 1998, the first ISO/IEC Functional Size Measurement standard was published (ISO/IEC 14143-1:1998). This standard defines the Functional Size as “a size of the software derived by quantifying the Functional User Requirements.” In 2007, it was updated and published as ISO/IEC 14143-1:2007.

ISO/IEC 14143-1 defines the fundamental concepts of Functional Size Measurement (FSM) and describes the general principles for applying an FSM Method. It does NOT provide detailed rules on how to:

- Select a particular method
- Measure Functional Size of software using a particular method
- Use the results obtained from a particular method

The definition of FSM in ISO/IEC 14143-1 is applicable when determining if a method for sizing software is a Functional Size Measurement Method. It does not prevent the development of various methods, but rather provides a basis for assessing whether a particular method conforms to FSM.

ISO/IEC 14143-1 distinguishes between two subsets of user requirements:

- Functional User Requirements
- Non-Functional User Requirements

The ISO/IEC 14143-1 definitions are as follows:

Functional Size

- A size of the software derived by quantifying the Functional User Requirements

Functional User Requirement

- A subset of the User Requirements. Requirements that describe what the software shall do, in terms of tasks and services.
- Functional user requirements include but are not limited to:
 - Data transfer (for example: input customer data, send control signal)
 - Data transformation (for example: calculate bank interest, derive average temperature)
 - Data storage (for example: store customer order, record ambient temperature over time)
 - Data retrieval (for example: list current employees, retrieve aircraft position)

Non-Functional User Requirements

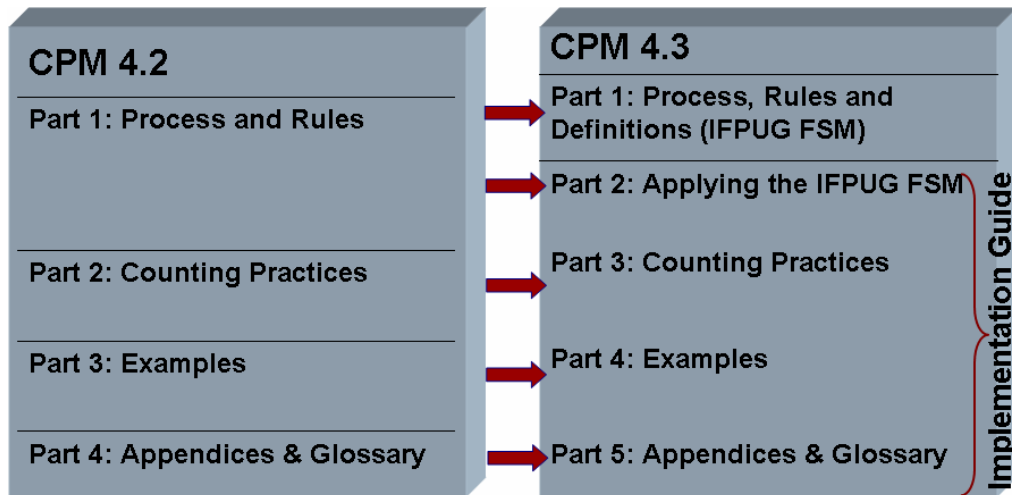
- ISO does not provide a definition for Non-Functional User Requirements, but gives some examples in a note.
- Examples of User Requirements that are Non-Functional User Requirements include but are not limited to:
 - Quality constraints (for example usability, reliability, efficiency and portability)
 - Organizational constraints (for example locations for operation, target hardware and compliance to standards)
 - Environmental constraints (for example interoperability, security, privacy and safety)
 - Implementation constraints (for example development language, delivery schedule)

Major Structural Change Areas in CPM 4.3

The major structural change areas in CPM 4.3 are:

- Replace existing Part 1 with the new ISO Standard (ISO/IEC 20926:2009)
- Created The Bridge - Applying the IFPUG Functional Size Measurement Method (now Part 2) which provides guidance in applying the process and rules as defined in the ISO Standard (now Part 1)
- Amend the remaining parts to be consistent with the revised Part 1
 - Counting Practices (Part 3)
 - Examples (Part 4)
 - Appendices and Glossary (Part 5)

The schematic below outlines the structural changes between 4.2 and 4.3.



OVERVIEW OF CHANGES IN CPM 4.3

Part 1 (new): The ISO Standard (IFPUG FSM)

- Is the ISO Standard “IFPUG FSM” in its entirety
- Is the former Part 1 Process and Rules of the CPM
- Condensed to 21 pages and using strict ISO wording and format
- Uses ISO Template
 - Different look and feel
 - Wording, format and sequence of actions changed
 - No repetition
- Minor modifications
 - Clarifications and simplifications
 - Added definitions for consistent state, self-contained, sorting and arranging
 - Common set of DET and FTR rules for all transactions
 - Uniqueness Test (i.e., same DETs, FTRs & Processing Logic) - removed from EI/EO/EQ rules & stated once
- Does not include GSCs or VAF

Part 2 (former part 1): The Bridge - Applying the IFPUG FSM Method

- Retained remaining chapters from Process and Rules
- Contains additional guidance to easily apply the FSM rules
- Moved GSCs and VAF to Part 5, Appendix C
- Changed wording and sequence of actions consistent with FSM
- Moved all discussion of Enhancement Projects to Enhancement Chapter

Part 3 (former part 2): Counting Practices

- Changed wording and sequence of actions consistent with FSM

- Provided additional guidance and examples for enhancements
- Added 5th Chapter for counting Conversion Activity

Part 4 (former part 3): Examples

- Changed wording and rule boxes consistent with FSM
- Added clarifications and additional examples

Part 5 (former part 4): Appendices and Glossary

- Now contain optional GSCs and VAF

CPM 4.3: What It Looks Like

Part 1: IFPUG FSM provides the function point analysis process for functionally sizing software following the IFPUG Method as well as the detailed rules for identifying and measuring data functions and transactional functions.

Part 2: Process and Rules provides an overview of the IFPUG Method, along with guidance in applying the rules for determining the type of count, establishing application boundaries and measuring data and transactional functions.

- Chapter 1: Introduction
 - The title of this chapter has changed to “The Bridge - Applying the IFPUG Functional Size Measurement Method”.
- Chapter 2: Overview of Function Point Analysis
 - Changed to “IFPUG FSM Method Overview”
 - Made extensive wording changes for consistency with the FSM
 - Added first step in procedural diagram: “Gather Documentation”
 - Replaced term “unadjusted function points” with “functional size”
 - Moved discussion of “unadjusted” or “adjusted” to Appendix C
- Chapter 3: User View
 - Changed to “Gather Available Documentation”
- Chapter 4: Determine Type of Count
 - Revised wording for definitions:
 - development project function point count,
 - enhancement project function point count and
 - application function point count
- Chapter 5: Identify Counting Scope and Application Boundary
 - Changed to “Determine Counting Scope and Boundary and Identify Functional User Requirements”
 - Made minor wording changes
- Chapter 6: Count Data Functions
 - Changed to “Measure Data Functions”
 - Repeated rules here for convenience
- Chapter 7: Count Transactional Functions
 - Changed to “Measure Transactional Functions”

- Repeated rules here for convenience
- Chapter 8: Determine Value Adjustment Factor
 - Moved entire chapter to Appendix C
- Chapter 9: Calculate Adjusted Function Point Count
 - Moved formulas to Appendix C
 - Moved Enhancement Example to Enhancement Chapter in Part 3

Part 3: Counting Practices provides detailed counting practices and enhanced examples to assist the practitioner in applying the rules to measure data functions and to size enhancement projects.

- Chapter 1: Code Data
 - Added ISO definition of Functional Size
 - Updated the definition of Functional User Requirements
 - Replaced the terms Quality Requirements and Technical Requirements by the ISO term Non-Functional Requirements
- Chapter 2: Logical Files
 - Changed the sequence and structure of steps consistent with the FSM
 - Moved evaluation of code data to prevent premature elimination of items that may look like code data, but are in fact not
- Chapter 3: Shared Data
 - Changed references to other parts and chapters of CPM
- Chapter 4: Enhancement Projects
 - Incorporated section from Part 1, Chapter 9 on Enhancement Project Count
 - Updated terminology to be consistent with FSM
 - Provided additional guidance and examples for forms of processing logic
 - In “Enhancement vs. Maintenance” section prefaced any reference to GSCs with “optional”
- Chapter 5 (New): Data Conversion Activity

Part 4: Examples provides examples of measuring data functions and transactional functions to illustrate the rules from Part 1.

- Chapter 1: Data Function Counting Examples
- Chapter 2: Transactional Function Examples

Part 5: Appendices contains the appendices and glossary.

- Appendix A: Calculation Tables
 - Removed term “Adjusted”
- Appendix B: Change from Previous Version
 - Reflects all changes in the document from 4.2.1 to 4.3
- Appendix C: Adjusted Functional Size
 - Contains guidance on applying the General Systems Characteristics and Value Adjustment Factor to determine Adjusted Functional Size
 - Contains all formulas

Impact Study

44 CFPS volunteers performed an impact study. They counted a case study using both CPM 4.2.1 and CPM 4.3, and then counted over 100 projects previously counted under CPM 4.2.1 using CPM 4.3. These projects were a mix of development, application and enhancement. Results were the same for both methods. Therefore, the conversion factor was determined to be 1.0 (i.e., no difference).

The Next CPM – Summary

The IFPUG FSM (CPM 4.3 – Part 1) meets the requirements of the ISO FSM Standard. It looks and reads significantly different from Part 1 of the CPM 4.2.1 while conveying the same rules and counting process in a much more succinct and unambiguous manner. The Implementation Guide (CPM 4.3 – Parts 2 thru 5) provides a bridge between the FSM and the familiarity of the “old” CPM; together, they will comprise CPM 4.3.

About the Author:

Janet Russac has over 25 years of experience as a programmer, analyst and measurement specialist in software application development and maintenance. In 2008 she started her own company, Software Measurement Expertise, Inc. (SME). She has worked for The David Consulting Group, Software Productivity Research, IBM Global Services and Prudential Insurance as a lead function point analyst, software measurement specialist and function point instructor. She has implemented software development measurement programs and used various software development metrics, including function points, to recommend business decisions and identify best practices and process improvements in client organizations.