



# Simplifying Simple Function Points (SFP)

Dan French, CFPS, PMP, CSM  
Domain Expert  
571-225-0380  
[dfrench@cobec.com](mailto:dfrench@cobec.com)

Carol Dekkers, CFPS (Fellow), PMP, CSM  
President, Quality Plus Technologies, Inc  
813-816-1329  
[Dekkers@qualiytplustech.com](mailto:Dekkers@qualiytplustech.com)

**Quality Plus Technologies, Inc.**

# Topics

- Introduction to metrics and software sizing
- History of functional size
- Two flavors of IFPUG Function Points
  - IFPUG FP
  - Simple FP
- When to use IFPUG vs Simple FP
- ConOps Example
- Do's and Don'ts with IFPUG FP and SFP
- Conclusions

# A Little About Us... Carol

---

B.Sc. in Mechanical Engineering from University of Calgary

---

25 years in software measurement and FPA

---

IFPUG CFPS (Fellow) → Over 20 years certified

---

Author of 75 articles, co-authored 10+ textbooks (ASQ, PMI, ISBSG, IFPUG, CrossTalk and others)

---

Member of U.S. delegation to ISO/IEC JTC1 SC7 since 1994

---

Co-developed “northern Scope” certification (Finland) → € / FP

---

IFPUG Past president, now Dir of Communications and Marketing

---

Presented / instructed > 30 countries

---

Project Management Institute (PMI) Project Management Professional (PMP)

---

Agile Alliance Certified SCRUM Master (CSM)

# A Little About Us... Dan

- B.S. in Economics from Virginia Tech
- Graduate of the Chubb Institute Top Gun Program
- Over 20 years experience in software cost estimation
- Counting function points for 24 years and been a Certified Function Point Specialist (CFPS) for 22 years (IFPUG Fellow)
- Experience in a number of estimation techniques and tools including SEER-SEM, COCOMO, SLiM, Delphi, and Estimating by Analogy
- Certification Director for the International Function Point Users Group (IFPUG) Board of Directors
- Former Chairman of the IFPUG Functional Software Sizing Committee (FSSC)
- GAO Agile and Cost guides expert team member
- Project Management Institute (PMI) Project Management Professional (PMP)
- Agile Alliance Certified SCRUM Master (CSM)

# What is a Metric ?



“A metric is just **an indicator**. Just as a high temperature reading on your dashboard indicates a problem with your engine or cooling system, a metric only lets you know that something might be wrong — forcing that *number* to change doesn't necessarily fix anything.”

# Software Size “Metrics”

## PHYSICAL SIZE

## RELATIVE EFFORT SIZE (Agile development)

## FUNCTIONAL SIZE

SLOC/ESLOC	Story Points	Function Points/Simple FP
<ul style="list-style-type: none"> <li>Objective Size Measurement</li> <li>Good for ROM analogy estimate</li> <li>Easy to collect</li> <li>Highly dependent on coding language and skill of programmer</li> </ul>	<ul style="list-style-type: none"> <li>Subjective Size Measure</li> <li>Relative measure                             <ul style="list-style-type: none"> <li>Determined by individual Agile Teams</li> <li>Cannot be compared across programs</li> </ul> </li> <li>Team level view</li> <li>Cannot be independently estimated / analyzed</li> </ul>	<ul style="list-style-type: none"> <li>Objective Size Measure</li> <li>Standard unit of measure                             <ul style="list-style-type: none"> <li>ISO Standard</li> <li>Comparable across programs</li> </ul> </li> <li>Long term view at the Program level</li> <li>Can be independently estimated / analyzed</li> </ul>

SLOC/ESOC: Software Lines of Code/Effective Software Lines of Code; ROM: Rough Order of Magnitude ISO: International Standards Organization

# What is Functional Size?

- Functional size is a size of software based on quantifying its functionality or business processes. Functional size is a quantified answer to “What does the software do?”
- It is not the same as physical size (Source Lines of Code – SLOC)
- Defined by the International Function Point Users Group (IFPUG) is as the “measure of the functionality that an application provides to the user...” (IFPUG, 2010).
- Software performs automated function as described by Functional User Requirements (FUR). These include:
  - What data to store
  - Which reports to run
  - What data to display
  - What data to send to other systems, et al.
- Functional size is based on an assessment of the FUR, and expressed in units of “function points”

# History of Function Points

- Mid-1970's: Function Points (FP) developed at IBM as an alternative to Source Lines of Code (SLOC) in response to new, more efficient software languages
- 1984: the International Function Point Users Group (IFPUG) founded & formalizes the methodology
- 1986: IFPUG releases IFPUG FP Counting Practices Manual v1.0
- 1998: IFPUG Function Points become the first International Standards Organization (ISO) Functional Size Measurement Method (ISO/IEC 20926) and reaffirmed to conform with IFPUG FP v4.3.1
- IFPUG FP method assigns FP to 5 function types



# History of Simple Function Points

- 2009: Dr. Roberto Meli of DPO introduces the Early & Quick Function Points (E&Q FP) based on the IFPUG method. New concepts:
  - Generic Functions
  - Typical Process (TP) (CRUD)
  - Generic Process (GP)
  - Macro Process (MP)
- 2010: Meli refined E&Q FP into Simple Function Points (SiFP) with 2 generic function types:
  - Elementary Process (EP)
  - Logical File (LF)
- 2019: IFPUG acquired the SiFP method
- 2021: IFPUG releases IFPUG Simple Function Point (SFP) manual v.2.1

# Key Terms (1 of 3)

- **Functional User Requirements (FUR)** - A sub-set of the user requirements; requirements that describe what the software shall do, in terms of tasks and services. **FUR** are those requirements that describe what the software will do: for example, what data to store, what reports to produce, which data to display, what information to send to other systems, to name a few.
- **User** - A user is any person or thing that communicates or interacts with the software at any time. A user could be a physical person, other software or hardware, or anything that sends or receives data that crosses the software's application boundary.
- **Elementary process (EP)** - "An Elementary Process is the smallest unit of activity, which is meaningful to the user, that constitutes a complete transaction, it is self-contained and leaves the business of the application being measured in a consistent state".

# Key Terms (2 of 3)

- **Logical file (LF)** A Logical File represents functionality provided to the user to meet internal and external data storage requirements. It is a user recognizable group of logically related data or control information maintained and/or referred within the boundary of the application being measured.” The term file here does not mean physical file or table. In this case, file refers to a logically related group of data and not the physical implementation of those groups of data.
- An **internal logical file (ILF)** is a user recognizable group of logically related data or control information maintained within the boundary of the application being measured. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being measured.
- An **external interface file (EIF)** is a user recognizable group of logically related data or control information, which is referenced by the application being measured, but which is 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 measured. This means an EIF counted for an application must be in an ILF in another application.

# Key Terms (3 of 3)

- An **external input (EI)** is an elementary process that processes Data or control information sent from outside the boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.
- An **external output (EO)** is an elementary process that sends data or control information outside the application's boundary and includes additional processing beyond that of an external inquiry. 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, create derived data, maintain one or more ILFs, and/or alter the behavior of the system.
- An **external inquiry (EQ)** is an elementary process that sends data or control information outside the 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.
- **Data Element Type (DET)** - A unique, user recognizable, non-repeated attribute.
- **File Type Referenced (FTR)** - A data function read and/or maintained by a transactional function.
- **Record Element Type (RET)** - A user recognizable sub-group of data element types within a data function

# IFPUG FP vs Simple FP (1 of 2)

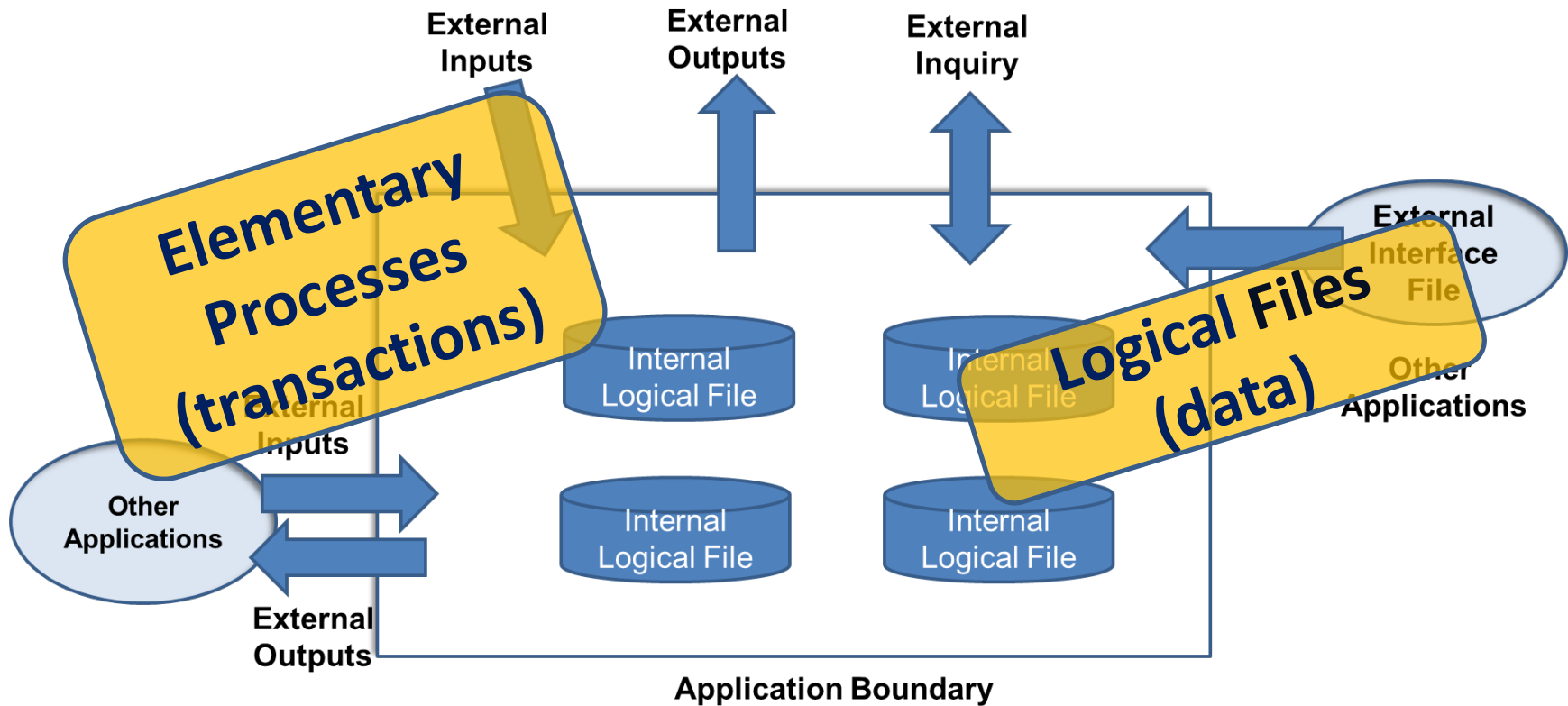
- International Function Point Users Group (IFPUG) SFP v2.1 (2021)
- Originally developed by Dr. Roberto Meli/Italian researchers v1.1 (2010))
- Simplifies functional sizing into two types of functions:
  - Generic elementary processes (transactional functions)
  - Generic logical files (data groups)

IFPUG Components	Low	Average	High	Simple Function Point SFP components	Weighting Factor
External Inputs	3	4	6	Elementary Processes EP (Transactional Functions)	4.6 SFP
External Outputs	4	5	7		
External Inquiries	3	4	6		
Internal Logical Files	7	10	15	Logical Files LF (Data Functions)	7 SFP
External Interface Files	5	7	10		

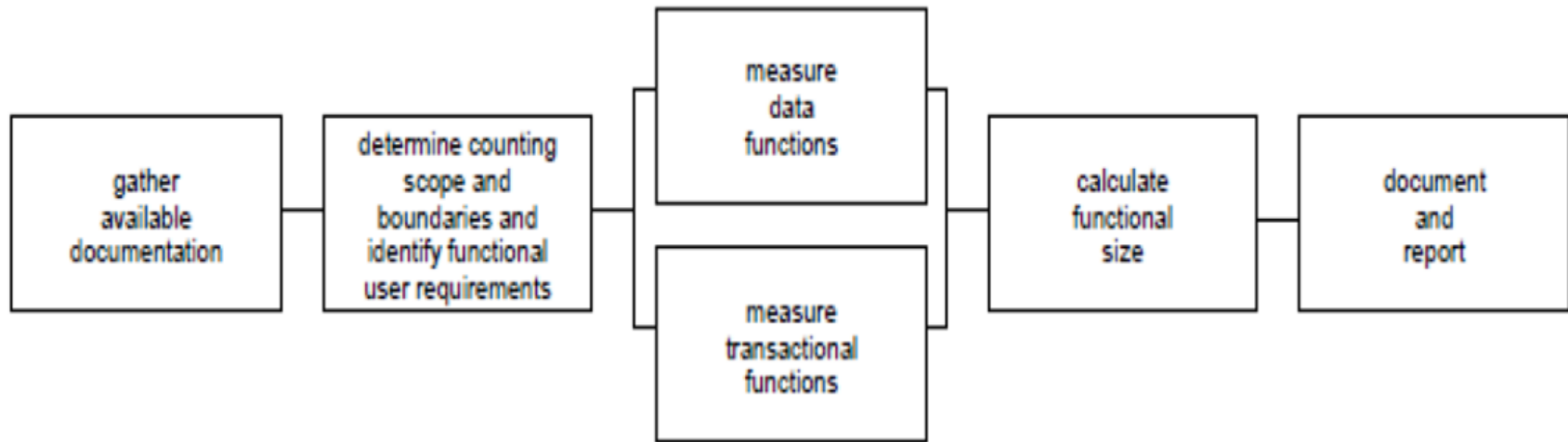
# IFPUG FP vs Simple FP (2 of 2)

## 1. IFPUG FP v4.3.1

## 2. IFPUG Simple FP v2.1



# IFPUG FP & SFP Measurement Process



# When to use IFPUG FP vs SFP

- Use SFP when:
  - Need for quick ROM sizing
  - Documentation low level of detail or not available
  - Analyst (s) is/are new to FP or not well trained in FP counting
  - Early in software development lifecycle or proposal phase
  - Budgeting
- Use FP when:
  - Need detailed, high fidelity and accurate sizing for estimating project costs and schedule
  - Documentation has high level of detail
  - Analyst (s) is/are experienced with counting FP, preferably CFPS or CFPP



# Can IFPUG FP or IFPUG SFP be used to Size Agile S/W Development? (1 of 2)

- Short Answer:



# Can IFPUG FP or IFPUG SFP be used to Size Agile S/W Development? (2 of 2)

- More detailed answer:
  - **Misconception** that function points cannot be used in Agile development, either SFP or FP
  - **FP are agnostic** to platform, language, developer skill and development methodology
  - **Can be used to size product backlog, use cases/epics/features, calculate velocity and sprint planning**
  - More reliable and verifiable than using subjective Agile sizing such as story points or T-shirt sizing
  - Can provide more reliable metrics as well such as productivity, FP/month and cost/FP

# ConOps Example (1 of 2)

## **ConOps contained functional requirements for an online book ordering system:**

- a. Create, read, update, delete (CRUD), and store customer records.
- b. Sys admin: CRUD, and store book catalog entries.
- c. Browse catalog by author or title.
- d. Select and display book details.
- e. Online order created by placing books in a shopping cart and saving.
- f. Display shopping cart summary showing books and order totals.
- g. Complete online order with a credit card.
- h. Generate an order summary receipt (calculations).
- i. Generate an order request to be fulfilled (calculations).



# ConOps Example (2 of 2)

Functional Requirement	IFPUG avg functions	IFPUG FP	IFPUG SFP functions	IFPUG SFP
a. CRUD & store customers	3A EI, A EQ, 1A ILF	26 FP	4 EP, 1 LF	25.4 SFP
b. CRUD & store catalog	3A EI, A EQ, 1A ILF	26 FP	4 EP, 1 LF	25.4 SFP
c. Browse books	1A EQ	4 FP	1 EP	4.6 SFP
d. Display book details	1A EQ	4 FP	1 EP	4.6 SFP
e. Create and save order	1A EI, 1A ILF	14 FP	1 EP, 1 LF	11.6 SFP
f. Shopping cart display	1A EO	5 FP	1 EP	4.6 SFP
g. Complete order with credit card	1A EI	4 FP	1 EP	4.6 SFP
h. Generate order summary	1A EO	5 FP	1 EP	4.6 SFP
i. Generate order fulfillment request	1A EO	5 FP	1 EP	4.6 SFP
<b>TOTAL</b>	<b>8A EI, 3A EO, 4A EQ, 3A ILF</b>	<b>93 FP</b>	<b>15 EP, 3 LF</b>	<b>90 SFP</b>

# IFPUG FP and SFP Usage Do's and Don'ts

## Do's:

- **Use properly trained analysts**, even if it requires hiring an outside analyst
- **Properly document** the function point count and all source documentation
- **Use IFPUG function points** if a high degree of accuracy in sizing is required for estimating or legal reasons, and there is sufficiently detailed requirements to support it
- **Use IFPUG SFP** when it is necessary to develop a quick sizing estimate with little documentation available

## Don'ts:

- Use SFP just because it is easier/quicker, ensure that it will meet other business needs for the software size
- **Use SFP** if using a parametric estimating tool to develop cost and schedule estimates as none currently on the market support native SFP sizing
- **Don't use traditional IFPUG function point sizing** when there is limited time or lack of resources to properly conduct the count
- **If sizing a waterfall method project** and the early phase sizing estimates are done using SFP it is recommended to transition to traditional IFPUG function points sizing when there is the available documentation to support, it.
- **Depending on the business need, it is not recommended to use SFP** for application counts, because all of the prerequisite details to do a formal IFPUG FP count should be available and known.

# Conclusions

- The IFPUG function point methodology is a tried-and-true software sizing method that is an ISO/IEC Functional Size Measurement standard
- IFPUG SFP Can be used even if requirements are high-level
- SFP provide a faster, simpler way to size FURs with reasonable accuracy when compared to IFPUG FP – especially on high-level ConOps or EPICS or user stories
- SFP are easier to learn and should be used when analysts are new to function point analysis or are not properly trained/certified
- IFPUG FP provide more accurate and defensible software size estimates and should be used when proper documentation is available, and analysts are appropriately trained

# Resources

- International Cost Estimating and Analysis Association (ICEAA) <https://www.iceaaonline.com/> - SCEBoK Lesson X includes a full sizing case study (multiple functional sizing methods)
- International Function Point User Group (IFPUG) <http://ifpug.org/>
- International Software Benchmarking Standards Group (ISBSG) <http://www.isbsg.org>
- Software Engineering Institute (SEI) [www.sei.cmu.edu](http://www.sei.cmu.edu)
- Systems and Software Consortium <http://www.software.org/>



ICEAA



The global and independent source  
of data and analysis for the IT industry



Software Engineering Institute

Carnegie Mellon



**THANK YOU**

# Simplifying Simple Function Points (SFP)

Dan French, CFPS, PMP, CSM  
Domain Expert  
571-225-0380  
[dfrench@cobec.com](mailto:dfrench@cobec.com)

Carol Dekkers, CFPS (Fellow), PMP, CSM  
President, Quality Plus Technologies, Inc  
813-816-1329  
[Dekkers@qualiytplustech.com](mailto:Dekkers@qualiytplustech.com)

**Quality Plus Technologies, Inc.**



# Backup: IFPUG FP vs IFPUG SFP (1 of 2)

Concept	IFPUG FP	IFPUG SFP
IFPUG standardized glossary	Yes	Yes, same
Intent to measure functional size based on FUR	Yes	Yes, same
Method owned by IFPUG	Yes	Yes
IFPUG FP measurement steps: 1. Gather available documentation 2. Purpose/scope/boundary, identify FUR 3a. Measure data functions 3b. Measure transactional functions 4. Calculate functional size 5. Document and report	Yes, but steps 3a and 3b involve additional sub-steps: subclassification into 3 types of transactional functions and 2 types of data functions, and a complexity classification (into Low, Average, or High) to get FP values	Yes
Base functional components (BFC): transactional functions and data functions	Yes: Transactional functions are subdivided into EI, EO, EQ, and Data functions are subdivided into ILF, EIF	Yes: Transactional functions are called “Elementary Processes” and Data Functions are called “Logical Files”

# Backup: IFPUG FP vs IFPUG SFP (2 of 2)

Number of different FP values allocated across function types	3 FP values allocated as Low, Average or High across 5 function types (total of 8 different values)	2 SFP values allocated, one each to two function types
Range of FP values by category	Transactional functions are worth between 3 and 7 FP depending on type and complexity. Logical files are worth 7 to 15 FP depending on type and relative complexity	All transactional functions are considered to be EP and assigned 4.6 SFP. All data functions are considered to be logical files and assigned 7 SFP
Unit of measure	Function Points (FP)	Simple Function Points (SFP)
Convertibility	1 FP = 1 SFP	1 SFP = 1 FP