

MetricVIEWS



IFPUG.org

A PUBLICATION OF THE INTERNATIONAL FUNCTION POINT USERS GROUP

RELEVANT SOFTWARE MEASUREMENT

DRIVES STRATEGIC VALUE

**IFPUG AD/M Benchmarking
Certification**

**Simple Function Points:
Your Questions Answered**

**The FP-Based
'Productivity Paradox'**



IFPUG
International **Function** Point
USERS GROUP

**NEW WEBSITE
COMING SOON**

Changing to Adapt to New Times

- Learning management system
- New look & feel
- Improved navigation
- Enhanced certification directory
- Archived webinars & virtual events
- And more!



Message from the President..... **4**

From the Editor's Desk..... **5**

Elvis Presley, the Big Mac®,
and Function Points **6**

Measuring the Value
of Software **8**

IFPUG AD/M Benchmarking
Certification **11**

Simple Function Points:
Your Questions Answered **13**

Functional and Non-Functional
Measurement and Reporting
with Scrum..... **17**

Advancements in Software
Development Productivity: the
FP-Based 'Productivity Paradox' **23**

COMMITTEE REPORTS

• Certification Committee..... **32**

• Functional Sizing Standards
Committee **32**

• Partnerships and Events
Committee **32**

• Communications and
Marketing Committee **33**

• ISO Standards
Committee **33**

• Non-functional Sizing
Standards Committee **34**

• International Membership
Committee **34**

• Business Applications
Committee..... **35**

METRICVIEWS EDITORIAL BOARD:

Carol Dekkers | Julián Gómez | Peter Thomas | Tom Cagley | Roberto Meli | Steve Woodward



As we begin a new year, I would like to express my gratitude to all of our volunteers who worked so diligently under the difficult circumstances of the worldwide pandemic. I hope that this letter finds all of our members and their families healthy and safe.

It is remarkable what our volunteers were able to accomplish in the past two years. We have a new website scheduled to deploy in early 2022. Our Task Forces have developed new programs and practices, including Benchmark Certification and Simple Function Points (SFP).

I would like to thank Immediate Past President Christine Green for her leadership during her term. Special thanks to Past President Mauricio Aguiar, who retired from the board after decades of service to IFPUG, as well as to Diana Baklizky also retired.

Welcome to our new board members Dr. Roberto Meli and Saurabh Saxena.

On behalf of the Board of Directors, I wish you all a Happy New Year, and we look forward to working with you, to constantly improve IFPUG and contribute to the software measurement community.

Charles Wesolowski

IFPUG President



I'm not much of a dreamer; maybe you are. That's good since we seem to need plenty of good ideas today. Systems thinking advocates suggest that the best way to quash bad ideas is to get lots of good ideas. You might find the following short story to be an anti-pattern of that thinking, or it may resonate in unexpected ways.

Somebody, maybe a committee or just a hoard of colleagues on a Zoom call, thought to offer me the role of the *MetricViews* Editor. At first, I reckoned, this might be a great way to establish another revenue stream for my grandchildren's college tuition; but no there was no money, nor crypto attached to the offer. Then I thought I could be a powerful force in the world of software measurements; but no, I actually would have no power. Next, I pondered the possibility of being a fame-fueled influencer so I could use my swag to get lots of free stuff. As you probably guessed, the role didn't come with any influencer incentives. Finally, I realized that I was doomed to be a volunteer, merely for the benefit of IFPUG and the betterment of our flagship publication *MetricViews*. I suppose that intrinsic reward is better than all of that other stuff anyway. Really? No, I haven't lost my mind; not any more than the multitude of other IFPUG members that volunteer their time, expertise, and passion to sustain the significance of software measurement and productivity.

This issue of *MetricViews* is possible thanks to the good ideas of authors, committee members, editorial board members, researchers, industry proponents, and pundits alike. CMA's efforts to organize this bucket of ideas to fruition are also noteworthy. Whether your interests are aroused by Simple Function Points, benchmarking certification, the value of software, productivity, paradoxes, or Scrum, this issue has some good ideas for you. And if that is not enough, there are even analogies to Elvis and Big Macs®. Thank all of you for your sacrifices and contributions to our expanding domain of measurement knowledge.

I said that I was not much of a dreamer. Loosely interpreted, Harvey McKay suggested that a dream became a goal when a date was attached to it. I prefer to have dates, deadlines so to speak. Probably like you, I also prefer to have some input as to what those dates are. *MetricViews* certainly had its share of dates for its development over the past few weeks. You, however, get to take your time consuming it. Enjoy.

Wait. I'm kind of thinking maybe Chief Editing Officer would be a better title for this role. Sounds like a dream.

Be well, stay well.

Joe Schofield

title under consideration . . .

ELVIS PRESLEY, THE BIG MAC[®], AND FUNCTION POINTS



By: Julián Gómez

Elvis is the King

When you look carefully into a domain you often find a king or a queen in that domain. For instance, if I ask you who the King of Rock is, there can be no doubt; it's Elvis Presley. Other names may have crossed your mind, but none of them is the King; Elvis still reigns today. Why is he still the King? For three simple reasons.

The first reason is that Elvis is the 4th highest-earning deceased millionaire in 2017. Impressive, isn't it? Well, more impressive is the second reason. Elvis is the musician who collects the most money for his songs every year, except for The Beatles. But these two reasons are nothing compared to the third, which is that some people think he is still alive. If that's not being the King, then what is?

As Elvis is the King of music, Product is the king that rules the world of software development; that is, the world of Apps. Why Product? For three simple reasons.

The burger is the king of measurement

The first reason is a burger. A burger is the main protagonist of an index published every year by the renowned newspaper *The Economist*; it's known as The Big Mac[®] Index. Using this Index, *The Economist* compares the currencies of all of the nations where a McDonald's is located, against the price of the Big Mac in the United States. It uses the cost of a Big Mac instead of the cost of a labor hour because it is a "product" that we buy, thus allowing us to scale the relative purchasing power in each of the participating countries.

In the same way that the burger is a unit of standardization better than other measures, the unit of quantity for software products (IFPUG function points) is the standard de facto for enabling us to manage our software-related assets.

The king of transformation

The second reason has to do with digital transformation. We are changing, in our companies, in your company, from a software development paradigm based on waterfall models to new agile paradigms: *Scrum*, *DevOps*, etc. Yet, many organizations choose to ignore the fabled philosopher's stone that allows us to track the ROI from our transformation investment to determine its success.



We need IFPUG function points as an objective, auditable, understandable measure that provides you all the benefits of being king.

Some people think that the price is the philosopher's stone. Others think that the effort, the Time to Market, or the user satisfaction, is their stone. Rather, the amount of software product (measured with IFPUG function points), will let us know if we are achieving our objective. How many burgers are we delivering? How many function points do I have? Comparing the quantity that each team is delivering to the business we can identify teams that may need help to improve their process.



At the same time, we are interested in the price of the final product. The quantitative sizing of the software facilitates our value analysis with suppliers and development teams. We can compare the value with similar data in reference databases to confirm intended value propositions.

We only trust the product

The third reason is that we can only trust the product. If someone offers you a low rate to build an App, is the price low or high? You don't know because you are missing the effort needed to make it happen. If they tell you that they will charge you one cent per hour to make the burgers, but they don't tell you how many hours are needed, you are missing key information for making your selection. What if they tell you that the burger requires 2,000 labor hours? A more complete picture of the size and cost gives us greater confidence in that choice. If we define the software product as the amount of product to be delivered using function points, and we price it, we will be better able to compare that option with competing suppliers. That is, we can assess our alternatives by prioritizing the product as king. While ordering in the restaurant, do you inquire about the hourly rate of the server or would you ask the price of the entree?

We need IFPUG function points as an objective, auditable, understandable measure that provides you all the benefits of being king. 

ABOUT THE AUTHOR



The Brand Manager of LedaMC & Quanter with more than 21 years of experience in IT Project and Services management, Julián is focused on helping companies to improve their software development processes. He has participated in Software Development projects, Quality Assurance projects, Sourcing, Benchmarking, Software supplier productivity management services, Process Development and Improvement, among others. As an IFPUG Certified Function Points Specialist, IFPUG Certified SNAP Practitioner, PMI Project Management Professional PMP®, PMI Disciplined Agile Scrum Master DASM® and Scrum Manager Autoridad, he has given conferences/trainings in Argentina, Brazil, Chile, Colombia, Italy, Spain, Panama, Peru and Romania. All these experiences have given him a broad vision of Project Management and its best practices. He is the author of two books *El Juego de Tronos de los Proyectos* and *Guía Práctica de Estimación y Medición de Proyectos Software: ¿Por qué? ¿Para qué? y ¿Cómo?* and blogger in El Laboratorio de las TI (<http://www.laboratorioti.com>).



MEASURING THE VALUE of Software

By: Bram Meyerson

Most executives agree that the global pandemic played a key role in accelerating the adoption of new technology solutions in most sectors. This resulted in an immediate change in the ways of work for employees as well as customer experiences. Subsequently, the pace of automation puts organizations under immense pressure to respond with a responsible and strategic approach.

We can already identify winners and losers, the risktakers and early adopters, as well as the laggards, in the technological revolution that continues to play out. The organizations that were able to pivot rapidly, backed by resources and support for digital transformation, are already proving their success in a post-pandemic world. Application technology is now a critical business capability and organizations can no longer tolerate slow delivery and poor returns.

As some economies show signs of recovery, successful companies have already taken the next steps needed to further extend their digital strategy. These steps include implementing flexible architectures and ensuring that they can leverage the value of the Intellectual Property (IP) embedded in legacy systems by seamlessly integrating new digital channels.

The challenge for finance leaders, in particular as they step back to consider the legacy of application solutions that keep their organizations running, is how to properly assess the value of their systems before large sums are invested into the next generation of technology.

Boards are demanding evidence from CFOs and CIOs of the value that enterprise IT solutions add before they allocate budget and resources strategically. So, what are the critical drivers for CFOs and CIOs to assess the value of the organizations' IT ecosystem

and dependencies, and what benchmarks are used for this assessment?

Approaches to addressing this challenge should include the latest practices in output quantification, based on Function Point Analysis, an international standard. The sizing of software is based on the business-functional capabilities of the software, such as the types and complexity of the business transactions the software is designed to process. Also evaluated are the richness or complexity of the maintained data, and the management information it provides. In reality, the value businesses obtain from such software is very much dependent on the value proposition of the business. However, what businesses can do, as a minimum, is judged upon whether their acquired/developed



Application technology is now a critical business capability and organizations can no longer tolerate slow delivery and poor returns.

software is being procured at a fair price. The foundation of these types of judgments is using techniques to "size" the amount of software procured and supported and relate this to the cost of the procurement and support. Effective management of software procurement, delivery, and support, requires a range of key dimensions of performance to be considered and measured, to help the business get the balance that meets its needs.

Like a risk audit, it will independently consider gaps of the current ecosystem against the IT transformation plan. Also included are elements for consideration such as the cost of maintenance and the ability for the system to support future growth. Importantly, like the services of a quantity surveyor, software buyers can use the outcomes to properly scope procurement projects, and determine the appropriate price points, utilizing benchmarks that



define acceptable and best practices.

This approach takes the guesswork out of the equation when making tough decisions. This means that we help CFOs and CIOs “speak the language of boards” and give them the insights and confidence needed to make highly informed and data-driven decisions.

The following is an example of a recent case study for a health insurance administrator.

Objective: To measure the value of their operating platforms (systems) for financial reporting purposes.

Method: The IFPUG FPA method was selected, excluding the analysis of complexity factors due to the time expected to calculate complexity factors. Instead, all FP components were set as “average” complexity. This approach is similar to the new Simple Function Point (SFP) method.

Approach: a) Our analysts sized the legacy member administration, policy, claims, and managed care systems using the method described above. b) Replacement costs for these applications were calculated using our unit cost benchmarks. c) A report was produced and presented to the Executive Management Committee.

Conclusion: As CFOs are concerned about the yield of their services and products, they need to think about measuring the value-for-money of supporting software applications that

underpin their revenue streams, whether owned, licenced, or provided as a service, which the organization then smartly orchestrates. 🏆

ABOUT THE AUTHOR



Bram Meyerson is the founder of Quantimetrix, established in 1992 with a presence in the UK and South Africa. He engages with Systems and Finance executives to address their challenges and needs. Bram has worked with some of the world’s leading telcos, financial services companies, and government agencies. Quantimetrix owns a vast benchmark database of information pertaining to software projects, underpinned with function point counts, and this is used by his clients to guide sensible decision-making and to evaluate and benchmark the cost of acquisition of bespoke software. Bram also specializes in de-risking and optimizing software application delivery and support processes.

IFPUG AD/M BENCHMARKING CERTIFICATION

By: Pierre Almén, on behalf of the AD/M Benchmarking Task Force



Having accurate and reliable data is critical for companies to survive in today's highly competitive business world. Companies must ensure they are focused on the appropriate goals by comparing their performance against the industry. IT organizations achieve that goal by using highly reputable benchmarking data and reports.

For more than 30 years, IFPUG has demonstrated a strong commitment to the IT industry by providing best practices, methods, and knowledge to sustain high levels of performance within the global IT industry. IFPUG continues its pursuit of IT

industry excellence by launching the **IFPUG AD/M (Application Development and Maintenance) Benchmarking Certification** program.

The IFPUG AD/M Benchmarking Certification represents a standard method through which IFPUG affirms that a benchmarking service provider has fulfilled the requirements deemed necessary to be competent to conduct an AD/M benchmark analysis, through the investigation of evidence that is based on criteria defined in applicable ISO/IEC 29155 tasks and activities.

IFPUG has concluded that the **ISO/IEC 29155 series** represents the most appropriate framework for IT AD/M benchmarking, as illustrated in the following figure:

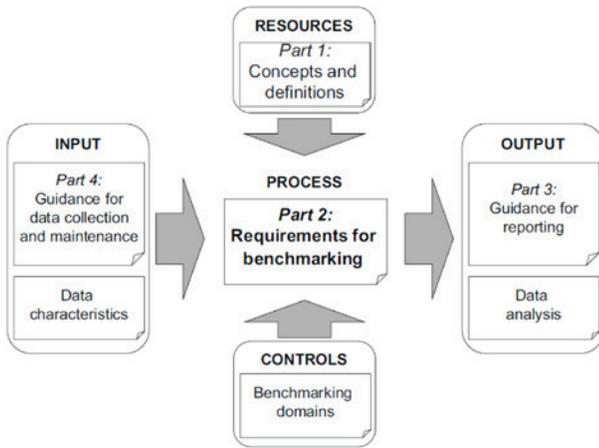


Figure 1 - ISO/IEC 29155 standards overview (extracted from Part 2: Requirements for Benchmarking, Page V)

Benefits

The following items describe the core benefits that an AD/M benchmarking provider can obtain from the IFPUG AD/M Benchmarking Certification:

- achieve a best-in-class corporate level benchmarking qualification,
- establish a standard for requirements, guidelines and qualifications for AD/M benchmarking,
- apply a consistent approach used by both benchmarking consultancy companies and software suppliers, and
- define the measure of the ability and the quality of their benchmarking process.

The Task Force

To manage the creation of this new certification process, IFPUG nominated members of a task force team. The team was responsible for the design of the certification process and the core associated products. The task force team was comprised of the following IFPUG members:

- Pierre Almén, ISBSG President and Chair of Business Applications Committee,
- David Herron, former member of the Communication and Marketing Committee, where he was the chief editor of MetricViews magazine,
- Don Beckett, member of the Certification Committee,
- Iván Pinedo, member of the International Membership Committee and Country Representative for Spain, and

- Sérgio Brigido, Board Director and liaison of the Partnerships and Events Committee and the Communications and Marketing Committee.

The task force designed two core products:

- Certification Application: A formal application, that includes a code of ethics and the disclaimer
- Checklist: A questionnaire to be used by the candidate company applying for certification. It contains 14 questions and evidence descriptions covering a selected set of ISO activities grouped by: Requirements, Execution, Outcomes.

Scoring

The checklist scoring supports the evaluation of the evidence provided for each of the 14 questions based on specific criteria. Six of the questions are identified as Essential and eight of the questions are identified as Regular. Upon examination of the submitted evidence, each question is rated as either Fully Achieved, Partially Achieved, or Not Achieved and assigned a pre-defined number of points based on their rating. Essential questions have a higher value than Regular questions and a Not Achieved rating on any Essential question results in a disqualification from the certification process. If there are no disqualifiers, then the awarded points are totaled. Certification requires that the candidate company score a minimum of 80% of the total point value.

Certification Policies

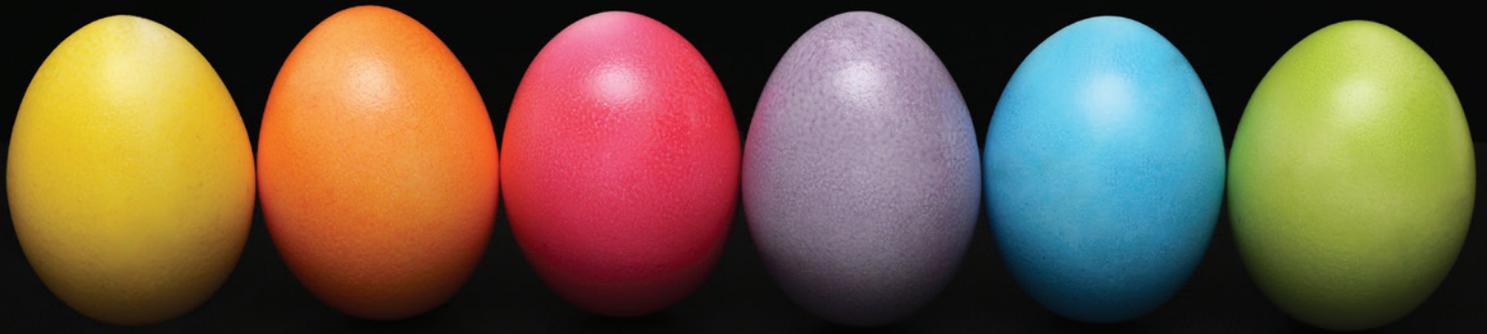
The following items represent the core policies that support the certification process:

- The certification is valid for three years
- Maintaining an IFPUG corporate membership is mandatory to retain certification
- If the candidate company fails to pass the assessment, it can reapply for certification at a 50% discount rate within six months

Pilot

During the period between July and September of 2021, the task force team submitted the certification process to a period of testing by conducting a pilot with a candidate company—LedaMC. The pilot assessment was completed in September 2021 and recommended the approval of LedaMC for IFPUG AD/M certification. In addition, the task force team identified several lessons learned and improvements that will be implemented in the core products and the certification process.

The official deployment including an updated website is planned for 2022. Anyone with interest in this certification program can contact ifpug@ifpug.org for more information and assistance. 🌐



Simple Function Points:

YOUR QUESTIONS ANSWERED

By: David Herron

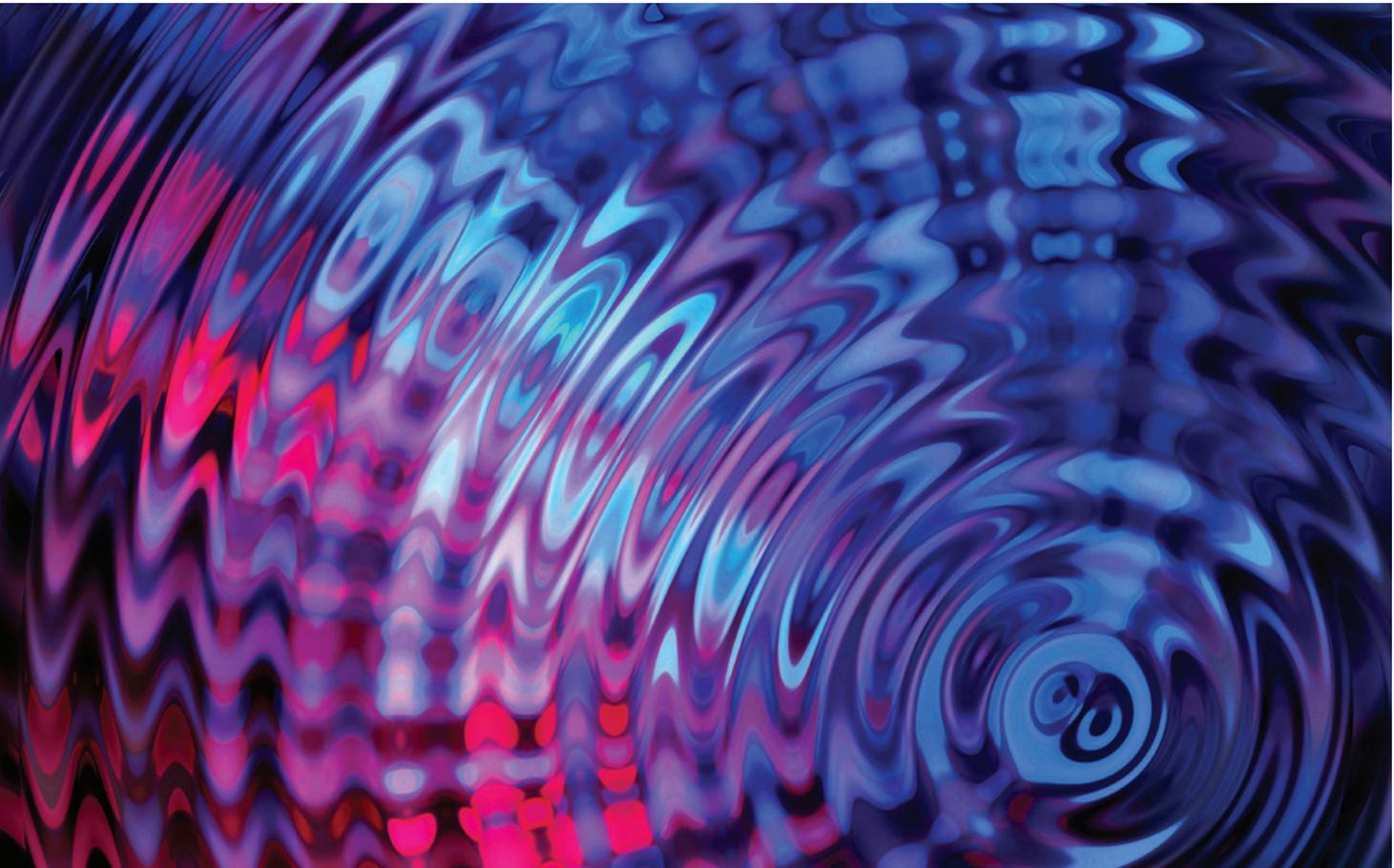
I**ntroduction:** IFPUG recently acquired the rights to Simple Function Points (SFP). To better understand how SFP are used and their potential impact on sizing, David Herron (DH) conducted an interview with the experts at Galorath. Galorath is an industry leader in software estimation and is well-versed in software sizing. Following is an interview with Dan Galorath, CEO of Galorath Incorporated, Karen McRitchie, Vice President for Product Development of Galorath Incorporated, and Bob Hunt, President of Galorath Federal Incorporated.

DH: Why is functional sizing important?

Dan Galorath: The question should be posed as “why is sizing important?” Size is a measure of scope and effective size is key to understanding progress, effort, and schedule.

The easier it is to determine/estimate the size of the intended system, modification, or maintenance, the more valuable the sizing measure becomes. And the simpler the method is, the easier it is to understand by technical personnel and management.

Sizing is a key component in the estimating of software deliverables. Functional sizing is a means by which the functionally being delivered to an end user is evaluated and measured. Function Point Analysis (FPA) is an accepted industry standard for functional size measurement. Functional sizing helps to make software sizing and estimating more transparent to the end-user. And SFP is the latest iteration of functional size. Pilot studies show SFP is substantially faster to count and hence more likely to be used.



DH: What are the standard size measures commonly used today?

Karen McRitchie: For many years the most common software sizing methodology has been counting the lines of source code; often referred to as SLOC. Beginning in the late 1990s Functional Size Measurement became the industry standard (at least in the commercial sector, many government projects continue to use LOC) and is used to compute the size of software functions by performing FPA. IFPUG's FPA is used among measurement and estimating experts worldwide. Variations on the IFPUG Functional Size Measurement methodology have emerged, and these standards are COSMIC Function Points, MK II Function Points, Nesma Function Points, and FISMA Function Points. Other variants of these standards include Object-Oriented Function Points (OOFP) and newer variants as Weighted Micro Function Points, which factor algorithmic and control-flow complexity.

DH: How different is a full-function point count versus an SFP count?

Bob Hunt: SFP counting is exactly what the name implies; it's simple. A full-function point counting process goes through several well-defined steps. Those steps include gathering available documentation, identifying the project boundary and scope, identifying the five data components (inputs, outputs,

inquiries, internal data stores, and external data stores), assessing complexity, applying appropriate weights, and computing the size. The SFP method executes some of those same steps. SFP differs by identifying and assessing the five data components and classifying them into one of two groups. One grouping includes data transactions (inputs, outputs, inquires) and refers to that group as Unspecified Generic Elementary Process (UGEP). The second grouping includes data stores (internal and external data groups) and refers to that group as Unspecified Generic Data Group (UGDG). The steps that include assessing the complexity and applying appropriate weights are greatly simplified by assigning a constant value to each unique UGEP and UGDG. SFP streamline counting by removing the complexity assessment step. UGEPs are assigned a value of 4.6 and UGDGs are assigned a value of 7.0. Research shows that there is a strong correlation between full function point counts and SFP counts.

DH: Do SFP give me sufficient visibility into the size that I can trust?

Bob Hunt: This involves two considerations: visibility and accuracy. Visibility into size means being able to identify all of the required software features and functions that the user is requesting and then accounting for each transaction type and each data group during the sizing process. With SFP you can demonstrate that you have accounted for every function and

feature. SFP counting requires that all functions are identified, classified, and counted. A user review ensuring that all features and functions have been counted and sized will create a level of trust. Think of visibility as transparency.

Similar to full function point counts, the accuracy of SFP counts is dependent on the availability of artifacts that properly describe the user requirements. By comparison, SFP counts have been statistically shown to have a strong correlation to full function point counts. That is to say that one method is no more or less accurate than the other. The bottom line is that if the user has properly identified all required features and functions, they will be included in the size calculation. If the requirements are incomplete or ambiguous then accuracy could be compromised.

DH: What if I am already using FPs; did I waste my company's resources by investing in function point counting?

Bob Hunt: Not at all. If you have successfully been using full function point counting practices, then there is no reason not to continue. SFP counting is not a better method, it is just a simpler method. When applied, it yields the same sizing results as a full count. There is much more detailed information required to conduct a full function point count and if that detailed information is of value to you and the amount of effort expended to gather that detailed information is worth the investment, then you should be satisfied with the overall process. But it should be noted that simple function points have been statistically proven to yield the same results with significantly less effort.

DH: What are some of the benefits of using SFP?

Bob Hunt: The best way to answer this question is to first look at what some of the challenges are with the full function point counting method. Full function point counting requires very detailed information regarding each unique transaction and data store. Early in the development lifecycle, this information may not be complete or may not exist at all.

Additionally, when counting full function points, the counter must be fully versed on the rules and guidelines that govern how to count various inputs, outputs, etc. And, of course, all of this takes time and can be

very labor-intensive. This is not the case with simple function point counting. The identification of data element types, record element types, and file types referenced is not required when using the simple function point method. Therefore, simple function points are quicker and easier to apply. In some cases, it provides sizing information quicker and earlier allowing for initial estimates to be generated earlier in the lifecycle.

DH: How are SFP applied in an Agile environment?

Bob Hunt: The easy answer would be to say, "SFP can be applied in an agile environment like any other functional measurement." However, the reality is not so simple. In the application

development world, software engineers often follow a waterfall model. In that environment, requirements are (theoretically) stated early in the lifecycle. Those functional requirements are then used to do a full or simple function point sizing.

In the agile environment, not all requirements are known early in the lifecycle of the project. Numerous iterations or sprints are conducted anywhere from a week to a month in duration. Stories are presented and sized using story points. This gives the agile team a mechanism with which to "size" how much work can flow through a given sprint thru to completion. Story points are not standardized and may not be consistently applied across agile teams. Instead, they are intended to facilitate a common understanding of the effort by the agile team members for a given story or set of stories within a sprint.

Functional sizing has limited value at the sprint/iteration level, however, there are several ways to apply functional sizing in an agile environment. The easiest application of function points would be at the end of each sprint when a working piece of software is delivered. Additionally, function point size can be used with other metrics to measure things like quality and productivity. At the beginning of an agile project, requirements likely exist at various levels of abstraction. The most immediate features are detailed while future features may not be as well defined. At that higher level of abstraction, some assumptions can be made, and high-level functional size can occur.

DH: Will management understand the counts when I discuss simple function points?

Bob Hunt: Absolutely. For management, the end game is not about computing a size metric. It is about being able to understand the estimated costs associated with the development and deployment of a given project. Should the conversation with management get to a level of detail where sizing is discussed, it will be far easier to talk about the SFP method vs. the full function point counting method.



SFP counting is not a better method, it is just a simpler method.

DH: Do tools like Seer use SFP?

Karen McRitchie: Yes. All of the off-the-shelf parametric models covered in the recent ICEAA training summit: SEER-SEM, True Planning, SLIM, and COCOMO, support the use of functional size as an input. As a result, they support using an SFP for estimating. SEER for Software has an SFP size metric that allows for counts for elementary processes and logical data groups to be entered directly. When estimating, oftentimes the counts are "estimates" of the count, which involves uncertainty. SEER for Software considers when, in the lifecycle, the count was established, factoring in lack of visibility into detailed requirements early on. Also, uncertainty can be expressed as a range on the count—using the least, likely, and

most inputs. This combined with Monte Carlo risk analysis will result in an estimated range by the confidence level.

DH: Do SFP require training/certification?

Dan Galorath: With any new tool or methodology, it is always best to learn the proper way to use the tool or method. Training on full function point counting is typically a two-day class. Since simple function points do not include many of the details, rules, and complexities of full function point counting, a training class in simple function points can be conducted in a single day or even a half-day.

And SFP appears to be an opportunity to make functional sizing mainstream, with a wider community of software leaders and developers on board than ever before. This is due to SFP's ease of use.

When SFP is mainstream, many users will not be certified but will pick it up. Providing quick, intuitive web-based training, such as on YouTube, could be sufficient for such users.

Additionally having a core user(s) who is certified can help ensure SFP is applied by those mainstream engineering.

Of course, the need for certification depends on the nature of your position and where you work. An IFPUG certification would certainly be valuable for those specialists. And consultants should be certified if they are offering SFP services.

DH: What evidence do you have that SFP are viable?

Bob Hunt: Table 1 shows the full IFPUG count for a major IT system within 8% of the SFP count. This comparison is compelling. While we didn't track the actual labor difference between the counts, the SFP count was substantially quicker. We are pleased that SFP appear viable and can potentially become the primary counting method or a sanity check on a full count.

The differences in size will not impact well-constructed software estimation models.

IFPUG High Level	Simple Function Points (SFP)	Difference
18.1k	19.7k	+8%

Table 1: Traditional IFPUG vs. SFP

DH: How can I find out more about SFP?

All: The best place to learn about SFP along with training and certification options is at www.IFPUG.org. In addition, any of the named partner companies on the IFPUG site are also available to you. 

ABOUT THE AUTHOR



David Herron is an authority in the areas of performance measurement, process improvement and functional size, among others. David has put into practice in a number of entities from different parts of the world, the use of metrics to monitor the impact of IT on the business and in governing outsourcing contracts. He has brought IFPUG and its concepts and add value to the highest levels. David was one of the founders of the David Consulting Group and has provided consulting and coaching services for a high number of IT companies. He has presented at IFPUG conferences on topics such as "Identifying Your IT Organization's Best Practices," "The Need for Auditing Your Measurement Program," "Early Lifecycle Identification of Software Quality Risk Factors" and "The Lighter Side of Software Measurement." He has authored articles and co-authored books such as *Measuring the Software Process: A Practical Guide to Functional Measurements* and *Function Point Analysis: Measurement Practices for Successful Software Projects*.

Functional and Non-Functional

MEASUREMENT AND REPORTING WITH SCRUM

By: Joe Schofield



Abstract: This article suggests an approach for the integration of functional and non-functional measurement for Scrum practitioners. Because Scrum is undeniably the most widely used of the agile frameworks,¹ clear articulation for measurement techniques with Scrum in general, and stories in particular, are a necessity for Function Point Analysis (FPA) advocates. Additionally, incorporating relevant measurement into organizational reporting for traditional cost, schedule, and scope is critical to enable and sustain agile cultural transformation.²

Why focus on Scrum and (User) Stories: Most organizations, as well as Function Point enthusiasts, struggle to transition from traditional “predictive”³ development methodologies to adaptive frameworks. Many Key Performance Indicators (KPIs) and Objectives and Key Results (OKRs) are tied to classic “iron triangle” components of scope, cost, and schedule. Meanwhile, 66% of agile teams use Scrum, with another 15% using Scrum with either Kanban or eXtreme Programming (XP).¹ As an interesting side note, many teams using Scrum with Kanban, or ScrumBan, do so because they reject timeboxing which requires them to stop, inspect, and adapt to alleviate workflow impediments. Ironically, timeboxing with continuous improvement is quite often the very remedy to their inability to complete work within a sprint, their initial motive to abandon Scrum. Another often-cited reason is their perceived limitation to release “on-demand” with Scrum. Despite these

curious criticisms, Scrum and its cited variants are used eight times more than all of the other agile frameworks combined. Organizations and teams utilizing Scrum are therefore the primary target audience for this paper. However, the common denominator for functional and non-functional measurement is a story, also used for instance, in eXtreme Programming as story cards.

Applying FPA to User Stories: Functional measurement quantifies value delivered to the business. The business is the recipient of a service or product created by the Scrum team as prioritized by the Product Owner, aka the “voice of the customer.” In 2018 I proposed that stories be written at the elementary process level corresponding to CRUD (create, retrieve, update, delete) activity offering several advantages:⁴

1. Writing user stories in the language of the business aligns with FPA by keeping the focus on business needs rather than collecting technical implementation details which, are often mistakenly captured as part of the story rather than as tasks required to fulfill a story. The business need to “take an order” translates easily into tasks to “create an order” during sprint planning.
2. Perhaps the most valuable reason to decompose stories to the elementary process level is to answer the elusive question of “when do we stop breaking a story down?” The absence of a decomposition *boundary* of an elementary

process can result in excessive decomposition often to the task level, at which point the story loses its business focus. Unnecessary decomposition is not lean, nor is it in keeping with the 10th agile principle.⁵

3. Duplicate functionality becomes more visible; re-use more intentional.
4. The story translates naturally as a transactional function for FPA.
5. The story is defined as “small enough” to fit within a sprint aiding teams that claim stories are “too big.”
6. Acceptance criteria are narrowly defined promoting clarity and tighter “coupling” with story completion.
7. The data functions are typically specified as part of the story or included in the acceptance criteria for the story.
8. The acceptance criteria for the story provides a “container” for non-functional needs and considerations for IFPUG’s SNAP.⁶
9. Teams can establish a taxonomy to retain and distinguish among the use of story points, function points, and even use case points for relevant audiences.⁷

Using the Acceptance Criteria in Stories for SNAP:⁶

Scrum stories capture the need of the business and when tasked during sprint planning, the work of the Scrum team. Briefly, not exhaustively, Scrum stories reside in the Product Backlog, are written in the language of the business, are owned and prioritized by the Product Owner, can be written in the Connextra style⁸ (As a...I want . . . So that . . .), are limited to the effort and duration of a single iteration or sprint. Acceptance criteria is an essential, some would argue mandatory, attribute of a story. A story without acceptance criteria is not “ready” to be pulled into a sprint backlog. Once committed as part of the sprint backlog, the acceptance criteria remain unchanged. The demonstration by the Scrum team of the story and its acceptance criteria is the basis of the “acceptance” or “rejection” of a story near the end of an iteration during the sprint review. The Product Owner adjudicates the “acceptance” or “rejection” of the story on behalf of the business and its stakeholders. Acceptance criteria that are common to the work of the Scrum team may be consolidated into the “Definition of Done” (DoD), which all stories are expected to satisfy before being included in the sprint review.

Acceptance criteria combined with the DoD constitute the “conditions of acceptance” of the story; that is, the functional and non-functional verification (on behalf of the producer) and validation (on behalf of the consumer). Almost all of the non-functional elements of SNAP are well-suited for inclusion as acceptance criteria. The following table offers guidance for using the acceptance criteria associated with stories for SNAP determinations.

Table 1: Suggested SNAP elements incorporated into User Story Acceptance Criteria

SNAP Elements	Testable Acceptance Criteria?	*As an example . . .
1. Data Operations		
1.1 Data Entry Validations	Yes	A valid date; a valid address within a city
1.2 Logical and Mathematical Operations	Yes	Precipitation prediction based on historic probability
1.3 Data Formatting	Yes	Different date formats; credit card data tokenization
1.4 Internal Data Movements	Yes	Use a date from another “partition”
1.5 Delivering Added Value to Users . . .	Yes	Use the zip code to capture the city and state information
2. Interface Design		
2.1 User Interfaces	Yes	Enter a date via voice
2.2 Help Methods	Yes	Hover over a “?” to see how to enter or select a value
2.3 Multiple Input Methods	Yes	QR codes scan, links, or “taps”
2.4 Multiple Output Formats	Yes	Send a receipt via e-mail, a robocall, or text message using the same functionality
3. Technical Environment		
3.1 Multiple Platform	Yes	Web- and app-based solution
3.2 Database Technology	**No	
3.3 Batch Processes	Yes	A monthly compliance scan that does not report any user data
4. Architecture		
4.1 Component Based Software	**No	
4.2 Multiple Input / Output interfaces	**No	

*The “As an Example” column contains cells that are notional, an expression of an idea. They are not intended to redefine or alter SNAP definitions per the Software Non-functional Assessment Practices Manual (APM).



**The Scrum development team is responsible for the solution it delivers. Neither the Scrum Master nor the Product Owner determines who does the work of the team or how that work is completed; this includes but is not limited to database and component usage. Granted, the existence of architectural components and standards, platforms, and environments may be “inherited” by the team from the broader organization. “Scaling” choices may also constrain the technical solution that would otherwise reside with the Scrum team.

From Measures to Metrics⁹: Alternatives for Agile Project Reporting

As a quick reminder, the word *measurement* is used to describe the act of collecting *measures*; that is, values like quantity, weight, value, and size. *Measure* is also a verb that depicts the comparison of a value to a standard like inches, pounds, and hours. *Metrics* are the meaningful comparison of two *measures* to derive a value for comparison to similar paired sets of *measures*. As an example, Scrum teams may conduct *measurement* by collecting *measures* such as expected task hours, actual task hours, expected story value, and story points. Teams may derive *metrics* to improve future expected times by comparing expected to actual task hours. In addition, they might sum story points “accepted” per sprint to derive velocity.

Precautionary reminder: The expressed intent of this article is to more easily assimilate the continued use of FPA and SNAP in story-based agile environments. Attempts to use velocity, as an example, to compare teams or to forecast

Scrum team completion dates by applying highly unstable and intentionally volatile product backlogs are harmful to self-organizing teams. This same misuse of measures impedes the desired trust between the Scrum team and its stakeholders. Often contrary to organizational desires, agile teams shun productivity measures within and among teams due to variation in the:

- target product or service,
- number and expertise of team members,
- physical and technical environments,
- profit-margin impacts,
- hidden investment in cross-functional development, and
- team role boundaries violated or honored by self-organizing teams.

Scaling frameworks that attempt to minimize some of the wariness and suspicion that accompany “productivity measures” can be met with skepticism, often well-deserved due to the organization’s past practices. Providing the Development team with work estimates hinders self-organization and team accountability. Productivity measures and estimates are easily and often misused and manipulated¹⁰ with story points and projected completion dates. Leadership teams convene around conferences tables (virtual or real) and ponder how to get Team A to produce more like Team B instead of how to remove barriers to success that Team A confronts daily. Frequently, leadership inhibits the cultural change necessary for teams to succeed.¹¹

Team Measurement

Measurement is conducted primarily for two reasons:

- 1) As clearly expressed in the 12th Agile Principle, teams measure to improve performance based on their continuous learning and ongoing improvement, the essence of heuristic thinking¹². The Principle reads . . .

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Observations made during retrospective events vary from teaming skills, to cross-functional growth, communication flows, tool knowledge, distraction management, self-imposed constraints, backlog refinement, and workflow. While measures can certainly be collected for each of these, other more readily, and often more practical measures are collected for the number of “accepted” stories, story points and expected and actual task time. From these measures, teams can if desired, understand variation in their estimates and actual task time, use the story points for a threshold in future sprint planning meetings, and minimize the effects of known causes for rejected stories. Each of these opportunities further illustrates the value of heuristic thinking.

Teams practicing Kanban are by definition more focused on work-in-progress (WIP) limits, as well as lead and cycle time optimization. While Kanban is not the focus of this article, it still promotes quantitative continuous improvement.

- 2) Measurement supports status reporting structures, traditionally for projects, programs, and portfolios. Less fortunately, measurement can be used as a defensive mechanism when the team is accused of being behind schedule, over budget, or outside the scope. But each of these “iron triangle” constraints is of decreasing value to agile organizations who find more meaningful metrics for understanding releases, value delivery, and prioritization.

Organizational Measurement

Project level measurements are almost always aggregated at some level in the organization. Mid-level management and senior leadership periodically scrutinize reports that consolidate key initiatives onto a dashboard. A classic summary of these project statuses include:

- projects in rows along a “y” axis,
- cost, schedule, and scope “columns” and
- red, yellow, green indicators in intersecting cells.

Reports like these provide decision-makers with a glimpse into what they believe to be the overall health of the work being undertaken, some financial confidence that dollars are not being expended without the completion of milestones (scope), and alignment with expected completion (schedule). Projects escape further surveillance when the cells are “green.” Questions arise when the indicators are “yellow” and often attract unwanted

attention when “red” or thought to be trending as such. “Fixing” the red indicators has for decades been addressed with change requests or exception reports that typically require an explanation for the deviation, some corrective action, and potentially a new forecast of completion, spending, or scope freezing. Organizations can develop onerous and invasive processes to raise leadership’s confidence that tomorrow will bring better outcomes. Sadly, agile projects that foster changing requirements and priorities late in development, driven by business innovation, become the shamed victims of reporting systems that focus on milestones and outdated schedules. This culture of reporting and corrective action obscures agile transparency and stymies adaption by viewing scrum product increments through historically-tinted bifocals. Agile progress indicators are rejected or ignored. What are they?¹³

Agile’s value-delivery emphasis is misrepresented by classic cost, schedule, and scope measures. Rather, that discussion needs to be recast toward *value-delivery, releases, and priorities*, respectively.

- *Value-delivery* is realized when “seed funding” encourages the team to create early value and then (funding) flows as value continues to be delivered. No or limited value delivery also serves to question the longevity of the work triggering the possible cancellation of work that has little chance of being completed. Contrast this scenario with projects that are funded for years that fail to deliver and where funds could have redirected toward opportunities with more promise.
- Early and frequent *releases* provide the business with early and frequent value since the business selects the capabilities to be delivered. Tracking releases and release velocity seems superior to setting and re-setting schedules.
- Scope has always been about *priorities*; “in-scope” is more urgent than “out-of-scope.” But tracking priorities in Scrum reminds us that *priorities* are subject to change at the discretion of the product owner, potentially honed each sprint during *grooming*, also known as *refinement*.

Grooming is the Scrum change control process. Agile organizations do not need the burdensome change control processes imposed by traditional and well-intended project management offices for expectedly evolving and constantly innovated business needs.

Alternative Team Metrics

Since each sprint creates some value as determined by the Product Owner, the inclusion of a value with each story by the Product Owner enables the derivation of value for each sprint. The value-delivered for completed stories in each sprint is represented in Exhibit A. Tying stories to Function Points and SNAP Points provides an added value for FPA enthusiasts.

The value delivered for each release can be embodied in a cumulative value-delivery chart. Release when determined by the business, considering the cost of delay and the transaction cost of the release.¹⁴ Again, the relationship of stories to releases and stories to Function Points facilitates FPA.

Exhibit A

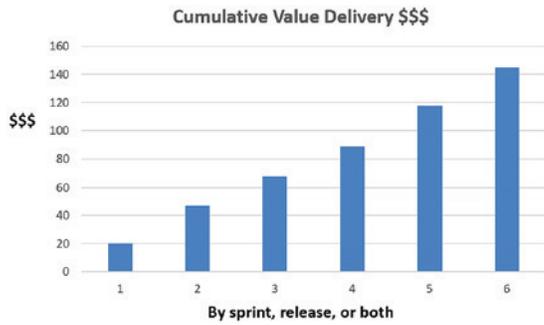
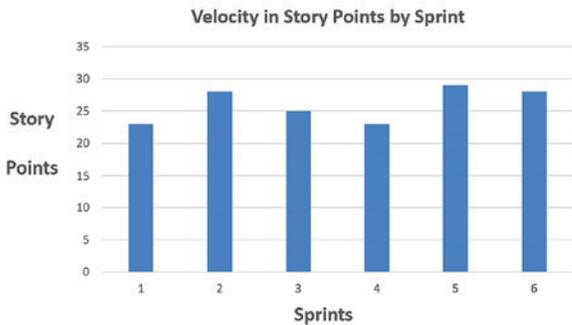


Exhibit B



Other insightful metrics might include defects per release, story point completion (velocity) to defects, and capacity (team

available hours per sprint) to velocity. This last metric is an obvious contributor for explaining variation in velocity (see Exhibit B as an example), yet is often ignored. Absent this information, senior leadership may seek explanations for swings in velocity or even worse, attempt to compare velocities among teams in search of some best practice to be imposed on all teams. Understanding the nature of each team's relative values with story points and derived velocity is too often missed at the organization level. Establishing standards for velocity across all teams has other detrimental, even disastrous effects. The organization prioritizing a story point value over Scrum team understanding during its own grooming provides numerous opportunities for negative outcomes.

Conclusion

Agile product development occurs worldwide. Scrum is without a doubt the most prevalent agile approach employed. Similarly, IFPUG's Function Points and SNAP share a broad international audience. Traditional reporting impairs Scrum's value-delivery approach; however, as proposed, agile relevant success measurements and FPA can be employed bridging historic and contemporary measurement systems. Organizations that include senior leadership, management, and practicing teams as part of their agile transformation have a brighter glimmer of hope to overcome the cultural change that so often disrupts adoption. 🍀

Special thanks: Talmon Ben-Cnaan whose helpful insights and expertise enhanced the quality of the "examples" in Table 1.



REFERENCES:

¹15th State of Agile Report; digital.ai; May, 2021; page 13 – Scrum used by 81 percent of survey respondents

²Ibid.; page 12 – the 2nd leading challenge to agile adoption is “organizational culture at odds with agile values”

³Aligning the PMO to Lead Agile Transformation; Schofield; Project Office Journal; IT Metrics & Productivity Institute; September, 2019

⁴Reflecting on Measurements in an Agile World; Schofield; MetricViews; International Function Points Users Group; Spring, 2018; pages 14 - 17

⁵<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>; Agile Principle # 10: Maximize the amount of work not done!

⁶https://en.wikipedia.org/wiki/SNAP_Points retrieved 10/4/2021; SNAP is the acronym for “Software Non-functional Assessment Process”, a measurement of non-functional software size. The SNAP sizing method complements ISO/IEC 20926:2009, which defines a method for the sizing of functional user requirements. SNAP is a product of the International Function Point Users Group (IFPUG) and is sized using the “Software Non-functional Assessment Practices Manual” (APM) now in version 2.4. Also, ISO/IEC 32430:2021; The SNAP methodology has the IEEE standard IEEE2430-2019.

⁷Function Points, Use Case Points, Story Points: Observations from a Case Study; Schofield, et al; CrossTalk; May / June, 2013

⁸Rachel Davies; Connextra; circa 2000

⁹<https://wikidiff.com/measurement/measure>; retrieved 10/5/2021

¹⁰Inflate Gate: Mastering Overestimation for Agile Software Projects; Schofield; Computer Aid’s Accelerated IT Success; (Featured Article); IT Metrics & Productivity Institute; August, 2015

¹¹Aligning People and Culture for Agile Transformation; Schofield; 2020; See Vignettes 45, 48, 53 – 56 – Executive Action Team

¹²<https://www.thoughtco.com/heuristics-psychology-4171769>; retrieved 10/5/2021

¹³Countering 5 Barriers to Organizational Enterprise Success; Schofield; AgileConnection; July 17, 2019

¹⁴The Principles of Product Development Flow, Second Generation Lean Product Development; Reinertsen, D Celeritas Publishing; 2009; ISBN-10: 19354010

ABOUT THE AUTHOR



Joe Schofield is an Authorized Training Partner, a Scrum Certified Trainer and Agile Coach with SCRUMstudy™. He has more than 80 published books, papers, conference presentations—including contributions to the books: *The IFPUG Guide to IT and Software Measurement* (2012), *IT Measurement*, *Certified Function Point Specialist Exam Guide*, *The Economics of Software Quality*, and his recently released *Aligning People and Culture for Agile Transformation*. He holds nine agile-related certifications: SCT™, SCAC™, SSMC™, SSPOC™, SMC™, SDC™, SPOC™, SAMC™, and SAFe 5. Joe is also a Past President of the International Function Point Users Group, a Certified Software Quality Analyst, and was a Lean Six Sigma Black Belt. He retired from Sandia National Laboratories as a Distinguished Member of the Technical Staff. He served the last 10 years of that 31-year career as the “Chief Process Officer” of an organization of 400 software engineers which was awarded a SW-CMM® Level 3.

Advancements in Software Development Productivity:

THE FP-BASED 'PRODUCTIVITY PARADOX'

By: Luigi Buglione and Carol Dekkers

Abstract: In the late 1980s, Capers Jones wrote about the *Paradox of Source Code Metrics* [1] hoping to stimulate organizations to adopt Function Point Analysis (FPA) as a preferred method of software sizing. Today, function point-based productivity has itself led to a “new” productivity paradox that presents challenges to data-based software estimation. This article describes these productivity paradoxes and outlines remedies and a most robust approach to software sizing in the future.

In the 1980s, Capers Jones wrote a well-known paper [1] about the “(productivity) paradox of source-code metrics,” and recommended that companies adopt Function Points (FP) to overcome it. This productivity paradox arose when calculating productivity using Source Lines of Code (SLOC)-based sizing. Productivity is typically expressed as a ratio of the output of a process divided by the input, for example in the U.S. automotive industry, miles per gallon (of gasoline) is a common metric. In software development, the term productivity is generally used without standardized definitions, and is expressed as output (software product size in standard size units of measure) divided by input (project work effort, typically in hours). With the advancements from traditional lower-level programming languages (e.g., Assembler or COBOL) to higher-level languages (e.g., Java, Dot Net or SAS), one would anticipate higher levels of productivity. The paradox of source code metrics occurs because the higher-level languages, which require less SLOC to deliver the same software compared to lower-level languages, actually appear to be less productive when the productivity ratios use SLOC for size. To illustrate, consider two fictitious projects that deliver the same piece of software using a different programming language: Project 1 uses Java, was sized at 1,000 SLOC, and requires 500 hours of effort (productivity = 2 SLOC per hour); project 2 used COBOL, was sized at 10,000 SLOC, and took 1,000 hours of effort (productivity = 10 SLOC per hour.)

The paradox of SLOC-based metrics implies that the COBOL project is more productive (i.e., more SLOC per hour.) Higher-level languages, such as Java, require less SLOC, but the effort allocation is not proportional. Another problem with SLOC-based productivity is that it rewards spaghetti code (poorly planned, excessive SLOC): the higher the SLOC count, the more productive the project appears to be.

To counter this paradox, Jones recommended a new unit of measure called FP. FP measure a software’s functional size based purely on its functional requirements (what the software does), independent of the technology or programming language. Function points also included an optional value-adjustment factor (VAF) that could adjust the software size up or down by up to 35% based on the non-functional aspects of the software. FP-based productivity thus normalized unit costs and productivity *on the basis of functionality*. Of course, productivity rates still varied based on the tools and technology used, but FP gave the practitioner a technology-independent way to overcome the SLOC-based productivity paradox.

At the same time that function point usage became mainstream over the past 30+ years, software complexity, demands for higher quality, and other non-functional software requirements (NFR) were also increasing. In the 2004 Springer *Perspectives on Software Requirements*, the paper titled Non-Functional Requirements Elicitation [2] states: “non-functional aspects have been treated as properties or attributes after the fact. While these properties have always been a concern among software engineering researchers, early work has tended to view them as properties or attributes of the finished software product to be evaluated and measured. Recent work offers the complementary view that they should be treated as requirements to be dealt with from the earliest stages of the software development process, and then throughout the entire lifecycle.”



In 2010, in recognition that non-functional requirements could be treated as a separate and measurable component of software product size, the International Function Point Users Group (IFPUG) introduced a new unit of measure called Software Non-functional Assessment Process (SNAP) Points to quantify the size of NFR. In 2020, IFPUG's SNAP methodology became an ISO/IEC and IEEE standard.

Today, the FP-based productivity paradox arises when functional size is used as *the singular unit of measure for software size*, (even if it is adjusted). The impact of non-functional requirements on software development costs can be double or more according to leading researchers including Dr. Barry Boehm (of the University of Southern California) and William Perry (of the Quality Assurance Institute). Adjusting the functional size as an attempt to quantify the impact of non-functional requirements (e.g., using the IFPUG VAF), does not adequately account for the increase in costs. As such, a new productivity paradox has arisen and is the subject of this article.

This paradox is illustrated with projects for which there is increased project effort not traceable to functional requirements,

but rather to the meeting of non-functional software enhancements. Additionally, there are the zero FP projects for which there are only non-functional requirements. While the customer value of the non-functional requirements (e.g., reduced response time, increased performance and reliability, better user interfaces), is often equal to or even exceeds the value of the functional enhancements, using purely FP-based productivity obscures this.

This new "productivity paradox" arose because FP-based productivity measures non-functional requirements only through an adjustment factor to an FP-based size. An example of a zero FP project would be:

- add security features/encryption to existing software (non-functional), and
- migrate all programs to a client-server platform from a mainframe environment (non-functional).

The productivity paradox is especially pronounced on software enhancement projects where the effort for non-functional requirements is obscured, and not accounted for. As an example,

consider two enhancement projects equal in functional size at 100 FP. Project 1 includes non-functional enhancements to improve the system performance, increase usability, tighten up existing security, and streamline multiple screens onto a central, user-friendly dashboard. (Note that none of these increased the adjustment factor, which was 1.0). Project 2 includes none of the non-functional requirements of Project 1. Obviously, the effort to complete Project 1 would be much higher, and its FP-based productivity would appear to be much lower than on Project 2 (implying that Project 1 was less productive). In terms of software project estimation, if FP-based productivity was used to estimate the two projects—the effort and cost would end up to be the same, despite the differences in non-functional requirements.

(Note: There are several commercial estimating tools that successfully use a combination of SLOC-based and FP-based productivity ratios, together with proprietary approaches that quantify the impact of non-functional and technical requirements, to produce reliable software estimates. The parametric and data-based approaches rely on a foundation of thousands of software development projects data points normalized and analyzed to ensure consistent results. This article is not intended to dissuade tool usage, but rather to illustrate how a more robust approach to software sizing, beyond mere functional size, can improve manual and analogous software estimates.)

Software development projects that are constrained by non-functional requirements (that describe how the software shall operate) and deliver more value to the project, will always appear to be less productive when using FP-based productivity. Reason: As illustrated in the example, when functional requirements are estimated using FP, but the NFR are not, the project effort will not go up according to the degree and impact of the NFR. To summarize: Higher FP = higher effort; higher NFR = higher effort as outlined in Figure 1. The overall Cost/FP will increase due to the non-functional requirements, that are not measured (or even considered). This is the new paradox.

Both authors began researching these issues over a decade ago; in Italy, the first author, in 2010, began to re-label what is typically called productivity, with a new label: “nominal” productivity [4]. He did so in order to correctly analyze and address the variability of values in the effort/cost estimation process and avoid a large “cone of uncertainty” [5]. The second author, around the same timeframe, began to address the Zero FP project issue in articles and with client engagements. Dekkers was concerned that the data from several large outsourcing engagements, underway in the telecommunications industry where FP-based productivity was being used, were overlooking the Zero FP paradox, and FP consultants at the time were advising that such projects should be done without payment (these were contracts for which the supplier payment was based on U.S. dollar per FP delivered) because “the numbers will even out (to be equitable).” This led to confusion, frustration, and some FP-based contracts were cancelled because FP-based productivity results were inconsistent [14]. The numbers did *not even out* and outsourced development teams started to rebel saying that FP-based contracts simply did not work. They were partially correct, when the only payment mechanism is based on FP delivery, Zero FP projects are an issue. The implication that non-functional requirements and technical requirements are of no value (because they deliver no FP) in such contracts creates issues as we further discuss later in this article.

To further explain this paradox and the ensuing issues, Buglione developed the “ABC schema” consisting of a three-tiered software requirements taxonomy, which was introduced and published in 2012 in *MetricViews*. [6] This schema was subsequently referenced in the joint IFPUG-COSMIC document about the NFR taxonomy [7]. Buglione and Dekkers further presented these concepts at the Capability Maturity Model Integration (CMMI) conference in Reston, Virginia, United States in 2018 [8].

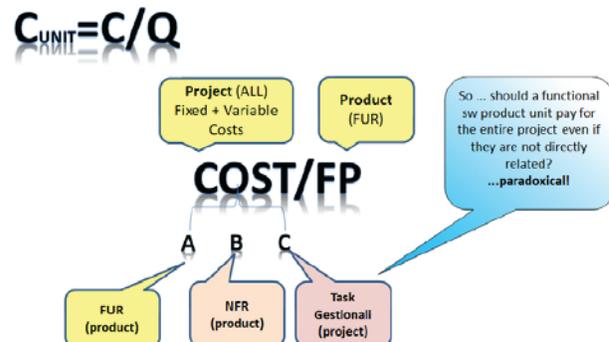
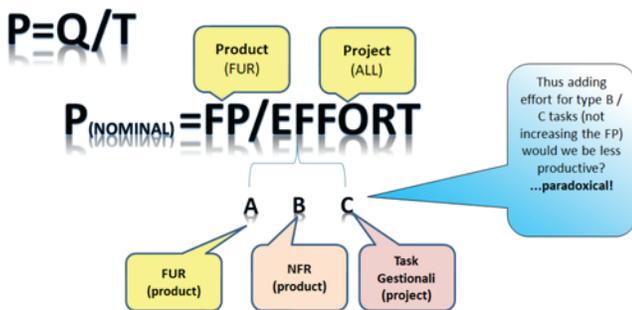


Figure 1: The 'Nominal' Productivity and the 'Nominal' Costs phenomenon [3]

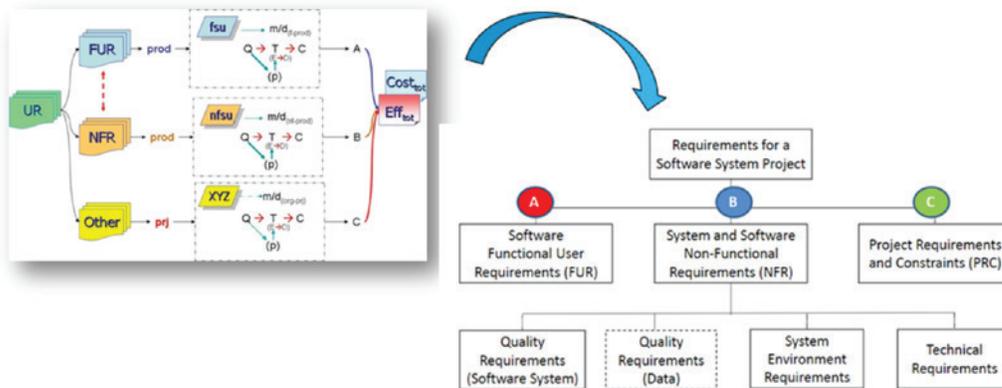
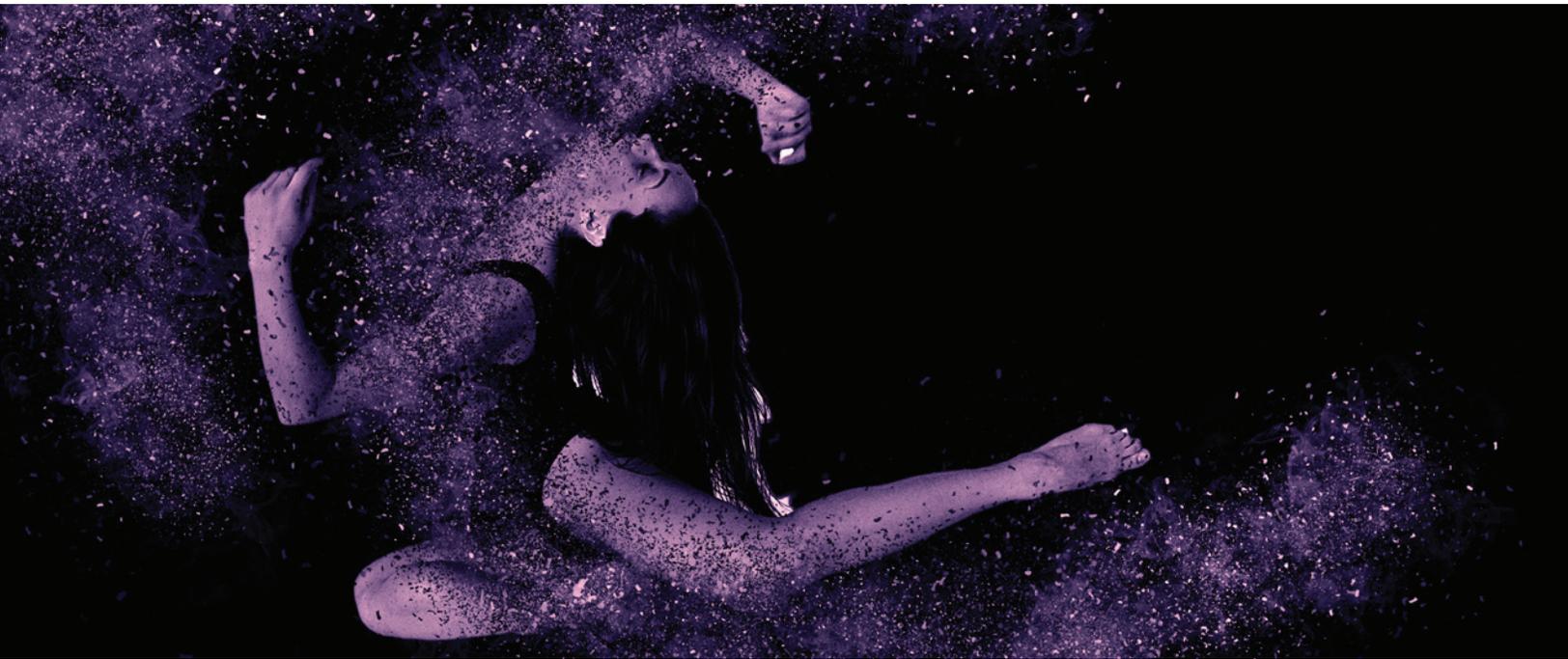


Figure 2: The ABC Schema in the IFPUG-COSMIC requirement taxonomy [6]

Is there a single formula for calculating software development productivity?

As discussed in [6] the problem with the “traditional” definition, FP-based productivity does not account for non-functional and technical work (i.e., it only considers the A in the ABC schema in Figure 2). Software development productivity, must account for *all* the work that needs to be done, thus addressing all three of the A, and B, and C, requirements as depicted in Figure 2.

This means that the software development productivity formula should evolve as shown in Figure 3, from the current (#1) definition to the (#3) definition (where “XYZ” stands for a possible, future software size unit of measure to quantify the technical or “org-project” requirements).

Today, most organizations that embrace FP-based productivity (nominal productivity) use the formula #1. Now that IFPUG SNAP and its equivalent ISO/IEC 32430 and IEEE 2430 have standardized SNAP Points as a non-functional unit of measure, organizations should be moving toward the formula #2, taking into account both the FP and SNAP project size. Someday, the industry will hopefully be able to develop and embrace both formula #2 and #3 below.

Of course, the more mature an organization is in its software estimating maturity, the more likely it will be able to collect the data to support the higher #2 formulation.

$\frac{FP_{FUR-prod}}{Effort_{prj}}$	$\frac{FP_{FUR-prod}}{Effort_{FUR-prod}} + \frac{SP_{NFR-prod}}{Effort_{NFR-prod / Org-Prj}}$	$\frac{FP_{FUR-prod}}{Effort_{FUR-prod}} + \frac{SP_{NFR-prod}}{Effort_{NFR-prod}} + \frac{XYZ_{Org-Prj}}{Effort_{Org-Prj}}$
Formula #1: Nominal productivity	Formula #2: Functional and Non-Functional Productivities (Level-1)	Formula #3: Functional, Non-Functional and Org-Project Productivities (Level-2)

Figure 3: Three formulas for calculating Software Development Productivity

Therefore, the “next frontier” for moving beyond FP-based productivity, and estimation is to stimulate organizations to:

- Prepare to use the formula #2: This involves measuring NFRs using SNAP and/or other NFR-based sizing units (e.g., using

measures described in such standards as ISO/IEC 25010 and 25012 about software Data Quality) and collecting commensurate data that support the NFR development in order to calculate distinct productivity values,

- consider different efforts from different requirement types within the project scope as part of the estimating process. Anything can be refined (see the formula #3 scenario) but taking one step at a time to embrace new thinking about estimating approaches, in an evolutionary manner, is usually the best.

From our collective years of software development and consulting experience, the authors note that the software development industry lacks both consistent and standardized guidance about how to measure the “software development productivity” of a project, not even mentioning the fact that the definition of “project” is itself variable and defined by being a measurable entity larger than its product(s)/deliverable(s).

Each project has different characteristics (and different software development productivity)

An important issue when estimating software development project effort (and cost) is to first identify what is the type of project:

- software development (DEV),
- operational improvement (OPS), or
- maintenance (SVC).

In the latter case, according to the ISO/IEC 14764 standard classification for software lifecycle processes: maintenance [11], and the IFPUG FPA counting practices manual version 4.3.1, it is important to define and standardize the type of maintenance being done. There are four standardized types of maintenance: adaptive, preventive, perfective, and corrective [12].

To help reconcile these three project types, an additional schema called the “123 Schema” was developed to harmonize the project type with the ABC schema presented previously [3]. The combination is depicted in Figure 4. The 123 schema helps to explain what the International Software Benchmarking Standards Group (ISBSG) Development and Enhancement (D&E) repository research has found: that a DEV project is more productive because it contains ABC (functional + non-functional + project constraints) requirements, while an SVC project is less productive because it contains proportionally more non-functional requirements (B type) and project constraints (C type) requirements relative to the FP size (A type requirements). Figure 4 also depicts graphically why corrective maintenance cannot be sized using functional sizing units (*fsu*): no FURs (A-type requirements) impacted.

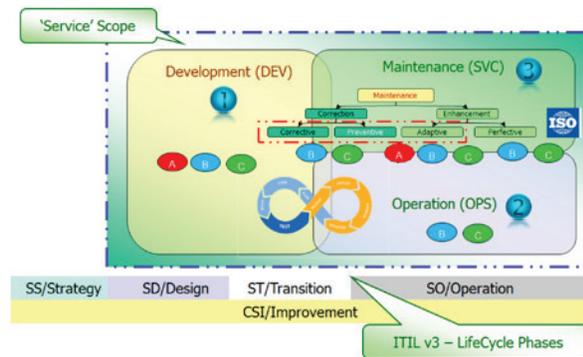


Figure 4: Project type (123 Schema) combined with Functional/Non-Functional/Project Constraints (ABC Schema) [3]

Note that adaptive maintenance may (or may not) include FURs, because this depends on each single project requirement’s classification using the ABC schema: it has an impact on the calculation of the percentage effort distribution by lifecycle phase and requirement type. Other benchmarking variables to consider for proper comparisons from the experience with ISBSG data are the year of the project (a 1996 Java project cannot be productive in the same way today, using tools/facilities as Eclipse, Junit, etc.), Industry sector, Organization type, Application Group, Application Type and—of course—the programming languages and technologies used.

Information and communications technology (ICT) contracts are often oversimplified because they apply a single nominal productivity value to an entire, multi-year contract, even though the contract consists of many different projects that span the entire “service scope” (which may/may not include the DEV part, and/or the OPS and the SVC part, as depicted in Figure 4). While the various contracting parties (customers and suppliers) prefer a simplistic approach, such oversimplification inevitably leads to contract disputes. Here’s why: Even if we could derive a single productivity rate to span all of the work being done across the 123 schema (ignoring the fact that DEV+OPS+SRV are different activities and work), a single FP-based productivity rates ignores the productivity paradox outlined in this article.

As described in the example with the outsourcing situation above, contracts that impose a single FP-based productivity rate are prone to problems, especially knowing that real, data-based software development productivity varies over time and by project type. As we’ve outlined, functional and non-functional development productivity is different and variable according to the nature of the User Requirements of each customer request. It follows then that it is unreasonable to consider a “static” breakdown of functional (FUR), non-functional (NFR) and PRJ (project constraints) requirements (i.e., per the “ABC schema”)

for the mid to long term, or for Agile projects, and across *all* the Sprints/iterations planned in a contract. However, this is the current way, at least in Europe, to handle such contracts, for example [9] the risk from “oversimplification” in using a single productivity factor compounds the FP-based productivity paradox as we outlined earlier with formula #1. There are multiple concepts that a software cost estimator must learn from Figure 4 and understanding and embracing the two schema (Figure 4) will take time and energy to gather and amass enough project and product data to support this approach to estimation. One small but possible, common-sense first step would be to stimulate organizations to gather their own project data to support their own productivity analysis over time. As with all approaches to software cost estimation, data-based regression always leads to better and more reliable estimates than theoretical models [10].

The difference between Productivity and Project Delivery Rate (PDR) in the ISBSG repository

Annually, since 1997 the ISBSG (www.isbsg.org) has released and managed best practices for benchmarking analysis in the form of a new D&E repository of completed software development projects. Guidance can be also found in the ISO/IEC 29155 Software Project Information Technology benchmarking family of standards and the creation of the D&E repository. Note that the “productivity” fields section of the D&E repository present productivity (size/effort) as a PDR. In actuality, these PDR fields are the inverse of what was introduced as nominal productivity [13]. Thus, considering the following Figure 5 (effort figures in person-hours), it could be confusing in a contractual document to read that project #11042 has a PDR equal to 38.2 while a nominal productivity of 0.026 if it's not clearly stated the formula and the units of measures applied for deriving an absolute value.

A	L	M	N	S	V	X
ISBSG Project ID	Primary Programming Language	Count Approach	Functional Size	Normalized Work Effort	Normalized PDR (ufp)	Nominal Productivity
11042	ABAP	IFPUG 4+	78	2976	38,2	0,026
11083	ABAP	IFPUG 4+	228	2877	12,6	0,079
11589	ABAP	IFPUG 4+	28	1037	37	0,027
11755	ABAP	IFPUG 4+	62	2640	42,6	0,023
12498	ABAP	IFPUG 4+	396	6051	15,3	0,065
13312	ABAP	IFPUG 4+	85	7800	91,8	0,011
15002	ABAP	IFPUG 4+	461	12149	26,4	0,038

Figure 5: Project delivery rate (PDR) and Nominal Productivity for a few sample projects delivered using the ABAP programming language [13]

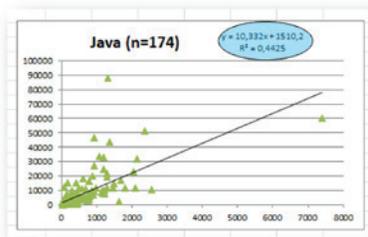
An example of how confusion can arise: consider the situation where there are multiple COBOL projects, for each of which the productivity was calculated at 0.5FP/person day. If PDR would be confused as synonymous with productivity, PDR would be equal to 2.0 (m-d/FP). One of the basic principles in any contractual arrangement is to clearly state any productivity conditions and parameters.

Zero FP projects: What to do? Let's use NFR-based sizing units (such as SNAP points)

As stated previously, not all maintenance projects involve functional changes (i.e., many maintenance projects address non-functional requirements only), and therefore would not be countable using FP, the so-called “Zero FP” projects [14]. As stated in the IFPUG CPM 4.3.1, typically only an adaptive maintenance project could include FURs, and be countable using functional size measurement. Thus, the inclusion of NFSU into a measurement program is fundamental for two main reasons:

- (1) it permits an organization to determine and maintain a second baseline (containing the non-functional size of projects) for asset management purposes; and
- (2) it permits the calculation of a non-functional productivity that can be used for estimating future projects.

Since sizing NFRs is still a rarity in software development, historical data and applying linear regression to analyze the data (e.g., using the general linear regression formula of $Y = AX + B$), is usually still reliant on the “X” cost driver being “the” functional size expressed in FPs. The higher the value of the “B” constant value, the worse the correlation for that project's data in terms of the determination coefficient (R²). Research shows the long-term negative effect of FP-based productivity: instead of considering that the effort on the y-axis is the summation of all three requirement types (A+B+C), adding “BC”-related effort to a project does not necessarily imply a loss of value. As stated earlier, such additional effort can be attributed to a “different” type of value that could be quantified using different sizing units, for each of the NFR and PRJ types of requirements.



Id	FP	Hrs	M/d	Prod(Nor ¹)
154	1188	8161	1020,125	1,165
155	1188	24639	3079,875	0,386
156	1198	33206	4150,75	0,289
157	1211	7840	980	1,236
158	1230	10152	1269	0,969
159	1285	11949	1493,625	0,860
160	1290	22045	2755,625	0,468
161	1300	19542	2442,75	0,532
162	1306	88555	11069,38	0,118
163	1374	43946	5493,25	0,250
164	1453	11683	1460,375	0,995
165	1484	15044	1880,5	0,789
166	1630	2303	287,875	5,662
167	1689	16896	2112	0,800
168	1800	12000	1500	1,200
169	2043	23271	2908,875	0,702
170	2091	11752	1469	1,423
171	2151	32170	4021,25	0,535
172	2351	51755	6469,125	0,363
173	2559	10528	1316	1,945
174	7400	60270	7533,75	0,982
Max	7400	88555	11069	5,662
Avg	523,64	6921	865	0,910
Median	287,00	3349	419	0,836
Min	21	167	21	0,055

Figure 6: Java projects from ISBSG D&E r2021: have they really a low productivity? [13]



The recommendation for software cost estimators is to consider multiple regression analysis with at least two (or more) sizing drivers, including both FP and SNAP Points. MS-Excel® can be sufficient for recording and storing the data. There are no requirements for specific statistical tools. Guidance about how to use the IFPUG SNAP is available from the www.ifpug.org site. Recently published as the ISO/IEEE/IEC 32430:2021 SNAP standard [15], the SNAP process outlines 14 sub-categories of Non-functional Requirements and how to evaluate each one to obtain the SNAP point size [16]. Alternatively, one could look at the ISO 25023 and 25024 suite of measures about software product quality and data quality for guidance. Of course, the definition of quality evolves over time based on newer technologies: it can be sufficient to look at the evolution of ISO/IEC standards from the initial ISO/IEC standard on software product quality (9126) published in the 1990s to the newest 25010:2021 version recently published about software product quality measures [17].

Currently, the data on actual application and results for SNAP Points is not prevalent in the software development industry. To stimulate the collection of SNAP and other NFR-related software development productivity and benchmarking data, ISBSG created a SNAP Data Collection Questionnaire (DCQ) on its website (www.isbsg.org). Collecting SNAP Point data, like its predecessor

FP, is a slow process. In a similar way to what happened 40 years ago when FPA was in its infancy, collecting SNAP and NFR-related data to create benchmarking repositories requires both time (project data collected over several years) and submitted projects. Corporations can start today with completed projects: collecting the data and then recording the NFR-related effort and SNAP size for software development projects. The next step would be to apply the new SNAP (NFR) productivity rates to refine and improve the overall estimation process. In addition, author Dekkers is involved in discussions with the University of Southern California researchers, consultants, and SNAP experts to incorporate non-functional requirements in the emerging COCOMO III software cost estimating model, currently under development.

How are NFSUs relevant to Agile/DevOps projects?

In Agile projects, the presence of NFRs is particularly evident at the beginning and end of any sprint/iteration as well as in some sprints, generating "Zero FPs," as stated previously. Applying a typical FP-based productivity analysis to an Agile project with many iterations or sprints could lead to a series of simplistic and potentially incorrect hypothesis. Because user stories in practice are often a collection of:

- functional user stories such as: “As a <user type> I want to <perform a function> so that I can <achieve a (functional) outcome>”;
- non-functional user stories such as: “As a <user type> I want the system to provide reliable and secure data so that I can <achieve a (non-functional) outcome>;
- project work or spikes (research work); and
- fixes to existing or in-development functions; only the functional user stories can be sized using FP.

Thus, the introduction of a NFSU of measure such as SNAP Points could improve an estimation process during sprint planning. In 2012, a presentation for an IFPUG Agile Interest Group session, Buglione introduced a method he coined as “the US2 (2nd generation-User Stories),” which included both FPs and SPs units of measure [18]. This sizing approach uses two sizing units to reduce a typical trend to under-estimate the effort needed per sprint/iteration. In a typical software development, the actual project effort is often higher than the estimates made solely on the basis of FPs. If the estimator, on the other hand, had considered also the NFR and PRJ requirements, and their associated productivities, as part of the estimation process, the estimated effort would be more realistic. This is the rationale for the question posed in Figure 7: would you like to play to the 15 or 16 puzzle? Any sprint should be allocated close to 100% of its limit but never exceeding it (15 game) allowing to potentially swap User Stories thru sprints/iterations. Realistically, one should slightly under-allocate the team effort, allowing the team to manage (and not suffer the consequences from) whatever kind of project risks [19]. Otherwise, we’d play to the 16 (or 17...) game with delivery slippages and an increasing Technical Debt carried over to the next sprints.



Figure 7: Would you like to play to the 15 (or 16) puzzle? [18]

Conclusions & Next Steps

Any functional size measurement method can be a very effective way to help organizations to improve the way they manage their requirements, with a focus on FURs for the software. Typically, the less we know about one object of interest, the less we are able to measure it: for this reason, there was ignorance (and avoidance) about how to measure NFRs in the past, even in organizations with established software measurement programs. As outlined in this article, there are several approaches and techniques to measuring NFRs including the IFPUG SNAP, among others. The introduction of such complementary measures for covering the “B” and “C” types of software requirements (NFRs and project requirements and constraints), as outlined here will facilitate the possibility of a win-win solution to realistic estimation to benefit both customers and suppliers, reduce the “cone of uncertainty” and facilitate more on-time and regular software deliveries.

The next step is just there...let’s walk and reach our next (business) goal!

“Computers are non-functional.” – Dr. Spock to Captain Kirk, *Star Trek IV – The Voyage Home*, 1986 🚀

REFERENCES:

- [1] Jones C., What are Function Points? SPR website, URL: <http://tiny.cc/mabkuz>
- [2] Cysneiros L.M., Yu E., Non-Functional Requirements Elicitation. In: do Prado Leite J.C.S., Doorn J.H. (eds) Perspectives on Software Requirements. The Springer International Series in Engineering and Computer Science, vol 753. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0465-8_6, 2004
- [3] Buglione L., Interpretare DevOps per misurare bene (e meglio) i progetti, ISIPM PMExpo conference, Presentation, October 2017, Rome (Italy), URL: <http://tiny.cc/4abkuz>
- [4] Buglione L., Some Thoughts on Productivity in ICT projects, WP-2008-02, White Paper, version 1.3, August 2010, URL: <http://www.semq.eu/pdf/fsm-prod.pdf>
- [5] McConnell S., Software Estimation: Demystifying the Black Art, Microsoft Press, 2006, ISBN 978-0735605350
- [6] Buglione L., The Next Frontier: Measuring and Evaluating the NonFunctional Productivity, IFPUG MetricViews, Aug 2012, URL: <https://www.ifpug.org/Metric%20Views/MVBuglione.pdf>
- [7] IFPUG/COSMIC, Glossary of terms for Non-Functional Requirements and Project Requirements used in software project performance measurement, benchmarking and estimating, v1.0, September 2015
- [8] Buglione L., Dekkers C., Improving Data Management WITH Functional Analysis: the FSM* way, Capability Counts 2018, May

2018, URL <https://stage.cmmiinstitute.com/capability-counts/2018>

[9] CONSIP, Guida al Contratto Quadro "Servizi Gestionali Integrati per le Pubbliche Amministrazioni", Lotto 4, ID SIGEF 1607, 2016 <http://shorturl.at/fuGY3>

[10] Briand L., Wieczorek I., Resource Estimation in Software Engineering, Technical Report, ISERN 00-05, URL: <http://tiny.cc/grjluz>

[11] ISO/IEC, International Standard 14764:2006 – Software Engineering – Software Life Cycle Processes - Maintenance, 2006

[12] IFPUG, Function Point Counting Practice Manual (CPM), release 4.3.1, January 2010, URL: www.ifpug.org

[13] ISBSG, Development & Enhancement (D&E) Repository, version 2021, URL: www.isbsg.org

[14] Dekkers C., FP Chaos – Making sense of Zero FP projects, Proceedings of the Software Measurement European Forum (SMEF) 2007, Rome (Italy),

[15] ISO/IEEE/IEC 32430:2021, Software engineering — Trial use standard for software non-functional sizing measurements

[16] IFPUG, Software Non-functional Assessment Process (SNAP) Assessment Practice Manual (APM), version 2.4, May 2017, URL: www.ifpug.org

[17] ISO/IEC, IS 25010:2011 – Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, International Organization for Standardization, March 2011

[18] Buglione L., Agile-4-FSM: Improving estimates by a 4-pieces puzzle, IFPUG Agile Interest Group, Presentation, May 17 2012, URL: <http://tiny.cc/zvdkuz>

[19] Buglione L., Gestione Agile: il Balloon Effect e il Gioco del... 16!, Tutto Misure, n.3/2021, URL: https://issuu.com/tutto_misure/docs/tm.3-2021/89

ABOUT THE AUTHORS



Luigi Buglione is a Measurement & Process Improvement Specialist at Engineering Ing. Inf. SpA in Rome, Italy. Luigi is currently the IFPUG Director for Sizing & International Standard (previously he was the Director for Conference/Education, 2013-2019), GUFPI-ISMA (the Italian Software Metrics Association) President and ISBSG (International Software Benchmarking Standards Group) Board member. He is a regular speaker at international Conferences on Software/Service Measurement, Process Improvement and Quality and is actively part of International and National technical associations on such issues. He achieved several certifications, included IFPUG CFPS, CSP and CSMS. He received a Ph.D in MIS and a degree cum laude in Economics. Further info and contact: www.semq.eu, <https://semqblog.wordpress.com/>, @lbu_measure.



Carol Dekkers, PMP, CFPS (Fellow), P.Eng (Canada) is the lead author of the ICEAA Software Cost Estimating Body of Knowledge (SCEBoK), and CEO of Quality Plus Technologies, Inc. She is best known for her software measurement expertise, and as the author of industry textbooks and articles. She is the past president of IFPUG and a member of the U.S. software engineering delegation to ISO/IEC standards. Carol is a CFPS (Fellow), PMP, P.Eng. (Canada) and a Certified Scrum Master. She holds a Bachelor of Science in Mechanical Engineering from the University of Calgary. Contact Carol by email at dekkers@qualityplustech.com or caroldekkers@gmail.com.

CERTIFICATION COMMITTEE

By Cinzia Ferrero, Committee Chair

The Certification Committee (CC) works daily to:

- Support IFPUG members taking the CFPS/CFPP (IFPUG FP) and CSP (IFPUG SNAP) exams.
- Help IFPUG members in applying the CFPS/CFPP Certification Extension Program (CEP) to maintain certifications without retaking the certification exam.

The CFPS/CFPP exam in English has recently been updated and enriched with new questions. During the exam, it is now possible to use an electronic calculator and a whiteboard.

Soon these upgrades will also be available for exams in Italian, Spanish and Brazilian Portuguese.

The CC also has been providing support to various other tasks and activities within IFPUG. The main tasks include providing support for:

- The Certified SNAP Specialist (CSS) task force with the Non-Functional Sizing Standards Committee (NFSSC). The task force works to enrich the currently available CSP exam and enable the creation of the SNAP Specialist exam. For those with active CSP certification, it will be possible to upgrade to CSS.
- KOSMA and Brightest to translate the CFPS/CFPP exam into Korean, which will be available soon.
- The SNAP Training Material task force with the NFSSC. The management process of FP training materials is under review as well.
- The IFPUG website redesign task force.

If you are interested in working with the CC, please send a volunteer form to ifpug@ifpug.org.

FUNCTIONAL SIZING STANDARDS COMMITTEE

By Daniel B. French, Committee Chair

Despite the continued challenges of working as a team during the pandemic, the Functional Sizing Standards Committee (FSSC) continues working on a wide variety of projects.

The new IFPUG-branded Simple Function Point manual was released in October and a follow-on multi-committee task force is being established to develop a training course, certification

exam and marketing and promotion materials to promote the new sizing methodology.

The FSSC is also working on a number of white papers and iTips including the joint project with the Non-Functional Sizing Standards Committee on Boundaries and Partitions, which will be published soon. Once that white paper is published, the committee will publish the Elementary Process white paper as well as the Mobile Applications white paper and counting Use Cases iTip. New projects are also underway including using function points with Agile development methodologies, cloud environments and Kanban continuous development.

The committee is pleased to announce the addition of five new members: Domenico Geluardi, Marcello Sgamma, Roberto Meli, Noemi Andre and Carlos Vasquez. We look forward to their valuable contributions to the committee and appreciate their support of the FSSC and IFPUG.

If you are interested in joining the committee or working as a non-member volunteer on any current or future projects, please complete the IFPUG Volunteer Form and send it to Michael Canino at mcanino@cmasolutions.com.

The committee appreciates the support of the IFPUG membership and is always looking for new projects to work on. We welcome suggestions from members on topics of interest. Please submit your suggestions to dfrench@cobec.com.

PARTNERSHIPS & EVENTS COMMITTEE

By Sushmitha Anantha, Committee Chair

The Partnerships and Event Committee (PEC) continues to arrange events for bringing our member base together for knowledge sharing.

On September 17, Luigi Buglione, Director of Sizing Standards at IFPUG, delivered a webinar “Measuring the Maturity and Capability of Measurement Practices in Your Organization.” The presentation examined the maturity of measurement programs and how to measure the reliability of measurement methods.

In collaboration with other committees, we celebrated the tenth anniversary of SNAP APM 1.0 on October 7 with a special webinar, “IFPUG SNAP—Past, Present, Future: 10 Years of Experiences.” During the event, senior member and Vice Chair of IFPUG Non-Functional Sizing Standards Committee Dr. Charley Tichenor and newly-elected board member Saurabh Saxena talked about various aspects of SNAP. Eligible SNAP Practitioners were offered a chance to extend their CSP certification by attending the event.

In early November, Lionel Perrot, the President of ASSEMI and IFPUG French Representative, and Jérémy Torrent-Bassin, Senior Consultant at Semantys, presented during the Knowledge

Café webinar. The topic focused on an innovative method for realizing business value through Function Point-driven performance management.

Pierre Almen presented the new IFPUG AD/M Benchmarking Certification Process during the Coffee Talk in early December.

We are very happy to share that Kiran Yeole from Amdocs has joined our committee and we look forward to working with him on interesting tasks. Special thanks to Kiran for his enthusiasm and for taking some time to contribute to our committee.

Please write to pec@ifpug.org with your suggestions for topics and speakers. We shall try to host them during our Coffee Talks. If you are interested in working with the PEC, please complete and send a volunteer form to pec@ifpug.org.

COMMUNICATIONS AND MARKETING COMMITTEE

By Julián Gómez, Committee Chair

The end of the year provides a timely opportunity to say thank you. While it is always a good time to say thank you, today is even better. Thank you to Diana Baklizky for those years serving our community as a Board Member and Communications & Marketing Committee (CMC) liaison. We at IFPUG will miss you a lot. Thank you to Antonio Ferre for your years as the MetricViews editor. Thank you Joe Schofield for accepting the challenge to serve as the MetricViews Editor.

I don't want to forget to say thank you to Sergio Brigido, our new CMC Board liaison. Together, and by partnering with the Partnership and Event Committee, we can make a better impact in our community.

We are excited because we have been able to consistently publish through social networks, carrying the news from IFPUG to all the people around the world. We incorporated new branding and imaging on social networks. Next year, we plan to launch the new IFPUG website.

We are working diligently to bring to life the new website, a place for all our community to exchange knowledge. We look forward to you seeing then, but you will have to wait.

Happy holidays. The coming new year has plenty of opportunities.

ISO STANDARDS COMMITTEE

By Carol Dekkers, CFPS (Fellow), Committee Chair

The ISO Standards Committee consisting of Talmon Ben-Cnaan, Steven Woodward, and I (Carol Dekkers) continue to work on several International Standards initiatives to provide value to IFPUG members and the international measurement community.

I remain active as the official IFPUG lead member on the U.S. Technical Advisory Group (US TAG) to ISO/IEC JTC1 SC7 Software and Systems Engineering standards—maintaining our strategic position as a U.S. domiciled member company in the standards arena. This involves attendance at quarterly U.S. meetings, participating in ongoing U.S. ballots and positions, and supporting international standards development beneficial to IFPUG. IFPUG pays an annual membership fee to be a part of the U.S. delegation, which has been instrumental in maintaining our market position with our international IFPUG standards (IFPUG 4.3.1 and SNAP).

Steven Woodward has been busy as Coordinator for the ISO/IEC JTC1 SC38 (Cloud) and SC7 (Software Systems) Liaison Coordination Group and other content for SC7, SC38 plus other standards groups such as IEEE, Cloud Security Alliance, itSMF and the Object Management Group. These are some highlights, excluding Non-Functional Sizing as Talmon will provide updates on this major topic area.

- ISO/IEC JTC1 SC 7 and SC 38 have been updating/reviewing several core standards, while also planning future work. This includes DevOps standards plus one subject area that I believe should be of interest to IFPUG—Artificial Intelligence (AI). Multi-cloud and cloud federation standards are also progressing.
- The development platforms and eco-systems today are much more complex than 10 years ago, where applying standardized methods, such as those developed by IFPUG help manage, organize, develop and maintain systems efficiently to align with business outcomes/objectives.

Talmon Ben-Cnaan has been busy as the new IEEE 2430 working group chair within the IEEE Standards Association for the emerging SNAP standard on behalf of IFPUG and IEEE. More information about this initiative will be included in future issues of *MetricViews*.

In this new year, 2022, measurement and IFPUG remain vital and relevant. Several IFPUG committees are currently seeking new volunteer members—and I encourage you to apply by

sending a note to the committee chair or to the IFPUG office offering your service. It is a great way to get involved in thought leadership and boost your own career and knowledge of software metrics. Currently, the IFPUG Certification Committee, the Partnership and Events Committee, and the new Business Applications Committee are looking for new members. I encourage IFPUG members to participate and become involved in multiple standards groups to influence and educate a wider audience of the value from effective software sizing the benefits/outcomes that can be realized.

Join a committee, meet new people, and make IFPUG the membership organization in which you remain proud to be a member! Wishing you continued success in 2022.

NON-FUNCTIONAL SIZING STANDARDS COMMITTEE

By Fabrizio Di Cola, Committee Chair

SNAP has become the ISO/IEC/IEEE 32430 full-use standard. This starts an important period for our committee. We are working on several tasks to complete everything necessary for a concrete and industrial use of the methodology.

- We support the Certification Committee in defining the CSS exam, which is an enabling step toward a certification extension process similar to that for CFPS.
- Together with the Certification Committee, we are working to prepare the documentation to support the training on SNAP.

For the next year, we have put into the roadmap some activities that give maximum visibility into how to use the SNAP methodology together with function points. This visibility will allow companies that have not yet adopted SNAP to perceive the advantage of using both metrics jointly. The first presentation we are preparing will be on the use of the subcategory "1.2 Logical and Mathematical Operations."

As written in the last report of the committee's work, we are working on two white papers that will help to define different counting scenarios with SNAP and the relationship between SNAP and function points.

Last but not least, all Non-Functional Sizing Standards Committee members, past and present, thank Talmon for the many years he served as Chair—years that saw the birth and growth of SNAP, under his leadership.

INTERNATIONAL MEMBERSHIP COMMITTEE

By Loami Barros, Committee Chair

IFPUG is undergoing several positive changes and so is the International Membership Committee (IMC).

We are pleased to share the following updates:

- Loami Barros from Brazil is the new Chairperson for the IMC. He will be replacing Saurabh Saxena who will continue as a Committee Member and Country Representative for the India region. We welcome Loami's new role and thank Saurabh for the last five years as Chairperson.
- Amir Sidek is now the Country Representative for the Malaysia region where he was volunteering. Our best wishes to Amir for the new role.
- The IMC is looking for an enthusiastic Brazilian Country Representative who will replace Loami. IFPUG members from Brazil are invited to volunteer for this role.
- The Board has asked the IMC to start a task force for managing Academics affairs. This task force will collaborate with universities and colleges around the world that use Software Estimations/Function Points in their curriculum.
- The IMC has been providing support to various other tasks and activities within IFPUG. The main tasks include providing support to:
 - o ISBSG/IFPUG reporting task force
 - o Benchmark Certification task force
 - o IFPUG website redesign task force
 - o The approvals and verification for the CPM 4.3.1 IFPUG manual translation into French.

Finally, the IMC has been acting as the primary contact point for all IFPUG-related queries and engages IFPUG members so that they continue to benefit from their memberships. We are more than eager to assist you with all IFPUG-related queries.

BUSINESS APPLICATIONS COMMITTEE

By Saurabh Saxena, Committee Chair

IFPUG is pleased to announce the launch of the new Business Applications Committee (BAC) and Pierre Almén as the new chair.

Pierre has 40+ years of IT experience with 20+ years of experience as an IT consultant in different roles, such as project leader/manager, test leader and management consultant. Since 1998 he has worked with benchmarking and software measurement as an employee and then in his own company since 2008.

Pierre was the first in northern Europe to become a Certified Function Points Specialist in 1994 and he has used this method internally at IBM since 1984 and at major well-known companies mainly in the Nordic region. Pierre has often been engaged as a function point trainer, as a support when starting using the method and as a counter and reviewer of function point counting result. Pierre is a committee member and project leader within IFPUG and is the president of ISBSG.

The purpose of the BAC is to contribute to C-level and management decision-making using quantitative approach.

The function of the BAC is to encourage and support the development and definition of standardized metric-based business practices utilizing unit of size based on IFPUG sizing standards such as:

- Value-based contracting
- Value-based price modelling
- Estimating models
- Cost modelling
- Measurement
- Productivity models and analysis
- Productivity and quality indicators
- Benchmarking analysis

To start with, the new committee will be responsible for the following two major programs:

IFPUG AD/M Benchmarking Certification

Having accurate and reliable data is critical for companies to survive in today's highly competitive business world. Companies must ensure they are focused on the appropriate goals by comparing their performance against the industry. IT organizations achieve that goal by using highly reputable benchmarking data and reports.

The IFPUG AD/M Benchmarking Certification represents a standard method through which IFPUG affirms that a benchmarking service provider has fulfilled the requirements deemed necessary to be competent to conduct an AD/M benchmark analysis, through the investigation of evidence based on criteria that were defined in applicable ISO/IEC 29155 tasks and activities.

The following items describe the core benefits that an AD/M benchmarking provider can obtain from the IFPUG AD/M Benchmarking Certification:

- Achieve a best-in-class corporate level benchmarking qualification
- Establish standard for requirements, guidelines and qualifications for AD/M benchmarking
- Apply a consistent approach used by both benchmarking consultancy companies and software suppliers
- Define the measure of the ability and the quality of their benchmarking process

Analytics of the ISBSG Function Point Database

The purpose of this program is to provide IFPUG users with helpful function point-based statistics and benchmarks from the August 2021 version of the ISBSG database based on IFPUG function point data submitted by volunteers worldwide. For those organizations which are without developed databases, these statistics and benchmarks can serve as useful benchmarks and expectations.

If you are interested in applying to join the BAC or have questions, please send a request to ifpug@ifpug.org. 



191 Clarksville Road
Princeton Junction, NJ 08550 USA

Contact IFPUG Headquarters at +1-609-799-4900
or ifpug@ifpug.org.