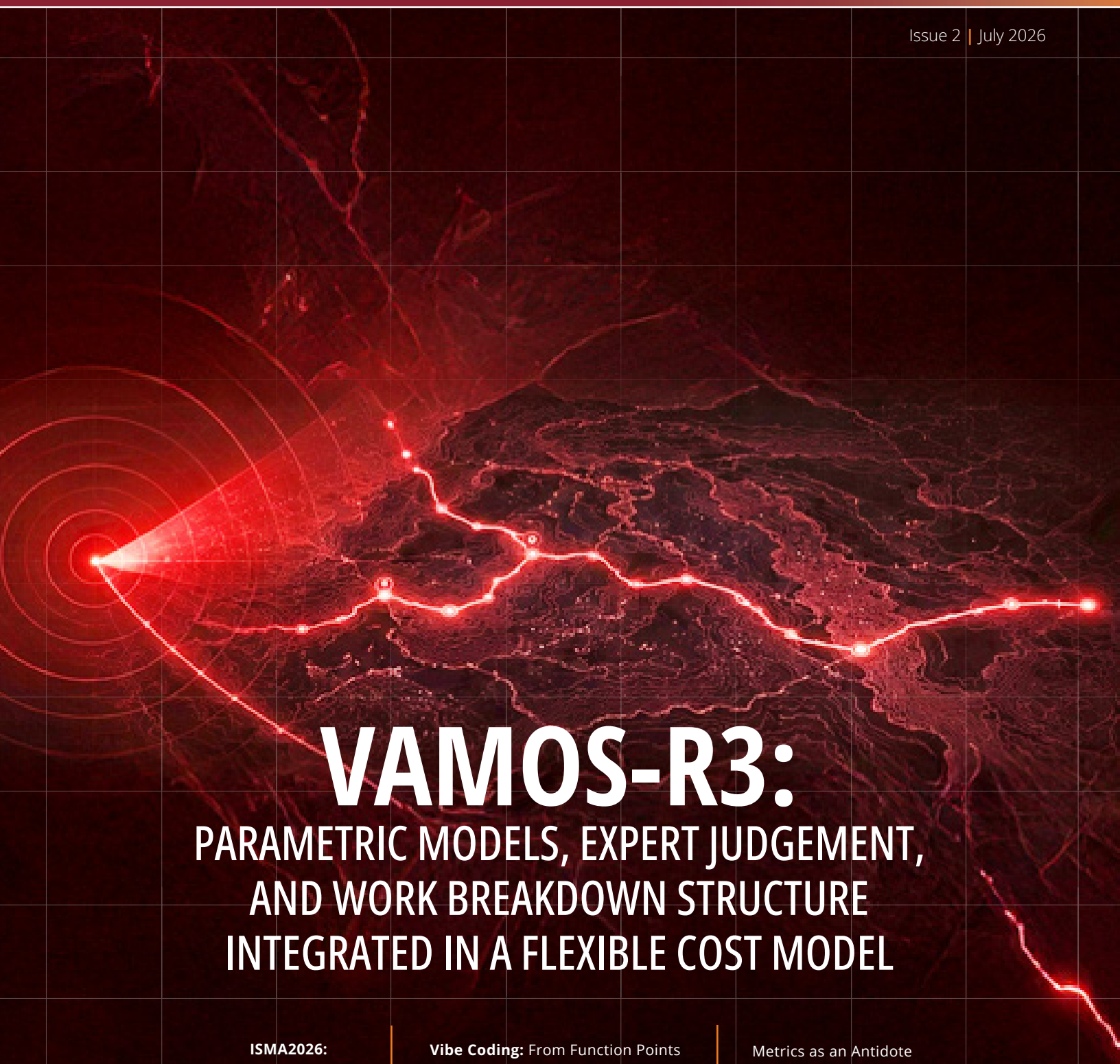
MetricVIEWS

A PUBLICATION OF THE INTERNATIONAL FUNCTION POINT USERS GROUP

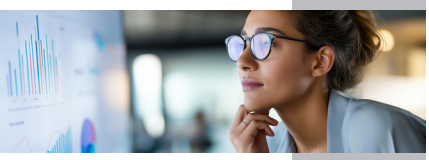
Issue 2 | July 2026



VAMOS-R3:

PARAMETRIC MODELS, EXPERT JUDGEMENT, AND WORK BREAKDOWN STRUCTURE INTEGRATED IN A FLEXIBLE COST MODEL


IFPUG
International Function Point
USERS GROUP



COMMITTEE REPORTS

IFPUG is committed to publishing timely articles related to function and non-functional software measurement in every issue of *MetricViews.* While each article is reviewed for relevancy and clarity, articles, especially those that are innovative and thought-provoking, are not necessarily endorsed by IFPUG.

BOARD OF DIRECTORS:

An Invitation to Shape the Future of Software Measurement: Join IFPUG Committees & Share Your Success Stories

Dear IFPUG Members,

I hope you are doing well and thriving in your professional journeys. As we continue strengthening IFPUG's global role in software measurement, I want to extend a personal invitation to each one of you to become an active contributor to our community's future.

IFPUG has always been built on the passion, expertise, and collaboration of its members. Today, more than ever, we need your voice, your experience, and your leadership. Whether you are a practitioner, consultant, researcher, educator, or organizational leader, your participation can make a meaningful difference.

Join an IFPUG Committee

Our committees—ranging from Functional Sizing Standards, Non-Functional Sizing (SNAP), Communications & Marketing, Education, Conferences, and more—play a vital role in advancing our mission. These committees are where ideas evolve into standards, where best practices are shaped, and where the next generation of measurement professionals is inspired.

By joining a committee, you will:

- Contribute to the evolution and global adoption of Function Points (FP) and SNAP.
- Collaborate with international experts and thought leaders.
- Expand your professional network and visibility.
- Stay at the forefront of emerging trends such as AI-driven estimation, automation, benchmarking, and productivity analytics.
- Influence the strategic direction of IFPUG and the industry at large.

Your time, whether small or significant, can help IFPUG grow as a global authority and collaborative hub for software sizing and estimation.

Share Your Success Stories – Inspire the Community

We also encourage you to share your stories of how Function Points and SNAP have created value in your projects and organizations. Your experiences—whether improving transparency, supporting fair pricing, optimizing budgeting cycles, enabling predictive analytics, or enhancing software quality—can inspire others around the world.

We are eager to highlight these stories through:

- Blogs and case studies
- Webinars and knowledge-sharing sessions
- Conference presentations
- Social media spotlights
- Member newsletters

Your success is the community's success. By sharing your journey, you help demonstrate the impact of quantitative sizing in today's digital and AI-driven landscape.

Together, Let's Strengthen the Future of Software Measurement

IFPUG stands at an exciting moment. With rapid advancements in AI, automation, and global digital transformation, the world needs reliable, unbiased, and standardized methods to measure and estimate software more than ever. By participating, you help ensure that our standards remain relevant, modern, and globally adopted.

If you are interested in joining a committee or submitting a success story, please write to ifpug@ifpug.org.

Before closing, I would like to take this opportunity to invite all our members to attend the ISMA 2026 Hybrid Conference, which will be held in São Paulo, Brazil, on 22–23 October.

Your participation will not only strengthen our global IFPUG community but also provide a valuable opportunity to engage with industry experts, exchange ideas, gain new insights, and learn about the latest developments in the field.

I look forward to meeting many of you in person in São Paulo and welcoming those who join us virtually. Together, let us make ISMA 2026 a memorable and enriching experience for all.

Thank you for being part of the IFPUG family. I look forward to your active participation and leadership.

Warm regards,

Saurabh Saxena
President, IFPUG

Over the last hundred years or so, okay, maybe a few less, IFPUG has published *MetricViews* in the spring and in the fall. I suppose one could argue that this *MetricViews* issue was released toward the end of spring, but this semi-annual schedule is primed for change. A new quarterly newsletter will provide twice as many opportunities for the Board, our active committees, and our vibrant membership to share the latest developments in the constantly expanding dimensions of computing and software measurement. These changes are often attributed to massive increases in data collection, AI, cybersecurity, robotics, hybrid clouds, and, of course, the unceasing evolution of software development.

You will find four relevant articles in this issue to complement most of these very trends. As examples, Tetyana reminds us that trust is a key component of vibe coding. Carlos and Thiago introduce the integration of estimation with an agile workflow. This issue includes the first segment of their article to be completed in subsequent IFPUG publications. Roberto explores a flexible cost model incorporating parametric modeling and work breakdown structures. Alcione strives to reduce burnout and enhance the mental health of the IT workforce. You will likely find other hidden gems of value from these contributors.

Most likely, by the time you are reading this, ISMA 2026 has just happened, the voting for the changes to the bylaws is complete, the inaugural quarterly newsletter is being developed, and the Board elections are being planned. It's a busy time for IFPUG and an exciting time in the IT world. Continuing to navigate that uncertainty together makes as much sense as ever before.

As with every issue, I extend my thanks to the authors for their thoughtful ideas and to CMA for their contributions to the content quality in *MetricViews*. Along with our President's message and updates from our geographically diverse committee members, we're highlighting the *international* in IFPUG.

Be well, stay well.

Joe Schofield
Editor, Past President, Honorary Fellow

The ideas expressed in MetricViews are those of the authors and may not reflect the consensus of the software measurement community

ISMA 2026 Virtual

Powered by IFPUG

By Thiago Monarca Leone D'Durand

The IFPUG Partnerships and Events Committee (PEC) hosted the ISMA 2026 Virtual conference. As an event dedicated to Software Measurement and Estimation, the conference brought together an impressive global audience of 252 members to discuss the evolution of software metrics, modern architectural boundaries, and the integration of artificial intelligence (AI) into software estimation.

Broadcasted concurrently across international time zones in Eastern Time (ET) and Central European Time (CET), the conference served as a key Certification Extension Program (CEP) eligible event, offering Certified Function Point Specialist (CFPS) and Certified Software Non-functional Assessment Process (SNAP) Specialist (CSS) extensions to dedicated participants.

The technical program featured an elite lineup of international experts who successfully balanced emerging technological trends with traditional core measurement metrics.

Julian Gomez opened the event by setting the stage for modern estimation challenges and later concluded the day by summarizing the key insights gained, firmly anchoring the value of the community's shared knowledge.

Luigi Buglione delved into foundational metrics, exploring the structural concept of "control" and analyzing its specific operational impacts on executing accurate Function Point Analysis (FPA) counts.

Andrés Gutiérrez provided a visionary roadmap for the future of enterprise structures, focusing on the workflows required to scale Generative AI safely within software engineering metrics.

Carol Dekkers & Daniel B. French delivered a critical reality check on automation, emphasizing that human expertise and standardized sizing rules remain vital because technology alone cannot fix flawed baseline logic.

Roberto Meli shared vital project management strategies, detailing how to leverage the scope concept in FPA and Simple Function Point (SFP) methods to optimize cost and effort estimations.

Harold van Heeringen's Framework for Output-Based Measurement addressed modern development workflows, presenting the NPR 5333 standard as a powerful framework for output-based measurement in Agile delivery environments.

Ifeanyi Echereobia investigated complex modern architecture, mapping out specific FPA boundary challenges encountered when measuring interconnected, API-dependent financial platforms.

Marcello Sgamma explored the intersection of user experience and non-functional requirements, demonstrating how to align modern user interfaces with the SNAP.

We would like to express our deepest gratitude to all who joined us virtually from all corners of the world, bringing their energy, perspective, and insights to the interactive discussions.

A special thank you goes to our incredible lineup of expert speakers for sharing their pioneering research and volunteering their time to help the metrics community stay at the cutting edge of industry compliance. Thank you all for making ISMA 2026 Virtual a resounding success, and we look forward to seeing you in Brazil at ISMA, in October.

About the Author

Thiago Monarca Leone D'Durand has more than 10 years of experience in IT and Business as well as expertise in agile methods, cybersecurity, growth hacking, strategy, technology, digital, coaching, and training. With a multidisciplinary background, he has applied his IT and business skills in various sectors. He holds 15+ international certifications in various domains, including Software Measurement. A member of IFPUG since 2009, and having passed the rigorous CFPS exam several times, he became a member of the CEC, PEC, and CMC committees where he contributed broadly to the IT community in several ways, such as helping organize webinars and the ISMA conference, evaluating the work of others submitted to IFPUG, and translating SFP manuals and tools.

SAVE THE DATE
ISMA 2026

HYBRID CONFERENCE

OCTOBER 22–23, 2026

SAO, PAULO, BRAZIL

EVENT SPONSOR

Quanter
SMART AI ESTIMATION

HOST SPONSOR

tiMÉTRICAS

VAMOS-R3:
PARAMETRIC MODELS, EXPERT JUDGMENT, AND WORK BREAKDOWN STRUCTURE INTEGRATED IN A FLEXIBLE COST MODEL

By Roberto Meli

Abstract

*Software measurements serve various purposes related to information system governance. Among these, a crucial goal for organizations is **estimating the effort required for new software production or maintenance**. From effort estimation, it's relatively straightforward to derive monetary costs by considering the unit rates of human resources involved in the work and their expected professional mix for different parts of the process. This aspect is relevant both for software producers and for those who need to evaluate or benchmark software purchases in the market.*

Numerous approaches and models exist to aid estimators. Fundamentally, however, two main types of models are commonly used:

1. **Parametric Models:** These models rely on **objective measurements of software products and processes**. They use actual data to estimate costs. For instance, they consider factors such as software functional size, lines of code, or implemented features. A well-known parametric model is **COCOMO** (Constructive Cost Model), which estimates costs based on a specific project size and other parameters.

2. **Analogous Estimation:** This approach is based on **experience** and often involves **expert judgment**. Estimators use comparisons with similar projects or past experiences to estimate effort. Some methods may be used to improve personal estimation performance like the **Delphi** or the **Shang methods**.

VAMOS-R3 integrates both these approaches within a **Work Breakdown Structure (WBS)**, a project management technique used to break down a project into smaller, manageable activities. In practice, **VAMOS-R3** acts as a **superset** of many existing models and methods. It also employs metrics such as **FPA** (Function Point Analysis), **SFP** (Simple Function Points), and **SNAP** (Software Non-functional Assessment Process).

In summary, **VAMOS-R3** is a model in the public domain that provides an open and customizable approach (framework) for effort and cost estimation throughout the software lifecycle. VAMOS-R3 combines objective measurements and expert judgment through the WBS structure. It serves as a flexible method for organizations assessing production or maintenance costs of software applications

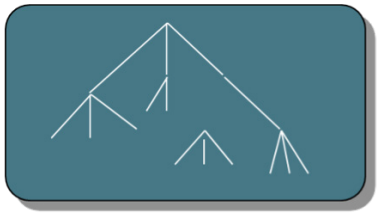
Premise

VAMOS-R3 is the acronym of VAlorization MOdel for Software. R3 indicates the three types of requirements defined in the COSMIC-IFPUG joint document called "Glossary of terms for Non-Functional Requirements and Project Requirements used in software project performance measurement, benchmarking and estimating." See also the international standard ISO/IEC 14143-1:2007: the Functional Requirements (FUR), the Non-Functional Requirements (NFR), and the Project Requirements & Constraints (PR&C).

The model is used by numerous Italian Public Administrations (Ministry of Culture, Aria, Innovapuglia, SIAG, Liguria Digitale, CINECA) and their ICT suppliers.

The underlying assumption consistent with the most widespread Project Management methodologies is that all the work must be carried out in a project represented by the Work Breakdown Structure (WBS), which therefore defines the entire scope of the project.

A WBS answers the question: what must be done to achieve the project objectives? It consists of enumerating all the necessary tasks and activities according to a structure organized by a detailed hierarchy. Graphically, it is usually represented as an upside-down tree, i.e., with the root at the top and the branches at the bottom, but it can also have a lateral representation. The WBS divides the project into gradually smaller work components and integrates their objectives, responsibilities, costs, and deliverables. Each element of the WBS has the characteristics of an activity to be completed. What changes between the various levels of the hierarchy is the granularity of the actions. You can go from macro activities or high-level production process phases, composed of dozens or hundreds of sub-activities, to the leaves of the structure called Work Packages (WP). The WPs can be considered elementary and no longer usefully decomposable, at least at the level of overall project planning. The WBS can be used for predictive purposes of the overall project effort because the structure enjoys an important property. That property enables certain scalar variables to be summed by aggregation going up the structure and to be subdivided by distribution going down the structure.



In bottom-up forecasting mode, the project's overall effort estimate is simply the sum of the efforts for all the leaves of the structure. The problem with this way of proceeding, however, is that when an estimate is requested, usually, detailed requirements are not known. The leaves of the structure (WPs) are too rough to allow a direct analogue estimate, called expert judgement. In other cases, the work of estimating hundreds or thousands of WPs can be very onerous and not very motivating. For this reason, parametric cost models have been developed. Based on the value assumed by some input variables to the model, they allow obtaining an indirect estimate relating to a macro aggregate of activities. In one fell swoop, an estimate of a large set of activities typical of a certain production process (in some cases even of the entire project) is obtained. This can happen because empirical studies have shown that the work necessary to carry out these macro activities is well correlated with these input variables. This is the case, for example, of effort models based on Function Points (FPA or SFP). Using data collected in the ISBSG database, we can develop models that predict project effort based on the estimate or measurement of FP that will need to be developed or maintained.

The crucial question then becomes: are we sure that all project work is always proportional to this size variable? If not, what is the share of effort that can be correlated to FP on average? The answer to the first question unfortunately, is partially negative. Most empirical studies on heterogeneous data, such as those of the ISBSG, agree that a measure of size in FP to be developed or maintained can explain, in statistical terms, only a part (sometimes a very good part) of the cost estimate for a project. The IFPUG Business Applications Committee (BAC) published "Analytics of the International Software Benchmarking Standards Group (ISBSG) Development and Enhancement Repository" in 2022. The R2 the BAC calculated was 0.73. This means that only 3/4 of the effort variability can be explained by the FP variability; the other quarter must be explained by other factors (non-functional, project / process / constraints requirements). These empirical results are, however, easily explainable if we consider that not all the work to be done in a project is dedicated to the creation of transactional and data functions. There are WPs that have the same value for both the realization of 100 FP and 1000 FP. For example, the preparation of an environment to test the software performance and response times of an application is independent of how many features we will have to test on it. Pretending that this effort is estimated by a parametric model based on FPs is a mistake.

There are also non-functional requirements that change the type of influence depending on the way we implement them. Let's take an example related to the usability of an application, a characteristic that belongs to the set of quality factors and therefore non-functional requirements. Let us assume that to meet a usability goal, some specific work on each transaction is needed, related to the video mask, colours, field placement, font size, and interaction method. Therefore, the effort required by this non-functional requirement will depend on the number of transactions and, consequently on their Function Points size. If we may obtain the same final usability result with the use of a style sheet, a template or an automatic formatting procedure, then the usability effort is no longer proportional to the transactional FPs. Instead, that effort is proportional to the difficulty of creating the style sheet, template, or automatic

process. Clearly, now, the non-functional requirements may impact the effort estimation in different ways. The following figure shows the type of more or less "viscous" behaviour that a non-functional requirement can have with respect to functional requirements and their "size."

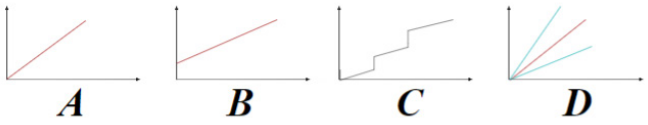


Figure 1 - Examples of NFR r PR&C) impact on total cost

A. NFR/PR&C do not affect the total cost; the cost is proportional only to the Functional Size

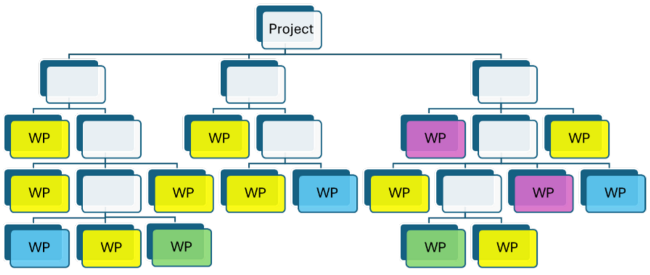
B. NFR/PR&C generate a fixed cost to be added to the cost proportional to the Functional Size

C. NFR/PR&C generate fixed costs that are added to the Functional Size-dependent cost in a stepped manner (determined by size thresholds, for example)

D. NFR/PR&C influence the Product Delivery Rate (PDR) by increasing or decreasing it

How can we integrate these lessons learned into a single model or framework?

It is possible to integrate different parametric models (based on FP and SNAP, for example) with direct WBS estimates for those parts not covered by any parametric model. For example, in the WBS graph shown in the figure, the yellow parts could be estimated by a cost model based on FPs, the light blue parts by cost models based on non-functional metrics, like SNAP, and the green and purple parts estimated directly with the judgment of experts. The overall effort estimate will be given by the sum of the different parts as long as they are disjoint or are not covered by more than one model at the same time.



Brief Description of the VAMOS-R3 Model

Function Points are a class of functional metrics for software that play an important role in the governance of software development and maintenance processes. In recent years, there has been a growing awareness that the effort required to produce software and, consequently, its cost cannot be determined by evaluating the Functional Requirements only. In fact, the Non-Functional Requirements and Project Requirements & Constraints contribute to this enhancement in a direct and indirect way. Therefore, the proposed Valorisation Model takes note of this awareness and provides the basis for the use of an appropriate cost model in the market context.

The image illustrates how the various approaches and parametric models mentioned can be integrated to produce a complete and appropriate effort estimate.

In Figure 2 we see that some NFRs and PR&Cs are covered in Service Level Agreements (SLA) which define minimum thresholds

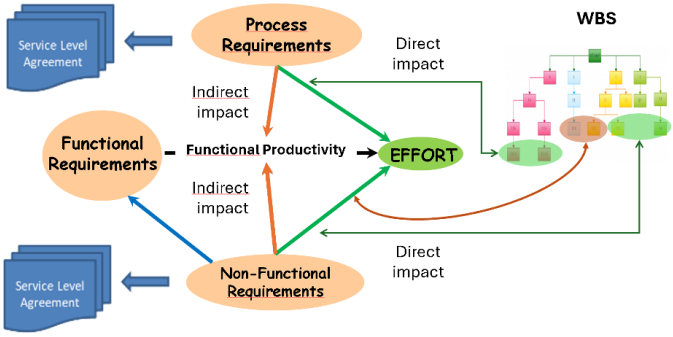


Figure 2 – Effort model underlying VAMOS-R3

that must be respected. But these requirements, sometimes, must be modulated according to needs and can participate in the cost model, determining the modulating variable effort depending on the different levels required. For example, software reliability is a quality factor that expresses the degree of absence of errors present in the code. We all want the codes to be defect-free, showing maximum reliability, but are we sure that we would always like to accept the high costs needed to ensure a priori and to guarantee a posteriori that there are no errors? Software on which human lives or the economic destinies of a nation depend cannot have the same degree of reliability as software for strictly personal use and whose shortcomings are of little consequence. Therefore, reliability is a (non-functional) quality requirement that cannot always take on the same value in an SLA. Instead, it must be modulated according to the business criticality, and that influences the production cost in a variable way, exactly as envisaged in the COCOMO model.

As already mentioned, according to the IFPUG analysis reported, the measurement of Functional Requirements (FUR) alone can statistically explain only around 75% of the effort value of a project (determination index R2 = 0.73).

The remaining part is associated with the influence of other types of requirements, including Non-Functional Requirements (distinguished from ISO in Quality Requirements and Technical Requirements) and Project Requirements and Constraints (also called Production Process Requirements, i.e., expressed on the production factors of a project). These requirements influence the relationship with effort in two different ways: proportional to FPs (indirect impact) and not proportional to FPs (direct impact).

In the first case (indirect impact), these requirements are well represented by productivity adjustment factors that take on

values around unity, and which are all multiplied together. This approach was adopted by the COCOMO model. For example, a non-functional maintainability requirement can result in a 10% increase in effort whether the software accounts for 100 FPs or 1000 FPs. A coefficient equal to 1.1 applied to the size-based effort function [Effort = f (FP)] allows this aspect to be considered.

Non-functional or process requirements that impact the work to be performed in a way that is not proportional to the functions (direct impact) must be considered separately. It should be estimated directly in person-time or using other types of parametric models and added to the previously obtained value. Here, several attributes of the SNAP method are useful, and Non-Functional Productivity (orthogonal to the functional productivity) may be used to make this part of the estimation.

Figure 2 also shows how some non-functional requirements, described at a very low level of granularity (detail), can be decomposed into detailed NFR but also into some functional requirements that will increment the overall FP size. For example, the management of user types, roles and responsibilities helps to achieve a good level of security of operations, but they are definitely FURs to be measured.

A parametric cost model (functional or non-functional) can allow you to estimate entire parts of the project WBS using a few input metrics and valuing them appropriately. It may not be possible to apply parametric cost models because there is a lack of sufficient empirical evidence of correlation between effort and deliverable measures. In such cases, it is necessary to break down the work in the WBS to levels of granularity that can allow a direct and reliable estimate by a subject expert. There is an international standard that regulates the WBS: ISO 21511:2018. The standard does not prescribe an absolute level of granularity (in-person time, for example) but links granularity to the "manageability" of the terminal element of the WBS (i.e., the Work Package - WP). This often leads to incorrect and incomprehensible practices. It is impossible to make a direct, intuitive estimate of a WP involving hundreds of person days of effort and which lasts six calendar months. Miller's law suggests that human beings can simultaneously hold attention to no more than nine elements at a time . The difference between 10 and 15 person days for the execution of an activity is still perceptible to the human mind, while between 125 and 130 person days it is no longer perceptible. This leads to the prescription of sizing the WPs around 15-20 GP at most, especially if they must be verified as appropriate by an auditor.

In the case of configuration interventions on packaged commercial software, IFPUG suggests using the SNAP Sub-category 1.5 "Delivering Added Value to Users by Data Configuration." In the case of a mixed intervention that includes configurations, customization and custom integrations, all the separately identified components must be added.

[1]George A. Miller empirically observed, in phycological experiments, that the number of objects the average person can hold in working memory is about seven. It was put forward in a 1956 edition of Psychological Review in a paper titled "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information"

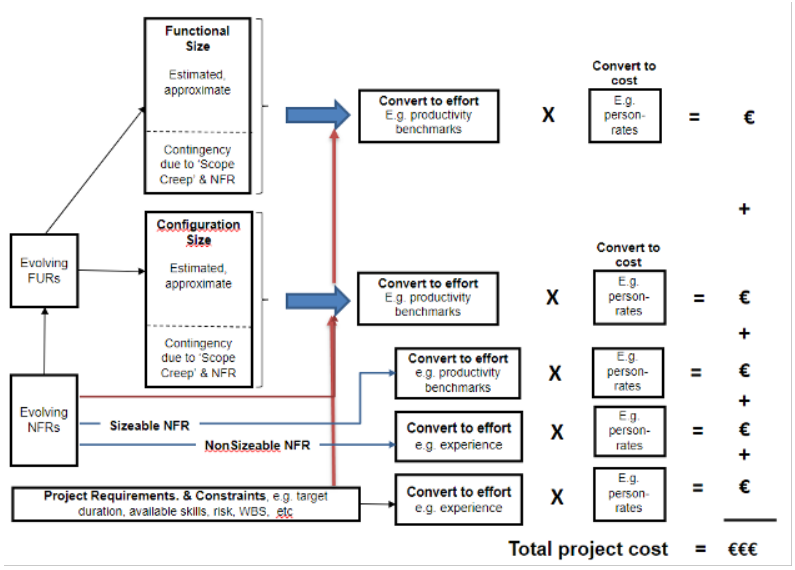


Figure 3 – Model VAMOS-R3

In this way and considering all the visible factors, we arrive at the most representative function of the relationship between the variables considered in the following:

→ **Effort = FE + CE + PE + Σj NFRDEj+ Σk PCRDEk**

Where:

***FE=** Functional Effort = FP*AvgFP* FGAF*

***FP=** Function Points (FPA or SFP)*

***AvgFP=** Average Functional Productivity (WorkHours/FP)*

***FGAF=** Functional General Adjustment Factor*

***CE=** Configuration Effort = SP* AvgCP* CGAF*

***SP=** SNAP Point (Sub-category 1.5)*

***AvgCP=** Average Configuration Productivity (WorkHours/SP)*

***CGAF=** Configuration General Adjustment Factor*

***PE=** Personalization Effort = FP* AvgPP * PGAF*

***AvgPP=** Average Personalization Productivity (WorkHours/FP)*

***PGAF=** Personalization General Adjustment Factor*

***NFRDE=** Non-Functional Requirements Direct Effort*

***PCRDE=** Process & Constraint Requirements Direct Effort*

In summary, we must:

*a) determine the effort component due to the functional imension of the software: Functional Requirements estimated or measured in FP (FPA or SFP), multiplying FP by the average Productivity (FP*AvgFP);*

[2]The entire set of formulas may be used with FPA values or SFP values but they shouldn't be mixed: FPA or SFP but not both of them in the same set.

b) determine the functional effort corrector due to NFR and PR&C (FGAF);

c) determine the effort proportional to the functions (FE) as the product of the two variables a, b;

d) add COTS configuration effort (CE) using a similar approach to steps a, b, c;

e) add COTS standard report customization effort (PE) using a similar approach to steps a, b, c;

f) add the direct effort, due to the NFR and PR&C, which is not affected by the proportionality to the functions;

g) sum all the effort components.

Calculation of Average Productivity

To estimate the average productivity for the creation of the software (development or functional evolutionary maintenance), the ISBSG database or other reliable sources in terms of quality and independence of the data collected can be used.

Calculation of the General Adjustment Factor (GAF)

The General Adjustment Factor is a factor that considers the general production conditions of the project being measured. Specifically, it considers the influence of non-functional and project / process / constraint factors on productivity. It can be applied to average or nominal productivity of functional (FGAF), configuration (CGAF), or personalization (PGAF). This factor is calculated starting from the classification of some of the degrees of influence of several general characteristics. Consider for instance, COCOMO II model, Caper Jones data or customized production factors, all of which may undergo updates over time depending on the evolution of scientific evidence. This factor serves to further characterize the project to determine a forecast closer to reality.

Activities Included in the Effort Forecast Proportional to Functionality

The effort forecast derived from the component proportional to the functional size expressed in formula as

FP*AvgFP* FGAF

includes the work necessary to implement the following activities:

Basic Activities of the Production Model

1.) Development coordination: project management activity linked to the coordination of the complete development cycle and measurement of the software

2.) Requirements Management

3.) Functional analysis

4.) Technical Design

5.) Test plan and test design

6.) Implementation, including performance tuning without changing user functionality

7.) Functional documentation (requirements, user manuals) and technical (manuals for administrators, Privacy risk analysis)

8.) Execution of Tests

9.) Operational configuration management

10.) Quality Assurance

At the same time, this component contains the impacts derived from the non-functional and process requirements included in the General Adjustment Factor illustrated previously and from the related WBS activities.

Direct Effort (NFRDE and PCRDE)

This category includes the effort generated by activities that cannot be considered included in the basic life cycle, proportional to the functionality (Σj NFRDEj+ Σk PCRDEk), such as those listed below by way of example and not exhaustively:

- Search engine optimization (SEO) and search
- Performance tuning
- Technical reengineering
- Architectural changes
- IT security
- Legacy data migration
- Check data quality
- Data cleansing
- Performance, integration and security tests
- User acceptance test
- External test support
- Support for operational deployment
- Generation of users/roles on demand
- Training
- Application monitoring in operation
- Ad hoc extractions
- Other professional services

The attributes of the SNAP method can be used in parametric models, where available. It is mandatory to pay attention to those attributes that may have been included in the "functional productivity model" based on FP analysis. For example, if we use the ISBSG data set for deriving the average productivity rate (PDR) the total effort collected in real projects supplied till now, includes the effort to implement the Data Entry Validation rules (attribute 1.1), the Logical and Mathematical Operations (attribute 1.2) and the Data Formatting operations (attribute 1.3). If we make an FP derived estimate using a ISBSG PDR and we add the effort derived by the quoted attributes, we are considering the same factors twice. Similarly, if we have adopted sizing guidelines that use FP to consider also the Multiple Input and Output methods or the Batch Processes, then we do not have to use more than one method to calculate the effort. For software maintenance where no functionalities are created, changed, or deleted (the so-called zero FP projects), then all the attributes may be considered without overlapping effect. At the

moment, a figure of SNAP points productivity for each attribute is not available in the international community, but they may be locally defined and used.

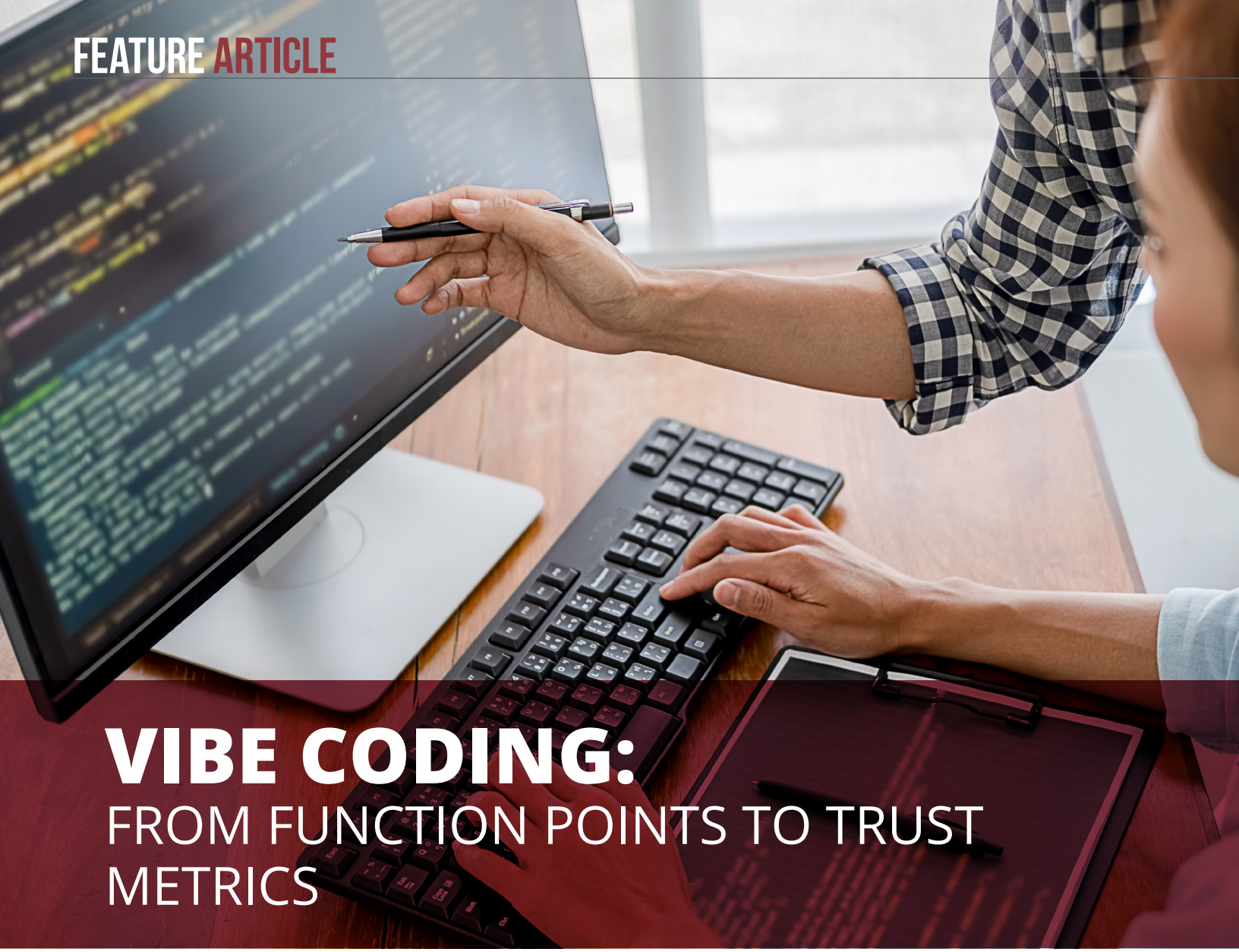
				Service Catalog				
Service Code	Service description	Deliverable	Size driver	Size	Unitary Effort (person day/SNAP Points)	Total Effort (person day)	Unitary staff cost (cost/person day)	Total cost
Category 1.	Data Operations							
1.1	Data Entry Validation		SNAP Points			0.0		€ -
1.2	Logical and Mathematical Operations		SNAP Points			0.0		€ -
1.3	Data Formatting		SNAP Points			0.0		€ -
1.4	Internal Data Movements		SNAP Points			0.0		€ -
1.5	Delivering Added Value to Users by Data Configuration		SNAP Points			0.0		€ -
Category 2.	Interface Design							
2.1	UI Changes		SNAP Points			0.0		€ -
2.2	Help Methods		SNAP Points			0.0		€ -
2.3	Multiple Input Methods		SNAP Points			0.0		€ -
2.4	Multiple Output Methods		SNAP Points			0.0		€ -
Category 3.	Technical Environment							
3.1	Multiple Platforms		SNAP Points			0.0		€ -
3.2	Database Technology		SNAP Points			0.0		€ -
3.3	Batch Processes		SNAP Points			0.0		€ -
Category 4.	Architecture					0.0		€ -
4.1	Component Based Software		SNAP Points			0.0		€ -
4.2	Multiple Input / Output Interface		SNAP Points			0.0		€ -

Conclusions

The most complete technique to predict software project costs is the WBS. A detailed version of the Work Packages needed to accomplish the work is not frequently available when an estimation is needed. Parametric models, either functional or non-functional in nature, may contribute to estimating huge subsets of the total work. The remaining part should be evaluated as a bottom-up estimation of the needed WPs. VAMOS-R3 is a flexible model to produce effective and complete estimation of effort starting from FUR, NFR and PR&C that integrates parametric, empirical models and direct WBS estimation.

About the Author

Dr. Roberto Meli graduated in Computer Science in 1984. In 1996, he became CEO of DPO Srl (Italy). Over the past 30 years he has developed focused competences in project management and software measurement and has written more than 75 papers for technical magazines and international conferences. He is a consultant and lecturer on project management and software measurement for many major Italian companies and public organizations. He developed the Early & Quick (E&Q) Function Point Analysis method and the Simple Function Point (SiFP) method. Roberto has held leadership roles over the last 30 years including: IFPUG Board of Directors, Chairperson of the GUFPI-ISMA (Gruppo Utenti Function Points Italia - Italian Software Measurement Association) board of directors, and coordinator of their Counting Practices Committee; Italian delegate to the MAIN (Metrics Association's International Network), Chairperson of the COSMIC Measurement Practices Committee, Conference Chairperson of SMEF (Software Measurement European Forum), and President of the Simple Function Point Association. He was a Certified Function Point Specialist (CFPS) from 1996 to 2004.



VIBE CODING:
FROM FUNCTION POINTS TO TRUST METRICS

By Tetyana Komarova

Imagine this scene. A developer opens a laptop at 10:00 in the morning. By 10:47, there is already a working application with authentication, a database, and a graphical user interface. Almost no code has been written manually. The system compiles, the tests pass, and the customer is satisfied. Then, at 11:03, the first difficult question arrives: "Can you explain why this function does what it does?" Silence follows.

This is *vibe coding*. It is not science fiction: it is already one of the ways many people, both experienced developers and non-developers, produce software today. A person describes to AI what they want to obtain, receives code, observes the result, asks for corrections, and tries again. The cycle continues until the system "seems to work."

Fascinating? Absolutely. Risky? Indeed, because code can start growing faster than human ability to read, understand, and verify it. And when this happens, speed is no longer an advantage; it becomes a hidden debt.

Code That Works... but That Nobody Really Understands

In the traditional software development model, the person who wrote the code usually understood its logic. They could explain constraints, compromises, and architectural choices. In vibe coding, this connection becomes weaker. The code may compile, pass tests, and show the expected behavior, while still containing unassessed dependencies, hidden complexity, hard-to-detect vulnerabilities, and unapproved assumptions.

The problem, therefore, is not only "Does the code contain bugs?" Instead, the questions become more uncomfortable. Who understood it? Who verified it? Who takes responsibility if something goes wrong? We could call this the risk of code *beyond human verifiability*. It is a concept that many traditional metrics do not capture. And this is exactly where a new challenge begins for the software measurement community.

Why Function Points Become Even More Important

Function Points do not measure lines of code. They do not depend on programming language, framework, or development tool. They measure the functional size of software from the user's point of view. This characteristic makes them especially valuable in the age of AI.

A functionality can be written by a human developer, by a mixed human-AI team, or almost entirely through automatic generation. But the functional questions remain the same. What service is provided to the user? What information is managed? What process is supported? Historically, we asked: how many Function Points were delivered? With what effort? With how many defects and at what cost? Now we must add new questions. How many of those Function Points were verified by an accountable human being? How many are traceable? How many can be explained? How many were delivered with an acceptable level of risk?

Consider the difference between a drug under development and a drug that has passed a verification process. Both are "real," but only one has a chain of evidence supporting it. Future Function Points may need the same distinction, not only functionality delivered, but functionality delivered in a governed, verifiable, and responsible way.

From Writing Work to Governance Work

As AI writes a growing share of code, human value does not disappear. It moves. The human being is no longer only the entity that produces code. The human becomes the person who frames the problem, clarifies the context, defines constraints, evaluates the result, and decides whether the risk is acceptable. The most valuable human contribution in AI-generated software may not be the line of code written manually, but the conscious decision to accept or reject an output. In other words, value moves from the keyboard to responsibility.

Can One AI Control Another AI?

One possible answer to the risk of generated code is to use more AI, but with different and deliberately separated roles. Imagine three agents:

- The Builder creates code, tests, and documentation.
- The Reviewer checks consistency, quality, security, and alignment with requirements.

The Auditor, perhaps the most interesting role, should not simply confirm the other two. Instead, it plays an adversarial role, actively trying to break the solution, exposing edge cases, vulnerabilities, hidden assumptions, and failure scenarios that may not emerge during standard review. The critical point is that two similar AI systems, with the same context, may make the same mistake and then convincingly confirm each other. Control should not be a simple consensus among artificial agents but rather, become a technical adversarial process. This does not remove human responsibility. On the contrary, it may become more explicit and more valuable. AI can help produce evidence. But the final decision must remain understandable, traceable, and attributable.

A Proposal of Metrics for Measuring Trust, Not Only Speed

The following metrics are proposed as a basis for reflection, not as a definitive standard. Some may be difficult to collect in certain organizational contexts; others may require significant adaptation. The goal is to open a conversation about what is worth measuring in the age of generated code, and to offer a concrete starting point for experimentation.

Cluster 1 - How Much Code Can We Really Trust?

This cluster measures control, understanding, and verification debt. It answers the question: Is the code we released really under human control, or are we silently accumulating risk?.

Metric	Proposed Formula	Indicative Target	Unit
Governed & Understood Code Rate	FP with requirement + prompt + tests + review traced / total delivered FP x 100	> 80%	%
Verification Debt	Unverified artifacts x estimated verification hours per artifact	-> 0	hours
Time to Trust	Verification hours / Generation hours	< 0.5	ratio

Concrete Example: Time to Trust

A team generates an authentication module with AI in 2 hours. Verification requires code review, security testing, and validation of edge cases: 6 hours in total. Time to Trust is 6/2 = 3.0, far above the indicative target of 0.5. This does not mean that the module is wrong, but it shows that the apparent speed gain was much lower than it seemed.

The point is not unique. A randomized controlled study (arXiv, July 2025) on 246 development tasks in mature open-source projects found that experienced developers took on average 19% more time when using AI tools than when not using them, while subjectively estimating that they were 20% faster. The gap between perception and reality is exactly what Time to Trust makes visible.

Cluster 2 - How Well Does the AI Behave?

This cluster measures model discipline, reliability, and traceability. It answers the question: Is the AI we are using merely fast, or is it also reliable, consistent, and responsible?

Metric	Proposed Formula	Indicative Target	Unit
Hallucination Tax	Hours lost on AI errors (false APIs, wrong configurations, etc.) / total development hours x 100	< 5%	%
Human Confidence Gap	Confidence declared by AI - confidence assigned by the team after verification (0-10 scale)	< 1.5	points
Prompt-to-Code Traceability	Artifacts linked to prompt + requirement + tests + documentation / total artifacts x 100	> 90%	%

Concrete Example: Hallucination Tax

During a 10-day sprint, a team tracks the time lost due to AI-generated errors: a cryptography library that does not exist in the indicated version (3 hours), an incorrect database configuration presented as standard (2 hours), and a convincing but wrong explanation of an authentication API (4 hours). Total: 9 hours out of 80 = Hallucination Tax of 11.25%, more than double the indicative target.

This phenomenon has been documented at scale. Veracode (GenAI Code Security Report 2025) found that in 45% of cases, AI chooses insecure methods even when a secure alternative exists, and this percentage did not improve over time. A study of 43,000 security advisories (Georgia Tech Vibe Security Radar, 2025-2026) tracked 74 Common Vulnerabilities and Exposures (CVEs) directly attributable to AI-generated code as of March 20, 2026. Researchers estimate that the real number may be 5 to 10 times higher than what was detected. Every CVE not identified in time is a Hallucination Tax paid in production.

Cluster 3 - What Data Feeds the AI?

This is the often-invisible dimension: the quality of generated code also depends on the quality of the data on which the model was trained and on the contextual data provided in prompts. Incomplete, obsolete, or unverifiable data produces unreliable outputs, even when the code "seems to work."

The metrics proposed here are particularly experimental. Collecting data on the origin and quality of input is still a relatively unexplored area in practice. But the signals are already visible: according to Informatica's CDO Insights Survey 2025, 43% of data leaders identify data quality and availability as the main obstacle to AI success. Gartner (2025) estimates that by 2026, 60% of AI projects will be abandoned because they lack AI-ready data. A Forrester survey of 500 enterprise data leaders found that 73% consider data quality and completeness the primary barrier. These numbers suggest that measuring input data quality is not an academic exercise, but a practical necessity.

Metric	Proposed Formula	Indicative Target	Unit
Data Completeness Rate	Required fields completed / total required fields in the dataset x 100	> 85%	%
Stale Data Ratio	Artifacts with last update date beyond threshold (e.g., 12 months) / total artifacts x 100	< 10%	%
Data Provenance Score	Verified and traced sources / total sources used x 100	> 90%	%
Siloed vs Shared Data Index	Datasets shared and validated across teams / total datasets used by AI x 100	> 70%	%

Note: the targets shown in all three clusters are purely indicative. Each organization should calibrate them according to its context, system criticality, and process maturity. The goal is not to reach a number, but to make visible what was previously invisible.

AI Engineering Quotient: Measuring the Intelligence that Really Matters

We often talk about how "intelligent" a model is. But in software development, this concept is too vague. We are not interested in whether AI seems intelligent in a conversation; we are interested in whether it produces correct, useful, secure, maintainable, and verifiable results.

We could call this capability AI Engineering Quotient: not an abstract measure, but a practical assessment. Does the code work? Does it respect constraints? Does it introduce vulnerabilities? Does it produce useful tests? Can it correct itself when it makes a mistake? Does it truly reduce human work, or does it simply move that work to correction?

The true intelligence of AI in software is not the brilliance of the answer; it is the verifiable quality of the artifact produced and the cost required to make it reliable.

Certifying the Process, Not Only the Product

The issue of responsibility will become unavoidable. If a system generates code and that code causes damage, it will not be enough to say: “AI wrote it.” It will not even be enough to ask whether the software “seemed to work” at the time of release.

We will need to reconstruct the decision chain that led to that release: which model was used, which version, which prompt, based on which requirement, with which constraints, which tests were executed, who approved the result, and which risks were accepted.

This direction is not entirely new. Capability models such as CMMI and several ISO standards have already taught the software industry that quality does not depend only on the final product, but also on the maturity of the process that produces it: traceable requirements, clear roles, repeatable controls, documented verification, risk management, and continuous improvement. With AI-generated code, this principle becomes even more important because part of the production activity may be performed by a non-human agent that is very fast, but not always transparent.

The issue is already concrete. In March 2026, Georgia Tech’s Vibe Security Radar reported 35 new CVE vulnerabilities attributed to AI-generated code. A CVE is a publicly identified security vulnerability tracked with an official identifier. This means that the problem is not only theoretical. Some vulnerabilities related to AI-generated code have already reached repositories and production systems.

For this reason, future certification may include not only the final software product, but also the way that software was generated, verified, and approved. Certifying the process does not mean adding bureaucracy; it means making responsibility visible. It means being able to demonstrate that in addition to the release there is not only working code, but a chain of evidence, controls, and conscious decisions.

The New Object of Measurement

The software metrics community is used to measuring projects and products. With vibe coding, a third object emerges: the software production system, which includes people, AI, prompts, requirements, generated code, tests, reviews, decisions, and responsibilities.

Function Points and SNAP can continue to provide a solid foundation for measuring functional and non-functional size. But around this foundation, we need new metrics to measure trust, human control and process quality.

In the world of vibe coding, the question "Who wrote the code?" becomes less important. The more relevant questions become who understood it, who verified it, who accepted its risk, and what evidence proves that the decision was responsible?

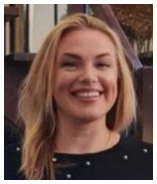
When code can be generated faster than a human being can read it, software metrics must become trust metrics.

References and sources

The following sources support the data and claims included in the article. The figures cited in the text come exclusively from these publications. The proposed metrics remain conceptual examples and are not derived from existing standards.

- Veracode, GenAI Code Security Report 2025 - 45 percent of AI-generated code introduces security vulnerabilities; security performance unchanged over time despite syntactic improvements. SecurityMagazine.com, July 2025.
- Georgia Tech SSLab, Vibe Security Radar (May 2025 - ongoing) - 74 CVEs directly attributable to AI-generated code as of March 20, 2026; 35 new CVEs in March 2026 alone, compared with 6 in January. research.gatech.edu, April 2026.
- Schreiber & Tippe, Security Vulnerabilities in AI-Generated Code: A Large-Scale Analysis of Public GitHub Repositories - analysis of 7,703 GitHub files with CodeQL; 4,241 CWE instances identified. arXiv:2510.26103, October 2025.
- METR / arXiv:2507.09089, Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity - randomized controlled trial on 246 tasks: experienced developers took 19 percent more time with AI while subjectively estimating that they were 20 percent faster. July 2025.
- Informatica, CDO Insights Survey 2025 - 43 percent of data leaders cite data quality and availability as the main obstacle to AI. informatica.com, 2025.
- Gartner, Lack of AI-Ready Data Puts AI Projects at Risk - 63 percent of organizations do not have data management practices adequate for AI; Gartner predicts that 60 percent of AI projects without AI-ready data will be abandoned by 2026. gartner.com, February 2025.
- Capital One / Forrester Research, survey of 500 enterprise data leaders (2024) - 73 percent identify data quality and completeness as the primary barrier to AI success. Cited in mountainadvocate.com, December 2025.
- CodeRabbit, analysis of 470 open-source pull requests (December 2025) - AI co-produced code presents approximately 1.7 times more significant issues than human-written code. Cited in getpanto.ai, April 2026

About the Author



Tetyana Komarova is based in Rome, Italy, and has more than 25 years of experience in IT. She has been a Certified Function Point Specialist (CFPS) since 2015 and has been a Certified SNAP Practitioner (CSP) since 2020. Tetyana works as a software developer, analyst, and software metrics expert at NTT DATA Gov & Tech Italia, a multinational company focused on innovative IT solutions. She is responsible for verification services for estimates and counts, as well as consulting services supporting the preparation of software estimates and functional size measurements. Through her international experience and collaboration with professionals from different countries and cultures, she has developed a strong ability to adapt communication styles and interpret business needs across diverse contexts. She is also passionate about behavioral economics.

METRICS AS AN ANTIDOTE TO IT BURNOUT?

By Alcione Jandir Candéas Ramos, Márcia Jacyntha Nunes Rodrigues Lucena, Thaís Bento Lima da Silva

Abstract

In Information Technology burnout is not only a human resources issue, but also a measurement problem. Drawing on global mental health data, neuroscience, and software engineering practice, this article shows how optimism bias, decision fatigue, and unrealistic schedules amplify psychosocial risk in technology teams. It proposes mature software measurement—especially IFPUG-based functional and non-functional sizing—as a practical mechanism to replace subjective pressure with defensible estimates, transparent scope control, and healthier delivery expectations. In this view, metrics are not merely management artifacts; they are safeguards for sustainable performance.

1. The Global Landscape of Workplace Mental Health

In the modern corporate era, mental health has transcended its traditional role as a human resources peripheral to become a core strategic driver of global economic productivity. The mental state of the workforce is no longer just a private concern: it is a structural element of organizational resilience. When professional well-being falters, the economic machinery slows, making the management of psychosocial risks a fundamental priority for any high-performance enterprise.

The scale of this challenge is reflected in global data from the World Health Organization (WHO) and the International Labor Organization (ILO), which estimate that mental disorders in the workplace cost the global economy approximately USD 1 trillion annually in lost productivity [1]. In Brazil, this reality is particularly acute. Recent reports indicate a 68% increase in medical leaves due to psychological disorders, totaling more than 472,000 cases and an economic burden exceeding R$ 3 billion. In response, the Brazilian legislative landscape has established Law No. 14.831/2024 and the updated Regulatory Standard No. 1 (NR-1/2025) [2].

The Three Pillars of Mental Health Promotion (Law No. 14.831/2024) [3]

Pillar	Focus Area
Mental Health Promotion	Proactive implementation of strategies to maintain cognitive and emotional health.
Employee Well-being	Creation of a supportive work environment that values the individual's integral health.
Transparency & Accountability	Clear reporting and organizational responsibility regarding psychosocial risks and safety.

This legislative shift acknowledges the transition from a **VUCA** (**V**olatile, **U**ncertain, **C**omplex, **A**mbiguous) world to a **BANI** framework. In a BANI environment—characterized as **B**rittle, **A**nxious, **N**onlinear, and Incomprehensible—the traditional pressures of the workplace are magnified [4]. This framework explains why professional exhaustion has become systemic: when systems are brittle and results are nonlinear, the human "software" between our ears is the first to experience a circuit break. This is nowhere more evident than in the Information Technology (IT) sector.

2. The IT Pressure Cooker: Burnout, Deadlines, and the 'Chaos' of Project Failure

The IT sector is uniquely vulnerable to psychosocial risks due to accelerated delivery cycles and constant innovation demands. Developers and engineers are not merely managing tasks: they are managing constant cognitive dissonance and the high-stakes demands of digital transformation.

The statistics are sobering: 80% of developers report symptoms of burnout, primarily driven by workload overload. Furthermore, Imposter Syndrome affects 52.7% of software engineers, a phenomenon directly correlated with reduced well-being and diminished productivity [5]. Historical data from the Standish Group "Chaos" reports reveal a persistent trend: more than 50% of all projects fall behind schedule and overrun their costs [6].

From a neuroscientific perspective, this is the "Neural DNA" of poor performance. Project failure is often rooted in **Cognitive Moderators**—filters that distort or block information—and specific neural short-circuits [7]:

- **Neural Short-Circuiting:** The brain "misses" critical information during planning that was predictable and preventable.
- **Present-Bias and Optimism:** A cognitive tendency to underestimate future obstacles while overestimating current capacity.
- **Decision Fatigue:** The loss of mental energy required to find and process complex information, leading to sub-optimal shortcuts.

Research indicates that deadlines are one of the most stressful aspects of the work environment, especially when the ability to meet deadlines is overestimated. Excessive time pressure impairs creativity, effectiveness, and overall performance.

A critical "So What?" analysis of unrealistic deadlines reveals a damaging behavioral cycle. Research by Ashley Whillans shows that employees are 31% less likely to ask for extensions due to a fear of judgment, even when they know more time would yield higher quality [8]. While methodologies have evolved, the "People Problem"—driven by neural biases—remains the central bottleneck.

3. A Neuroscience-Informed Model for Sustainable Software Engineering

To break this cycle, organizations must reframe the human factor as a structural element of software quality. By optimizing brain function and mitigating cognitive overload, we can foster a sustainable innovation culture [9].

Strategic Interventions for High-Performance Environments

Intervention	Neuroscientific Basis	Impact on Software Quality
Sustainable Cognitive Activation	Neuroplasticity & Variety	Improved decision-making and defect reduction
Neuroarchitecture	Sensory Stress Reduction	Sustained focus and increased creative output
NeuralPlan Workflows	Bias Mitigation	More accurate scope capture and fewer "surprises"

But subjective well-being initiatives remain "fragile" (BANI), unless they are anchored by the "Hard Truth" of objective software sizing.

4. The Missing Link: Why Resilience Requires Mature Metrics

Resilience without realism is a strategic danger. Preparing professionals to support chaos without fixing management maturity only leads to delayed burnout. Real resilience is built on mature metrics. When a manager lacks measurement maturity, they cannot produce a defensible estimate, leading to the "firefighting" culture that destroys mental health.

Immaturity in estimations is a root cause of projects exceeding budget and schedule. Carol Dekkers lists 10 principles for saving software development [10]:

Principle	Description
1. Data on Opinions	Estimates based on historical data are more defensible than theories or expert opinions.
2. Professional Discipline	Estimation is complex and requires a structured approach (Estimation Maturity Model).
3. Scope Definition	It is the basis for producing meaningful and accurate estimates.
4. No "One-Size-Fits-All" Approach	There's no single approach. The methods should vary depending on the purpose, time frame, type of estimate, and available data.
5. Size Matters	Software size is the main cost driver (CERs—Cost). Estimating Relationships.
6. Standard Units	Use of Units of Measurement (UoM) and early requirement documents (ConOps, Epics).
7. Non-Linearity	Software costs follow an "S" curve; doubling the project size doesn't just mean doubling the effort.
8. Impact of the Methodology	Agile, waterfall, and hybrid approaches alter the cost drivers.
9. Hybrid Solutions	Projects that combine custom code, COTS, and SaaS require multiple estimates.
10. Leadership Costs	They should be included, especially in large-scale complex systems.
Transparency & Accountability	Clear reporting and organizational responsibility regarding psychosocial risks and safety.

The International Function Point Users Group (IFPUG) has served as a stabilizing force in IT history for more than 40 years. Standards like Function Point Analysis (FPA) can act as neuro-regulators; by replacing subjective, amygdala-activating negotiations with objective data, they reduce the "Fear of Judgment" and "Cognitive Dissonance" that trigger stress.

IFPUG Standardized Units of Measure

Standard	Full Name	Primary Use Case
FPA	Function Point Analysis	Detailed sizing of functional requirements; historical benchmarking.
SNAP	Software Non-functional Assessment Process	Sizing non-functional requirements (Security, Architecture etc.).
SFP	Simple Function Points	Early-stage, rapid estimation for ROM (Rough Order of Magnitude).

These metrics create a "normalized historical base" that supports humanized management. They provide a "line in the sand" to protect the team from arbitrary, high-cortisol deadlines.

5. Integration Strategy: The PSM Framework and Standardized Sizing

Proper decision-making also requires a Balanced Multiple Perspective (BMP) approach. We must transition from raw Data to Information, Knowledge, and finally Wisdom (the ITIL DIKW model). Wisdom in IT management is knowing that software development costs are Non-Linear (Principle 7 above). Due to diseconomies of scale, effort increases disproportionately as size grows; ignoring this leads to the "Scrambling Phase" where teams burn out trying to defy mathematical reality.

To maintain this equilibrium, Luigi Buglione proposes that managers should adopt the Practical Software and Systems Measurement (PSM) information model [11]:

1. Early Estimation: Use Simple Function Points (SFP) during project initiation for rapid, early-stage Rough Order of Magnitude (ROM) estimates.
2. Detailed Management: Apply FPA and Software Non-functional Assessment Process (SNAP) as requirements mature to identify functional and non-functional cost drivers.
3. Accuracy Monitoring: Calculate the Relative Error (RE) to refine future estimates:
    - o RE = (Estimate - Actual) / Actual
4. Defensible Defense: Leverage normalized historical data to defend realistic timelines to leadership, bolstering the team's Emotional Security.

6. Call to Action

By balancing the biological needs of the developer with the mathematical precision of mature measurement, we create an industry that is not only productive, but profoundly sustainable.

- Invest in Neurobehavioral Management: Train teams to identify the "Cognitive Moderators" that distort project planning.
- Adopt well-established metric models: Use PSM Framework, FPA, SNAP, and SFP to move from opinion-based to data-driven, defensible estimates.
- Implement Regenerative Design: Design workplaces that prioritize cognitive health and reduce sensory stress.
- Cultivate the Skill Triad: Value Hard (technical), Soft (emotional), and Mad (creative/hobby-based) skills to foster a resilient culture.

To promote mature applications in society, we can also foster state-of-the-art software metrics within the university environment, bridging the gap from academia to global industry standards. The CFPS certification and the metrics community serve as vital conduits for this knowledge transfer to the next generation of professionals.

Conclusions

Reducing burnout in IT requires more than well-being initiatives; it requires better management decisions grounded in measurable reality. When effort, scope, and deadlines are negotiated without disciplined sizing, pressure escalates and delivery risk increases. Mature metrics offer a practical antidote by strengthening estimation, improving transparency, and creating a more defensible basis for planning and governance. By combining neurobehavioral awareness with IFPUG measurement practices, organizations can reduce avoidable stress, improve delivery confidence, and achieve more sustainable performance. In this sense, measurement maturity is not only a technical advantage but also a leadership responsibility.

References

[1] World Health Organization; (2024); *Mental health at work;* https://www.who.int/news-room/fact-sheets/detail/mental-health-at-work

[2] Ministério do Trabalho e Emprego; (2025); *NR 01 - Disposições gerais e gerenciamento de riscos ocupacionais* (Portaria MTE n 765, de 15 de maio de 2025); https://www.gov.br/trabalho-e-emprego/pt-br/acesso-a-informacao/participacao-social/conselhos-e-orgaos-colegiados/comissao-tripartite-partitaria-permanente/normas-regulamentadora/normas-regulamentadoras-vigentes/nr-01-atualizada-2025-i-1.pdf

[3] Philos Org (2024); Anuário Saúde Mental nas Empresas 2024; https://philosorg.com.br/anuario-saude-mental-nas-empresas-2024-3/

[4] Cascio, J.; (2020, April 29); Facing the Age of Chaos; https://medium.com/@cascio/facing-the-age-of-chaos-b00687b1f51d

[5] Guenes, P., Kalinowski, M. and Baldassarre, M. T.; (2024); Impostor phenomenon in software engineers; In Anais do XV Concurso de Teses e Dissertações em Engenharia de Software (Mestrado) – Congresso Brasileiro de Software: Teoria e Prática (CBSoft 2024) (pp. 5–6); Sociedade Brasileira de Computação; https://doi.org/10.5753/cbsoft_estendido.2024.4132

[6] The Standish Group; (1994–2020); CHAOS reports; https://www.standishgroup.com/

[7] Ramirez, J.; (2024, May 14); The neural DNA of bad-performing projects; https://medium.com/@drjoshramirez/the-neural-dna-of-bad-performing-projects-9c9da8371bef

[8] Whillans, A.; (2021, December 10); Go ahead and ask for more time on that deadline; Harvard Business Review; https://hbr.org/2021/12/go-ahead-and-ask-for-more-time-on-that-deadline

[9] Ramos, A., & Lucena, M.; (2025); Enhancing hackathon practices in software engineering: A neuroscience-informed model for team well-being and software quality. In Anais do XXXIX Simpósio Brasileiro de Engenharia de Software (SBES) (pp. 839–844); Sociedade Brasileira de Computação; https://doi.org/10.5753/sbes.2025.11606

[10] Dekkers, C.; (2025); What agile program managers need to know about cost estimating; MetricViews (pp. 12-15); International Function Point Users Group; https://ifpug.org/wp-content/uploads/2025/06/IFPUG_MetricViews_Spring_2025_v7_Pages.pdf

[11] Buglione, L.; (2025); Balanced measurement for proper decision-making; MetricViews (pp. 16-18); International Function Point Users Group; https://ifpug.org/wp-content/uploads/2025/06/IFPUG_MetricViews_Spring_2025_v7_Pages.pdf

About the Authors

Alcione Jandir Candéas Ramos is a Brazilian IT leader with a career spanning 40 years, including executive roles at SERPRO, Central Bank of Brazil, and the Ministry of Education. A renowned expert in software metrics, she is currently pioneering the integration of Neuroscience into Software Engineering to "design the pressure out" of development teams. Her credential portfolio includes CFPS, NPPQ, PMP, PRINCE2, PM4R, and SFPC. Through her Ph.D. research at UFRN, Alcione matures industry standards by merging technical rigor with neurobehavioral management. Her work transforms high-pressure environments into sustainable innovation hubs by aligning rigorous engineering with the fundamental neurobiology of high performance. She is also conducting a bridge into the Academia-Industry Gap to foster "critical mass" in the next generation of IT professionals, to avoid repeating current market failures, and reduce burnout.

Márcia Jacyntha Nunes Rodrigues Lucena is an Associate Professor at UFRN and a researcher in Software Engineering. She holds a Ph.D. from UFPE and conducted postdoctoral research at the University of Waterloo. Her work focuses on Requirements Engineering, Agile Methods, and Neuroscience applied to development teams. With more than 70 publications and 40+ supervised researchers, she also served as UFRN's Software Engineering program coordinator (2021–2025) and remains a permanent member of the PPGSC graduate

Thaís Bento Lima da Silva is a distinguished Gerontologist, researcher, and educator specializing in cognitive health and aging. She holds a Master's and a Doctorate in Neurology from the Faculty of Medicine at the University of São Paulo (FMUSP), with a focus on Cognitive and Behavioral Neurology. She is a researcher with the Cognitive and Behavioral Neurology Group at FMUSP. Her expertise focuses on cognitive training, dementia, and improving

INTEGRATING AND USING
AN ESTIMATION METHOD AND AN AGILE WORKFLOW IN SOFTWARE DEVELOPMENT

By Carlos Simões, Thiago Monarca Leone D'Durand

Abstract

One of the main challenges faced by organizations is establishing a consistent and agile workflow for software development, particularly when incorporating estimation methods. The use of a unified platform with predefined processes that integrates an estimation model can enable greater efficiency and effectiveness in project governance and management. By implementing structured workflows, organizations enhance their management capabilities, operational efficiency, financial control, and performance monitoring through planning and monitoring tools, supported by a standardized, repeatable, and auditable measurement method. Additionally, this approach supports compliance with high-maturity process models, such as the latest version of CMMI (v3.0) and MPS. BR SW (Melhoria de Processo Sofware Brasileiro -SW/Brazilian Software Process Improvement- SW), fostering continuous process improvement, quality assurance, and competitiveness in industrial and commercial contexts.

This article presents a simple, practical, and easily implementable approach, characterized by low costs and risks. Such an approach allows organizations of various sizes to integrate estimation methods and agile workflows in software development. Furthermore, it demonstrates the potential of this strategy to mitigate risks, optimize resource allocation, and maximize the value delivered in projects.

1. Introduction

Organizations face challenges in defining agile, easy-to-follow work procedures that align with their objectives. These procedures must provide management with visibility and governance over the execution and performance of established activities.

To achieve this, it is essential to allocate resources, align competencies, and establish clear guidelines for employees. This approach reduces risks, leverages opportunities, and ensures effective communication with stakeholders. The adoption of an agile and user-friendly workflow, along with planning and monitoring tools, is crucial for organizational success. It enhances project management, keeps plans current, and ensures control over activities, resources, risks, deadlines, and responsibilities.

A well-structured workflow also enables the collection and storage of data for continuous analysis, facilitating adjustments when necessary and supporting strategic decision-making by updating and utilizing the organization's measurement repository.

The use of these tools offers benefits such as:

- standardization and utilization of best practices;

- a unified integration platform between IT, suppliers, and business areas;
- improved management, financial control, and performance indicators;
- support for Capability Model Maturity Integration (CMMI) 3.0 and MPS BR SW assessments;
- root cause analysis and issue resolution before completing activities;
- comparison of results and continuous improvement; and
- feeding a historical measurement database.

The following premises must be considered for developing workflows and tools: i) simple, agile, and easy to use; ii) based on agile software development concepts; iii) compatible with functional and non-functional size estimation methods, ensuring effective planning, precise monitoring, and continuous improvement; and iv) aligned with CMMI 3.0 and MPS BR SW models.

The proposed workflow and tools do not aim to cover the Practice Areas and Practices that apply to the organization as a whole, such as the OT (Organizational Training), II (Integrated Implementation), PQA (Process and Quality Assurance), GOV (Governance), PAD (Process Asset Development) Practice Areas of the CMMI 3.0 model [CMMI 2024], among others, despite covering almost all of them (Figure 10). The same consideration applies to the MPS BR SW model [MPS 2024]. Therefore, the focus is exclusively on achieving the expected outcomes related to business requirements development. Thus, they do not replace configuration management systems, audits, or supplier and proposal selection criteria but rather complement software management by providing structure and control for projects.

This work presents a set of tools to support information systems development, including: i) a simple and intuitive workflow; ii) a planning and monitoring tool aimed at supporting the adoption and institutionalization of the workflow using software measurement concepts; and iii) a risk and opportunity management tool. All elements follow the best practices of the CMMI 3.0 and MPS SW Level C models, ensuring an efficient and auditable process.

2. Initiating the Development or Maintenance of an Information System

When addressing a business need, whether for new development or maintenance of an existing system, the budgeting process can be based on information outlined in a document containing the initial business requirements. This document may include i) a high-level scope description and the purpose of the business need; ii) problem definition, proposed solution, and high-level resources, which help set expectations and reduce risks; iii) information gathered during the demand analysis, such as business objectives, key characteristics of the business need, technological aspects, and risks, guiding the development process; iv) presentation of Functional and Non-Functional Requirements; v) assumptions and constraints.

The budget is also supported by additional information, such as: i) desired start and end dates for addressing the business need; ii) hourly rate and productivity levels established by the organization, based on the desired solution characteristics and team profile; iii) available budget. From this data, it is possible to calculate other key elements for an accurate budget, such as: i) number of business days between the desired start and end dates; ii) number of available working hours corresponding to the business days; iii) total work hours derived from the relationship between budget, hourly rate, and productivity; iv) average team size, calculated based on the combination of work hours and available hours; v) functional and non-functional size, determined by the relationship between work hours and established productivity levels [Simões and Silva 2024] [Simões and Silva Presentation 2024].

3. Initial Planning with Agile Methodology

Initial planning based on a detailed understanding of business requirements requires full knowledge of both functional and non-functional requirements before development begins. However, this complete knowledge is not a prerequisite for agile development. In an agile approach, planning and prioritization are based on an initial backlog often referred to as a *product backlog*, which evolves with more details and new requirements throughout the development cycle.

3.1 Clarification on Agile Method Adoption and Estimation Methods

As presented by Simões and Silva [Simões and Silva Presentation, 2024], several misconceptions about agile methodologies need to be clarified when applying them:

- There is no schedule → However, the user defines a desired start date for the development of the business need and a target delivery date for its use.
- There is no deadline → However, there is a desired final date set by the user and a pre-established budget that will be consumed within a specific timeframe.
- There is no fixed scope → However, the first step is to define the Initial Product Backlog, which evolves in knowledge and detail throughout development cycles.
- It is impossible to use an estimation method in Agile → Before defining at least the Initial Product Backlog, there is only a broad idea of what is desired (e.g., a system for service order management). At this stage, adopting any estimation method is impossible unless relying on guesswork or intuition as an "Estimation Method."

3.2 Scrum

It is common that when interest in an information system arises, there is no complete understanding of the business needs required to achieve the desired outcome. Agile methods are well-suited for

this scenario, as they prevent excessive time spent specifying requirements that may change and naturally evolve throughout the development cycle [Scrum 2020].

Scrum employs an iterative and incremental approach (Figure 1) to optimize predictability and control risk. Within Scrum, critical decisions are made based on the perceived state of its formal artifacts. If artifacts lack transparency, decisions may reduce value and increase risk. Transparency enables inspection, and inspection without transparency is misleading and wasteful.

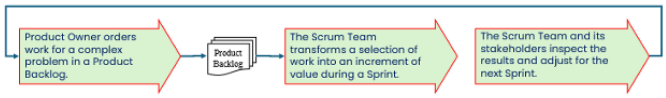


Figure 1 - SCRUM Cycle

3.3 Kanban

Kanban is not a methodology or a process framework. Instead, it is a management method or approach that can be applied to an existing process or workflow. The question is not about using Kanban versus a methodology or framework. Rather, Kanban is always added to an existing methodology, framework, or mode of work. Kanban helps to consistently meet customer expectations by managing work more effectively and improving service delivery. It is a means to enhance what and how things are already being done (sometimes called a current state), not a replacement for existing methods [Kanban 2020].

In Kanban, demand dictates the pace of production according to customer consumption levels. It is more flexible, agile, and easier to plan and monitor than using Scrum with Sprints, as it doesn't require grouping backlog items within a time-box. The established prioritization determines the sequence of execution of backlog items, providing more autonomy and independence to the development team.

For successful planning and managing workflow, it is critical to use a method for estimating, measuring, and controlling the entry and completion of work items. When requirements are not fully understood or defined, an initial estimate can be made, and as the cycle progresses, estimates can be adjusted and execution re-planned quickly.

Breaking down business needs into smaller parts (granularity) makes planning easier and helps in understanding business requirements, estimating and controlling production capacity, and facilitating the distribution of tasks to the team in a continuous, agile, short-term delivery flow, ideal for Kanban. Additionally, it helps detect defects early, contributing to product quality improvement by reducing the number of defects found during testing and those reported by customers. Workflow represents the time an item takes from entry to exit, depending on performance levels and the size of the work [Simões and Montoni 2014].

3.4 SAFe

SAFe (Scaled Agile Framework) (Figure 2) [Scaled Agile 2023] is an agile framework designed to help large organizations implement agile practices at scale, ensuring strategic alignment and continuous value delivery. It is structured into four levels:

1. Team Level: Teams use Scrum or Kanban to deliver work increments in short cycles, ensuring rapid delivery and adaptability.

2. Program Level: This level coordinates multiple teams through the Agile Release Train (ART), ensuring continuous delivery of value across teams.

3. Large Solution Level: Focuses on more complex solutions that involve multiple ARTs working together on large-scale projects.

4. Portfolio Level: This level manages projects aligned with the company's strategic vision and ensures that all initiatives are ontributing to business objectives.

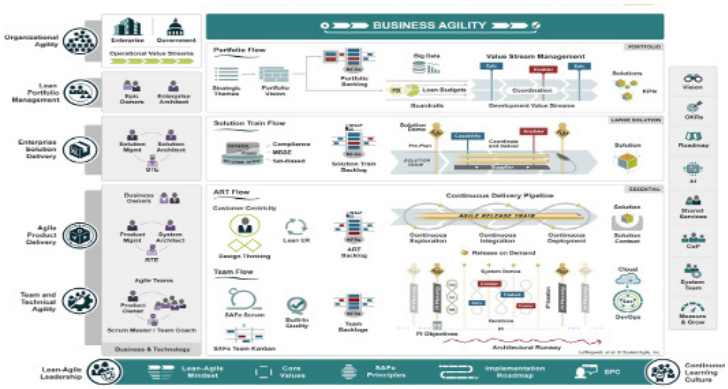


Figure 2 - Full SAFe 6.0 Configuration

SAFe combines Scrum and Kanban to provide agile governance and effectiveness. At the Team level, Scrum organizes work into sprints, while at the Program level, Kanban manages the flow between teams, prioritizing continuous delivery and handling dependencies. This hybrid approach enables flexibility and predictability, making it easier to adapt to changes without compromising efficiency.

The latest version, SAFe 6.0, released in 2023, enhances organizational agility with improvements in collaboration, acceleration of value flow, and integration of emerging technologies like AI, Big Data, and Cloud. Companies such as Intel, Cisco, and Phillips have adopted SAFe to strengthen innovation and continuous value delivery. The GAO (U.S. Government Accountability Office) also adopted agile methods, reflecting the growing acceptance of this approach in the government sector [U.S. Government Accountability Office 2020] .

In Planning Poker, the ideal approach is to combine experience with one or more estimation methods (whether functional or non-functional, but parametric, repeatable, and auditable), with each method validating the other. The goal is not to guess, but to estimate based on historical data and a standard of functional and non-functional sizing. No estimation method is perfect, so cross-validating them ensures more reliable and accurate estimates.

4. Software Development Estimation

In the context of software development, it is important to understand that a system has dimensions. The functional dimension involves business requirements measurable by functional size measures, as defined by ISO/IEC 14143, enabling the estimation of effort and cost by relating them to productivity and hourly rate. The non-functional dimension covers technical and qualitative requirements measurable by non-functional size measures, following ISO/IEC/IEEE 32430:2025, to facilitate effort and cost estimation. Additionally, there may be requirements that do not fit within these dimensions, such as post-production support or vendor selection.

In agile development, an initial backlog is planned and prioritized, evolving with more details and new requirements throughout the development cycle. Estimates are updated as the initial backlog evolves, incorporating new knowledge and details.

Ultimately, a total functional and non-functional size is delivered, rather than a predefined scope. Therefore, it is not the scope that is fixed but the number of work hours and budget, derived from an anticipated total size. Size is a combination of information about the budget, hourly rate, deadlines, and productivity [Simões 2024], as well as the estimated hours for requirements that are not measurable in terms of functional and non-functional size. These may include training or knowledge transfer activities.

When implementing a new mindset focused on quality and productivity, it is unacceptable to develop systems based on "feeling" [Simões and Silva 2024] [Simões and Silva Presentation 2024]. The absence of performance indicators—such as financial, quality, productivity, effort, and deadline indicators—hinders or makes it impossible to effectively manage software development. The Scrum Master should support the Scrum Team to ensure accurate estimates, using methods that are standardized, auditable, and repeatable, resulting in security, reliability, and objectivity.

4.1 User Story Points - USP

A User Story Point is a relative unit of measure, decided and used by individual Scrum teams to provide relative effort estimates for completing requirements. The primary purpose of Story Points is to facilitate team estimation.

Instead of looking at a product backlog item and estimating it in hours, teams only consider how much effort a backlog item will require in relation to other items in the backlog. A Story Point is an estimate of the effort involved in completing something. This estimate should be based on various factors, including the amount of work, the risk or uncertainty inherent in the task, and the complexity of the work [Cohn 2025].

Story Points have no value to a company because they cannot be used to predict costs or delivery dates outside the team using them. Initially, the team cannot make any predictions about how many Story Points can be completed in a sprint (velocity) until they have a few sprints under their belt [Cohn 2025].

The use of Story Points in Planning Poker has the central purpose of reflecting individual opinions on the feasibility of the work. Additionally, it promotes discussion, revealing potential difficulties. Estimates that are higher (or lower) than the rest of the team's should be explained in order to reach a consensus. However, using Story Points in this way makes it impossible to audit the results or compare the estimates with others made within the organization or across organizations.

Sizing a User Story, characterized as a functional requirement, in terms of the number of Story Points, so that the estimate can be auditable, requires preliminary work to establish classification and characterization standards. The following definitions and actions are key to successfully institutionalizing and measuring User Stories in terms of functional size in a standardized, repeatable, and auditable way:

- Classify User Stories as functional and non-functional requirements.
- Establish standards and criteria to define the granularity of the User Story, as described in ISO/IEC 14143-1 or ISO/IEC/IEEE 32430:2025.
- Define standards and criteria for the complexity of the User Story, associated with a scale, for example, Fibonacci.
- Describe procedures for measuring functional and non-functional size and train those involved.
- Characterize the team and the development environment.
- Perform functional and non-functional size estimations for User Stories.
- Perform validation audits of the estimates.
- Develop User Stories, collecting measurements, characterizations, and hours worked.
- Use statistical techniques to determine productivity (control limits).

Since this is a scale established and used by the organization itself and not by a globally recognized consensus organization, it is not possible to make comparisons with other organizations.

4.2. Non-Functional Requirements Sizing

Non-functional sizing encompasses the technical and qualitative aspects of software, such as performance, security, usability, and interoperability. The ISO/IEC/IEEE 32430:2025, published on February 6, 2025, defines a method for measuring the non-functional size of software. This standard complements ISO/IEC 20926:2009, which deals with measuring the functional size. It ensures that functional and non-functional measurements are made separately, avoiding duplication in measuring distinct dimensions of the software. The adoption of this standard allows for more accurate estimates of effort and cost for non-functional requirements, such as performance, security, and usability. Additionally, standardization facilitates communication among

teams and stakeholders, promoting a common understanding of non-functional aspects in software projects.

According to the standard, the Software Non-functional Assessment Process (SNAP) [SNAP 2014] can be used to evaluate non-functional requirements. This process organizes non-functional requirements into four main categories:

1. Data: Evaluates the complexity of data manipulation and storage.
2. User Interface: Considers aspects such as usability and accessibility.
3. Processing: Measures the effort needed for calculations, business rules, and security.
4. Environment: Covers interoperability, performance, and specific technical requirements.

The use of SNAP enables more precise estimates of effort and cost, improving planning and control in software projects.

SNAP complements functional analysis by quantifying the effort needed to meet non-functional requirements, aligning with ISO/IEC/IEEE 32430:2025 – Software Engineering and Software non-functional size measurement [ISO 32430 2025]. The application of this method helps determine the effort (person-hours, HH) and costs needed for non-functional needs, enhancing the accuracy of estimates.

A comparative analysis (Figure 3) was conducted using the presented Non-Functional Requirements Sizing measurement techniques—SNAP Function Point (SNAP), Man-Hour (MH), and Story Point (SP)—to evaluate their applicability and effectiveness in different project scenarios. To ensure a comprehensive assessment, we established 21 criteria based on internationally recognized standards, including CMMI, MPS.BR, and ISO/IEC/IEEE 32430:2025. This structured approach allowed us to identify the strengths, limitations, and best use cases of each method, enhancing the accuracy of estimations and aligning them with both agile and traditional project management frameworks.

Criteria - Non-Functional Requirements	SNAP	Man-Hour	SP
Enables support for distinguishing between functional and non-functional requirements	2	0	0
Has standards and criteria to define the granularity of non-functional requirements as described in ISO/IEC/IEEE 32430:2025	2	0	0
Has standards and complexity criteria associated with a scale for non-functional requirements	2	0	0
Enables complexity to be obtained through comparison of information relevant to business needs for non-functional requirements	2	0	0
Has documented procedures for measuring non-functional size	2	0	0
Allows comparison between organizations for performance and quality indicators used for non-functional requirements	2	0	0
Enables comparison of indicators related to non-functional requirements between teams within the same organization	2	0	0
Enables audit validation of effort and cost estimates related to non-functional requirements	2	0	0
Supports team characterization related to non-functional requirements	2	0	0
Allows the use of statistical techniques to obtain productivity (control limits) related to non-functional requirements	2	0	0
Can be used to support contract development	2	0	0
Enables the definition of a Service Level Agreement (SLA) to monitor task execution related to non-functional requirements	2	0	0
Supports calculations for determining the number of hours required to perform activities related to non-functional requirements	2	2	0
Supports calculations for determining the timeframe for completing activities related to non-functional requirements	2	2	0
Supports the definition of performance indicators related to non-functional requirements	2	0	0
Supports the definition of quality indicators related to non-functional requirements	2	0	0
Enables training for the team to understand predefined procedures related to non-functional requirements	2	0	0
Supports the presentation of easily understandable information to senior management	2	0	0
Easy to apply based on predefined procedures related to non-functional requirements	2	0	0
Supports the development of an auditable timeframe for business need development related to non-functional requirements	2	2	0
Supports cost estimation for business need development related to non-functional requirements	2	2	0
Total	42	8	0
	100%	19%	0%

Figure 3 – Non-Functional Requirements Sizing measurement techniques comparison

4.3. Functional Requirements Sizing

The following are some methods of estimating functional size.

4.3.1. Function Point Analysis (FPA)

Function Point Analysis (FPA) is an estimation method established by IFPUG [IFPUG 2010], that includes criteria for estimating the functional size and aligns with the ISO/IEC 14143-1 standard [ISO/IEC 2012]. It is a parametric, repeatable, and auditable method that can be applied at any stage of the software life cycle [Wolfart 2012]. The ISO/IEC 14143 standard defines the principles for measuring the functional size of software (FSM – Functional Size Measurement). It establishes fundamental concepts to measure the functional size of a software system in an objective and standardized manner, regardless of the technology or methodology used in development. One of the key concepts is the granularity of the elementary process, which satisfies the functional requirement and represents the smallest significant functional unit for the user, which cannot be divided without leaving the functional requirement incomplete.

Functional measurement is based on functional requirements and classifies the basic functional components (BFC) into five categories: Elementary processes must be classified as External Inputs (EI), External Queries (EQ), External Outputs (EO), and logical data Internal Logical File (ILF) and External Interface File (EIF). Each component is assigned a weight according to its complexity [Wolfart 2012] [IFPUG 2010]. By applying FPA, the result is the functional size, which, when associated with the established productivity rate, allows the determination of the required person-hours (HH) for developing the business need. Applying the hourly rate gives the costs within the functional dimension of the requirements.

4.3.2. Simple Function Points - SFP

The Simple Function Point (SFP) is a method for measuring functional size, designed by Roberto Meli in 2010 in compliance with the ISO/IEC 14143-1 standard. SFPs were adopted by IFPUG in 2019, and is currently in version 2.1 as of 2021 [SFP 2021]. The Simple Function Point is a simplification of Function Point Analysis and aims to express a new unit of measurement—the Simple Function Point—rather than just a solution to approximate the size in function points. It is a parametric, repeatable, and auditable method that can be applied at any stage of the software development lifecycle. It adopts the concept of granularity for the elementary process, compatible with Function Point Analysis [IFPUG 2010] and the ISO/IEC 14143-1 standard. The Simple Function Point considers only two basic functional components:

1. Elementary Process (without delving into whether it is an External Input, External Output, or External Inquiry);
2. Logical File (without delving into whether it is an Internal Logical File or External Interface File).

In the assessment of complexity within the Simple Function Point, there is no need to identify Data Element Types (DET), File Types Referenced (FTR), or Logical Record Types Referenced (RET). This greatly simplifies the application of the method from the very

beginning of identifying the business need and developing the initial product backlog. Some advantages of the Simple Function Point include: i) quick to use; ii) can be applied earlier in the life cycle; iii) requires fewer details; iv) easy to learn; v) standardized, auditable, and repeatable; vi) can be used to support function point estimation using FPA.

With the application of the SFP, the result is the functional size, and when the established productivity is applied, it leads to the number of effort hours (HH) needed for the development of the business requirement, and consequently, the costs.

The U.S. Government Accountability Office (GAO) recently released its updated Agile Assessment Guide: Best Practices for Adoption and Implementation. In this guide, the SFP was recognized and endorsed as one of the best practices for software cost estimation.

A comparative analysis (Figure 4) was conducted of the presented Functional Requirements Sizing measurement techniques—Function Point Analysis (FPA), Simple Function Point (SFP), Hours (HH), and Story Point (SP)—to evaluate their applicability and effectiveness in different project scenarios. To ensure a comprehensive assessment, we established 21 criteria based on internationally recognized standards, including CMMI, MPS.BR, and ISO/IEC 14143-1. This structured approach allowed us to identify the strengths, limitations, and best use cases of each method, enhancing the accuracy of estimations and aligning them with both agile and traditional project management frameworks.

Criteria - Functional Requirements	FPA	SFP	Man-Hour	SP
Enables support for the classification of functional and non-functional requirements	2	2	0	0
Has standards and criteria to define requirement granularity as described in ISO/IEC 14143-1	2	2	0	0
Has standards and complexity criteria associated with a scale	2	0	0	0
Enables complexity to be obtained through comparison of information relevant to business needs	2	0	0	0
Has procedures for measuring described functional size	2	2	0	0
Allows comparison between organizations	2	2	0	0
Allows comparison between teams within the same organization	2	2	1	0
Enables audit validation of estimates	2	2	0	0
Supports team characterization	2	2	0	0
Allows the use of statistical techniques to obtain productivity (control limits)	2	1	0	0
Can be used to support contract development	2	2	0	0
Enables the definition of a Service Level Agreement (SLA)	2	2	0	0
Supports calculations for obtaining the number of hours required to perform an activity	2	2	2	0
Supports calculations for determining the timeframe for completing an activity	2	2	2	0
Supports the definition of performance indicators	2	2	0	0
Supports the definition of quality indicators	2	2	0	0
Enables training for the team	2	2	0	0
Supports the presentation of easily understandable information to senior management	2	2	0	0
Easy to apply based on defined procedures	2	2	0	0
Supports the development of an auditable timeframe for business need development	2	2	2	0
Supports cost estimation for business need development	2	2	2	0
Total	42	37	9	0
	100%	88%	21%	0%

Figure 4 - Functional Requirements Sizing measurement techniques Comparison

Editor Note: The remainder of this article is targeted for publication in forthcoming quarterly IFPUG Newsletters.

References

[CMMI 2024] CMMI Institute. *CMMI Model V3.0.* Available at: https://cmmiinstitute.com/resource-files/public/cmmi-model-materials/cmmi-model-release-notes. Accessed on: March 3, 2025

[MPS 2024] Software Process Improvement Group (Softex). *Guia Geral MPS de Software: 2024.* Available at: https://softex.br/download/guia-geral-mps-de-software2024/. Accessed on: March 3, 2025.

[Scaled Agile 2023] Scaled Agile, Inc. (2023). *SAFe 6.0 – The World's Leading Framework for Business Agility*. Available at: https://scaledagile.com/2023-march-launch-details.

[Scrum 2020] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide.* Available at: http://scrumguides.org/.

[Kanban 2020] *Kanban Guides: The Official Kanban Guide.* December 2020. Available at: https://kanban.university/wp-content/uploads/2023/04/The-Official-Kanban-Guide_A4.pdf.

[IFPUG 2010] IFPUG. (2010). *Counting Practices Manual*, Version 4.3.1. January 2010. Available at: https://ifpug.org/. Last accessed: November 20, 2023

[SFP 2021] IFPUG. (2021). *Introducing Simple Function Points (SFP).* October 26, 2021. Available at: https://ifpug.org/2021/10/26/introducing-simple-function-points-sfp.

[SNAP 2014] IFPUG. (2014). *Software Non-functional Assessment Process (SNAP).* Princeton: IFPUG.

[NESMA 2015] Onvlee, J., & van Solingen, R. (2015). *Scrum and Function Points: Friends or Foe.* Available at: https://nesma.org/2015/03/scrum-function-points-friends-foe/. Originally posted in AutomatiseringGids (in Dutch). Available at: https://web.archive.org/web/20161105053844/http://www.scrum.nl/media/scrum_pub_06/$FILE/Scrum%20en%20functiepunten.pdf.

[ISO/IEC 2012] ISO/IEC 14143-1:2012. *Information technology – Software measurement – Functional size measurement – Part 1: Definition of concepts.* Available at: https://www.iso.org/standard/51717.html.

[ISO/IEC 2009] ISO/IEC 20926:2009. *Software and systems engineering – Software measurement – IFPUG functional size measurement method.* Available at: https://www.iso.org/standard/51717.html.

[ISO/IEC/IEEE 2025] ISO/IEC/IEEE 32430:2025. *Measurement of Non-functional Software Size.* Available at: https://www.iso.org/standard/86303.html.

[ISO/IEC 2011] ISO/IEC 25010:2011. *Systems and software engineering – Systems and software quality models.* Available at: https://www.iso.org/standard/78176.html.

[LEI 13467] Lei Nº 13.467, de 13 de julho de 2017. Available at: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/l13467.html. Accessed on: March 3, 2025.

[Cohn 2025] Cohn, M. (2025). *It's Effort, Not Complexity.* Available at: https://www.mountaingoatsoftware.com/blog/its-effort-not-complexity. Accessed on: March 9, 2025.

[Simões and Montoni 2014] Simões, C., & Montoni, M. (2014). *Applying Statistical Process Control in Small-Sized Evolutionary Projects: Results and Lessons Learned in the Implementation of CMMI-DEV Maturity Level 5 in Synapsis Brazil.* Journal of Software Engineering Research and Development, 2:2. Available at: 2195-1721-2-2.pdf (springer.com).

[Gonsalves 2024] Gonsalves, W. (2024). Available at: https://www.linkedin.com/posts/wilfred-gonsalves-169821270_knowledgevsexperience-professionalgrowth-activity-7276114774923300864-uAKT?utm_source=social_share_video_v2&utm_medium=android_app&utm_campaign=share_via.

[Simões 2023] Simões, C. (2023). *Modelo de Referência de Processo para Terceirização de Força de Trabalho de TI.* Tese de Doutorado, Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática.

[Wolfart 2012] Wolfart, D. (2012). *Estimativa de Tamanho de Software por Meio da Técnica de Análise de Pontos de Função.* Monografia (Especialização em Engenharia de Software). Universidade Tecnológica Federal do Paraná.

[Simões and Silva 2024] Simões, C., & Silva, T. S. da C. (2024). *Agile Planning and Monitoring with Kanban and Measurement.* IFPUG Metric Views, June 2024. Available at: https://ifpug.org/learning-and-events/knowledge-cafe-webinar-series.

[Simões 2024] Simões, C. (2024). *A Difícil Arte de Planejar e Monitorar Custo, Desempenho e Qualidade Quando da Adoção de Métodos Ágeis.* Workshop Anual do MPS – WAMPS.

[Simões and Silva Presentation 2024] Simões, C., & Silva, T. S. da C. (2024). *Agile Planning and Monitoring with Kanban and Measurement.* Available at: https://ifpug.org/learning-and-events/knowledge-cafe-webinar-series. Accessed on: March 15, 2025.

[U.S. Government Accountability Office 2020] U.S. Government Accountability Office. (2020). *Agile Assessment Guide: Best Practices for Agile Adoption and Implementation (GAO-20-590G).* Available at: https://www.gao.gov/products/gao-20-590g.

About the Authors

Carlos Simões has a PHD from the Federal University of the State of Rio de Janeiro and a Master's in Systems Engineering and Computing from COPPE/UFRJ. He is a Consultant for PMO, Agile Methods, Project Management, and Metrics for the Enel group in Brazil.

Thiago Monarca Leone D'Durand has more than 10 years of experience in IT and Business as well as expertise in agile methods, cybersecurity, growth hacking, strategy, technology, digital, coaching, and training. With a multidisciplinary background, he has applied his IT and business skills in various sectors. He holds 15+ international certifications in various domains, including Software Measurement. A member of IFPUG since 2009, and having passed the rigorous CFPS exam several times, he became a member of the CEC, PEC, and CMC committees where he contributed broadly to the IT community in several ways, such as helping organize webinars and the ISMA conference, evaluating the work of others submitted to IFPUG, and translating SFP manuals and tools.

MEASURING SOFTWARE PRODUCTIVITY IN THE ERA OF GENERATIVE AI

By Emanuele Richiusa

Abstract

The widespread adoption of Generative AI (GenAI) is transforming software development practices, particularly by accelerating implementation activities and reshaping the distribution of effort across the software lifecycle. This evolution has led to renewed questions about the relevance of traditional productivity measurement approaches and, more specifically, about the continued applicability of Function Point Analysis (FPA).

This article argues that software productivity measurement remains not only valid but essential in AI assisted contexts, provided that productivity is interpreted correctly as the ratio between functional output and effort. While GenAI affects how software is produced, the functional output delivered to users remains largely invariant. For this reason, IFPUG FPA continues to represent a stable, technology agnostic reference for productivity measurement, governance, and sourcing decisions. The article further discusses the role of GenAI in supporting functional measurement activities and highlights the importance of a prudent, evidence based approach for interpreting productivity changes in diverse technological and organizational contexts.

1. Introduction

GenAI is rapidly becoming embedded in software development lifecycles. AI assisted coding tools, autonomous agents, and intelligent automation increasingly support activities such as scaffolding, refactoring, documentation, and test generation. These changes have reopened a long standing debate within the software measurement community: does productivity measurement still make sense when a significant portion of software production is automated?

This debate often arises from a misunderstanding between the means of production and the value produced. The purpose of software measurement is not to track how code is written, but to understand the relationship between the value delivered to users and the effort required to deliver it. From this perspective, the introduction of GenAI does not invalidate measurement practices; rather, it reinforces the need for methods that are independent of technology and execution mechanisms.

2. Productivity as Output Over Effort

Software productivity can be formally expressed as:

Productivity = Functional Output / Effort

Much of the current discussion around GenAI emphasizes effort reduction at the level of individual activities, particularly coding. However, productivity improvements cannot be inferred from localized efficiency gains alone. A reduction in effort for specific tasks does not necessarily imply an increase in overall productivity, especially when effort is redistributed to other lifecycle phases.

To assess productivity meaningfully, it is therefore essential to maintain a stable definition of output and to observe how effort changes relative to that output. Without a stable output measure, productivity discussions risk degenerating into tool specific claims that lack comparability and empirical grounding.

3. Functional Size as a Stable Measure of Output

IFPUG FPA measures software size based on functional user requirements, independently of implementation choices. This principle is particularly relevant in the context of GenAI, where implementation mechanisms evolve rapidly but functional intent remains anchored to business and user needs.

From a functional perspective, GenAI does not alter:

- the types of user transactions supported by the software;
- the logical data structures maintained;

- the boundaries between the application and its users or external systems.

External Inputs, Outputs, Inquiries, Internal Logical Files, and External Interface Files continue to represent the same functional behavior required by users, regardless of whether the software is implemented manually or with AI assistance.

This invariance makes Function Points a stable numerator for productivity measurement, allowing organizations to compare:

- projects developed before and after AI adoption;
- different delivery models;
- different vendors and sourcing arrangements

4. Effort Redistribution Across the Software Development Life Cycle

One of the most significant effects of GenAI is the redistribution of effort across the software development life cycle, rather than a uniform reduction of total effort.

4.1 Compressed Activities

Empirical evidence shows substantial efficiency gains in:

- code generation and scaffolding;
- refactoring;
- generation of baseline tests and documentation.

These activities increasingly function as execution layers that can be partially or fully automated.

4.2 Expanded Activities

At the same time, effort expands in areas such as:

- requirements clarification and intent definition;
- functional and semantic validation;
- evaluation of AI generated behavior;
- security, compliance, and risk controls;
- release governance and adoption monitoring.

As a result, faster implementation does not automatically yield higher overall productivity.

5. GenAI as Support for FPA

Beyond software delivery, GenAI can also be leveraged to support functional size measurement activities. AI techniques may assist in:

- analyzing requirements and user stories;
- identifying candidate functional processes and data functions;
- supporting early or incremental FPA;
- enabling continuous review of functional scope in iterative delivery models.

From a methodological standpoint, this evolution does not modify the foundations of IFPUG FPAs. Boundary definition, functional component identification, and adherence to the Counting Practices Manual remain essential.

AI supported measurement must therefore operate within a Human in the Loop framework, where automated outputs are reviewed and validated by qualified practitioners. In this sense, GenAI acts as a productivity enhancer for measurement activities themselves, accelerating analysis while preserving rigor, consistency, and comparability.

6. Contextual Variability and the Case for a Prudential Measurement Approach

A recurring theme in discussions on GenAI is the expectation of productivity gains. While such expectations are reasonable, experience shows that AI impact is highly context dependent.

Key factors influencing productivity outcomes include:

- the maturity of the technological stack (legacy versus modern architectures);
- the delivery model (sequential versus iterative or agile);
- the quality and stability of functional requirements;
- organizational maturity in validation and governance practices.

In many contexts, AI reduces effort in implementation while increasing effort in validation and control. This variability makes it methodologically unsound to assume productivity improvements a priori. Instead, productivity gains must be observed, normalized, and validated over time using stable output measures.

FPA provides the necessary anchor for this prudential approach, enabling organizations to distinguish:

- local efficiency gains from structural productivity improvements;
- temporary acceleration from sustainable performance change.

Rather than constraining innovation, this approach allows organizations to translate AI adoption into verifiable and governable outcomes.

7. Productivity Measurement as a Governance Mechanism

In AI assisted contexts, relinquishing output based productivity metrics often leads organizations back to input based measures such as headcount or capacity. These approaches obscure the real impact of AI and weaken governance, particularly in sourcing

and outsourcing arrangements.

Anchoring productivity to Function Points supports:

- transparency in cost per unit analysis;
- comparability across vendors and delivery models;
- evidence based decision making.

From this perspective, productivity measurement is not merely a reporting exercise but a central governance capability that preserves accountability in an increasingly automated and complex delivery environment.

8. Learning from Historical Measurement Data

Consistent application of FPA enables organizations to build and maintain historical productivity datasets. Over time, these datasets:

- improve forecasting accuracy;
- support trend analysis;
- make learning curves observable.

Such data also provides a foundation for more advanced analytics, including AI assisted pattern detection and scenario analysis, without replacing the underlying measurement framework.

9. Conclusions

GenAI profoundly affects software development practices, but it does not invalidate functional size measurement principles. On the contrary, increased variability in effort distribution makes structured, technology agnostic measurement more critical than ever.

IFPUG FPA continues to provide a stable measure of functional output, allowing productivity changes to be interpreted correctly as transformations in effort rather than changes in delivered value. By adopting a prudential, evidence based measurement approach, organizations can govern AI adoption responsibly, avoiding both unwarranted optimism and resistance to innovation.

In this sense, functional measurement is not a legacy practice, but a key enabler for sustainable productivity governance in the era of GenAI.

About the Author

Emanuele Richiusa is a Director at Business Integration Partners (BIP), a global management consultancy supporting large organizations in technology transformation, value governance, and sourcing strategies. He leads a Center of Excellence focused on Software Productivity, Measurement, and IT Should Cost, with more than 15 years of experience across enterprise and public sector environments.

His professional expertise covers functional and non functional software measurement, productivity analysis, cost and effort estimation, and vendor governance, with particular attention to modern delivery models. He is actively involved in international software measurement communities and contributes to initiatives related to the adoption and evolution of Function Point–based metrics.

Emanuele holds multiple professional certifications, including Certified Function Point Specialist (CFPS), Certified SNAP Practitioner (CSP), Specialista di Misurazione (SdM – Measurement Specialist), and ITIL.

CERTIFICATION COMMITTEE

By Cinzia Barbara Ferrero

The Certification Committee remains fully dedicated to empowering IFPUG members worldwide. Our core mission focuses on supporting candidates through their CFPS/CFPP and CSP/CSS examination journeys, while meticulously managing the Certification Extension Program (CEP) to help professionals retain and showcase their hard-earned credentials.

To meet the evolving needs of our global community, we are actively driving several key initiatives:

- **Exam Security and Innovation:** Following the successful launch of randomized question pools for the English and Italian CFPS exams, deployment is currently underway to extend this randomization standard across all other available language versions.
- **Quality Assurance:** A continuous improvement process is actively being applied to all exam texts, ensuring maximum linguistic clarity, professional relevance, and a standardized testing experience.
- **Methodological Synergy:** Close collaboration is ongoing with the Simple Function Point (SFP) Task Force to design and deliver the SFP certification exam.
- **CEP Modernization:** A new version of the CEP program has been released, alongside an updated extension application form. Professionals are strictly reminded to always download and use the latest available version of the application before submitting it to the Certification Committee. Looking ahead, the committee is actively working on a fully automated online service to entirely streamline application submissions and payment processing directly through the IFPUG website in the coming months.
- **Digital Credentialing:** Certified professionals can now instantly celebrate and share their achievements. Digital badges for both CFPS/CFPP and CSS/CSP certifications are now automatically issued and available via the Brightest platform.

Join Our Team!

The committee is actively looking for passionate individuals to join our ranks. If you hold a valid CFPS certification and want to contribute to shaping the future of software measurement standards, we invite you to volunteer.

Finally, I would like to express my deepest gratitude to all committee members. Their steadfast dedication, exceptional competence, and profound professionalism continue to be instrumental to the global success of IFPUG.

COMMUNICATIONS AND MARKETING COMMITTEE

By Cristina Nogués

The Communications and Marketing Committee (CMC) plays a vital role in strengthening IFPUG's global presence and ensuring that its value reaches key stakeholders around the world. Our committee is responsible for promoting IFPUG's mission, activities, and benefits to both current and prospective members, while also keeping the community informed about important news, events, and opportunities. Through our work, we help build awareness and engagement across the international function point community.

The CMC oversees several essential communication channels that support IFPUG's visibility and outreach. These include the *MetricViews* publication, the IFPUG website, email eblasts, and other communication tools designed to connect with members and stakeholders. Each of these channels plays an important role in sharing relevant content, highlighting achievements, and maintaining a consistent and professional message that reflects the organization's goals and values.

One of our key responsibilities this year is to bring IFPUG closer to people in all regions of the world through culturally aware strategies. We recognize that effective communication is not only about sharing information, but also about understanding the different contexts, preferences, and needs of our global audience. By adapting our messaging and outreach efforts to diverse cultural environments, we aim to make IFPUG more accessible, relevant, and engaging for professionals everywhere. This global and inclusive approach is essential to expanding our reach and reinforcing the international identity of the organization.

We also warmly invite more people to get involved and contribute to this important mission. If you have ideas about how IFPUG can better connect with professionals from different parts of the world, we would be delighted to hear them. New perspectives, creative approaches, and culturally sensitive strategies are always welcome, as they help us improve the way we communicate and grow as a truly international community. Joining the CMC is an excellent opportunity to share your voice, collaborate with others, and make a meaningful impact within IFPUG.

The CMC is always looking for volunteers. If you want to be part of the most creative side of IFPUG, please complete the IFPUG volunteer form on the IFPUG website: Volunteer - IFPUG - International Function Points Users Group.

PARTNERSHIP & EVENTS COMMITTEE

By Thiago Monarca Leone D'Durand

The Partnerships and Events Committee (PEC) continues to expand its contribution to the software measurement community through educational initiatives, strategic partnerships, and innovation proposals that strengthen the IFPUG ecosystem.

Knowledge Sharing Through Webinars

During the first half of 2026, the PEC successfully hosted four international webinars covering relevant topics in software measurement, estimation, and project management:

1. January: A Proposal for the Interpretation of Elementary Process by Josiane N. Pereira.

2. February: How Brazil Transformed Function Points into an Effort Metric for Software Contracts by Eduardo A. Oliveira.

3. March: Software Metrics in Hybrid Project Management by Analia Irigoyen

4. May: Simple Steps to Group Entities into Logical Files by Cleber Ferrareze

The webinar series will continue with the upcoming presentation:

5. Revisiting the Experience Factory in the Agile Era: Metrics, Effort Estimation, and Organizational Learning — PhD Ajax

These sessions have provided valuable opportunities for professionals worldwide to exchange experiences and discuss emerging practices in software measurement and management.

ISMA 2026 Initiatives

The PEC has actively supported the organization and execution of ISMA Virtual 2026, helping ensure the success of this important global event for the measurement community.

At the same time, preparations are underway for ISMA Brazil 2026, scheduled for October, reinforcing IFPUG's commitment to expanding regional engagement and promoting knowledge exchange across Latin America.

Strategic Partnerships

The PEC has also strengthened collaboration efforts through new partnership initiatives, including engagement with universities, creating additional opportunities for professional development and community growth.

Innovation and Community Development

Beyond events and webinars, several strategic initiatives were developed during the period:

- Creation of a proposal for a University Membership Program, designed to increase academic participation and strengthen relationships between industry and educational institutions.
- Submission of a proposal for the development of a new professional certification, expanding IFPUG's portfolio and responding to emerging market needs.
- Development of a webinar certificate monetization model, creating new opportunities for value generation while supporting educational activities.
- Establishment of a formal article evaluation and review process for ISMA, improving consistency, quality, and transparency in conference submissions.

Looking Ahead

Through education, partnerships, innovation, and community engagement, the PEC continues to support IFPUG's mission of advancing software measurement knowledge worldwide. The committee remains committed to creating opportunities for learning, collaboration, and professional growth while helping shape the future of software metrics and estimation practices.

FORECAST AND SOFTWARE ESTIMATING COMMITTEE

By Emanuele Richiusa

Over the past months, the Forecasting and Software Estimation Committee (FSEC) has transitioned from its initial setup phase into active execution, focusing on delivering tangible contributions to the IFPUG community.

Following the leadership transition at the beginning of 2026, the committee has been successfully reactivated, establishing a regular bi weekly cadence and aligning members around clear objectives and priorities.

A major milestone has been the practical validation of the resource and tool endorsement framework, moving from conceptual design to real use. This has enabled the refinement of evaluation templates, the introduction of standardized scoring mechanisms, and the consolidation of a transparent and repeatable endorsement process for tools and resources in the estimation domain.

At the same time, the committee has strengthened collaboration with NFSSC and FSSC, ensuring alignment on methodology compliance and reinforcing the coherence of the initiative with IFPUG standards.

In parallel, FSEC has defined the target model for the integration of endorsed resources within the IFPUG website, including the structure for publication, the application workflow, and the requirements needed to support the future roll out of the endorsement program.

The committee has also initiated its contribution to IFPUG's knowledge ecosystem through the uTips initiative, with a first contribution focused on the application of IFPUG methodologies to estimation in AI driven contexts, focusing on approach and process, an area of growing interest and demand across the industry (coming soon).

Looking ahead, FSEC is entering a key phase focused on the deployment of its initiatives and their adoption within the community. In this context, a recruitment effort is currently underway to strengthen the committee's team, in line with the increasing number and ambition of its objectives.

The committee is seeking new volunteers interested in contributing to software forecasting and estimation practices, including activities such as resource evaluation, development of best practices, research on emerging topics, and creation of educational content.

The FSEC continues to position itself as a driver of innovation and methodological rigor within IFPUG, contributing to the evolution of software estimation practices and supporting the community in addressing new challenges in an increasingly complex landscape.

If you are interested in contributing and shaping the future of software estimation within IFPUG, we warmly invite you to join the FSEC.

FUNCTIONAL SIZING STANDARDS COMMITTEE

By Esteban Sanchez

The Functional Sizing Standards Committee (FSSC) continues to advance several major initiatives that strengthen IFPUG's body of knowledge and support the evolution of functional sizing practices. Our primary focus this cycle has been the Counting Practices Manual (CPM) Modernization Program, where the team has completed the first draft of the modernization backlog. This represents a significant milestone: for the first time, we now have a structured, user story driven inventory of enhancements, clarifications, modern examples and new guidance needed to bring the CPM forward to the current state of the practice when it comes to software development.

The team is actively refining and prioritizing this backlog, expanding user stories, and aligning them with the modernization charter. Momentum continues to grow as more contributors join the effort. In the coming weeks, we will be issuing an e blast to the entire IFPUG community, inviting members to submit ideas, pain points, and improvement proposals. This outreach will ensure that the CPM modernization reflects the needs of real practitioners across industries and geographies.

In parallel, the committee has completed the White Paper on Multiple Media, which provides updated guidance on how to size functionality delivered across diverse media outputs. Stay tuned: This paper is now entering the final editorial cycle.

Progress also continues on two additional papers: Robotic Process Automation (RPA) and Function Points for Object Oriented Systems. Both documents are moving steadily through research, drafting, and internal review. These papers aim to clarify how functional sizing applies to modern automation patterns and object oriented architectures—topics that are increasingly relevant to today's software development environments.

The FSSC remains committed to delivering high quality, practical guidance that supports the global measurement community. With the CPM modernization gaining traction and multiple publications advancing in parallel, the committee is entering one of its most productive periods in recent years.

The FSSC is looking for volunteers. If you want to be part of the team driving these important initiatives, please complete the IFPUG Volunteer Form on the IFPUG website: https://ifpug.org/about-us/committees/volunteer.

Our mission is to serve IFPUG and its members, and we love to innovate. If you have feedback or suggestions for new projects, we definitely want to hear from you. Please submit your comments to esanchez@galorath.com.

INTERNATIONAL MEMBERSHIP COMMITTEE

By Sérgio de Quintal Brigido

The mission of the International Membership Committee (IMC) is to act as the voice of the membership. The IMC is responsible for enhancing IFPUG members' experience, increasing membership value, and supporting global growth. Through a network of country representatives, the committee provides localized support in native languages, helping members navigate procedures and engage more effectively with IFPUG resources.

From December 2025 through May 2026, the IMC made significant progress in strengthening its internal processes and advancing strategic initiatives focused on engagement and membership expansion.

A key achievement during this period was the implementation of a standardized monthly activity reporting process, improving visibility into regional interactions and enabling better tracking of member needs across countries. This initiative has supported more consistent communication and alignment among country representatives, even in periods of low activity.

In parallel, the IMC initiated groundwork for the Membership Engagement Task Force, a strategic effort aimed at defining actionable plans to increase membership globally. Although the initiative started gradually, by early 2026 the committee reinforced the importance of proactive contributions from all country representatives, positioning them as key drivers in identifying potential individual and corporate members.

Membership growth efforts produced tangible results, particularly within the academic segment. The IMC supported multiple university membership requests in Brazil, including the successful approval of the Federal University of Rio Grande do Norte (UFRN). Additional institutions and training organizations also expressed interest, demonstrating increased awareness of IFPUG within the academic community.

The IMC expanded its focus to continuous improvement initiatives, including proposals to enhance university membership models and the development of a membership survey to better capture feedback and expectations from the global community.

Another important milestone was the collaboration with the Communications and Marketing Committee (CMC) to support the translation of articles, posts, and membership instructions into multiple languages, aiming to reduce language barriers and improve accessibility for members worldwide. This initiative reinforces IFPUG's commitment to inclusivity and global reach.

In summary, the IMC's work during this period reflects a strong commitment to operational excellence, strategic growth, and enhanced member experience. These efforts position the committee as a key contributor to IFPUG's ongoing global expansion and engagement strategy.

NON-FUNCTIONAL SIZING STANDARDS COMMITTEE

By Marcello Sgamma

The IFPUG Non-functional Sizing Standards Committee (NFSSC) continues to educate the industry on Software Non-functional Assessment Process (SNAP), including what it is, the benefits of using it, how to measure specific sizing scenarios and how to train future SNAP trainers. Do you want to use SNAP in contracts and need a certification that has a renewal process, similar to the Certified Function Point Specialist Certification Extension Program (CFPS CEP)? This would help ensure your staff are properly trained and prepared to size softwarebased on non-functional user requirements measured by SNAP. IFPUG launched the Certified SNAP Specialist (CSS) certification, which complements the existing Certified SNAP Practitioner (CSP) certification. This will help you introduce non-functional measurement from a contractual perspective. New in this latest period, the CSS certification and Assessment Practices Manual (APM) have been translated into Italian, making this opportunity more accessible to native Italian speakers. Taking the CSS certification now could give measurers a competitive advantage in a market where certifications are often required for certain contracts and the number of CFPS-certified professionals is already very high. We are getting the translation of the APM into Spanish, Chinese and Korean underway!

- To help maintain CSS certification, the NFSSC periodically prepares valid CEP presentations: Measuring NFRs in Application Security Functionalities, presented at 2025 IFPUG conference, ISMA2025
- Non-functional Measurement – AI Chatbot, presented at 2025 Hybrid IFPUG conference, ISMA 2025 Hybrid
- The mystery of ADD, CHG and DEL in SNAP revealed, presented in 1EM2026 GUFPI Event
- How to Measure the User Interface with IFPUG SNAP, presented at the last IFPUG conference, ISMA 2026

We have released and are working on publishing a series of examples of SNAP uses that will help organizations complement IFPUG function points by measuring the non-functional dimension:

- A white paper on Measuring the Size of Security Requirements, published on the IFPUG website
- A white paper on "Sizing Microservices Using Integrated Approach of FPA and SNAP," approved and under publication
- Examples of sub-category 1.2 with measurements made in the field
- Examples of using sub-category 1.5, especially in the context of configurable software (like SAP, oracle application etc.)
- Using Gantt Charts to Track SNAP Work Effort (already released)
- Guidelines on sizing non-functional batches (sub-category 3.3) and multiple Input / Output interfaces (sub-category 4.2)

To continue evolving the methodology when needed, we rely on your SNAP measurement data. So, we strongly encourage you to populate the ISBSG database with data on SNAP or contact us if you prefer to make it available in other forms for the NFSSC. We will continue to enrich this list.

We are also beginning to create a structured way to estimate the number of SPs, both in the early stages of a project and to make a quick estimate of the number of SPs in some circumstances.

Other important committee activities either complete or near completion include:

- Working on a white paper with IFPUG's FSSC on measuring AI applications, from the functional and non-functional point on view.

Growing use of our first five YouTube videos providing an overview of the SNAP method, which have received 1,442 views as of this writing. We encourage you to access them by searching YouTube for “IFPUG SNAP,” “sizing non-functional software” or similar terms. Please “like” the videos if you find them helpful.

We are also working to explore another important benefit of using SNAP: the measurement of quality indices such as technical debt. Now is the time to measure this as well to enable companies to manage quality in a systematic way.

Don’t let the cost of introducing SNAP discourage your company. Many large industries are already adopting this approach and recognizing its benefits, from improving effort estimation and measuring non-functional baselines to using SNAP subcategories as a common language for communication between business areas and customers. On this point, we invite you to view the presentation from ISMA 22, “Evaluating a Change in Software Architecture: A SNAP Approach.”

The NFSSC gives you the opportunity to connect with leading professionals in the measurement of software non-functional requirements, understand the reasoning behind the practices you will later apply within your organizations and share this knowledge in academic settings.

The measurement of non-functional dimensionality in software is absolutely one of the hottest topics in the industry in recent years. For those interested in working with us on a groundbreaking topic such as non-functional dimension measurement with SNAP, please send in your application by going to https://ifpug.org/about-us/committees/volunteer.

If you would like to contact us, you can do so at nfssc@ifpug.org.

50 Millstone Road
Building 200, Suite 215
East Windsor, NJ 08520 USA

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