MetricViews | Issue 1 | July 2021

MetricVIEWS

IFPUG.org

A PUBLICATION OF THE INTERNATIONAL FUNCTION POINT USERS GROUP

How to Spread Function Point's Word

How Can Communication and the Systemic Approach Help Us for Software Measures?

FUNCTIONAL SIZING

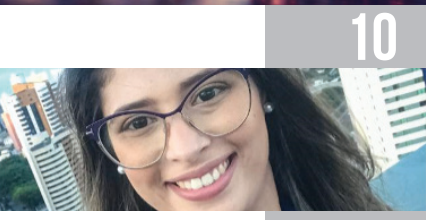
OF AGILE PROGRAMS AT U.S. DEPARTMENT OF HOMELAND SECURITY

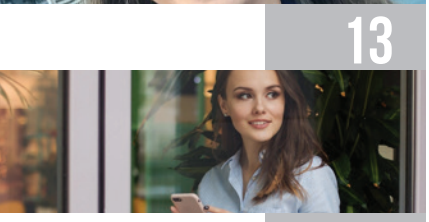
IFPUG
International Function Point USERS GROUP

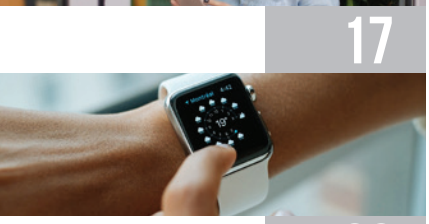
NEW WEBSITE COMING SOON

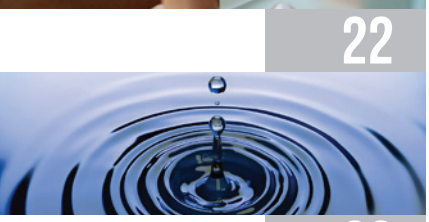
Changing to Adapt to New Times

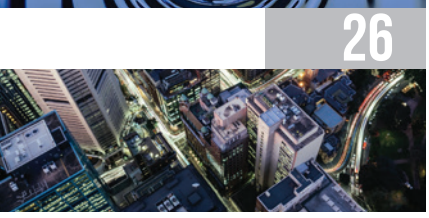
- Learning management system
- New look & feel
- Improved navigation
- Enhanced certification directory
- Archived webinars & virtual events
- And more!











COMMITTEE REPORTS

Since my first IFPUG event in 1999, I have always been buzzing with new input, comprehensive knowledge and insight whenever I join an IFPUG event or read *MetricViews.* The information and contribution show that we are all working toward a common goal—improving the success and quality of software delivery. I bring my knowledge to leadership to make them see the obvious in the advantages just like I do.

Some say that IFPUG, function point analysis and SNAP are the world's best-kept secrets. In this edition of *MetricViews,* IFPUG highlights the testimonials, benefits and competitive advantages that IFPUG standards have supported. This is exactly the type of information we need in order to share our knowledge with leadership around the world. Maybe we can get the leadership around the world to feel inspired.

Looking at the testimonials, benefits and competitive advantages shared in this issue of *MetricViews,* we clearly see that IFPUG sizing standards are current and relevant.

However, we still have less than 10% of all companies and projects using software measures and estimating techniques using these measures. The decision-makers seem to be hard to convince. I have heard words like expensive, hard to accomplish and not relevant for us. Would they be just as hard to convince on subjects like budgeting if it were not required by them? The countries, organizations and companies that have implemented usage of IFPUG sizing standards as a contractual demand to ensure transparency and competitive pricing seem to have seen the benefits and it is growing and that it is not a secret. Companies that use IFPUG sizing standards for benchmarking do not see it as hard, costly or unnecessary. For these companies, it is a way to grow their market and increase their compatibility—getting the benefits of huge cost savings on their IT budget.

We need to spread the words to this world's leadership and make a buzz about all the good things IFPUG sizing standards can support.

The information shared in this edition of *MetricViews* will support the statement that IFPUG is current and relevant and will support any leadership vison of successful software delivery. I can only hope that leadership and decision-makers will see the benefits of using both the processes coming from IFPUG, but surely also the skillset you gain from using any of our members—individuals as well as companies.

Christine Green
IFPUG President

MetricViews, the IFPUG magazine that perhaps you are reading just now in a computer, in a tablet or in a smartphone, has as objective to promote the use of IT metrics in the world, and to emphasize into the importance to measure the software for "improving." The IT size (so functional as non-functional) is the base for the most strategic metrics and IFPUG is the most world-prestigious organization when we talk about IT metrics and IT size.

If we turn back the clock 10 years, we will discover that a lot of IT products, tools, systems or programming languages that just two decades ago were in vogue as first players now they are just history. Even more…a lot of current IT technologies and trend topics existing in the year 2021 will be again history in the year 2031. But if we turn back the clock 200 years, even when computers were just fiction or the word IT was inexistent, quality, productivity and size concepts existed and were managed. Those are atemporal concepts, and for sure has the same objectives as in the beginning, in 1986.

In this changing IT world, the International Function Point Users Group (IFPUG)—as a worldwide organization that promotes the use of IT metrics and brings standard software sizing methods such as IFPUG FSM or SNAP—has the same objectives as the in beginning in 1986. The IFPUG aim (to improve the IT products/projects/enhancements measuring them) has remained stable for decades. At the same time, IFPUG has been the father of the most used software sizing methods (such as IFPUG FSM and SNAP) and the inspirator of other ones.

Perhaps you are reading those words from the north hemisphere or from the south hemisphere, from the east or from the west. In all the cases, I think that all of us have a common denominator: for sure that you are a "metrics" believer. Under this universality, in spite that this edition could be named "summer edition," this would not be a right approach: due to the aforementioned universality, in some parts of the world is "summer" but in other zones it is "winter." IFPUG is a truly worldwide and atemporal organization: the concepts that IFPUG promotes are not related to a concrete point of time nor a geographical space.

In this issue you will find a set of articles written by first worldwide metrics players, dealing with trending topics such as Agile, about experiences linking Agile and functional size, or about systemic and software metrics. Other important topics included are the importance of better size data and better estimates, flow metrics or an interview of Denise Alencar about how to spread function points in the world and more concretely in Brazil, a country where IFPUG function points are widely used.

IFPUG is an organization based on the volunteer concept; a lot of people from different parts of the world work in different committees without any kind of return. Who guides them (I include myself) is the passion for the IT metrics and to spread the importance about how they act positively in the IT projects management. I would not like to miss the opportunity to remember and to give thanks to Paul Radford, who passed away on 30 March 2021. Paul, from Victoria, Australia, was the editor of IFPUG *MetricViews* for years, member of the IFPUG Communications and Marketing Committee along the history and a great contributor of the software metrics in the world: Paul, thanks for everything you did throughout history for IFPUG.

Antonio Ferre
MetricViews Editor

FUNCTIONAL SIZING OF AGILE PROGRAMS AT U.S. DEPARTMENT OF HOMELAND SECURITY

By: Katharine Mann and Ryan Hoang

Each year, the U.S. Department of Homeland Security (DHS) invests billions of taxpayer dollars into everything from helicopters for Customs and Border Protection (CBP), vessels for the U.S. Coast Guard, baggage screening equipment for the Transportation Security Administration (TSA) and complex software systems for such purposes as administering FEMA grants, processing U.S. citizenship applications and monitoring the enforcement of illegal immigration. A 2017 report by the Government Accountability Office (GAO), noted that "in fiscal year 2016, the department's IT budget of approximately $6.2 billion USD was the third largest in the federal government." (GAO, 2017) Like many other federal agencies in the U.S. government, DHS has struggled with estimating the cost of, and establishing realistic schedules for, large IT programs.

One of the primary challenges experienced by DHS is accurately estimating the size of software development efforts. Many of these efforts result in public-facing systems and have many stakeholders with various needs, leading to complex sets of requirements. In the cost estimating field, developing an estimate is not conceptually difficult, as estimates are often just build-ups of labor; the justification of those inputs is what presents the major challenge. Agile development principles often conflict with established processes in the traditional acquisition lifecycle framework. Development teams will continually shift or add requirements as directed by the customer to deliver working software, but how can a program tell that it has completed what it originally set out to do? Understanding the true scope of programs is the missing piece to improving program management practices.

In 2017, the DHS Under Secretary for Management (USM) charged the Cost Analysis Division (CAD) under the DHS Office of the Chief Financial Officer (OCFO) to find a way to improve cost estimates for Agile software development programs. There were two primary objectives:

1. Enhance the credibility and accuracy of a software development estimate, and
2. Decrease the time required to develop the estimate.

At the time, DHS had designated five software development programs as pilots for implementing Agile processes and best practices and providing lessons learned for other DHS endeavors. In addition to being highly visible major acquisitions, these programs were at various stages of the acquisition lifecycle and had experienced common challenges with cost, schedule and performance. This provided a timely opportunity for the CAD to expand its technical knowledge of software development and attempt some novel estimating methods. From discussions with industry and government partners, CAD learned of the benefits of functional sizing techniques and identified functional sizing as a promising solution to the current dilemma.

> "CAD learned of the benefits of functional sizing techniques and identified functional sizing as a promising solution to the current dilemma.

DHS CAD has based our sizing approach on the open-source Simple Function Point (SiFP) method as published by Dr Roberto Meli (v1.01) and this forms the basis for our custom software

cost estimation process called Simple Software Estimation (SiSE). The SiSE Estimating Process combines functional software sizing (i.e., quantifying business function/transaction types, system interfaces and requirements counts from high-level acquisition documentation) together with software productivity rates (e.g., hours per function point) to determine Agile software development effort and costs. Note that we realize that there are Diseconomies of Scale (DoS) associated with software development estimating; however, we currently assume a simplified linear relationship between software functional size and productivity.

CAD research illustrates that functional requirements are typically expressed as action verbs (e.g., "submit," "maintain," "receive"), which can be decomposed to one or more weighted components (a grouping of generic unspecified transactions and/or generic unspecified data groups) based on the Simple Function Point method. Work done by functional sizing experts produced a lexicon of 140+ action verbs and their associated components. The appropriate weighting factors (a number of equivalent Simple Function Points) are then applied to these components to produce a total functional size estimate.

To understand a software system's business transactions and estimate software requirements, CAD uses a program's Concept of Operations (CONOPS), a high-level acquisition document developed early in our acquisition lifecycle that describes what functions the completed system will do. The CONOPS is reviewed and validated by the DHS requirements and technical communities to ensure all required capabilities are captured before a program moves further through the acquisition lifecycle. The SiSE sizing step leverages the action verbs used in the CONOPS written functional requirements to quickly estimate a Simple Function Point size of the software. Once the initial size estimate is calculated, additional factors and risk may be applied to the estimated size to anticipate software growth, complexity and program uniqueness.

The program office and the appropriate technical communities should then validate the final size estimate to ensure a consistent interpretation of the requirements used for the estimate. CAD uses analogous historical and industry data to determine a throughput/productivity rate used with the estimated Simple Function Point size to estimate the total software development effort for the program. This is then time-phased across the schedule to estimate the software development cost for a program's Life Cycle Cost Estimate (LCCE).

As part of the Simple Software Estimating process, the Simple Function Point estimate quantifies the size of the functional requirements for a development effort. This number, when used by a cost estimator, provides a justifiable input for estimating cost. But after performing this sizing effort to produce just one number—the Simple Function Point estimated size, is that it? No! There are many ways that our Simple Function Point size estimate, when utilized effectively by a program office, can provide maximum value by influencing many aspects of program management activities.

Developing Schedules

One of the first items that a program needs to have agreement on is the realistic duration of the software development effort. There have been studies conducted that provide metrics on development rates for functional sizing (ex: FP/team-month, etc.). Using a standard approach like Simple Function Point

with an appropriate productivity rate to estimate our software development effort (in hours or person months) we can then estimate, using historical development rates, the schedule duration to complete the software development. If a schedule was already assigned to a program, this schedule estimation can assess the reasonableness of existing development timelines. The program can then justify to decision-makers why pre-assigned milestones (or deadlines!) may be unrealistic and should be delayed or re-evaluated.

Estimating Resources

If timelines are already established, SiSE can assist program management with an easy way to quantify how many resources will be required to meet those deadlines. Using software development rate metrics together with the Simple Function Point size estimate the assigned milestone dates, an analyst can estimate the required team size and quantity to meet the desired schedule. If the available team size is insufficient, the analysis provides solid, objective justification to ask for additional program resources and funding.

Planning Agile Sprints

If based on good requirements (i.e., unambiguous, clearly stated, functional requirements), a cost analyst can use the Simple Function Point estimate, with a relevant productivity rate to estimate the effort required to develop each of those requirements. Because each requirement is objectively quantified, Agile teams can appropriately divide tasks when planning sprints and minimize potentially over-assigning work. Program managers can also use this approach to assess team throughput and ensure that they are all producing similar amounts of functionality. This approach is far more objective and applicable across teams than the alternative velocity metric (expressed as Story Points per Sprint) typically used by Agile development teams.

Reviewing Vendor Proposals

This paper describes how programs can use SiSE to assess internal schedules and resourcing. This estimating process can also be applied to assessing vendor proposals for software development services, to validate that the scope of work is mutually understood between the government and contract

offerors. The Simple Function Point sizing estimate can provide a quick cross-check to the overall amount of effort proposed, as well as gauge reasonableness of the delivery timeline and the staffing proposed to meet those dates. This will allow programs to better evaluate best-value proposals when awarding contracts.

Tracking Progress

The results of SiSE combines with other noted analyses to produce a baseline for accurate tracking of development progress. An initial cumulative Simple Function Point's "estimate to complete" chart can be plotted to project completion dates and effort, using assumed development rates and proposed staff. Plotting cumulative, delivered Simple Function Points completed after each sprint against this initial projection can provide a program manager with valuable information in the form of a Burn Up Chart. Program and project managers can track current development progress and see if the project progress is trending as planned. Deviations will provide an early indication of potential issues and allow the program to react pre-emptively. Establishing a visual representation of such progress also provides an instrument to initiate useful communication with leadership and focus the conversation on issues that require attention.

DHS leadership has supported the use of SiSE as a methodology to estimate software development size, effort and cost. They have recognized the objective nature of functional size measurement and the standardized calculations, as well as the link to functional requirements. Tracking using SiSE has begun to focus discussions of development progress on capabilities delivered rather than deadlines promised. SiSE has been used to date on five independent cost assessments with successful results. In a May 2019 memo from the DHS Acting Chief Financial Officer (CFO), all new or re-baselining major acquisition programs are required to use functional sizing for estimating software development effort. SiSE streamlines and formalizes this approach by providing a standardized software estimation process.

CAD believes SiSE offers many benefits to Agile acquisition programs. SiSE provides a faster, more reliable and repeatable process for cost estimators to produce credible estimates of functional size and development effort. The methodology leverages high-level documents created early in the acquisition lifecycle, allowing long-term analysis of system capabilities without being impacted by Agile processes that can shift development priorities. Lastly, integrating functional sizing into other aspects of program management provides additional value to program managers by tying all activities to the same requirements and can be communicated consistently to leadership and decision-makers.

The SiSE methodology is still a "work in progress." We seek to improve this methodology based on data and lessons learned by programs as they progress through software development. All CAD efforts referenced in this paper are ongoing, with the hope that the SiSE process will soon become a standard not only within DHS, but across the U.S. federal government.

ABOUT THE AUTHORS

Katharine Mann, CCEA®, is an Operations Research Analyst for the Department of Homeland Security, Cost Analysis Division (CAD) IT/Agile Software Development Team. Before joining DHS, Ms. Mann supported the NATO Communication and Information Agency (NCIA) in Brussels, Belgium, and various DoD programs. She is the Vice President of the Washington Capital Area Chapter of the International Cost Estimating and Analysis Association (ICEAA). Ms. Mann has undergraduate and master's degrees in Industrial and Systems Engineering from Virginia Tech. Ms. Mann has been a member of IFPUG since 2018.

Ryan Hoang, CCEA®, is an Operations Research Analyst for the Department of Homeland Security, Cost Analysis Division (CAD), IT/Agile Software Development Team. Since 2015, he has had experience supporting major acquisition programs across many DHS Components, including the Transportation Security Agency (TSA), Cybersecurity and Infrastructure Security Agency (CISA), and others. He has B.S. and M.Eng. degrees in Chemical Engineering from Cornell University.


indra

HOW TO SPREAD FUNCTION POINT'S WORD:

Denise Alencar, a Volunteering Experience in Brazil

By: **Christine Green, Sergio Brigido and Antonio Ferre**

During a metrics conference held in São Paulo-Brazil previous to the COVID-19 pandemic, Christine Green, IFPUG president, and Sergio Brigido, currently on the IFPUG board, met with Denise Alencar. Both were a little bit impressed with a short conversation maintained with her. Denise is a metrics professional who lives and works in João Pessoa, state of Paraíba, Brazil, the city where the sun rises first because it is located in the easternmost point of the American continent. She is a computer scientist who has been working in the IT market for more than two decades, acting since 2016 with software metrics, IFPUG CFPS and IFPUG CSP (SNAP) certified and currently is a consultant on metrics and software projects.

During this conference, Denise reported the situation of a shortage of professionals with knowledge in function points in the city of João Pessoa. As a way to meet this demand for knowledge, she mentioned her experience in disseminating that knowledge through a voluntary initiative that was offered to young students of IT from the Instituto de Educação Superior da Paraíba (IESP), where she had the opportunity to transmit a little of her knowledge about function points through a series of workshops, held previous to the COVID-19 pandemic period. Her enthusiastic experience about metrics and function points led Christine and Sergio to interview her in Brazil, and follow up with other questions to Denise (online, April 2021).

Denise, how did the idea for this training came about?

From the need to hire analysts with function points knowledge, it was identified how scarce the market for professionals with this knowledge is in João Pessoa. In view of this scenario, I jointly decided, with other professionals from the metrics team of the company I work for, to promote training in order to disseminate function points knowledge, as well as raise awareness of the city's IT market about the relevance of the metric in IT contracts.

And how did you put this idea into practice?

I needed to count on institutional support from other companies/entities, and that is where the partnership with IESP came from. The company in which I work already has a partnership with this university, which provides facilities for some company´s internal training. So, we proposed to do it for students and professionals who met some minimum criteria, for example, having already attended the discipline of software engineering. The university saw this initiative as an opportunity for students to engage in something fundamental in software development projects.

How did you transmit the benefits that students would have from participating in this training?

During the negotiation process with the university, we presented to the university the importance of the professional with knowledge in software measurement, based on the scenario in which the company I work. There is inserted into a significant number of the contracts payment for software development using function points, as well as measuring the team´s productivity. Thus, the university understood that this qualification would bring a differential to its students when they enter into the professional market.

How was the content of the workshop prepared?

With the sponsorship guaranteed for the training, initially, the concepts and the function point counting process were approached. At the end, a test was applied, where students made a case study counting and reviewed a count made by a CFPS, followed by a debate about the differences found. The workshop was held on Saturdays and had a total duration of 32 hours.

How was the training experience?

During the training, the participants discussed many topics, the most recurring were about the concept of code data and its applicability. Another topic very much discussed was how a software development effort is paid and how the financial value

of one function point is calculated from a function point unit of measure. After the training, the students positively evaluated the training with comments such as "well-crafted content, seeking to use practical examples and always accurate in answering doubts and questions about the subject addressed" or "congratulations on the work."

Denise, you forgot to mention other comments received, such as "instructor with excellent technical level, very accessible and humble." What about the feedback from the Instituto de Educação Superior da Paraíba?

The feedback from the university was very positive, since before the training started, the enrollment had already reached its maximum capacity and there was a great demand from the students. This fact motivated the university to have plans to adopt the workshop as a periodic extension course in the future. After the training, we continued to have positive feedback from the university, where we were thanked by the students for having had the opportunity to know and go further into the Function Point Analysis world.

Denise, would you do it all again?

This was undoubtedly an experience with a very positive professional impact, as it was a pioneering action in the João Pessoa market. We went from a scenario where many students and professionals who were not aware of the Function Points Analysis technique started to become interested in the method and spread the experience to other colleagues. I am putting together an action plan to promote metric workshops in the city of João Pessoa in partnership with other universities, where besides function points I also intend to bring SNAP, as well as other metrics.

Denise, how do think being IFPUG CFPS or SNAP certified impacts your professional career?

Achieving CFPS and SNAP certifications was a big step in my career, not only by learning the metrics themselves, but also by deepening knowledge in other aspects of the IT projects with which the metrics are related. With these certifications, I have the opportunity to work on strategic projects in the company I work for, where I have been widely recognized. Due to the scarcity of professionals with this knowledge here in the region, I am also having the opportunity to assemble and lead a team of specialists in software metrics in the company. My expectation is that the more and more I can contribute to the dissemination of knowledge of IFPUG metrics in the regional market, other professionals and companies can monitor how these metrics can aggregate in IT projects.

And how do you think IFPUG metrics can help to manage IT projects?

Usually in scenarios where these metrics are not used, there may be doubts about how much software is being delivered to the customer, whether the customer is paying for something he/she is not receiving from the supplier and back and forth. With the use of these metrics, it is possible to know how much software is being delivered, with greater assurance. In addition, it is possible to improve project planning, derive productivity, cost and effort. These data values are important for the analysis and identification of areas for improvement within the software development process.

How do you think that teleworking is impacting the developer productivity during these COVID-19 days?

I believe that for projects already underway and with an already structured and well-integrated development team, there is no significant negative impact on the teams' productivity. As for projects that are started in this COVID-19 pandemic scenario, I realize that they tend to have a negative impact, especially if the team is composed of professionals with little experience in projects. For these cases, actions are needed to minimize the impact and to adapt the execution strategy to this reality, where a greater management effort is also required.

IFPUG sincerely thanks and congratulates Denise for this incredible achievement! These are the type of initiatives that make IFPUG and the Function Points Analysis technique recognized around the world!

PRACTICING AGILE:
SCRUM, KANBAN OR SCALING? HOW DO YOU KNOW?

By: Joe Schofield

"Please don't tell me you're doing Agile" began the article *Keep the Baby*[1] some seven years ago. That article exposed the challenge to verify claims of agility, due to variation in definition, understanding and practice. While the Agile community has evolved since the earlier days of "if it's written down, it's not Agile," the ambiguity associated with Agile terminology has continued to proliferate. As examples:

- Are we doing iterations or sprints?
- Do we conduct Sprint 0 events?
- Is a spike an increase in effort or something else altogether?
- Wait, a scrum master is no master?
- What does a minimal viable product mean?
- Is that a scrum board, a task board or a Kanban board?

Phrases like "we do Kanban" or "we are scaling" or "we use a tool" could be interpreted with a wide variation among actual practices. Would some standardization reduce ambiguities in the Agile world today? For function point advocates, do functional and non-functional measurement standards inspire more consistency or more resistance in tribal-knowledge-driven cultures?

Let's unravel some of those topics. The notion of "if it's written down, it's not Agile" is likely rooted in the second value statement of the *Manifesto for Agile Software Development*[2] which exhorts "working software over comprehensive documentation." This phrase from 2001, seems to have been an attempt to

differentiate "Agile" from more rigorous software development approaches captured in IEEE[3] or ISO[4] standards for software engineering, or then, the Software Capability Model® (SW-CMM)[5]—the forerunner to the CMMI℠.

The *Scrum Guide*[6] might serve as an example of efforts to both write something down while minimizing documentation. The sixth version of the *Scrum Guide* was released in November of 2020; however, the first version wasn't released until 2010, nine years after the development of the manifesto, and 15 years after Schwaber and Sutherland's Scrum-themed presentation at OOPSLA in Austin, Texas.[7] Compare this 15-year gap with the first Agile principle touting "early and continuous delivery."[8] While the current version of the *Scrum Guide* contains no glossary of terms, the document is itself depicted as the *definitive* guide to scrum. The *Scrum Body of Knowledge (SBoK) Guide*™[9] by contrast, has more than 50 pages of terms in its glossary. Its first release was circa 2013. An industry standard for Scrum would promote consistency of practice in the growing number of industries where Scrum is being adopted. Many of the Scrum practices in use today have evolved ancestrally from the likes of eXtreme Programming, and it from test-driven development, while others from folklore (estimation techniques using dogs, come quickly to mind). The opportunity to consolidate and perform consistently, more predictably, persists. Technology itself seems afflicted with a preference toward innovation and trendiness over leveraging and improving. No one should anticipate any attempts to standardize Agile or Scrum soon as the branches of application and the growing number of special interest groups aren't likely to converge any more than during the codification of the manifesto. The limited output of two days of work in Snowbird, Utah, that resulted in the manifesto, could be characterized as "not how much they knew about software development, but rather, how little they agreed!"

Often overlooked in the zealous application of "working software over comprehensive documentation" is the life-long work of Takeuchi and Nonaka captured in the Socialization, Externalization, Combination, Internalization (SECI) model.[10] First released in 1996, the model strongly endorses writing down tacit "knowledge" to make that knowledge "explicit" and therefore more easily shared and assimilated with other knowledge domains. The SECI model encourages the exact opposite behavior of the "documentation avoidance syndrome," a battle cry still of many Agile enthusiasts, whose fondness of product or process documentation is asymptotically null. Ironically, the earliest use of Scrum and rugby, and a non-waterfall-like approach for product development came 10 years earlier from the same pair of authors, Takeuchi and Nonaka in their seminal article *The New New Product Development Game.*[11] To highlight and chronologically sequence these events consider:

- 1986 – Takeuchi and Nonaka reference rugby and Scrum as inspiration for new work approaches.
- 1995 – Schwaber and Sutherland, influenced by the work of Takeuchi and Nonaka, present Scrum for software development at a conference in Austin, TX.
- 1996 – Takeuchi and Nonaka's publish the SECI model for capturing knowledge explicitly, that is, writing it down.
- 2001 – 17 software "leaders" convene and develop the Agile manifesto, which explicitly exalts working software over comprehensive documentation, seemingly contrarian with the SECI model authored by the same duo that also inspired Scrum as a framework. Schwaber and Sutherland were in attendance.
- 2010 – Schwaber and Sutherland (finally) release a written description of Scrum, 15 years after introducing it.
- 2020 – Schwaber and Sutherland release the current Scrum guide celebrating its shrinkage from 19 to 13 pages despite 25 years of Scrum learning.

To a lesser degree, function points suffer a similar fate. Specialization and compartmentalization have given rise to spin-offs of functional counting. Nonetheless, IFPUG's Function Size Measurement Method[12] captures the essence and steps for functional counting enveloped in a standard since 2009. Thus far, SNAP[13] has not encountered a splintering audience of stakeholders to the extent of competing standards. Competition is often healthy for communities of practice spawning fresh ideas. Sometimes less healthy is the re-spinning of the past with a twist, sometimes for fame or a few more followers.

Kanban has become a trendy, if not the fashionable *framework* for workflow management, per Kanban activists. But the use of Kanban has grown marginally by 2-3% since 2017 and its close cousin ScrumBan by the same slim growth since 2013.[14] Distancing ourselves from words like processes and methodologies, the use of the word *framework* is used to characterize concepts, values and principles, that suit less formally structured workgroups. Teams may substitute Scrum or task boards with Kanban boards, though they often dis-include work-in-progress (WIP) limits that help to ensure flow when optimizing cycle time. The fundamentals of flow, pull, WIP limits, and value stream optimization are seldom evidenced among Kanban teams. Teams are rarely self-organized and cross-functional—two of six "characteristics" Takeuchi and Nonaka recorded.[11] Kanban column heading become silos of specialties diminishing the impact of flow and further prolonging the paradigm shift to generalists. The lack of defined roles is often supplemented with titles that conflict with self-organization and team responsibility for product. The absence of collaboration (see Agile Principle 4) from undefined meetings leads to the adoption of scrum-like stand-ups, planning and grooming sessions, as well as reviews and retrospectives—the lack of which impairs the 10th Agile Principle. In all, eight of the 12 Agile principles that impact people and teams go unaddressed in Kanban, in part due to a lack of a standard (ISO, IEEE, ANSI, etc. as examples).

Scaling occurs naturally when teams need to share dependencies, risks and release components among each other. Yet scaling too has evolved into an array of customizations and hybrids. Certainly, there are market leaders with Agile scaling "frameworks." Most

are built around Scrum and Kanban, which should give one pause for concern given the maturity of most organizational practices. With more than 80% of organizations using one form of Scrum or Kanban, a mere 16% report a high level of Agile competency or practices enabling greater adaptability.[15] Scaling with unstable or less defined frameworks could be compared to setting a building's foundation on sandy soil. A few considerations when scaling include:

- Scope of effort: projects, programs, portfolios, organization(s), enterprise
- Fully committed business co-ownership; pigs only need apply
- The loss of agility at the team level as the organization scales "up" (e. g., sizing, iteration durations, releases and dependencies, team practices, tools, and feeding reporting hierarchies)
- Affordability given the overhead associated with new architect and engineer roles, unique values and principles sometimes portrayed as "lean"
- Assessing whether to scale to fit the culture or scale to upheave the culture
- The assortment and cost of new certifications, some with annual renewals
- Where to invest training dollars: becoming better at value delivery for your business or enhancing your expertise around scaling (hint: they may not be the same!)
- Using less sophisticated scaling approaches that align with Scrum-based organizations and that may be a more natural transition based on the needs of the organization

Turning to the last phrase "we use a tool" is cause for "alert." Over at least the past five decades, tools have been used to jumpstart organizational software engineering practices. In many cases tools have served as a substitute for the underlying knowledge necessary to develop and sustain more mature engineering practices. Leadership is often pacified if not enamored, with dashboards and colorful trending charts. Unforgettable was a presentation in 2006 where I first heard the saying "a fool with a tool is still a fool.[16]" That phrase can be traced to Ronald Weinstein not later than 1989[17]; it was used in the context of medical research.

Tools aren't the issue; not knowing what you want to achieve as an organization and believing that a tool will help define that—that's an issue! Tools are facilitators for intentional and purposeful processes, but their upkeep and feeding can distract from the value-added work of the organization and be a burden to teams. They often replace the needed daily face-to-face interaction (Agile Principle 6) with reminders from the Scrum Master or (hopefully not) a project manager to update tasks and hours in a tool so more accurate status charts can be viewed. Accountability among self-organized face-to-face team members gets supplanted with email reminders around "due dates and task times." Virtual meetings

become a hangout for disinterested parties to "multi-task" rather than participate. Collaboration is diminished. Cross-functional growth is thwarted. Competence plateaus. Teaming becomes another apparent victim of the pandemic aftermath. Increasingly, tools can replace the thinking and engagement that was the heart of our value delivery.

Standardization has enhanced almost every area of our lives: devices, appliances, housing, power sources, fuel, therapies, product quality, food, nutrition, automobiles, car seats, even software. ISO alone has more than 21,000 different standards developed by hundreds of working groups from almost every country.[18] The absence of documented process and products, the rudimentary application of sparsely-depicted frameworks, the stealthy displacement of team engagement by tools that become our process, the urgency to scale outcomes that disrupt team-level agility are all exacerbated without some standardization.

The benefits of standardization such as context, consistency, completeness and correctness are worthy of our attention.

By contrast, the adoption of IFPUG's Function Point Analysis for use in governments for software planning, estimation and costing has been recognized from Brazil to Italy, Japan, South Korea, Mexico, Malaysia and Poland. International standards have a global impact on defining activities and product realization. Pioneers like Takeuchi and Nonaka remind us to transform the tacit to the explicit, discovery to knowledge, experimentation to learning, data vagueness to information richness, unpredictability towards certainty, team formation into team performance and tolerance for mediocrity into market dominance. Please don't tell me you're doing...

REFERENCES:

[1]Keep the Baby (Examining Agile); Schofield; *MetricViews;* International Function Point Users Group; Winter, 2014 / 2015

[2]https://agilemanifesto.org/; retrieved 3/22/2021

[3]IEEE/ISO/IEC 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes

[4]ISO/IEC 9126:2001 Software engineering – Product quality

[5]http://vhg.com/standards-compliance/sw-cmm-software-engineering-capability-maturity-model/

[6]https://scrumguides.org/revisions.html; retrieved 3/22/2021

[7]https://agileforgrowth.com/scrum-history/

[8]https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

[9]https://www.scrumstudy.com/sbokguide

[10]https://www.knowledge-management-tools.net/knowledge-conversion.php

[11]https://hbr.org/1986/01/the-new-new-product-development-game

[12]https://www.iso.org/standard/51717.html

[13]https://www.ifpug.org/about-snap/

[13]https://standards.ieee.org/standard/2430-2019.html

[14]Annual State of AgileTMReport; VERSIONONE, digitial.ai; 2013 – 2020

[15]Annual State of AgileTMReport; digitial.ai; 2020; pgs. 10, 9

[16]*A Fool with a Tool: Improving Software Cost and Schedule Estimation;* Ian Brown; ISMA Conference; San Diego, CA.; September, 2006

[17]*Perestroika, fashion, and the universal glue;* W M Thurlbeck; The American Review of Respiratory Diseases; 1989 May; quotes Ronald Weinstein, eminent academic pathologist in 1989 "a fool with a tool is still a fool"

[18]https://www.quora.com/How-many-ISO-Standards-are-there

ABOUT THE AUTHOR

Joe Schofield has more than 80 published books, papers, conference presentations and keynotes including contributions to the books The IFPUG Guide to IT and Software Measurement, IT Measurement, Certified Function Point Specialist Exam Guide, The Economics of Software Quality and his recently released Aligning People and Culture for Agile Transformation. He has facilitated ~200 teams in the areas of software specification, team building, organizational planning and Agile transformation. He holds eight Agile-related certifications: SA, SCT™, SSMC™, SSPOC™, SMC™, SDC™, SPOC™ and SAMC™. Joe was a CMMI Institute-certified instructor, an IFPUG Certified Function Point Specialist (CFPS) and a Lockheed Martin-certified Lean Six Sigma Black Belt. He is a past IFPUG President and for more than a decade he served as the "Chief Process Officer" of an organization of 400 software engineers.

HOW CAN COMMUNICATION AND THE SYSTEMIC APPROACH HELP US for Software Measures?

By: Tetyana Komarova and Paola Billia

Have you ever heard of a **systems approach** in the world of software metrics?

The systems thinking approach is based on the generalization that everything in the world is inter-related and interdependent. A system is composed of related and dependent elements which, when in interaction with each other, form a unitary whole. A system is simply an assemblage or combination of things or parts forming a more complex whole.

According to this perspective, the IFPUG Function Point (FP) sizing methodology can be seen as a **complex system consisting of several interrelated subsystems.** Here is a list of the components involved in the IFPUG FP process:

- The software development project documentation, from which functional and non-functional requirements can be identified
- The set of applications, modules and parts involved in defining the boundaries
- The specific terminology (ILF, EQ, boundaries, etc.) relating to the IFPUG FP method
- The architecture of the software applications (data, functions, processes, etc.)
- Concepts relating to software maintenance
- International Organization for Standardization (ISO) norms and standards
- Functional and non-functional measurement dimensions
- The purpose and scope of FP counting
- Project stakeholders
- The FP counting documentation
- The presentation of FP counting results

A system is not only characterized by its components, but also by the links between them. In this article, we examine the link between the presentation of FP counting results and the stakeholders of the project. What is the link between these two parts of the system? The link is the communication between those who perform the IFPUG FP process and present the results, and the stakeholders of the project. The former is often on the supplier side of a software development project, while the latter is often on the customer side. This linkage, if not well managed, can become the Achilles heel of the entire system, with results ranging from success to downright failure of a software development contract. The question arises: how can two parties who both desire successful outcomes in a contracting environment end up in conflict over the presentation of FP counting results? This article analyzes how communication and understanding of human behavior can streamline FP-based contracting between suppliers and their customers.

Case study: Determining the cost of an enhancement project of an application

Let us focus on the case where the purpose of counting is to determine the cost of an enhancement project. At the completion of an enhancement project, the supplier presents the measure of the functionality added, changed or deleted, but it may happen that the points of view of the customer and supplier are different. For example, some functionalities have been modified and the supplier considers the changes made as improvements, while the customer considers them as corrections.

The stakeholders involved in this activity can be divided into two groups: those who develop the project (**the supplier**) and those who sponsor the project (**the customer**) who must pay the supplier for what has been achieved. Communication between the two is an essential part of project success. The objective of the supplier's group should be to carefully collect all software products and documents that have been delivered in order to include them in the FP count. Practically, the **primary** objective of the supplier's group is to satisfy the customer, collect their requests, communicate with them, check which functions need to be added, changed or deleted due to customer requests considering all related projects and follow them up promptly when change requests occur during the SW development cycle. The project documentation used to support the count is developed for a **secondary** objective, that of providing a record of the software development, and as such, the descriptions of certain functionalities are often omitted, even if requested by the customer, because they are considered implicit and taken for granted as part of the development. Examples include descriptions of the application backend, and the logical files inherent to the operating system (textual log files, system clocks, etc.). What this means is that there may be software functionality that is not explicitly stated in the project documentation that may be missing when one does the FP count.

What should be kept in mind for good communication about the FP count?

In most cases, the FP counting results are presented and then discussed in a team meeting of supplier and customer representatives. It is often the case that neither the supplier presenting the results nor the customer receiving the results have participated in the IFPUG FP process, but both parties need to understand what the results mean. During these discussions, the human factor comes into play and can lead to conflict, especially if the final FP count varies greatly from the initial estimates. The discussion of such increases in FP count results, if they are presented without adequate context or explanation, can result in unpleasant surprises and conflict for one of the parties involved in the project. Let's explore how this can happen, and what can be done to mitigate any potential negative consequences should conflicts occur.

Cognitive biases should be kept in mind in order to reach the goal

Wikipedia defines cognitive bias as: "[...] *a systematic pattern of deviation from norm or rationality in judgment. Individuals create their own 'subjective reality' from their perception of the input. An individual's construction of reality, not the objective input, may dictate their behavior in the world. Thus, cognitive biases may sometimes lead to perceptual distortion, inaccurate judgment, illogical interpretation or what is broadly called irrationality".* In short, the errors of our mind are not perceptible to ourselves.

Curse of knowledge

One such cognitive bias is called the "curse of knowledge." This is a cognitive bias that occurs when an individual, in communicating with others, unwittingly assumes that everyone in the discussion has the same basis of understanding about a particular subject.

What does this mean in our specific case? We can deduce that both the measurement specialist and the developers are convinced that the customer receiving our information is in full possession of the logic, terminology and everything else that is "behind the scenes" of the FP count for the project.

Invalid arguments may arise because the participants are unaware or uninformed about what the FP number means, and also about the FP counting process, what goes into it and what the count actually represents (the size of the software being delivered, worked on or enhanced). Suppliers often argue that FP counts should be higher (implying more work) while customers often argue that FP counts are too high (implying that they should be paying less) all the while not understanding the concepts involved in the first place. If we do not address this "curse of knowledge" directly, it can result in misunderstandings, arguments and broken trust—all of which could have been avoided by presenting more information to explain the FP count (it is better to provide a primer on fundamentals to go along with any new concept—even as background reading—than to risk people not understanding and not asking about those things of which they have little knowledge. Note that as a supplier, we need to realize that it can be intimidating and embarrassing to ask what may seem like a basic question in a room full of suppliers and customers. Make it safe to ask questions by being patient and putting yourself in your customers shoes).

So, what should we do to avoid falling into the trap of the "curse of knowledge?" We should stick to the IFPUG Counting Practices Manual that tells us to "gather all the documentation we can," but when we cannot find it, supplement it with knowledge and input from subject matter experts who know about the software's functionality. In any case, we should consult with the development team (supplier group) to ensure that the documentation gathered for the count is complete and does not exclude countable software functionality. In addition, we need to realize that FP counting is not easy for people to understand (even some seasoned FP practitioners don't really understand that it is not the entire size of software!).

In our case study, it is important to:

- Define very well and fully document the boundaries of the applications included in the project FP count.
- Document and count any implicit requests for functionality that are fulfilled by the project, even if they are not traced in the documentation.
- Remember that logical files may not have corresponding tables within the databases.
- Understand how software components interact with each other so that elementary processes are not overlooked during the FP count.
- Check the congruence between the counted data and the corresponding transactional functions, because it is difficult to have new data that is not treated by any function (for example, an ILF that has no External Input (EI) associated with it would indicate either: a) that the EI transaction function is missed; or b) that the ILF is actually used only for reference and should have been counted as an EIF).
- Carefully assess the functionality of an enhancement project in order to exclude corrective and perfective maintenance that do not contribute to the enhancement project functional size.

Also, if people use terms in their language that are unknown to you, do not hesitate to look up their meaning or ask the person using them.

Clear shared language is the basis for effective communication. Avoid using terms unknown to others or share them before starting the conversation.

Fundamental attribution error

Another bias to take into account when talking to the customer would be "fundamental attribution error."

Wikipedia defines it as "the tendency for people to under-emphasize situational explanations for an individual's observed behavior while over-emphasizing dispositional and personality-based explanations for their behavior."

What does this mean? It means that often the supplier/customer discussion might become "heated" not because the count is

incongruent or incorrect, but because we have instead uttered a sentence or used a tone that might seem offensive to the other person.

Each person has a perspective and position based on their previous experiences, beliefs, values and culture. It is therefore to be expected, that people, even in working environments, have opinions, expectations and needs that are different from our own. This is a fact. Given that this bias is present in every discussion involving two or more people, a good strategy is to accept that because of these differences, it is important not to insist that your way is the only right way in every discussion.

So, let's try to identify common interests and focus on the goals that the other person needs to meet, and look for solutions that are inexpensive for our group to resolve (the supplier) and have a satisfactory value for our customers.

A practical example

If we present to the customer a FP-based cost estimate for fixing or redoing the front-end interface of the application (corrective maintenance) and it seems to be too expensive, we know that there may be a disconnection between our understandings of the project. Additionally, the customer complains about current system performance and the fact that our supplier proposal does not seem to indicate that we planned specific interventions for their improvement. Our new proposal outlines the solution and would include shifting part of the query load directly to FE components by preloading the data and thereby achieving improved performance. The supplier/customer misunderstanding came about because of the aforementioned fundamental attribution error—differences in understanding and opinion. If we had not recognized this bias, it could have ended in conflict. When we sat down and discussed the proposal, we explained that this performance solution was planned into the proposal, and the solution does not increase the development costs for the customer, and the customer had the advantage of improved performance at the same time the other front-end maintenance was done.

Win-Win approach

In customer/supplier joint meetings, the first thing to remember is that the best negotiation is a win-win negotiation that results in a win-win solution for both parties. When it comes to discussions about the FP count, this means reaching a consensus agreement that satisfies everyone, even though it may happen that the final count does not match the software size that was initially thought by all of the parties.

In a practical example, a customer/supplier joint meeting where a FP count based invoice was presented to the customer, the customer was surprised that the amount of the invoice was higher than anticipated (due to scope creep and implicit requirements).

The proposed solution that satisfied both parties was to split the invoice into two parts—before the payment of the measure estimated and subsequently the difference with respect to the estimate. In this way, the customer has time to foresee an increase in the budget to cover the FP that had not been estimated.

Negotiations lead to future agreements

Always remember that obtaining a mutual benefit from a negotiation forms the basis for any future agreement.

Before starting the discussion on estimations, we have to keep in mind that the user's point of view may differ from that of the supplier. Alternative solutions to those we are going to propose, which are equally acceptable and valid, should be considered. In this way you will have a back-up proposal to present in case the other party does not like "plan A."

Conclusion

During supplier/customer meetings, always keep in mind that the other party might have very different goals from yours, and their perspective may be quite different from what you have imagined.

Keys to understanding the real needs and expectations of the other party lie in using empathy and active listening to understand the other person's perspective. Try to relate to them effectively, ask focused questions to understand if there are any implicit requirements that need to be evaluated and highlight the added value of FP counting to illustrate, in customer language, the size and scope of the work being done.

Happy counting!

REFERENCES:

1. Camerer, Colin; Loewenstein, George; Weber, Martin (1989). "The Curse of Knowledge in Economic Settings: An Experimental Analysis". Journal of Political Economy

2. Fischhoff, Baruch (1975). "Hindsight is not equal to foresight: The effect of outcome knowledge on judgment under uncertainty". Journal of Experimental Psychology: Human Perception and Performance

3. Lagdridge, Darren; Trevor Butt (September 2004). "The fundamental attribution error: A phenomenological critique". British Journal of Social Psychology

4. Bell, S. T.; Kuriloff, P. J.; Lottes, I. (1994). "Understanding attributions of blame in stranger-rape and date-rape situations: An examination of gender, race, identification, and students' social perceptions of rape victims". Journal of Applied Social Psychology

5. Getting to Yes: Negotiating Agreement Without Giving In Hardcover – April 30, 1992

by William L. Ury (Author), Roger Fisher (Author), Bruce M. Patton (Author)

6. Myshlenie. Sistemnoe issledovanie (Russian) Hardcover – January 1, 2018 by Kurpatov A. (Author)

7. General System Theory: Essential Concepts & Applications Di Anatol Rapoport

8. Features of the system approach to management of social and economic development of regions. Sherzod Mustafakulov

9. Heuristics and Biases: The Psychology of Intuitive Judgment 2002 Thomas Gilovich, Daniel Kahneman

10. Wikipedia: https://en.wikipedia.org/wiki/Cognitive_bias

11. Wikipedia: https://en.wikipedia.org/wiki/Fundamental_attribution_error

12. The Art of Systems Thinking: Revolutionary Techniques to Transform Your Business and Your Life (Joseph O'Connor, Ian McDermots)

ABOUT THE AUTHORS

Tetyana Komarova (Rome, Italy) has more than 20 years of experience in IT. She is CFPS since 2015 and is certified in CSP (SNAP) since 2020. Tetyana is the SW developer, analyst and metrics expert at NTT Data Italia, a multinational company focused on innovative IT solutions. She is in charge of verification service for estimates and counts and consulting service for the preparation of estimates and counts. Working in foreign countries and collaborating with different professionals, she has learned to easily relate to people of different nationalities and cultures, interpreting and adapting communication and requests to different personal, religious and cultural sensitivities. She is passionate about the world of behavioral economics.

Paola Billia (Rome, Italy) is Project Manager at NTT Data, expert in software measurement methods. She is CFPS since 2015 and is certified in CSP (SNAP) since 2020. Since 2017, she has been appointed to the Executive Council of the Italian Software Metrics Association (GUFPI-ISMA). She has 30+ years of experience with ICT first in the world of banks and later in telecommunications. Always interested in the topics of the software measurement and quality, she is a point of reference for the metrics, providing support and coaching on all issues regarding this topic. She is advisor on sizing issues for different customers: Telco & Media, Public Administration, Energy and Banks.

Better Software

STARTS WITH BETTER SIZE DATA AND BETTER ESTIMATES

By: **Carol Dekkers, CFPS (Fellow), PMP, CSM**

One of the most elusive and challenging questions for software experts today remains: what combination of project "ingredients" results in the most successful software projects today? In the U.S., we have often relied on the Standish Group's regularly published CHAOS reports, as a pseudo-gauge of software project success since 1996, as a guidepost to far as why some projects succeed while others fail. Over the past decade, the reports show that Agile software development projects are typically better at being on-time and on-budget, but even then, success happens less than 50% of the time.

Given a quarter century of knowledge and reporting, combined with advancements in technology and tools, one would expect to see industry improvement and better results over the years, but that trend simply has not materialized.

How can this be? Doesn't it seem ironic, given that software engineers and development teams boast some of the brightest minds on the planet, that delivering projects on time and on-budget seems to be so difficult?

I have been personally puzzled as I have watched our industry improve with new tools, better, more productive collaboration, dedicated individuals and a commitment to success on projects—yet the percentage of successful projects simply flatlines around 40%.

That was, until I started my current contract role as the lead author for the International Cost Estimating and Analysis Association (ICEAA) Software Cost Estimating Body of Knowledge (SCEBoK) in May 2021—and then it occurred to me… Software projects end up over-budget and behind schedule, partially because the cost estimates and schedules on which their contracts and projects were based were both unrealistic and overly-optimistic from the start. It is not rocket science to realize that unrealistic estimates lead to under-funded projects and unreasonable deadlines, but without updated knowledge of how to do better estimating, nothing will improve. It is like the adage attributed to Albert Einstein that is something like "insanity is doing the same thing over and over and expecting different results." While the software development industry as a whole is responsible for delivering high-quality software better, faster and in an efficient and effective manner, shouldn't it be up to software cost estimators to create realistic and reasonable estimates so that said projects can succeed?

The ICEAA and its members realize that better program estimates result in better program outcomes, and the inclusion of SCEBoK as a core competency (and an upcoming new certification program) are proof that software cost estimating has emerged as an important profession.

This article outlines how this emerging SCEBoK positions the software industry to create better software estimates based on solid data. This data includes, for the first time in a professional cost estimating curriculum, Functional Size Measurement (function points) as a viable software sizing unit of measure.

Why is Software Cost Estimating Important?

Increasingly, software is as important a component in product development as is hardware, facilities and other components. Software plays a major role in products ranging from cars to smart highways to bridges to satellites to weapon systems to embedded aircraft systems where automation plays a part. Traditionally, these types of program estimates relied on the high-dollar construction or hardware components as the drivers of program cost and schedule, however, software is emerging as big-ticket items in software-intensive programs, even when the software does not play a dominant role. Delays or cost overruns on software components can, and do, impact the delivery of entire programs. For example, on a Netherlands toll-highway project, delays associated with the toll-booth software caused the entire highway project to miss its anticipated opening. No one had predicted or planned for delays in software to impact the overall project's critical path.

As our world becomes more inundated with "smart-everything-products," software cost estimating plays an ever-more-important role.

Software Development Estimates are a Function of Software Size

Unlike hardware, where the design and fabrication of the first product is cost-intensive and followed by a production period where unit costs as the volume increases, software estimates are driven by the design and development of a single product—a software system and experience diseconomies rather than economies of scale as the size increases. One of the major contributors to software-intensive program cost and schedule, lies in software development, for which labor costs are a major contributor. Software development effort is a function of software size, for which units of measure range from the traditional equivalent source lines of code (ESLOC) to Functional Size Measurement units (function points), to Reports, Interfaces, Conversions, Extensions, Forms and Workflows (RICE-FW) for packaged software. Depending on the type of estimate being created, when in the software life cycle the estimate is being created and the type of software acquisition (procured or developed or a combination), software size is often an important factor. For procured or purchased software packages that must be tailored and/or configured and/or customized and/or requires "glue code" for proper implementation, the choice of sizing methods may vary. Different estimation techniques are specific to different types of software development, software procurement, software sustainment and depend on the availability of both historical and project data.

Software cost estimators must be able to work with function point experts and development teams, as needed, to validate that the software size is aligned with the estimate that is being created. Confidence in the data that are the foundation of the estimate is key to creating a realistic software estimate.

While software size is one of the most important cost drivers, it is also important to note that it is not the only one. When analyzing historical data to develop appropriate cost estimating relationships (CERs), software cost estimators must also consider characteristics pertaining to the complexity of the software, the capability of the development team, as well as contextual project characteristics. Data analysis and normalization, while not often discussed in function point circles, cannot be overemphasized when creating software estimates—we need an apples-to-apples comparison when data are to be made comparable.

Realistic software development cost and schedule estimates can be created through statistical analysis of relevant and comparable historical software development data, but a foundation of solid data must be present. In lieu of such historical data, published CERs and/or data repositories such as the International Software Benchmarking Standards Group (ISBSG) development and

enhancement repository or commercial estimating models can be important estimating tools. The best estimating technique for your estimating job relies on good data and equations that are appropriate for the estimate.

Scope and Purpose of the Estimate are as Important in Cost Estimating as They are to Function Point Analysis

As a software measurement practitioner or Certified Function Point Specialist/Practitioner, you know that defining the scope and purpose for a function point count/estimate is the first step to performing a function point count, establishing the scope and purpose for a software estimate is the first step to creating a realistic software estimate. Knowing what components are to be included in an estimate and what type of estimate is to be created are pre-requisites to any estimate and are fundamental to creating a realistic estimate.

Document Assumptions, Set Ground Rules, Establish a Qualified Estimation Team

Once the scope, purpose and fundamental aspects of the software estimate are established, it is important for the cost estimator to document their assumptions, establish ground rules and find the necessary data for the current project as well as any data that are available for similar completed projects. The amount, quality and availability of data, for both your current project and that of historical data, can make the difference between a realistic estimate and one based on supposition. Ensuring that data are high quality, reliable and understandable are significant considerations for selecting how to create the estimate as well as how good the resultant estimate will be.

As we progress through the guidance and lessons in the SCEBoK, we follow a standardized estimating process and rely on established best practice advice gleaned from the software industry, measurement and cost estimating experts, academia, benchmarking and commercial tool vendors, and established industry guidebooks and standards. The goal of SCEBoK is to increase the overall maturity of software cost estimating through straight-forward, formalized steps and techniques, that cover the major gaps and missing pieces that come from ad hoc, informal estimating practices. The SCEBoK curriculum includes the following major lessons (currently being finalized):

- Introduction of SCEBoK – audiences, intended usage, why software cost estimating is important
- Software development paradigms
- SCEBoK five-step estimating process
- Creating the software estimate and its components: equations, approaches, techniques and examples
- Software sustainment (including DevSecOps, software maintenance, software changes and overall sustainment activities)
- Software procurement and packaged software
- Software size (and considerations)
- Software productivity (and its considerations)
- Commercial estimating models
- Summary: templates and checklists

SCEBoK Data Sources

SCEBoK is based on size-agnostic guidance and presents the software cost estimator with choices about the approaches and techniques for estimating different types of software acquisition ranging from 100% custom software development efforts to 100% procured software solutions to various hybrid solutions along the spectrum. Data and knowledge content for SCEBoK were based on sources considered to be industry leaders including:

- ICEAA's established Cost Estimating Body of Knowledge (CEBoK), which is the foundation for SCEBoK
- Published handbooks including the U.S. Government Accountability Office (GAO) Cost Estimating Guide 2020, the GAO Agile Guidebook (2020), U.S. Federal Agencies and U.S. Department of Defense Cost Estimating Handbooks and standards (including the Defense Acquisition University Software Cost Estimating curriculum BCF-250 and others
- ISO/IEC Software and Systems Engineering Standards including ISO/IEC 12207 (Software Life Cycle Processes), ISO/IEC 29148 (SLC processes: Requirements Engineering), ISO/IEC 14143 Functional Size Measurement and ISO/IEC FSM standards including IFPUG, Nesma, COSMIC)
- University of Southern California's COCOMO II published CERs and research
- CMUniversity's Software Engineering Institute (SEI) research papers and models
- ICEAA, SCAF, ISPA/SCEA and IT CAST presentations and research by worldwide software cost estimating experts
- Many other published and research sources

Why is SCEBoK and Software Cost Estimating Important for Function Point Practitioners Worldwide?

As a Certified Function Point Specialist (Fellow), my 25+ years of software measurement experience has taught me that function point measurement is only valuable to clients if it serves a worthwhile purpose and delivers value beyond a mere number. For software development, the developed function point size quantifies the size of the software that was developed or built (and includes conversion functionality), while the delivered (installed) function point size quantifies the size of the software application being sustained, maintained and supported. When it comes to customization of procured software, an enhancement project function point size quantifies the size of customizations or glue code required to piece together different software packages.

Knowing what function points to count based on the counting (or estimating) purpose and scope are important elements in software measurement. When we use an estimated software size (in function points) as the basis for current estimates, we also need to know how big previous software development efforts and how much effort the development required, as well as the characteristics that put such efforts into context.

Given the many components, activities and breadth of software-intensive programs, it is no wonder why some software cost estimates take weeks and months to prepare—there are so many different (moving) parts with so much depth, requirements uncertainty and activities involved. Having better and more reliable historical software size, effort and contextual data available, and better measures of our proposed project size will lead to better estimates overall. Combine that with a solid, formalized estimating method and we are on our way to higher levels of software estimating maturity.

Summary of SCEBoK and Good News for Function Point Professionals

There are many more aspects beyond what I have the space to cover here in this article that are part of the SCEBoK. Join me as we finalize and begin to roll out the ICEAA SCEBoK and prepare to play a larger role as a function point practitioner and specialist as more software professionals and software cost estimators discover the power of software functional size measurement.

After SCEBoK is released in the second quarter of 2021, I am looking forward to working with software professionals who are interested in improving software estimating practices in their own companies both here in the United States and abroad. We can do better to estimate software projects and achieve better levels of software success. When projects are based on realistic, rather than overly optimistic, estimates, the whole industry will benefit.

ABOUT THE AUTHOR

Carol Dekkers is founder of Quality Plus Technologies Inc. with more than 25 years of experience leading initiatives for companies that want to succeed with software measurement and project management. She has co-authored several text-books (The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement; Program Management Toolkit for Software and Systems Development; Practical Software Project Estimation and others) and has published articles in a wide variety of software industry journals. Currently, Carol is the lead author of the ICEAA Software Cost Estimating Body of Knowledge (SCEBoK) and is an IFPUG Certified Function Point Specialist—CFPS Fellow. Ms. Dekkers also holds certifications as a PMP, Professional Engineer and Certified Scrum Master, and is an IFPUG Past President and Chair of the IFPUG Industry Standards Committee.

SIX IMPORTANT FLOW METRICS

By: Thomas Cagley

Getting the most value out of a process is important to any leader. Balancing getting the most value with getting value sooner complicates the discussion. In some cases, getting some value sooner is worth more than the same value delivered later. Guiding the delivery of value is more complicated than rank ordering a list of user stories and then magically hoping that everything will happen in the most effective and efficient manner possible. Measurement is an important tool to help teams and organizations ask the right questions. To borrow an idea from Daniel Vacanti's *Actionable Agile Metrics for Predictability,* measurement helps people ask the right questions sooner. The following six flow metrics provide process transparency into organizations that leverage continuous flow, scrumban and/or Scrum as the basis for their Agile implementations:

1. Work in progress
2. Story Escape Rate
3. Cycle time (two ways)
4. Throughput
5. Velocity

The six flow metrics, while useful in their own right, provide even more transparency when linked via Little's Law. Little's Law is a mathematical observation that the number of customers in a system equals the average arrival rate multiplied by the average time the customer is "in" the system. The assumptions that allow Little's Law to hold for flow metrics are an excellent definition of the discipline needed to make Agile techniques effective and efficient (in addition to being responsive). The six flow metrics provide useful feedback to teams about the process compromises that teams are making in the heat of delivery. The basic assumptions Vacanti identified are:

1. Average input and output are equal
2. All work started is completed
3. WIP should roughly be the same at the beginning and end of the period being measured (this will not be true under Scrum, which follows an s curve)
4. Average age of WIP is not increasing
5. WIP, cycle time and throughput are measured using consistent units of measure

Assumptions 1 and 2 reflect the conservation of flow.

Assumptions 3 and 4 equate to system stability.

The fifth assumption is all about math.

Metrics:

- **Work in progress (WIP)** is defined as the amount of work that has arrived to be worked on in a system and has not yet exited the system regardless of whether the item is actively being worked on or being delayed.

Formula: WIP = total number of stories accepted into the sprint at ANY time between start and end of the sprint that is not done (deployable).

WIP should be tracked on a daily basis.

Notes: In a Scrum-based approach, WIP will begin at zero and end at zero. In continuous flow or Kanban, the WIP should be relatively constant. (Test assumptions 1 and 3)

- **Story Escape Rate (SER)** is defined as the number of stories that are not done (deployable) by the end of the sprint.

Formula: SER = Stories Not Done / (Stories Done + Stories Not Done)

Review at retrospective and publish to other teams.

Notes: All work started should be completed in a single sprint. This metric is oriented to Scrum or Scrumban. SERs greater than zero (especially if widening) need to be addressed in team retrospectives. (Test assumption 2 Velocity)

- **Cycle time (CT)** has two competing definitions. In the more typical definition, CT is defined as the amount of elapsed time that a work item spends as WIP. Cycle time is a direct reflection of the calendar, which is the one element every customer understands. Cycle time (also called lead time or flow time even though there might be slight differences in the definition) includes ALL of the calendar time between starting and completing. The second and perhaps more important definition of CT includes lead time in the equation. This version of the cycle answers the question

of how long it takes for a piece of work to be imagined (put on the backlog) and then to be delivered.

Formula 1: CT = Date Item is Completed – Date Item is Started

Average CT = Sum (Date Item is Completed – Date Item is Started)/Number of Items

Formula 2: CT = Date Item is Completed – Date Item is Put on the Backlog (use the original item for stories that are broken down)

Average CT = Sum (Date Item is Completed – Date Item is Put on the Backlog)/Number of Items

Review at retrospective and publish to other teams.

Notes: Increasing cycle time metrics are a sign of demand outstripping capacity and/or process bottlenecks. (Test assumption 4 noted in Part 1)

- **Throughput (TP)** is a measure of the number of items that transverse the process in any given period. Throughput can be thought of as departure rate, i.e., how many work items complete and leave the process for a given period.

Formula: TP = Number of Items Completed Per Sprint

Review at retrospective and publish to other teams.

Notes: Throughput is a tool to predict how much work a team can accept. Throughput compared to arrival rate provides input on the stability of the process (higher arrival rate will negatively impact cycle time and throughput). (Test assumption 1 and 4 noted in Part 1)

- **Velocity (V)** is the average number of Function Points a team delivers in an iteration. Velocity is affected by changes to the team's capacity.

Formula: V = Sum (Function Points Completed)/Number of Sprints in Sample

Review at retrospective and publish to other teams.

Notes: Velocity is an adjunct to the throughput metrics and is used as a tool to predict how much work a team can accept.

A bit of lagniappe (a little extra): **flow efficiency** is the ratio of total elapsed time that an item is actively worked on compared to the total time it takes for something to be completed. This metric is similar to the burden rate (the ratio of non-engineering time/total time to develop and deliver).

We measure to ask the right questions. The information gleaned from everyone the metrics noted above provides input and feedback into the process of managing the flow of value.

Metrics alone are rarely sufficient; we still need a mind (or minds) to weigh context before making decisions. The six (if we don't consider flow efficacy) metrics provide a powerful set of tools to generate information about the flow of value. Delivering value is the only reason Agile teams exist.

ABOUT THE AUTHOR

Tom Cagley is a consultant, speaker, author and coach who leads organizations and teams to unlock their inherent greatness. He has developed estimation models and has supported organizations developing classic and Agile estimates. Tom helps teams and organizations improve cycle time, productivity, quality, morale and customer satisfaction and then prove it. He is an internationally respected blogger and podcaster for more than 11 years focusing on software process and measurement. His blog entries and podcasts have been listened to or read more than a million times. He co-authored *Mastering Software Project Management: Best Practices, Tools and Techniques* with Murali K. Chemuturi. Tom penned the chapter titled "Agile Estimation Using Functional Metrics" in *The IFPUG Guide to IT and Software Measurement.* His certifications include CFPS, IT-CMF Tier 2 Certified Associate, CSM, SAFe SPC, TMMi Assessor and TMMi Professional.

CERTIFICATION COMMITTEE

By **Cinzia Ferrero Committee Chair**

The Certification Committee (CC), in addition to the normal IFPUG CFPP/CFPS and CSP exam support activity, is working hard on a variety of projects. An analysis is underway to make the online exam process more efficient in order to make it easier for candidates to take the exams. For this reason, a series of tools available during the exam are being updated and a video has been created that explains how the PearsonVue platform works and technically simulates the exam. Soon the video, already available in English, will also be available in Italian. A short exam simulation (Definition, Implementation, Case Study) is being prepared to make it clearer to candidates how to best prepare to take the exam effectively. The simulation will be available in English, Brazilian Portuguese, Italian and Spanish.

The CC is working to translate the CFPP/CFPS online exam into French and Korean as well. The overview and guidelines, published on the IFPUG website, for the CFPP/CFPS and CSP exams in English, Italian and Brazilian Portuguese, are being revised and updated. The revision and updating of the CFPS Certification Extension Program is underway (in particular for the introduction of online events).

Some members are working in conjunction with the Non-Functional Sizing Standards Committee on the creation of the Certified SNAP Specialist (CSS), on the creation of a program that allows the upgrade from CSP to CSS, and on the creation of a CEP program for the Certified SNAP Specialist. The CC collaborates also with the working group that created and is testing the forthcoming Benchmark Vendor Assessment Process.

We are also pleased to announce that former CC Chairman Mahesh Ananthakrishna has kindly agreed to serve as the new Vice Chairman.

FUNCTIONAL SIZING STANDARDS COMMITTEE

By **Daniel B. French**

As the pandemic has continued into 2021, the Functional Sizing Standards Committee (FSSC) has diligently continued working on our many projects. The case study update and UML white paper have been published. The new IFPUG branded Simple Function Point manual is soon to be published as well as a joint project with the NFSSC on Boundaries and Partitions. Work is progressing on the Elementary Process paper, which should be published shortly, and Mobile Applications white paper. New projects are also underway on a counting MicroServices white paper and counting Use Cases iTip.

I would like to thank the members of the SiFP task force, Chuck Wesolowski, IFPUG Vice President, who led the task force; Talmon Ben-Cnaan, NFSSC Chair; FSSC member Esteban Sanchez and the SiFP creator Roberto Meli who were all instrumentational in the aligning of the manual with the IFPUG Counting Practices Manual (CPM) and IFPUG rebranding of the Simple Function Points manual, thanks to IFPUG. Once the SiFP manual is published, a task force, under the leadership of Talmon, will be working with the methodology and develop a certification exam.

We are sorry to announce that one of our newest members, Kiran Yeole, has left the committee. If you are interested in joining the committee or working as a non-member volunteer on any current or future projects, please complete the IFPUG Volunteer Form and send to ifpug@ifpug.org. The committee appreciates the support of the IFPUG membership and is always looking for new projects to work on. Some topics under consideration for our next projects include Agile and Cloud, which will likely be assigned to one of our new members. We welcome suggestions from members on topics of interest. Please submit your suggestions to dfrench@cobec.com.

PARTNERSHIPS & EVENTS COMMITTEE

By **Sushmitha Anantha**

Events and strategic partnerships with other organizations continue to be the key focus areas for the Partnerships and Events Committee (PEC). We started 2021 with Sergio de Quintal Brigido as our new board liaison. Former board liaison Filippo De Carli has taken up the role of Treasurer of IFPUG.

PEC hosted some interesting webinars in late 2020 within the Knowledge Café series, including “Navigating the Minefield – Estimating Before Requirements are Complete” by Carol Dekkers and “Trends in Agility – Implications for Function Measurement” by Joe Schofield. We opened 2021 with a talk on “Driving Culture Change Through Software Functional Metrics” by Steve Woodward, Kristin Curran and David Lipton. In the month of March, we hosted an introductory session on SNAP: “SNAP for Beginners: Why, How, What” delivered by Talmon Ben-Cnaan, Chairperson of IFPUG Non-Functional Sizing Standards Committee.

We organized the ISMA 18 virtual conference on 24 June 2021, which was a short version of ISMA. The event opened with IFPUG President Christine Green’s address in which she covered IFPUG functions and activities in detail. In addition, there were three presentations covering various aspects of function points sizing and project management concepts. This virtual conference offered IFPUG members a one-year Certification Extension Credit for attending the event. The event was energetically driven by Tom Cagley as a host and was attended by more than 300 people.

If these events caught your interest, please write to pec@ifpug.org with your suggestions for topics and speakers and we shall try to host them during our next webinars. If you are interested in working with PEC, please send a volunteer form to pec@ifpug.org.

COMMUNICATIONS AND MARKETING COMMITTEE

By **Julián Gómez**

A lot of changes have come to IFPUG. Starting from myself as new Chair of the Communications and Marketing Committee (CMC). For that, my first words are to appreciate the labor of Antonio Ferre in those years as Chair of this committee. Now it's my turn and I´d like to share with you an old Spanish adagio that says: *los cambios nunca vienen solos* (changes never come alone).

This moment is one of the most challenging moments in this committee because we are working on several interesting and important tasks. We are redesigning our website, our home on the internet, the home for all of the people that love our methodology and want to exchange knowledge with our partners. We need to rebuild our home to convert it into the best place to meet, to collaborate and to spread the word of our measurement methods all over the world.

At the same time, we are helping to create the new branding of IFPUG and to extend its use. Believe me when I say to you that great moments come to IFPUG even during challenging moments within our committee.

A lot of changes come to IFPUG; a lot of opportunities to grow.

INDUSTRY STANDARDS COMMITTEE

By **Carol Dekkers**

The Industry Standards Committee continues to focus our work on the International Organization for Standardization (ISO) standardization of IFPUG and software measurement related to international standards in the following areas:

- Cloud standards and Quality (ISO/IEC 25000 series of standards) standards in development
- IEEE 2430 SNAP (Software Non-functional Assessment Process), which is now advancing as ISO/IEC/IEEE 32430 in ISO/IEC JTC1 SC7 WG6

As standards work continues to expand and emerge globally with more countries joining the international work, we, your Industry Standards Committee of Steven Woodward, Talmon Ben-Cnaan, and myself, continue to ensure that IFPUG standards (functional and non-functional) are aptly represented and remain relevant and present in the international standards arena. IFPUG continues to have presence both as a formal member of the USA Technical Advisory Group (TAG) through our participation and membership through INCITS, and also as an influential industry group internationally.

Contact me or your local country standards association involved in ISO/IEC JTC1 SC7 work if you'd like to know more or to represent your country in the development of systems and software engineering standards development.

NON-FUNCTIONAL SIZING STANDARDS COMMITTEE

By **Talmon Ben-Cnaan**

The following white papers may help users in their work:

SNAP and the General System Characteristics (already in IFPUG store)

- A guideline on how to use the three sizing methods (unadjusted function points, SNAP and modified GSCs). This paper will guide users on how to separate the non-functional part so that FPA, SNAP and revised GSCs will provide better information needed for estimation.

Boundaries and Partitions (will soon be published)

- This paper is intended both for function point and SNAP experts and is a detailed guideline on how to interpret the CPM statement that the purpose of the count influences the boundary (CPM 4.3.1 Part 2, page 5-2).

Sizing Security (will soon be published)

- A white paper that describes security requirements and how to size them using IFPUG methodologies. The document covers password strength, protecting URL injection, disclosing information from unauthorized users, handling concurrent users, insecure cookies, data leakage prevention and more.

SNAP as an International Standard

- IEEE 2430, which is the international standard for non-functional sizing, is in a process of being an ISO standard, and will become ISO/IEC/ 32430. Expected publication date is around December 2021.

INTERNATIONAL MEMBERSHIP COMMITTEE

By **Saurabh Saxena**

The mission of the IFPUG International Membership Committee (IMC) is to provide help to the IFPUG membership and resolve any kind of requests, doubts or queries related to anything and everything with IFPUG.

Apart from existing country representatives: Lionel Perrot (France), Giovanni D'Alessandro (Italy), Cao Ji (China), Loami Barros (Brazil) and Iván Pinedo (Spain), we welcome Mr. Amir Sidek (Malaysia) as a volunteer member in the IMC. Malaysia is a growing region with huge potential for IFPUG members and we are pleased to have our presence here as well.

Based on lots of feedback, we are improving IFPUG's volunteer process. The new volunteer form is on the way and with IMC's presence, we hope to improve communications and engagements between IFPUG volunteers and IFPUG.

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

- ISBSG/IFPUG reporting task force
- IFPUG website redesign task force
- The approvals and verification for the "CPM 4.3.1 IFPUG manual translation in French language."

IMC is working with other committees to ensure that IFPUG not only grows in new geographical regions but also retains its existing members.

NOMINATING COMMITTEE

By **Mauricio Aguiar**

First, I want to announce our Annual Meeting will occur on October 4, 2021. You will be provided with information on how to attend the virtual meeting soon.

As the Chair of the 2021 IFPUG Nominating Committee, I ask each of you to consider nominating a qualified member to serve on the IFPUG Board of Directors. We are seeking four (4) exceptional candidates for two (2) Board of Directors openings. Terms will commence November 1, 2021.

Submit your nomination of candidates for the IFPUG Board of Directors to the IFPUG email address ifpug@ifpug.org or by mail to be received by the IFPUG Office no later than August 5th, 2021. In order to submit a nomination, you and the candidate must be current IFPUG members according to Article XVI – Miscellaneous – of the Bylaws, so please clearly indicate your name and contact information together with the nominee's name and contact information.

Download the Nomination Form

When providing a nomination, please first ensure that you have received the permission of the candidate. All nominees must meet the eligibility requirements specified in the IFPUG Bylaws (see below). The Nominating Committee will review all nominations received and will formulate an election slate no later than August 26th, 2021. A summary of the election key dates follows:

- July 6 – Call for nominations mailed by IFPUG office
- August 5 – Nominations due to IFPUG office
- August 26 – Ballots emailed to membership (paper ballot available upon request)
- September 27 – Ballots and selections due to IFPUG office by close of business
- October 4 – Election results will be presented at the 2021 Annual Meeting

For further details about the nominations process, requirements and campaigning options, please visit https://www.ifpug.org/annual-meeting-notice-and-call-for-board-of-directors-nominations/.

191 Clarksville Road
Princeton Junction, NJ 08550 USA

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