MetricViews | Issue 2 | November 2023

MetricVIEWS

IFPUG.org

A PUBLICATION OF THE INTERNATIONAL FUNCTION POINT USERS GROUP

RETHINKING SOFTWARE DEVELOPMENT ESTIMATES

ISMA21: A SNAP-CENTRIC VIRTUAL CONFERENCE

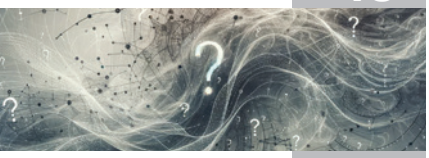
THE COMPLEXITY IS UNKNOWN


IFPUG
International Function Point
USERS GROUP

6

12

18



21

COMMITTEE REPORTS

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

MESSAGE FROM THE PRESIDENT

This edition of *MetricViews* focuses on the broader picture of the Metric Usage. Metrics cannot be used in isolation without having a deeper understanding of the architecture, its complexity, or the processes to which it is being applied. The three articles in this *MetricViews* focus on highlighting the need to understand the context of measurement before applying it. One of the articles also calls out the fact that metrics may need adjustment if context has evolved over time.

I think these articles are so timely as I take up my new role at IFPUG as the president. Not only the metric needs to evolve with time if the context changes, but the organization and the processes need to have a point-in-time relevance sanity check. In October 2023, IFPUG board members met in Dublin for the same purpose and to understand how and where IFPUG, as an organization, needs to evolve. Our fundamental question is how we can add more value to our membership. We understand the membership is looking for a holistic and simplified solution with data backing up the guidelines. With this purpose in mind, we have agreed on restructuring the IFPUG committees to align with various strategic areas of work like Forecasting and Estimation, Training, Data and Business Application beyond the certification and membership work.

Accurately estimating the costs, resources, time, and price of software projects is crucial in the ever-changing software industry. In Dublin, the IFPUG board agreed to add a new Software Forecasting and Estimation Committee. The goal of this committee would be to research the newest methods, tools, and approaches for forecasting, end-to-end estimating, and software estimation and create a Point of View for the industry. It will also encourage open communication with IFPUG estimation partners and publish content about measurement benchmarks, software productivity, and forecasting tools. Your experience in the industry can really help to provide the best-in-class guidance and insights from the ground level. So please feel free to join this committee as a volunteer. Nominate now by writing to ifpug@ifpug.org.

Training has been another area where a challenge has been reported by the IFPUG membership. Soon you will see a trainings-related taskforce to bring in the required support for training for our members in different parts of the world.

To maintain the growth of the organization and raise the value of IFPUG around the world, we depend on the assistance of volunteers from the Board of Directors, committees, taskforces, and other contributors. I want to take this moment to thank Charles Wesolowski for his leadership of IFPUG as the president and all the work he has done for years on the board.

I would like to congratulate Loami Xavier de Barros and Julián Gómez for being newly elected to the Board of Directors. They will succeed Christine Green and Sergio Brigido, who have completed their terms. On behalf of IFPUG, I would like to thank Christine Green and Sergio Brigido for their amazing efforts and time servicing IFPUG on the board. Christine will continue to lead some other initiatives for IFPUG. I would also like to congratulate Joe Schofield for being nominated as the Honorary Fellow and Fabrizio di Cola and Kiran Yeole as the Volunteers of the Year.

MetricViews has always been one place where members can find or share their experiences in estimation. Looking forward to seeing more great articles from you as we step into the new year.

Sincerely,

Roopali Anand Thapar
IFPUG President

BOARD OF DIRECTORS:

Roopali Thapar
President

Saurabh Saxena
Vice President and Director of Business View

Charles Wesolowski
Immediate Past President

Luigi Buglione
Secretary and Director of Partnerships & Universities

Cinzia Ferrero
Treasurer and Director of Certification

Sushmitha Anantha
Director of Non-Functional Sizing Standards

Loami Barros
Director of Membership, Research & Education

Julián Gómez
Director of Communications & Marketing

Roberto Meli
Director of Functional Sizing Standards

My eldest granddaughter started junior high this year. Here in Arizona that school district started in July, in the middle of the record-setting 55 days of 110 or more-degree highs. Hold on. This is not a story about global warming. She was told she didn't live far enough away to utilize the bus that goes through her neighborhood. Hold on. This is not a story about privilege or poverty. Finding alternative transportation required her mom to clear her schedule of patients twice a day. All of the other neighborhood kids could ride the bus. Hold on. This is not a story about being a "victim." When queried, the school district quipped that "as the crow flies" my granddaughter resided within a mile of the school. Her mother responded that her daughter was not a crow, and that readily available apps confirmed that she lived over a mile from the school.

This issue of *MetricViews* has three articles about measurement. One suggests increasing the number of Unadjusted Function Points based on the complexity of modern platforms and architectures. Carol Dekkers' article taps into her years of industry insight to strengthen the practices for software cost estimation. My article focuses on the unknown capabilities of team members in relation to future development tasks calling into question the credibility of estimation, especially in a truly Agile development environment. Published research is cited to support this potentially startling claim. Chances are that this issue will provide you with information that you will like, or, disagree with entirely. Regardless, I hope you enjoy the discourse.

Using crows to determine student eligibility for transportation to schools isn't much different than using another team's (or organization's) data to predict your outcomes. Replacing paper maps that include scale information slipped into a corner, with apps accessing current and precise location coordinates is an obvious improvement. Similarly, using products developed with outdated technology as a reference for your future performance doesn't make sense either.

My granddaughter rides the bus today. Someone questioned the antiquated approach for determining her eligibility, and someone else examined and replaced what they thought to be true with truth. Hold on. This is not a story about exploring new ideas and improving what we do; or is it?

Be well, stay well.

Joe Schofield
MetricViews Editor

SOFTWARE COST ESTIMATING:

WHERE'S THE FIT BETWEEN FUNCTIONAL SIZE, AGILE DEVELOPERS, AND THE NEW BODY OF KNOWLEDGE CEBOK-S?

By: Carol Dekkers

Abstract

Cost estimates are fundamental to large-scale software development for both government agencies and public corporations. While the shift to agile methods has changed *how* software is developed, it has not changed the needs of software professionals for early cost and schedule estimates. Those on the product development side argue that the inherent lack of specificity of software requirements on projects negates the value that early estimates provided on to traditional, requirements-first development. Even worse, state the agilists, estimates become unattainable targets during software development. The friction between cost estimators and developers reached a crescendo several years ago with the rise of the "no-estimates" movement in the agile community. Today, software cost estimating remains a critical prerequisite for software development, and advancements in the profession present opportunities for quality and development professionals alike.

This paper presents insights into the role that software cost estimation plays in the delivery of large-scale, high-quality software, regardless of development methodology, and presents a collaborative approach beneficial to both software cost estimators and the ensuing development teams.

The following questions are addressed:

- Why projects never seem to finish on time or on budget (and it's not YOUR fault!)
- Why befriending the cost estimating team can be good for your project
- Why do cost and schedule estimates take so long to prepare when all the data are right there in their tools? (And why do they ask such elusive and invasive questions about our finished projects?)
- How can communication between the two disciplines (cost estimating and development) lead to better outcomes?
- Why is a professional and collaborative approach to software cost estimating necessary in today's changing software development environment?

Introduction

Software has become pervasive in every aspect of our lives from smart homes with remote controlled appliances to self-driving cars to smart highways; software is everywhere. Even projects where software is not dominant (such as a toll-highway construction project) can be impacted by delays in software development. In addition, the volume of software "code" now

involved in major programs such as the F-15 military aircraft program has increased so exponentially that software (once a minor component of the product) is now as costly and dominant as hardware and other product components.

In 2001, *The Agile Manifesto,* written by 17 prominent software developers, revolutionized the approach to developing software and presented guiding principles to streamline software development and increase the value of delivered software. This changed how software is developed by minimizing the upfront, protracted requirements definition of the earlier waterfall approach, and allowed the detailed scope definition to evolve iteratively instead of monolithically. This means that software product development today does not come about using a fixed scope of software requirements, but rather through incremental "deep dives" into the software requirements as prioritized by the product owners (the business.) What has not changed, however, is the need for advanced, high-level cost and schedule estimates in order to initiate, plan and begin software projects—often 2-3 years in advance. As such, software estimates remain a non-negotiable requirement for agile and non-agile projects alike. This creates friction within the agile development community and resistance to the entire concept of estimates by agile software developers who argue that product development cannot be properly estimated because of its creative nature and lacking a predefined scope or set of features to be delivered.

At the same time, the ongoing definition of project success according to the annual Standish Group CHAOS report continues to be those projects that are delivered "on-time," "on-budget," and meeting the user needs (based on estimates.) The degree of success for agile software development projects has improved over the traditional waterfall (linear, fixed) approach to building software, however, despite the approach, still less than 50% of projects are deemed successful.

Since the original CHAOS report in 1994, researchers postulated that challenged and failed projects lacked customer-centric processes and tools, management support, and user engagement. While the reports through the year did mention that poor estimates and overly ambitious goals were in the top 10 causes of project challenges, researchers did not focus on them as main areas for process improvement. In this author's ICEAA CEBoK-S research, three factors of software cost estimating—over-optimism, lack of good historical data, and immature estimating maturity—can doom even the most advanced and well-trained project teams to failure. Perhaps the reasons that so many of the CHAOS report projects were over budget and/or behind schedule was not so much that the project failed to stay within the contracted budgets and schedules, but that they were not based on data-founded, realistic, or defensible estimates for delivering a project of that size in the first place.

For software development to succeed in meeting project goals of on-time and on-budget delivery, the software cost estimates must be reliable, realistic, and data-founded. An exploration of how to professionalize and formalize software cost estimating follows, with easy steps for agile software development teams to productively engage with cost estimating teams. The time is now for software developers and customers to embrace software cost estimating as a critically important profession and engage to guarantee project success for the future.

Background

Software development is relatively young, compared to other product development industries such as construction. In the 1950s software was designed to automate manual business practices and to automate the processing of large volumes of data using written computer "code" or programs, but without standardized software development life cycle processes. Over the decades, technological advancements and the application of automation to more industries added complexity to software development and a variety of sequential, formal approaches were used. In 1970, Winston Royce published his original paper introducing the "waterfall" model in *Managing the Development of Large Software Systems,* which became the software development norm, and is still in use today. (It is worth noting that in his original definition of the waterfall model, Winston Royce did say that if software was built this way, it would fail, and instead he recommended iterative development, iterating between phases and across the whole software development lifecycle.)

In 2001, a group of developers who realized that software requirements often changed after the design state, wrote the Agile Manifesto which espoused a dozen customer-centric concepts such as valuing working software over documentation, and people over processes, and the Agile Software Development movement was born. Today, software development is a multi-billion-dollar industry touching almost every aspect of human life from deep sea exploration and space travel to self-driving cars and smart highways. The software development landscape today includes both waterfall (sequential, adverse to change) and agile (evolving scope) approaches, with the majority of large scale government agency software development embracing a hybrid approach to software product development that can include Commercial Off The Shelf (COTS) software packages, custom software development, glue code (to put the pieces together), software as a service (think rented software) and configured enterprise resource planning (ERP) implementations. Regardless of the specific project, software development costs are often dominated by two major cost drivers: the size of the software and the productivity (complexity of the software, skills of the team, and tools) to complete the project. Software and its sustainment are now a major cost center for businesses worldwide.

As a case in point, during the winter of 2022, the U.S.-based carrier Southwest Airlines (SWA) suffered major business loss due to flight disruptions caused by out-of-date software that had not been updated (a business decision). The software itself was not SWA's major business, however, it supported (and then didn't) their business of getting people from point a to point b during a busy holiday season. Countless other businesses are disrupted or seriously challenged by software issues. This is peripheral

to the point of software cost estimates; however, it serves to demonstrate how integral software is to our daily life.

The current state of software development

The annual Standish Group CHAOS research report tracks software project success as on-time and on-budget delivery and since 1994 (the first year of the report) the rate has never exceeded 45%. Challenged projects are deemed as those that are either over-budget or behind schedule, while downright failures are both. Additionally, the average cost growth hovers around 40-50% and occurs on 80% of development projects, while the average schedule delay for the same projects is between 60-80% and occurs 90% of the time, according to Dr. Christian Smart in his 2021 book *Solving for Risk Management: Understanding the Critical Role of Uncertainty in Project Management.* It is plain to see how overly optimistic and unrealistic (wishful thinking) software cost and schedule estimates—a major root cause of software development "failure"—can be regardless of the development approach.

Why projects never seem to finish on time or on budget (and why it's not YOUR fault)

As outlined in the abstract, much attention and process improvement effort is spent on addressing the process and management issues related to failed and challenged projects. Software process maturity models, development tools and methods, and management practices certainly improve the odds for project success, however, much of the fault lies directly with immature estimating practices and over-optimism, coupled with poor historical cost and schedule data. Added to the known "Cone of Uncertainty" where preliminary software development estimates can vary by as much as +/- 400% with actuals, it is a wonder that even 40% of software projects succeed.

According to the Standish Group, unrealistic estimates resulted in $81B USD in cancelled software projects, and $59B USD in budget overruns (2015 Standish Group CHAOS report, one of the last publicly available reports).

The good and bad news about status quo cost estimating practices

The good news is that if you are a software developer or agile tester or even a product owner on a software development project that is over-budget and/or behind schedule, it is likely not your fault.

The bad news is that until software cost estimating is embraced as a bona-fide, structured professional endeavor, and the process around collecting historical data (actuals) is improved, software development is trapped in a cycle similar to a dog chasing its tail:

1. **Few strong inputs & low estimating maturity.** Poor requirements documents and apathy (or downright disdain for estimating) on the part of the software development team are poor inputs to the estimating process. This leads to
2. **Weak, and unreliable estimates.** Immature software cost estimating practices (ad hoc, non-data-founded, non-standard processes) together with lack of good data lead to overly optimistic and unrealistic software size and effort estimates. This leads to
3. **Unrealistic plans (cost, effort, duration).** Lack of standardized estimating process (ground rules & assumptions, cross-checking, verification by the development team) leads to
4. **Impossible contracts/internal projects.** Development begins based on flawed contract values (and undocumented) assumptions and both the customer and supplier seek to make changes/clarifications to lower their risk, which leads to
5. **Out of control projects.** Both sides end up at odds (there is no win-lose in software development—only win-win or lose-lose) and
6. **Adverse project outcomes:** Uncontrolled project "growth," failed & cancelled projects, litigation, wasted investment. Project is cancelled, restructured, descoped, or failed. Then the actuals data is not collected consistently. Few lessons learned, until…
7. **A new project emerges/or a cancelled project restarts at step 1,** with little change except for the promises/optimism to do better. The cycle continues with the same well-intentioned, but unrealistic estimates based on flawed assumptions and poorly documented software requirements.

Lessons Learned About Software Cost Estimating

The following subsections comprise the author's top 10 lessons learned based on her years of leadership in the software estimation field, including research and development work with the International Cost Estimating and Analysis Association (ICEAA), the U.S. Department of Defense's Defense Acquisition University (DAU), the U.S. Government Accountability Office (GAO), international software cost estimating subject matter experts, cost estimating tool vendors, the University of Southern California COCOMO II and III teams, and others.

- Lesson 1: Historical data-founded estimates are more reliable than those based on theoretical or expert opinion. Of course, this implies that the data are of high-quality and can be normalized. This predicates a planned process whereby the data are properly reported, captured and stored. In lieu of good historical data, one can use a publicly available and high-quality historical data repository such as those contained within commercial estimating tools (such as SEER-SEM®, SLIM®, or TruePlanning®) or maintained by the International Software Benchmarking Standards Group (ISBSG.)

• Lesson 2: The Estimating Maturity Model co-authored by Dan Galorath and Esteban Sanchez outlines how increased formality and professional practices fall into the model. The lesson learned is that organizations should be aware that they and the majority of software development organizations (worldwide) are typically at Level 1 of the model. Higher levels of estimating maturity result in more reliable and realistic estimates as depicted in Figure 1 below.

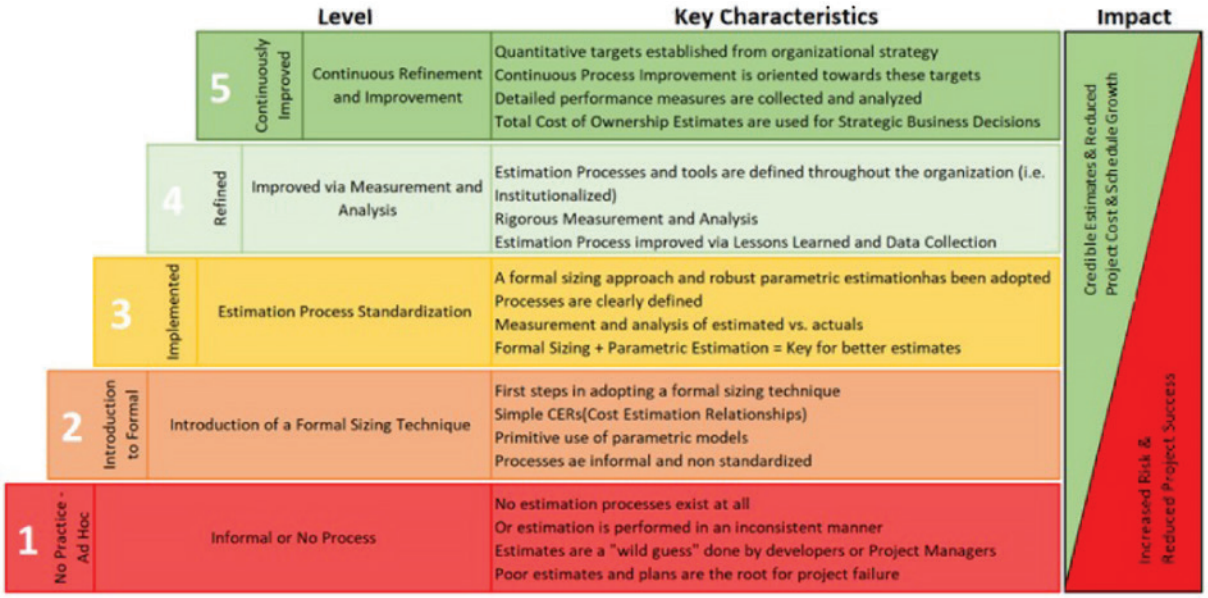


Figure 1: Software cost estimating maturity model (Galorath, Sanchez)

• Lesson 3: The estimation scope is critical to the context, understanding and validity of the estimate especially for analogous and parametric cost estimates.

• Lesson 4: There is not a one-size-fits-all estimating approach for all software projects, but rather approaches that are appropriate for the development phase when the estimate is being done. For example, at the earliest stages of development where few details about the software to be developed are known, a Rough Order of Magnitude (ROM) estimating approach may be the only possibility.

• Lesson 5: Software size is a major driver of software development cost and schedule, as is productivity. As such, a formal, standardized approach to estimating software size is fundamental to preparing a defensible software estimate.

• Lesson 6: Agile project size estimates can be developed based on the product backlog and using standard functional size measurement approaches such as IFPUG Simple Function Points (SFP).

• Lesson 7: Software development effort estimates are subject to Diseconomies of Scale. (Note: This is different from hardware estimating and other product development where volume of widgets comes into play. Analogous (linear) estimates are possible only within a specified (narrow) range of application size.

• Lesson 8: The software development approach (waterfall or agile) brings in different cost considerations, as well as whether the development is done in-house or by third-party vendor(s).

• Lesson 9: Most software development is a mix of hybrid solutions, and multiple estimates are generally needed. The type of development, availability of technical baseline data, historical data, level of quality required in delivered software, etc. should all be considered when preparing and presenting the estimate. Beyond software development, sustainment, integration, deployment, and other factors must be considered to create and provide the context for a realistic estimate.

• Lesson 10: The software development versus procurement continuum, (and what is included) are oft overlooked and can impact the development of a realistic estimate. Additionally, the level of leadership required can impact costs (bigger programs need more leadership—e.g., running a large software development program that involves systems of systems is impossible without a strong leadership team to guarantee that quality software will be delivered).

Where Does This Leave Agile Testers and Product Developers?

As an agile tester or agile developer, the software cost estimating team needs to collaborate with you/your team to develop software estimates (4 parts) that are the most reasonable, realistic, and inclusive within the context for which they are intended. This requires a joint effort to ascertain what will be the resulting software product delivery and the process starts

with project initiation documents (often a Concept of Operations: ConOps or project charter) to create the original estimate. Estimates are then refined as additional development artifacts such as development release strategies, software requirements specifications, contract documents, change and configuration management documents. Finally at the delivery/release stage, the project actuals should be collected and stored. Data collection and maintenance of a repository are also collaborative efforts where the quality team and project management organization (PMO), in conjunction with the cost estimators, should play a lead role.

4.1 Developing professional software cost estimating competency within your company

There is a certification: Software Cost Estimating Certification (SCEC), and an entire set of formal documentation for the ICEAA Cost Estimating Body of Knowledge for Software (CEBoK-S). The documentation covers the aforementioned lessons learned and also the following major concepts:

- Types of software cost estimates and considerations (lifecycle cost estimate, ROM, software development estimate, software sustainment, risk factors, etc.);
- How to select the best approach for doing a software cost estimate;
- Important concepts: estimating maturity model, size, productivity, ground rules & assumptions. Data normalization and analysis;
- Steps to preparing a good estimate;
- Cross-checking, context, presentation to management;
- Ten modules of materials plus backup slides (over 800 slides).

How Can Cost Estimators and Agile Software Teams Work Together to Achieve Success

Given that software cost estimates remain a requirement for acquisition and software development funding, it is critical that the best possible, data-founded, and realistic software estimates are developed.

While at first glance it might seem that a software cost estimate would be a range of numerical values representing the estimated software development size, effort, cost, and schedule (duration), a good software cost estimate should consist of much more.

At a minimum, the software cost estimate should contain the following information in sections:

1. Contextual information about the software project/release/phases included and:
    - o Program/project/release identification (name);
    - o Description of same;
    - o Type(s) of cost estimate(s) and scope of each;

- o Source documents and subject matter experts included in the formulation of the estimate;
- o Basis of estimate;
- o Ground rules & assumptions;
- o Technical baseline;
- o Scope of the estimate (e.g., software development, procurement, sustainment, maintenance, and any other inclusions. Exclusions from the estimate should also be stated explicitly).

2. Software Size estimate:

- o Size(s), unit(s) of measure and sizing method used (e.g., International Function Point Users Group Simple Function Points or other);
- o Growth factor(s) applied;
- o Source document(s);
- o List of included software requirements;
- o Templates or shortcuts.

3. Software Effort, Cost and Schedule estimates:

- o Estimating approach(es) (e.g., analogy, published parametric equation, derived Cost Estimating Relationships (CER);
- o Assumptions not covered in 1. above;
- o Effort estimate and unit(s) of measure (person hours);
- o CERs /Schedule Estimating Relationships (SERs) used and range of applicability;
- o Productivity assumptions;
- o Historical data used (as applicable);
- o Cross-checks and assumptions;
- o Growth factor(s), risk, and uncertainty considerations (and confidence levels).

Conclusions

As outlined in this article, poor cost and schedule estimates can doom a project or initiative from the beginning. While software cost and schedule estimates are especially challenging to cost estimators and the team due to the unique considerations of developing software as compared to hardware or other program aspects, communication between the two disciplines of cost estimating and software development can lead to far better and successful software products. As such, the recognition that software cost estimation is a professional endeavor that can be practiced, formalized, and standardized can provide projects with contracting environments and funding to increase successful on-time and on-budget software deliveries. While estimates are the best guess of what the project should cost and how much effort it should take, under standard, historically proven conditions, they are still estimates rather than targets or guarantees of actual project performance. Estimates based on realistic historical data and data-based CERs give projects a baseline on which to base changes, and a fighting chance for success.

ABOUT THE AUTHOR

Ms. Carol Dekkers is a Certified Scrum Master (CSM), a Certified Function Point Specialist (CFPS-Fellow), a Professional Engineer (P.Eng-Canada) and a Certified Software Cost Estimator (SCEC). She is also the lead author of the International Cost Estimating and Analysis Association (ICEAA)'s new Cost Estimating Body of Knowledge for Software (CEBoK-S), and a long-standing member of the U.S. delegation to the International Organization for Standardization (ISO) subcommittee for writing Software and Systems Engineering standards. Ms. Dekkers' expertise spans software development, software measurement, quality engineering, project management, and software cost estimating and she has shared her insights with technical and non-technical professionals worldwide through keynotes/ presentations, textbooks, and articles published in industry journals. Ms. Dekkers has received numerous awards for her industry contributions and thought leadership including Computing Magazine Global Leader in Consulting – Pro Bono (2023), ICEAA Educator of the Year (2022), IFPUG Honorary Fellow (2022), Brazil and Korean metrics associations special awards (circa 2010) and was named one of the 21 New Faces of Quality for the 21st Century by the American Society for Quality, ASQ (2000.)

Rethinking Software Development Estimates:

FOCUS ON TEAM MEMBERS

By: Joe Schofield

Abstract

What this article is not. It is not a survey, not an evaluation, not a recommendation, not a comparison of estimation and forecasting tools, platforms, methods. Rather, this article focuses on the need for deep understanding of contributing team members, their individual competencies as they relate to the work they perform, their physical proximity and access to fellow team members, their willingness to collaborate amongst themselves, and their limited distraction to work commitments outside the project being estimated. And finally, the still unrelenting conflict between traditionally-based culture and project management with Agile projects that are discovery-oriented and truly practicing "changing requirements even late in development".

BACKGROUND

"Estimates are not intended to be accurate" is a phrase oft used to explain variances for cost and schedule, for work in general and software specifically. How we establish initial estimates, who develops the estimates, the purpose of the estimate (to make a proposal seem attractive to a potential client, or to predict a most realistic timeline for fulfilling that proposal), the potential risks associated with the work itself, and the use of internal historic data or benchmark data, all influence the integrity of the estimate. Sadly, but not atypically, once rendered, the estimate itself is frequently interpreted as a delivery promise. Actual performance data for project management constraints like cost and schedule—while often reported as dismal[1]—is often accepted as reality. But is our desired outcome to excel at

delivering value, or to create a reasonable bid for a fixed price contract, or to appear to have mastered inflating time and cost estimates?

Software estimation is not for novices. Software development companies can swell profit margins or damage their brands and existence as a result of their estimation processes. For internally developed software, reputations, careers, bonuses are more likely at stake. Models [COCOMO, COCOMO II], benchmarking [ISBSG], internal historical data, and tools are intended to help improve our accuracy. Organizations [ICEAA, IFPUG, BFPUG, GUFPI], functional measurement standards [ISO], and conferences [ISMA, PMI Global Summit, et al], further hone estimation processes, practices, and knowledge sharing respectively. Books, articles (still another), training, podcasts[2], and certifications attempt to plan, monitor, and control the management of such undertakings. Platforms, development languages, risks, tools, environments, margins of errors, past performance, and requirements are often included in those estimates. Team capability too is often incorporated; sometimes portrayed as an anticipated team productivity. A reflection on five decades of software development practices provokes the question "how well do we understand components of team productivity variation"?

When we don't know what we should know. Before delving into opportunities for enhancing the integrity of estimates, it is reasonable to dispel some of the beliefs that may prevail. Here are two examples of *when bad estimates appear to be good,* and their uniqueness remains doubtful.

1. Inflating estimates to provide software development teams with a larger margin of error is a decades-old ploy. However, even nascent Agile teams are already applying "estimation inflation" masterfully, and it's unrelated to economics. Refactoring, unnecessary tasks, inflated task hours, and deflated productivity rates, provide the team with an excessive of hours to complete their work. Less obvious however, the business is deprived of potential value for which they were paying. When applying these same techniques to release and project planning, such antics only increase the potential loss of value delivery.[3]

   A somewhat different spin on *estimation inflation* ensures those estimates are accurate by slowing progress on tasks to closely align with actuals. This is a "win, win, win." Myopically, the business wins because they are getting work completed "on time." The team wins because it appears that they are performing according to the plan. The project manager wins because he/she is recognized as a competent estimator. In the long term, the organization loses: value is under-delivered, actual data is specious, future estimates are based on inaccurate data, and contributors gain a false sense of self-worth.

2. How often do we hear of projects being completed on time and within cost? Extreme instances of variances between estimated and actual costs and schedules make for eye-catching headlines. They also propagate the need for project management tools and governance. Granted, some percentage of work is actually delivered on time and on budget, but this too can be deceptive. As an example, fixed-price contracts may overrun their budgets--in some cases, by substantial amounts. However, to meet the expectations of their funding agencies as well as their leadership, they discontinue charging for the actual time required to deliver a release. Often times, overtime is accrued but not reported. The project lead or the team as a whole receives accolades for a job well done despite their failing to produce in conformance with their own estimates and labor plans. Once again, the impact of intentionally hidden costs of overruns is the unwarranted optimistic cost projections for future deliveries. The unreported cost overruns of the past become "tribal knowledge" passed along to selected future team members or is lost.

This third example is also unique. In this instance we can't be sure that the estimate is good or bad because team performance is clearly lagging. At least in this example, estimation is not to blame. Early in my career I recall attending project-status meetings with my peers and our second-level management. One of my more senior peers reported that their schedule slippage would not reoccur because they were "smarter now." After a couple of meetings with the same "smarter now" antidote, I chimed in that if they got any smarter, they would never deliver. Sneers and frowns were often my reward for stating the apparent. Defects and delays were also likely the reward for those who ignored the signals emitted by their slumping performance. Poor estimating wasn't their Achilles heel; incompetence and lacking accountability were more likely suspects.

The three examples above, may not seem to evidence a lack of understanding the team's role as a source of variation; that is, not without further explanation.

EXAMPLE ABOVE	BUT A CLOSER VIEW OF THE TEAM REVEALS . . .
A – the team was employing estimation inflation	Several new team members Naiveté of all team members on a scrum project Supportive but complacent novice Agile management
B – the team delivered on time and within budget	Large program with multiple teams, all with some new membership Burnout from uncompensated overtime triggered massive team turnover after every release Executive leadership either was misled about or ignored actual progress
C – the "we're smarter now" team	Some new team members Accommodating team leadership tolerated minimal expectations Key skills absent from the team

Granted, some team capability is included in estimation models [COCOMO II, SEER, SLIM as examples] and tools offering a range of values that attempt to reflect the ability of the team to perform the work. These are *team- and organization-based indicators.* They reflect a high-level of confluence of past performances by projecting similar performance onto future work based on degrees of similarity. This is precisely why understanding *individual team member capabilities* is necessary. Summary research evidence depicts why.

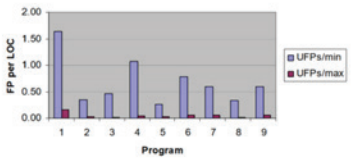
Data from software developers writing the same code, using their preferred language, each with relevant advanced degrees, in a classroom setting, subjected to the same validation tests, revealed that more than ½ of the participants wrote a program that had the fewest lines of code, and subsequently for a different program, also the most lines of code.[4] With the same control factors for each participant, a logical, perhaps the most logical, conclusion is that the specific problem was subject to the specific domain knowledge of the solution, on a developer-by- developer basis. Further, the research suggests that no "weak link" (underperformer) was evidenced in the study, and surprisingly, no "strong link" either. Outcomes were driven on a case-by-case basis specific to each participant. This "experiment" was repeated under the same control factors on three occasions. In one instance of 49 participants using "C" code, variation soared to as high as 2200%. The smallest variation was 960% and the average was 1500%. The nature of this research is still cited by the National Institute of Standards and Technology.[5] A key slide from that presentation is included below. Unlike many statistical samples, as the population grew, so did the range of variance.

Min and max values for "C" code
compared to Function Point size over 9 programs (n = 49)

	*P1	P2	P3	P4	P5	P6	P7	P8	P9
Min	22	20	15	13	27	23	25	21	25
Max	221	311	336	289	270	306	242	383	284
UFP$_S$	36	7	7	14	7	18	15	7	15
UFP$_S$/min	1.64	0.35	0.47	1.08	0.26	0.78	0.60	0.33	0.60
UFP$_S$/max	0.16	0.02	0.02	0.05	0.03	0.06	0.06	0.02	0.05
Variance Range	10.05	15.55	22.40	22.23	10.00	13.30	9.68	18.24	11.36	14.76

"C" Min/Max Function Point Range

Largest min to max variance is > 22, smallest is almost 10, average is almost 15.



Note that in these three examples, variance and averages increased as the population increased.

17

Individual team member competence associated with each program undertaken is an example of a *"micro" indicator.* I am unaware of any method, tool, standard, body of knowledge, or certification, that takes this possibility—individual team member expertise with the yet-unknown and unassigned task—into account. It may be impossible to know such, and thus is a cause of highly probable variation cloaked in other variables, masked by other more discernible "productivity" factors.

The Tuckman model,[6] best known for its forming, storming, norming, and performing stages of group development seems to reflect the impact of micro indicators as they pertain to team composition. Project Managers may relate to this as the Tuckman Ladder. The Examples A, B, and C above all "suffered" from the impact of frequent team turnover and therefore the consequences of groups reverting to the *forming* stage. Team advancement to the storming, norming, and performing stages is seldom linear. It is often interrupted with significant change to vision or approach, or more often, the loss or addition of team members. Since productivity is limited until the *norming* stage, and due to change in desired outcomes and the team itself, anticipating reversions to the *forming* stage is unreliable, typically unpreventable, and definitely unpredictable. Yet this highly impactful variable is seldom if ever taken into account when project plans are devised. Risk management and mitigations don't eliminate this major source of variation.

> "Your organization, your teams' recent, relevant, and valid actual data are the best predictors of future performance.

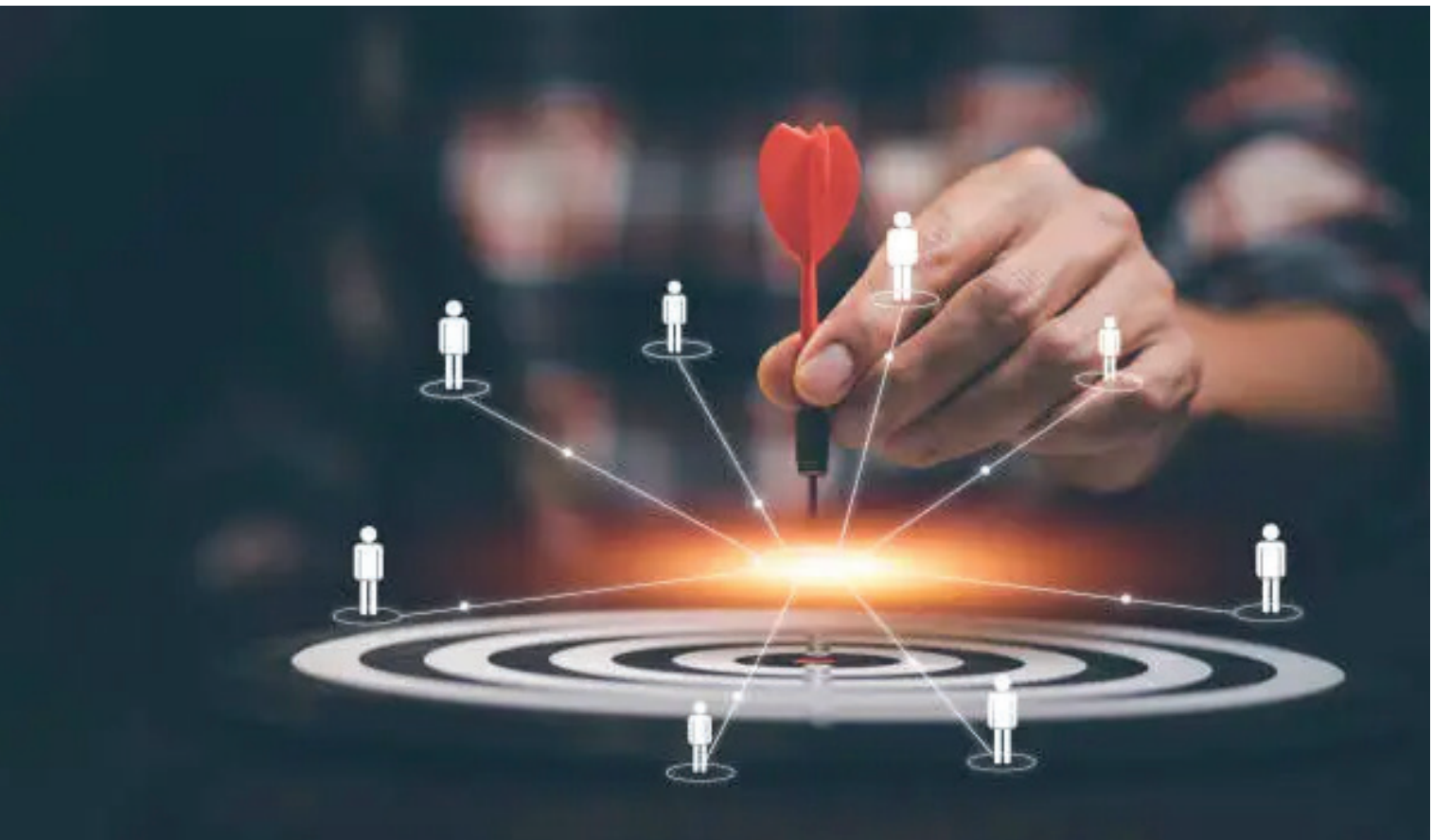
Behaviors in the *forming* stage-like variability itself doesn't cease with a frozen team. The strain on group maturity is exacerbated by the ability for individual team members to collaborate, grow cross-functionally, address their competency and skillset gaps with evolving project needs, and of course, collocate (enabling osmotic communication,[7] relationships, and mentoring) in today's upheaved work environment. More often an even larger source of variation is introduced when team members are shared across multiple projects. The phrase "I'm always in meetings" signals that team members are shared beyond their expected time to make meaningful and timely contributions. Scrum team members, for instance, lose about five hours a week of development time just to attend the minimal array of team ceremonies—per team on which they participate. For non-Agile teams, their time in meetings often grows even more quickly.

A final point on team dynamics. Large teams trigger more interactions and increased complexity. More interactions consume more time, which also negatively impacts the ability to deliver. One study concluded that 4.6 persons was the perfect team size.[8] And then there's the Ringelmann effect which demonstrated that the more folks were added to a team, the less effort each member exerted.[9] Quantifiable or not, there exists a substantial list of variables that drive productivity sky high, or into the ground, including:

- the size of teams,
- the velocity of team member turnover,
- the fractionalizing of team members across multiple teams,
- work from home vs. collocation (like osmotic communication [Cockburn]) and avoiding of isolation (it's hard to feel necessary when you're physically disconnected),[10]
- the speed at which teams can progress through Tuckman stages,
- the number of times a team reverts to Tuckman's forming stage,
- the familiarity of the team with the business, their own methods, and the target technical solution.

Recommendations towards improving current estimation practices

- Use recent data; that is, data that is three years or less to help account for constantly evolving platforms, technologies, and security needs.
- Use relevant comparisons. Traditional "predictive" examples may add data points for statistical confidence, and may actually have some relevant project management data, but
    - o if the project is developed with Agile frameworks, use Agile data,
    - o if the project is "firm-fixed-price" do not compare it to Agile projects that honor the 2nd Agile principle regarding the acceptance of "welcoming changing requirements even late in development"; this may also put the kibosh on estimation practices that are driven by a functional size value, because it only exists after each sprint planning session for discovery-driven and evolving Agile projects
    - o if the project plan calls for a single release, don't compare it to projects that "deliver working software frequently", all of these are very different in their development approaches.
- Sad to say, some projects don't record their actual data. That's not to say they don't record data. Team members may provide updates that favor what they are being evaluated on, or yesterday's marching orders. See Examples A and B above.
- Develop deep understanding of your organization's personal and team development capabilities. Build cross-functional skillsets to reduce the pain of team members that leave suddenly, are otherwise indisposed, or spikes in certain skillsets. How are hybrid and work-from-home options impacting product delivery? (Hint: the top technology companies are requiring their staff to return to the office.[11,12]) Is the spreading of team members over multiple projects hindering delivery commitment, integrity. What size teams work best for what types of work? How collaborative are teams? How quickly are new team members assimilated? What practices and activities speed cohesion?[13,14]
- Use your data. Your organization, your teams' recent, relevant, and valid actual data are the best predictors of future performance. Your data best reflects your cultural accelerators and nuances, the development environment, the capabilities and skillsets of your team. Unfortunately, few organizations do this. Those that do may have data



corrupted by inaccurate data, don't want to be compared to other teams, don't know how to use the data once its available. Most estimation-related consulting groups would argue that this lack of measurement focus isn't unfortunate at all; it has become their livelihood to the benefit of the software industry.

In closing

You can't estimate until you identify all work variables.

You can't estimate with integrity until you identify all of the sources of variation in those variables.

You can't estimate well until you associate performer capabilities with the task, assigned or selected.

Cost is a variable. Labor rates are a component within cost. Team member quality (competence) is a variable within labor rates that influence those rates. This variance is evident but not often explained when comparing and contrasting planned vs. actual values. Experience is also a contributor to competence, as is education. Focus, or the ability to focus, is a contributor to competence. Splitting people across multiple projects limits focus. Collocation, collaboration, cross-functionality are all also contributors to competence. Until all of the contributors of competence are understood and quantified in a meaningful and relatively accurate model or algorithm, labor rates will vary by task and by those completing the task. Competence will also impact quality that will impact defects, which will impact rework, cost and schedule as high-level variables. While competence is a variable that is frequently identified in labor-related estimates, others identified in this article may help to improve our understanding of the relationship between team member performance and successful delivery.

> The best estimates result from alignment of team member skillsets with each task undertaken by that team member.

The literature often highlights poorly performing projects that are staffed with teams of 10, 100, or 1000 developers, span multi-years, experience high turnover, are well over budget, and are led by a revolving door of management consultants and inexperienced developers.[15,16] Rendered solutions often point to better project management rather than more stable and capable teams. This is especially true and dangerous with Agile development teams when traditional project management metrics are forced upon non-traditional Agile teams. Senior leadership and culture are the most significant contributors to impeding Agile adoption which, includes Agile thinking. I have found that the best data for measuring teams' performances is their own data. Regrettably, few teams and organizations keep meaningful data. The best estimates result from alignment of team member skillsets with each task undertaken by that team member. The best execution of development practices come from collaboration among that team and the client, not historic benchmarks that rely on dated past performance and varying solution development approaches.

Yogi Berra was right, "prediction is hard, especially about the future."[17]

Further reading: https://en.wikipedia.org/wiki/Software_development_effort_estimation (great resource for history and survey of tools)

Special thanks to Karen McRitchie, Colin Hammond, and Larry Putnam who accepted my invitation to discuss this subject and whose comments enhanced the content of this article.[18]

REFERENCES

[1]https://management.simplicable.com/management/new/why-your-estimates-are-always-wrong; Why Your Estimates Are Always Wrong; Anna Mar, February 17, 2013

[2]70 – 80 percent of all SPaMCASTS touch on software estimation, project management, team development; Tom Cagley; 9/11/2023

[3]Inflate Gate: Mastering Overestimation for Agile Software Projects; Computer Aid's Accelerated IT Success; (Featured Article); IT Metrics & Productivity Institute; August, 2015

[4]The Statistical Case Against the Case for using Lines of Code in Software Estimation; 4th World Congress on Software Quality; Bethesda, MD.; September 17, 2008;

[5]https://www.nist.gov/itl/ssd/software-quality-group/metrics-and-measures; retrieved 9/11/2023

[6]https://en.wikipedia.org/wiki/Tuckman%27s_stages_of_group_development; retrieved 9/13/2023

[7]https://trustedinstitute.com/concept/agile-project-management/crystal-methods/osmotic-communication/ ; retrieved 9/26/2023

[8]https://knowledge.wharton.upenn.edu/podcast/knowledge-at-wharton-podcast/is-your-team-too-big-too-small-whats-the-right-number-2/ ; retrieved 9/26/2023

[9]Ringelmann effect — measured individuals and teams pulling a rope; as more folks were added to the "pull" less effort was exerted by each team member. In his day, this helped to explain "social loafing." https://knowledge.wharton.upenn.edu/podcast/knowledge-at-wharton-podcast/is-your-team-too-big-too-small-whats-the-right-number-2/; retrieved 9/14/2023

[10]"Malcolm Gladwell: https://www.foxbusiness.com/lifestyle/malcolm-gladwell-says-people-must-return-office-regain-sense-belonging; 8/8/2022

[11]https://www.cnn.com/2023/06/10/tech/silicon-valley-return-to-office-tensions/index.html; retrieved 9/29/2023

[12]https://www.cnn.com/2023/08/07/business/zoom-return-to-office/index.html; retrieved 9/29/2023

[13]https://www.foxbusiness.com/technology/amazon-ceo-staff-resisting-returning-office-probably-wont-work-out; retrieved 9/2/2023

[14]The move to bring workers back to the office for three in-person days came from a judgment call that Amazon executives made. Amazon said Jassy told employees at the meeting earlier this month. The company decided to go that route after looking at how teams were collaborating, the company's culture, performance of the business and other factors. https://www.foxbusiness.com/lifestyle; retrieved 9/2/2023

[15]https://blog.kintone.com/business-with-heart/blog/why-projects-fail-unreliable-estimates; Why Projects Fail: Unreliable Estimates; Euna Kim; Oct 28, 2020

[16]Why Big Software Projects Fail: The 12 Key Questions ; Watts S. Humphrey; The Software Engineering Institute; CrossTalk; March, 2005

[17]https://www.goodreads.com/quotes/261863-it-s-tough-to-make-predictions-especially-about-the-future; retrieved 9/12/2023

[18]I was Impressed with deep understanding of tools developers like SLIM and SEER, attempts at precision, ability to filter by various parameters, and ScopeMaster's emphasis on requirements as the basis for scope.

ABOUT THE AUTHOR

Joe Schofield SCT, SCAC, SSMC, SSPOC, SMC, SPOC, SDC, SAMC, CSQA, CSMS, SA

Independent Consultant – Enabling Organizational Capability

Scrum Certified Trainer | Certified Agile Coach | Certified SAFe® 5.0 Agilist

Past President, International Function Point Users Group

2023 National Champion; Powerlifting America; 74kg, Master IV

2022 National Champion; USA Powerlifting ; 75kg, Masters IV

Selected Key Roles: As a Distinguished Member of the Technical Staff at Sandia National Laboratories, Joe worked with the Departments of Defense, Energy, and Interior, and the NNSA. During his 31-year career he served for over a decade as the "Chief Process Officer" of an organization of 400 software engineers, which was awarded a SW-CMM® Level 3. He continued in that role to CMMI® Level 4 until his departure. Joe is a Past President of the International Function Point Users Group.

As an enabler and educator: . . . Joe is an Authorized Training Partner with VMEdu, a Scrum Certified Trainer and Agile Coach with SCRUMstudy with over 80 published books, papers, conference presentations and keynotes—including contributions to the books: The IFPUG Guide to IT and Software Measurement (2012), IT Measurement, Certified Function Point Specialist Exam Guide, The Economics of Software Quality, and his recently released Aligning People and Culture for Agile Transformation. He has taught over 100 college courses in software engineering and strategic decision making; 75 of those at the graduate level. Joe has facilitated ~200 teams in the areas of software specification, team building, organizational planning, and Agile transformation.

Lifelong learning: . . . Joe holds nine Agile-related certifications: SCT™, SCAC™, SSMC™, SSPOC™, SMC™, SDC™, SPOC™, SAMC™, and SAFe 5. Joe was a CMMI Institute certified Instructor for the Introduction to the CMMI®, a Certified Function Point Counting Specialist, a Certified Software Quality Analyst, and a Lockheed Martin certified Lean Six Sigma Black Belt. He completed his MS in MIS at the University of Arizona in 1980.

THE COMPLEXITY IS UNKNOWN

By: Ciro Coppola

Introduction

This "challenging" (maybe a little provocative) article tries to point out that *(and this is a common belief in many business contexts)* the functional complexity of the data and transactions is too small.

New technologies have subverted what Alan Albrecht and his team assumed in the early 70s; the methods of the time were aimed at processing methods, very complex for that period, but methods have changed.

Today we have the following complexity organization for ILF, EIF, EI, EO, and EQ:

	FUNCTION SIZE		
	LOW	MEDIUM	HIGH
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Table 1

Size of data functions

Data type features (ILF, EIF) contribute most of the Function Points (FPs) based on complexity. However, we often find it difficult to correctly identify the type of data:

- Business Data
- Reference Data
- Decoding Data

And the complexity (7 to 15 for ILF and 5 to 10 for EIF) does not describe the "effort" behind these numbers.

We also have big problems like evaluating "Decoding Data." This data is very useful, but is not evaluated in Functional Size Measurement (FSM).

Loading, updating or deleting (using scripts) Decoding Data is an effort that is not recognized using Function Point Analysis. It cannot always be traced back to effort and not evaluated in the Function Point count.

It is also necessary to consider how much life cycle (Collection of documentation, Analysis and Design, Test, Realization, Testing, Preparation of testing environments, etc.) is behind these complexities.

Size of transactions

Figure 1 captures the complexity of transactional functionality, a labyrinth of new technologies and methodologies:

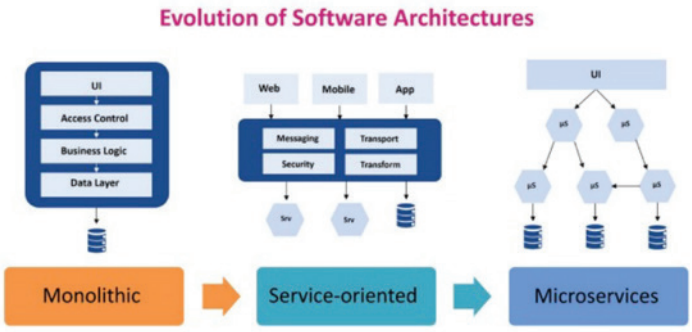


Figure 1

Boundary

Why doesn't functional size measurement (FSM) provide clarity when considering boundaries in the case of microservices?

Consider for example, having a single boundary for all microservices, which is useful when the purpose of counting is to determine all of the features provided to the user, ignoring the architecture (monolithic, SOA or microservices). By IFPUG's own admission ([IFPUG 2022] end of section 6), this way of proceeding is incorrect when the purpose of the count is to evaluate the value of the software in terms of complexity, production cost and maintenance cost savings. In this case, the boundaries shall be defined either at the level of:

1. Individual microservices, but this approach, in practice, can lead to overestimation of the measurement of FPs;
2. Homogeneous groups of services, i.e., grouping services according to the operations they perform, always considering the functional user requirements to identify homogeneous groups and neglecting the purely technical aspects (e.g., frontend-backend).

Scope of effort predicted by size

Requirements

In [Boehm 2000] [USC 2000] the estimate that is made starting from the size of the software (UFP) does not include the Plan and Requirements phase which is added as 10% to the total commitment of the Design, Programming and Integration and Test phases.

This article suggests increasing the complexity of each individual function by 10% as follows:

FUNCTION SIZE			
	LOW	MEDIUM	HIGH
ILF	8	11	17
EIF	6	8	11
EI	3	4	7
EO	4	6	8
EQ	3	4	7

Table 2

The IFPUG Counting Practices Manual reads as follows:

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

The objectives of function point analysis are to measure:

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

> One of the issues triggering difficulties is that the customer believes that the Testing phase should be included within the effort estimated from the Function Point size.

I want to focus attention on the second bullet above. Certified Function Point Specialists are very "devoted" to this statement, but in recent years it is not always applicable, because we cannot abstract from the technology used. While Microservices is mentioned above (see Boundaries), BlockChain and Artificial Intelligence are two additional examples with similar implications.

Testing

One of the issues triggering difficulties is that the customer believes that the Testing phase should be included within the effort estimated from the Function Point size; i.e., that the complexities indicated in Table 1 should represent the phases of:

- Collecting requirements
- Testing, including regression
- The creation of all documentation

Economic models for software, e.g., [Boehm 2000], attribute a statistical value of the percentage effort associated with each phase and here we consider the values given in [Consip2220 2022] (v. Table 3):

PHASE	COMMITMENT	CUMULATIVE PROGRESS
Definition	10%	10%
Analysis	25%	35%
Drawing	15%	50%
Realization	40%	90%
Test	10%	100%

Table 3 Phases of the SW life cycle and related percentage commitment

ISBSG periodically publishes reports containing statistical data on the average cost of function points and hourly costs, per country [ISBSG 2016], [ISBSG 2016a]. The reports address the possibility of increasing the commitment of a project which is not always possible to do using the Function Point measurement method.

Public administration in Italy has embraced this strategy in its contracts [G.Santucci 2023]. In the technical specifications, it is typical to indicate the size of a development or maintenance activity by establishing an overall ceiling, divided into UFP function points and person days where the flat rate cost of a function point is of the order of 200 euros (auction base).

We are talking about very small amounts that cannot cover what is requested by suppliers, but this is the situation of Italian public administrations.

An alternative could be to use SNAP metrics, but this is not yet sufficiently taken seriously; hence, there is no flat rate for a SNAP point.

Conclusions

The challenges associated with understanding boundaries are often complex and unlikely to sefl-resolve. I believe that IFPUG should continue to improve the Counting Practices Manual to incorporate new and constantly evolving technologies and platforms.

This article suggests that FPA may not represent the true Data/ Transactional functional complexity. This happens because the effort that concerns the "Definition" phase (see Table 3), does not reflect all of the complexity.

REFERENCES

[Boehm 2000] Barry Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, and Bert Steece. Software Cost Estimation with COCOMO II. Englewood Cliffs, NJ:Prentice-Hall, 2000. ISBN 0-13-026692-2

[USC 2000] COCOMO II Model Definition Manual – Version 2.1 http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

[Consip2220 2022] Open tender procedure pursuant to d. Lgs. 50/2016 and S.M.I., for the acquisition of development, management and maintenance services of the e-procurement system to support the PA purchasing rationalization program

[ISBSG 2016] International Software Benchmarking Standards Group. “The ISBSG Special Analysis Report:

Software Project Costs and Hourly Rates”, ISBSG, Hawthorn, VIC, June 2016

[ISBSG 2016a] International Software Benchmarking Standards Group. “The ISBSG Special Analysis Report:

Software Development Analysis by Country”, ISBSG, Hawthorn, VIC, June 2016

[G.Santucci] Functional and non-functional requirements for application software developed on behalf of public administrations; Department of Computer Engineering and Management “Antonio Ruberti” LA SAPIENZA University of Rome

ABOUT THE AUTHOR

Ciro Coppola has been a CFPS for more than 20 years (the first certification dates back to 1999), Almaviva employee (in Italy) and PM of various projects. He translated into Italian version 4.3.1 of the "Counting Practicies Manual" (2010).

His experience in the field of Function Points has been put at the service of numerous Italian state organizations that, on many occasions, have found benefits with the use of Function Points, often meeting the favor of the customer and finding a point of agreement from the user point of view.

ISMA21 – A SNAP-CENTRIC VIRTUAL CONFERENCE

By: **Kiran Yeole, Fabrizio Di Cola, and Charley Tichenor**

The purpose of the ISMA (International Software Metrics & Analysis) Conference is to provide educational and networking opportunities to IFPUG members and software measurement professionals in general, by learning and sharing knowledge in the world of software measurement. The Partnerships and Event Committee (PEC), along with the IFPUG's Non-functional Sizing Standards Committee (NFSSC), is sponsoring the virtual conference **ISMA21** on December 1, 2023. The **theme of ISMA21** conference is **Non-functional Sizing.** The reason for this theme is to increase the reach of IFPUG's non-functional sizing method (Software Non-functional Assessment Process—SNAP), create awareness about the Certified SNAP Specialist (CSS) certification, and provide a platform to share the experience of SNAP implementation with other IFPUG members and measurement community.

As most of us know, non-functional sizing is gaining popularity as many organizations and their clients want to have measurement models that cover functional and non-functional sizing. Some organizations are already implementing non-functional sizing along with the IFPUG's functional sizing method (function points).

Let's go a little deeper—what is the SNAP non-functional sizing method, how it is integrated with IFPUG's function points, and what are the benefits?

Paraphrasing ISO, "functional" user requirements describe "what" the software will do in terms of tasks and services. (ISO/IEC 14143-1) IFPUG interprets these aspects as external inputs, external outputs, external inquiries, internal logical files, and external files. From these, the functional size of software can be measured. "Non-functional" software requirements, by standard ISO/IEEE/IEC 24765:2017, describe not what the software will do, but how

the software will do it. Non-functional user requirements are part of almost every software application, and in many cases can represent the most complex or time-consuming aspects of the software development. IFPUG's method of measuring the size of non-functional software is by recognizing four categories and 14 subcategories of these non-functional requirements from which the non-functional size of software can be measured. These ISMA presentations will provide attendees with an overview of the SNAP method and important practical uses of SNAP.

IFPUG currently has two levels of certification related to SNAP: Certified SNAP Specialist (CSS) and Certified SNAP Practitioner (CSP). The CSS certification was introduced in 2023 as the new standard for Software Non-functional Assessment Process (SNAP) expertise and recognizes individuals who have mastered the SNAP method. The CSS designation indicates that the accredited individual, by passing a rigorous exam and achieving a score of 90% or better, has displayed the fundamentals of SNAP at a significant depth. The CSP designation recognizes the initial level of knowledge and skills in the SNAP. The CSP designation will be granted if the individual scores at least 80% overall correct with at least 70% correct on each section of the Certification Exam.

The agenda of ISMA21 includes four presentations on non-functional measurement and one presentation related to functional measurement.

ISMA21 will feature the following interesting sessions:

Overview of the Non-functional Assessment Process (SNAP)

Presented by Charley Tichenor

Abstract: Functional software describes "what" the software will do. IFPUG interprets these aspects as external inputs, external outputs, external inquiries, internal logical files, and external files. Non-functional software describes "how" the software will do it. In many situations, the work effort to develop the "how" portion of the software can be more complex and time-consuming that that of delivering the "what." The non-functional size of software can be measured by considering four categories and 14 subcategories of software requirements. The purpose of this presentation is to describe the thinking behind non-functional software, its sizing, and how sizing non-functional software can contribute to better cost and schedule forecasts for software development work. We will also reference the corresponding international measurement standards.

Speaker: Charley Tichenor has been a member of IFPUG since the early 1990s. He is currently Vice-Chair of the Non-functional Sizing Standards committee and member of the Business Applications committee. He has participated with the SNAP development team since 2011. He is a semi-retired Adjunct Professor at Marymount University in Arlington, Virginia, USA.

Sizing Zero Function Point Projects

Presented by Manjusha Misra

Abstract: Starting with some scenarios involving application enhancements that do not involve changes to functional requirements, we will analyze how the same can be measured effectively with SNAP taking into account changes to non-functional requirements. The scenarios we will cover are the technology migration scenarios that impact the application, e.g., UX migration or transition from monolithic application to microservices application. Is SNAP suitable for measuring all of these? The answer is yes, and we will explore together why ...

Speaker: Manjusha Misra is presently working as Productivity Analyst with Civica, one of the Global leaders in public sector software, head-quartered in the UK.Manjusha has 23 years of IT expertise, 13 of which have been spent measuring software deliveries. She earned her Master of Computer Application from NIT (National Institute of Technology), one of India's most prestigious universities. Prior to joining Civica, she worked with Mphasis and Vodafone at various roles such as Developer, Team Lead, Quality Analyst and Senior Project Manager. Manjusha is a member of NFSSC (Non-Functional Sizing Standards Committee) and certified in Function Point Analysis as well as SNAP.

> The agenda of ISMA21 includes four presentations on nonfunctional measurement and one presentation related to functional measurement.

My Organization Works with FP. How Do We Start Using SNAP?

Presented by Alfonso González Mateo

Abstract: There are many, more and more, organizations that are committed to use the size of software to help themselves to make their development models more efficient. From budgeting stages to the vendors management using productivity and quality controls.

However, throughout all these years it has been common to find a black box, the "non-measurable" software, coming from non-functional requirements.

Until now, there was little we could do beyond monitoring and controlling generalized behavior and its evolution over time.

Fortunately, today we have SNAP, and it really helps, not only to monitor and track this type of development, but also to make it more efficient over time. How do we implement it? Throughout this presentation we will see what aspects we should take into account, recommendations, and lessons learned derived from a real implementation of SNAP in a client used to using Function Points.

Speaker: Alfonso González Mateo is a Computer Engineer, and he holds a Master's Degree in Software Engineering Management and Project Direction. Graduate from Universidad de Alcalá (Spain) and Wrexham University (Wales), he has +15 years of experience implementing and defining productivity management models and vendor governance. He is currently a member of the Business Applications Committee (BAC) to contribute to C-level and management decision-making using quantitative approach. Since 2008, IFPUG CFPS, he also holds the CFL certification by COSMIC since 2009 and became certified as SNAP Practitioner in 2014. Alfonso is currently a partner at Leda MC and works as Account Manager in charge of several worldwide Productivity Services. He also leads internally the Benchmarking area and has extensive knowledge of data standardization, management, and exploitation. Throughout his professional career he performed as an FPA instructor and consultant in Spain, Italy, Belgium, Portugal, and Mexico. Currently, he specializes in model management definition and vendor governance using FPA in agile environments.

Tech Debt – Through Functional and Non-functional Sizing Perspectives

Presented by Sushmitha Anantha

Abstract: The topic of technical debt is very diverse. This term is used for wide spectrum of problems related to deferred work that may or may not be evident from functionality, however, often becomes apparent as reduced quality of the software product. Devastating impact of accumulated technical debt is more readily observable through increasing software maintenance costs due to significant rework needed to fix latent defects and quality issues.

Due to the very nature of technical debt, it is almost impossible to measure it entirely. Even with indirect methods, only partial aspects of technical debt be measured. Through this paper we shall try to conceptually address applicability of functional and non-functional sizing in the context of technical debt and to call out alternatives when functional and/or non-functional sizing do not or partially cover the dimensions of it.

Speaker: Sushmitha Anantha has been volunteering in IFPUG since 2015. She is currently Director of the Non-functional Sizing Standards committee and volunteer in Conferences and Events Committee and other taskforces. Sushmitha is working for Accenture Solutions India as a project manager.

"What's the Right EP Type?"

Presented by Luigi Buglione

Abstract: In the IFPUG FPA method an Elementary Process is characterized by three elements: set of DETs, set of FTRs and set of Processing Logic. This last element is the least often treated in the counts and is used to verify the nature of an EP, as well as the cases in which a EP is to be considered different from others or in a state of CHGA. During this presentation, we will verify together with a small "game" how to facilitate this analysis, in line with the IFPUG CPM v4.3.1.

Speaker: Luigi Buglione is currently an IFPUG board member, Secretary and Director for Partnership and Academic Affairs. He is also president of GUFPI-ISMA (the Italian Software Metrics Association) and Vice-president of ISBSG (International Software Benchmarking Standards Group). Luigi is a Measurement & Process Improvement Specialist at DXC Technology in Rome, Italy.

The ISMA21 conference will be approved as an eligible event for certification extension credits towards CFPS and CSS certifications. View the ISMA webpage for guidance on how to earn extension credits for the event.

To review the full conference schedule and register for the event, visit https://ifpug.org/learning-and-events/isma

BUSINESS APPLICATIONS COMMITTEE

By **Pierre Almén, Chair**

The Business Application Committee (BAC) is proud to announce a new member, Aman K. Singhal from IBM Consulting in India. Aman has more than 25 years of experience within IT services and he is a global leader for Client Services Excellence. Welcome Aman to the BAC team!

The BAC team continues to work with the following prioritized tasks:

- A new version of the document FP as Assets with more focus on the business—the new planned document title is "IT Value Metrics, Key Enablers to the Business." Examples of chapters are Market Risks and Risks Premiums, Flow Metrics, Voice of the Client Indicators, Trends and Early Warning Signs and Retrospective Analysis. The main target group is C-levels and similar kinds of managers. We are now at the end of the creation phase and the review process is going to start soon. In addition to an upcoming release of the document, we will try to create some white papers where we can go deeper and explain the content in the chapters. The target group for these white papers are those who are working with these topics.
- Application Development & Maintenance (AD/M) Benchmarking Certification
    - o The changed documents and process for a re-certification have been sent to the IFPUG Board of Directors and been accepted. Check the benchmarking certification page for more information.
    - o LedaMC from Spain has now been certified for two years. The BAC member Christine Green interviewed Alejandro Hernandez from LedaMC in a Knowledge Café webinar October 26. He shared their experience and the benefits of having the IFPUG AD/M Benchmarking Certification. The recording of the webinar is available now in the IFPUG Learning Center.

CERTIFICATION COMMITTEE

By **Daniel B. French, Chair**

The Certification Committee works daily to:

- Support IFPUG members to take the CFPS/CFPP (IFPUG FP) and CSS/CSP (IFPUG SNAP) exams.
- Assist IFPUG members in applying the CFPS CEP (Certification Extension Program) to maintain certifications without retaking the certification exam and evaluating their submissions for extension approval.

The committee has been working diligently and has several important updates to report:

We have added a new member to the committee, Rodrigo de Asis Vidal from Brazil. Welcome Rodrigo to the Certification Committee, we look forward to your contributions.

Work continues, working with the Non-Functional Software Standards Committee (NFSSC) on developing the training materials for the CSS/CSP certifications.

The committee has also successfully launched the development Certified SNAP Specialist (CSS) exam, and existing Certified SNAP Practitioner (CSP) holders will have to take and pass the entire examination to receive the CSS designation.

A dedicated Certification Extension Program will be applicable to this certification. The CSS CEP will also allow the certification to be renewed beyond its three-year validity, as is already possible at present with the CFPS certification.

Translation of the APM into Italian is nearly complete and the CSS/CSP exam will be offered in Italian as well.

Work has also begun on creating the certification for Simple Function Point (SFP) measurement. When it has been finalized, notifications will be sent out and information posted on the IFPUG website.

The Certification Committee is also working with the Functional Sizing Standards Committee (FSSC) to develop SFP training materials.

It has come to the attention of the Certification Committee and the IFPUG Board of Directors that there are a number of websites purporting to provide IFPUG exam questions and preparation materials and guarantee passing the exams. Please note that these sites are fraudulent and are not in any way endorsed or supported by IFPUG. If you have any questions about the validity of a website offering IFPUG materials, please contact the Certification Committee.

A big thank you to all the members of the committee fortheir dedication, competence and professionalism and the great contributions you all make to IFPUG!

COMMUNICATIONS & MARKETING COMMITTEE

By **Julián Gómez, Chair**

Take a little time to think about this subject: this organization is made by volunteers. Let me highlight it again: by volunteers. People who offer their time and do not expect anything back. But nothing back?

I think that they receive a lot, we receive a lot, but of course, not money or other material goods. They receive a lot of learning, a lot of experiences, and a lot of new friends who share with them knowledge, thoughts, and solutions to common problems. It is a treasure.

Without volunteering, there is no growth.

Without exchanging ideas, there is no improvement.

Today, I want to take this space in *MetricViews* to ask you to go forward and volunteer in our organization. Look at what could be the best for you based on your interests and help us to grow and help this community.

Everything from our annual events, ISMA20 and ISMA21 SNAP (coming in December), to this review is conducted by the work of our colleagues and is a collaborative work conducted by volunteers.

The more visions we have, the more diversity, the more ways to resolve problems and the more ways we have to succeed in this uncertain world.

You can collaborate as the representative of your country or participate in defining the following white paper inside FSCC or NFSSC. You can contribute to our new Education & Research Committee or collaborate in the Communications & Marketing Committee.

As one famous president of the United States said: Ask not what IFPUG can do for you—ask what you can do for your IFPUG community.

Be well, be safe.

FUNCTIONAL SIZING STANDARDS COMMITTEE

By **Esteban Sanchez, Chair**

The Functional Sizing Standards Committee (FSSC) is devoted to IFPUG and the entire community of function point practitioners around the globe. Our team works with passion and commitment to maintain and augment the guidelines in the Counting Practices Manual (CPM) and ensure that the standard can be applied universally to both, traditional and new technologies. The world is under constant innovation, and we strongly believe that the same guidelines originally conceived by Mr. Albrecht several decades ago, continue to be successful in measuring the functional size of emerging software technologies.

Our most recent publication, "Elementary Processes and User Stories" is a masterpiece in the realm of Agile methodologies; the paper provides examples of common scenarios for counting function points in Agile Software Development (ASD) and their analysis according to the rules of the CPM. The white paper brings scenarios that may raise doubts in the function point community regarding the correct way to evaluate users' functional requirements in accordance with the CPM rules. This paper is already available from the IFPUG Learning Center and we are also planning a complementary webinar for January 2024.

Still on the Agile domain, our most recent webinar "Accelerating Agile Success: Unleashing the Power of Function Point Analysis with Decoupled Cadences and Kanban" provides a summary of the core recommendations to apply function points in a Kanban environment.

The FSSC recently put the magnifying glass on the topic of system clock and other platform data (information provided by the operating system to the applications). The result of this analysis will be a comprehensive paper with guidelines on how to count system clock and other platform data. The paper will include examples and recommendations on what things can continue to be treated as functional and hence covered under the umbrella of the CPM, and what aspects should be considered non-functional and therefore approached thru the Software Non-Functional Assessment Process (SNAP).

On the back burner is a case study on the topic of Mobile Applications. This will be a comprehensive work that illustrates the application of function points to a full mobile application with cloud backend.

The FSSC is a catalyst in the adoption and empowering of Simple Function Points (SFP). We believe that SFP is the perfect complement to full Function Point Analysis. Particularly useful in situations where the requirements are not detailed enough

to perform a full count, and still successful in providing a good indicative of the functional size. Several of the artifacts we have under development will involve SFP. For example, we are developing a paper on the topic of SFP for Agile Software Development. Stay tuned for a paper on artificial intelligence and software bots.

If you want to be part of the team that is making all these great things possible, just get in touch with us. Please complete the IFPUG Volunteer Form on the IFPUG website: https://ifpug.org/about-us/committees/volunteer.

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

INTERNATIONAL MEMBERSHIP COMMITTEE

By **Loami Xavier de Barros, Chair**

First of all, my special thanks to the representatives of Brazil (Cristiane Baccarin) and Italy (Paola Billia) for their tireless work over the last few months supporting their country members!

The International Membership Committee (IMC) is currently responsible for:

- **Simple Function point manual translation process:** We are verifying if the translation group is following all the procedures specified on it:
  - Italy, Japan, China, and Spain translation teams are currently working on this.
  - The Brazilian team has already finalized it and it is in the process of an internal review by IFPUG.
- **Volunteer process:** We are involved in every step of the process between volunteer and committee chairs.
- **Translate all the most common user requests into English:** We have received it and it will be added to the IFPUG site as FAQ; this will simplify the support needs by all.
  - Brazil has already finished this.
  - Representatives from other countries are working on this.
- **IMC support:** We act as the main point of contact for related queries and interact with IFPUG members so that you continue to benefit from your memberships, and we are more than eager to assist you with all IFPUG-related queries. Feel free to send us your IFPUG improvements and suggestions.

We currently have representatives for France, Spain, Brazil, China, India and Italy. In the near future, we are planning to have a representative for both Argentina and Colombia.

Representatives from other countries are very welcome. If your country does not have a representative, let us know or you are welcome to join our team too.

Please visit the IFPUG Volunteers page if you would like to participate in the IMC or other committees by visiting https://ifpug.org/about-us/committees and looking under **"Get Started as an IFPUG Volunteer Leader."**

Join our team and become an IMC representative in your country too!

INDUSTRY STANDARDS COMMITTEE

By **Carol Dekkers, Chair**

The IFPUG Industry Standards Committee (ISC) currently includes several IFPUG leaders: Carol Dekkers, who is the U.S. National Body Representative to the ISO/IEC software and systems engineering standards, and Steven Woodward, who is involved with the Cloud Computing efforts with NIST and is on the Canadian delegation to ISO/IEC software and systems engineering standards, in addition to other IFPUG members who participate as part of the Institute of Electronic and Electrical Engineers (IEEE) standardization efforts.

Our collective ISC work continues to include outreach in several areas including:

- Ongoing participation as part of the INCITS (U.S. technical advisory work) to ISO/IEC JTC1 SC7 Systems and Software Engineering Standards);
- Subcommittee 38 work (by Steve Woodward of Canada, who works with both SC38 and as the liaison to SC7); and
- IEEE and ISO/IEC standards development work. Currently, work is underway with the international standardization of a joint work item between ISO/IEC and IEEE for the 32430 SNAP standard (under the leadership of IEEE representative Talmon Ben-Cnaan.) Supporting the effort are Cinzia Ferraro and myself who have been variously serving as officers to the joint working group (I as a co-project editor and U.S. national body representative, and Cinzia as the comment resolution group chair and a member of the Italian national body.) In addition, Dr. Charley Tichenor and NFSSC Chair Fabrizio Di Cola, together with a cast of supporting experts, have provided much appreciated expertise.

Without boring you with more acronyms, ISO procedures, and tedious meeting notes, it is sufficient to say that this project has been a huge team effort of many international parties both within and outside of IFPUG membership. International standards are seldom straight-forward and involve multiple rewrites/drafts and committee ballots and a multi-year journey. I

am personally thankful for every person who has provided their input, comments, and expertise. I look forward to a 2024 publication date.

If you have read this far, you may be wondering how you can help. One easy way is to consider using the SNAP standard as part of your own corporate measurement initiative and submitting your SNAP data to the ISBSG database (https://www.isbsg.org/submit-data/). The more data that are available, the better the research findings will be to support SNAP. While SNAP usage is increasing worldwide, it remains important to the work to have an increased amount of real-life data available to industry researchers.

Thank you.
Carol

NON-FUNCTIONAL SIZING STANDARDS COMMITTEE

By **Fabrizio di Cola, Chair**

The IFPUG Non-functional Sizing Standards Committee (NFSSC) continues its activities to explain to the industry what SNAP is, the benefits from its use, how to measure certain sizing scenarios, and to train future trainers on SNAP.

Do you want to use SNAP in contracts and need a certification that has a renewal process, similar to the CFPS CEP? This way you can be sure that you have staff properly trained and ready to measure software non-functional user requirements measured by SNAP. IFPUG launched the Certified SNAP Specialist (CSS) certification, which complements the existing Certified SNAP Practitioner (CSP) certification. This will help you introduce non-functional measurement from a contractual perspective as well. So, you are only a short time away from having the opportunity for this certification level.

We are preparing a series of examples of SNAP use that will concretely help organizations complement IFPUG function points by measuring the non-functional dimension.

- Example of measuring developments and enhancements.
- How to move from an FPA count to a SNAP count.
- Examples based on a range of scenarios involving the "Zero function point project."

And we will continue to enrich this list.

In a short time, the SNAP manual will be released in a new language—Italian! We are finalizing the translation of the APM and, together with the Certification Committee, we are translating the CSS/CSP exam into Italian!

Other important activities either completed or nearly completed during this period include the following:

- The writing of a new white paper that will give guidance on how to apply SNAP to applications built with microservice architectures.
- The finalizing of an important white paper on measuring security requirements.
- The growing use of our first five YouTube videos overviewing the SNAP method. We have 786 views as of this writing. We encourage you to access these by either searching YouTube by "IFPUG SNAP," by "sizing non-functional software," or something similar. Please "Like" them if you do.

We always need your help. Working in the NFSSC allows you to be in contact with some of the best professionals in the measurement of non-functional requirements for software, know the background of the choices you will later apply in your organizations, or speak in universities. The measurement of non-functional dimensionality in software is absolutely one of the hottest topics in the industry in recent years. For those interested in working with us on a groundbreaking topic such as non-functional dimension measurement with SNAP, please send in your application by going to https://ifpug.org/about-us/committees/volunteer.

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

PARTNERSHIPS & EVENTS COMMITTEE

By **Kiran Yeole, Chair**

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

Events:

The IFPUG Knowledge Café series is an exclusive platform to share your ideas, innovations, and experience in the field of measurement with other IFPUG members and the measurement community. This platform also provides an opportunity to learn from each other's experiences and networking.

In this calendar year of 2023, we have already conducted six knowledge café webinars, and we are planning a few more soon.

Following are the recent two webinars which we conducted during September 2023.

1. **Carlos Eduardo Vazquez** (software measurement & Agile expert, founder of "FATTO Consulting" a member of IFPUG Functional Sizing Standards Committee,) and **Esteban Sanchez** (estimation & cost analysis expert and chairperson

at the IFPUG Functional Sizing Standards Committee) presented the topic **"Accelerating Agile Success: Unleashing the Power of Function Point Analysis with Decoupled Cadences and Kanban."** Speakers helped us to discover the winning formula that drives unparalleled agility and productivity in software development.

2. **Luigi Buglione** (measurement and process improvement specialist at DXC Technology and IFPUG board member) presented the topic **"Diversity & Inclusion (D&I) KPIs."** In this webinar, we discussed the "backfiring," which describes the correspondence between lines of code (LOC) and the average (or median) FP for programming language/ environment and Luigi walked us through some application scenarios taken from the daily reality of measurement.

IFPUG Knowledge Café series brought you the sixth webinar of this calendar year on **October 26, 2023.** This webinar was about the **"Benefits of the IFPUG AD/M Benchmarking Certification"** presented by Christine Green (a senior consultant & advisor) and **Alejandro Hernández** (partner and member of LedaMC's Management Committee). In this webinar, we delved into the success story of LedaMC, the Spanish Benchmark Company that has achieved the esteemed IFPUG Benchmark certification and became the first company to achieve this. Christine Green (immediate past president of IFPUG) interviewed the leader of LedaMC, Alejandro Hernández, aiming to comprehend the motivation behind obtaining this certification and the significant impact it has had on the ability to market benchmark services.

Partnerships:

As you know, IFPUG is advancing the partnership with Netherlands Software Metrics Users Association (NESMA). As part of this, IFPUG announced the agreement with NESMA recognizing that we share specific objectives. IFPUG and NESMA have agreed to cooperate in overlapping domains of expertise, as well as mutually work on endorsement of the sizing standards, mutual development of content, facilitating professional networking opportunities and joint development and promotion of educational activities in software sizing, metrics, and measurement.

As part of this partnership, IFPUG and NESMA have agreed to work together on the development of a new white paper on **"What Really Matters in the Agile World."** We believe that this joint work will benefit the entire measurement and sizing community to a great extent. The joint taskforce with members from both IFPUG and NESMA are working on this whitepaper, and we are hoping to publish the white paper in coming months.

We regularly offer platforms for interesting topics to be discussed at our Coffee Talks and ISMA conferences. Please write to pec@ifpug.org with your suggestions for topics and speakers. If you are interested in working with the PEC, please complete and send a volunteer form to pec@ifpug.org or submit the form using https://ifpug.org/about-us/committees/volunteer

We want to take this opportunity to welcome our two new volunteers who recently joined the Partnerships and Event Committee. **Thiago Silva da Conceição Sr.** from Brazil and **Cleber Ferrareze** from Belgium.

FORECASTING AND SOFTWARE ESTIMATION COMMITTEE

By **Christine Green, Chair**

In the ever-evolving realm of software development, IFPUG heralds a groundbreaking initiative with the establishment of the Forecasting and Software Estimation Committee (FSEC). This strategic move is dedicated to enhancing the precision of forecasting and estimation in software projects—crucial for predicting cost, time, and resource allocation.

Watch out for the newsletter on this new committee with more details and how to volunteer for this committee.

TRAINING PROGRAM TASKFORCE

By **Christine Green**

Exciting news from IFPUG—we're launching a new Training Program Taskforce! This team, spearheaded by Christine Green, past IFPUG President, will craft a forward-thinking strategy for training in our three core sizing standards. To truly reflect the community's needs, we're seeking volunteers from diverse backgrounds and geographies who have hands-on experience with Function Point Analysis (FPA), Software Non-functional Assessment Process (SNAP), or Simple Functional Points (SFP).

This is your chance to help shape IFPUG's educational future and make a tangible impact. Engage with top-tier professionals, enrich your understanding, and help us steer the course of our training programs. Ready to join? Expect bi-weekly meetings, collaborative design sessions, and regular updates.

For a detailed dive into the initiative and how to get involved, keep an eye out for our upcoming newsletter. Your expertise can help define our path forward. Volunteer for the taskforce and be at the forefront of IFPUG's training transformation!

Stay tuned and volunteer to be at the heart of IFPUG's training evolution!

191 Clarksville Road
Princeton Junction, NJ 08550 USA

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