INTERNATIONAL
FUNCTION POINT
USERS GROUP

191 Clarksville Road
Princeton Junction, NJ 08550
USA

September 2018

Vol. 12 • Issue 2

www.ifpug.org

MetricViews

SOFTWARE MEASUREMENT MEETS AUTOMATION

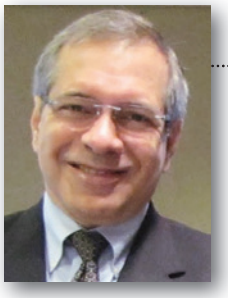
In this Edition 2
Message from the President 3
From the Editor's Desk 4
IFPUG's Position on Automated Tools … The Promise of an Exciting Future 5
Function Point Automation: Humans Always Matter? 6
Human Centric Automation 9
Measurement Automation Using M.A.R.IN.E 12
Automation Tools for Real-Time Embedded Software Functional Size Measurement 16
Think You're Ready for a Measurement Tool? Top 10 Questions to Ask Before Going Shopping 21
Simulation Models Based on ISBSG Benchmark Data 24
Automation and Too Many Metrics Sharing the Word "Points" 28
Thanks, ISMA15, Rome: Inspired by Others, Sharing Ideas and Synergies 32
Managing Agile Activities Using Standardized Measures and Managing Agile at Scale: A Briefing for Software Executives and CIOs 36
Certification Committee 38
Communications and Marketing Committee 38
Conference and Education Committee 39
Functional Sizing Standards Committee 39
International Membership Committee 40
Industry Standards Committee 40
Non-Functional Sizing Standards Committee 41
Committee Roster 42
CFPS Certification Listing 43
CFPP Certification Listing 44
Fellow 44

IN THIS EDITION

Here is a snapshot of the exciting articles you will find in this edition of *MetricViews:*

- **IFPUG's Position on Automated Tools…The Promise of an Exciting Future** (Chuck Wesolowski, Director of Counting Standards, and Carol Dekkers, Director of Communication and Marketing) A word on IFPUG's position on automated tools from the Board of Directors.
- **Function Point Automation: Humans Always Matter?** (Eduardo Orefice and Sara Terrani) Learn what the problems are that prevent full automation and what's possible in future developments.
- **Human-Centric Automation** (Sujatha Sivaraman) Autonomation stands for Automation with a Human Touch. Learn how measurement can become more consistent with less manual effort.
- **Measurement Automation Using M.A.R.IN.E** (Kevin McKeel) A tool for improving and automating the requirements analysis and software sizing process.
- **Automation Tools for Real-Time Embedded Software Functional Size Measurement** (Hassan Soubra) Measurement with automated tools eliminates possible measurement variances caused by different measurers, which may lead to different measurement results for the same set of requirements.
- **Think You're Ready for a Measurement Tool? Top 10 Questions to Ask Before Going Shopping** (Carol Dekkers) "A fool with a tool, is still a fool." A software tool should be looked at as an investment to help streamline your work. Top 10 questions to ask.
- **Simulation Models Based on ISBSG Benchmark Data** (Milan Rao) Data-driven models help PMs make better decisions early in life cycle of the projects.
- **Automation and Too Many Metrics Sharing the Word 'Points'** (Antonio Ferre Albero) Too many metrics sharing the word "points" for naming completely different concepts which creates confusion and even misunderstandings.

And be sure to enjoy these two additional articles: Thanks, ISMA15, Rome: Inspired by Others, Sharing Ideas and Synergies and Managing Agile Activities Using Standardized Measures, and Managing Agile at Scale: A Briefing for Software Executives and CIOs. ■



Message from the President

Mauricio Aguiar

On Making Justifiable Decisions

Both at IFPUG and at my job, I've often been faced with a decision where I have to argue for or against a request. In those cases, just making a decision is not enough — I need to justify the decision by explaining the rationale that led to my conclusion.

One technique that has helped me comes from a book originally written for lawyers: *Trial and Practice Skills in a Nutshell*, by Kenney F. Hegland, a law professor from the University of Arizona. The book recommends using three criteria (my interpretation follows):

1. FACTS (Do the facts support the request?): Here the point is to assess "merit." Do the available facts confirm this is a good idea? Is it fair and correct to follow this route of action? Is there objective evidence this is an idea that will contribute to our goals?

2. PRECEDENT (How similar requests were handled in the past): Here the important point is to keep all judgments consistent. If similar requests were rejected in the past, there should be a justification for this one to be approved. If there are precedents, one should be able to demonstrate this request is similar to one or more that were approved before.

3. POLICY (What is the future impact of this decision?): Here one should review the consequences of approving the request under judgment. Will the approval create a precedent that may lead to the approval of similar ones in the future? What are the consequences of this? How will this impact the organization? Is it safe to approve the request from this perspective?

Example: An employee wants the company to pay for his or her expenses to attend a conference . In this case, company policy allows paying for the expenses of employees selected to speak at conferences, provided there is a potential benefit to the company. The employee argues that even though he/she will not speak, he/she will collect useful market information and prepare a report for management.

Sample questions to help make a decision:

1. FACTS: Will the information collected be useful? Is the conference relevant to the company? Is the report potentially useful to justify the costs? Are the costs reasonable when compared to other alternatives to reach the same goals?

2. PRECEDENT: Is this an exception, or have we approved this kind of request before? Is there a precedent? Is there a past judgment that may render this idea unacceptable?

3. POLICY: If this request is approved, will other similar requests tend to occur in the future? How will the organization handle such requests? How will approving this request affect the existing policy? Will the current policy have to be changed?

The decision will depend on the answers to the questions above. Even though a request or idea may be considered from many different perspectives, the structure described may help to reach justifiable conclusions.

Mauricio Aguiar
IFPUG President

Note: Thanks to Professor Hegland for allowing me to use his ideas. His book can be found at amazon.com/Practice-Skills-Nutshell-Kenney-Hegland/dp/0314257306/

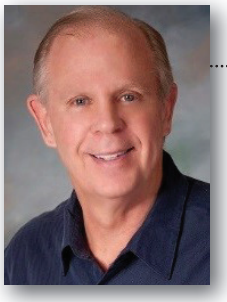
MetricViews
Published twice a year by the International Function Point Users Group (IFPUG), headquartered in Princeton Junction, New Jersey, U.S.A.

MetricViews September 2018

Editor
David Herron

Please submit all articles, news releases and advertising to:

IFPUG/*MetricViews*
191 Clarksville Road
Princeton Junction, NJ 08550
(609) 799-4900
ifpug@ifpug.org



David Herron

From the Editor's Desk

When the *MetricViews* topic of measurement automation was first suggested, there was some hesitation over concerns with what we would receive in response to our request for articles. Would there be general interest in the topic? Would there be enough variety of articles? Would companies use this as a way to market their products and services? As you read through the various articles in this edition I think you will find that there is a lot of interest along with a variety of initiatives focused on the quest for automating measurement and in particular, automating software sizing. The truth of the matter is that we received more articles than we can fit into this edition and so we will see those articles published in future editions of *MetricViews*, helping to continue the conversation around this topic. You may notice that some of the articles border on the marketing/advertising side of the line, but why not? We should all be for advancing the need for and benefits of measurement automation and that takes a bit of salesmanship.

Two thoughts came to mind as I read through the current group of articles. First, I noticed that the authors of these articles truly represent an international community. The desire for and the resources invested in measurement automation are not unique to one region or to one country. Companies and individuals within those companies are putting forth the effort to create and to share advances they are making in automation. The second thought I had was that somewhere down the road we are going to need an agreed upon set of standards regarding how software is sized. Sizing approaches and specifications currently based on OMG, COSMIC or IFPUG standards are appropriate within their respective user communities. We are still very much in the early stages of size automation and so current efforts should not be constrained by demanding alignment with any single approach. However, there will be a time when we will need to collaborate and join forces in the quest for automating the task of sizing software.

David Herron
Communications and Marketing Committee

IFPUG'S POSITION ON AUTOMATED TOOLS...

The Promise of an Exciting Future

By Chuck Wesolowski, Director of Counting Standards, and Carol Dekkers, Director of Communication and Marketing

On behalf of the IFPUG Board, Chuck and Carol wrote the following words to outline where your board stands in relation to automated tools. We welcome reader comments (email: cmc@ifpug.org).

This issue of *MetricViews* addresses several different approaches to automating software measurement. This idea is not new, although traditionally it seems the subject has been met with some skepticism amongst Function Point practitioners.

To us personally, and to others in the IFPUG community at large, it seems a logical next step beyond standardization is to automate aspects of software measurement. After all, our community is comprised of software people who specialize in different facets of its development, operations and maintenance.

Since the first IFPUG counting practices manual 1.0 was released in the late 1980s, there has been interest in automating everything from recording Function Point counts to full-blown Function Point counting tools. IFPUG supports this development and advises our members and readers to validate all promises by tool providers, especially those claiming IFPUG compliance.

As a service to both our members and tool providers, IFPUG offers a certification program for software tools. As such, IFPUG defines three categories for "IFPUG-Certified Software." These definitions have not changed in 20 years, and they are worth a review:

Type 1 Software provides Function Point data collection and calculation functionality, where the user performs the Function Point count manually and the software acts as a repository of the data and performs the appropriate Function Point calculations.

Type 2 Software provides Function Point data collection and calculation functionality, where the user and the system/software determine the Function Point count interactively. The user answers the questions presented by the system/software and the system/software makes decisions about the count, records it and performs the appropriate calculations.

Type 3 Software carries out an automatic Function Point count of an application using multiple sources of information such as the application software, database management system and stored descriptions from software design and development tools. The software records the count and performs appropriate calculations. The user may enter some data interactively, but his or her involvement during the count is minimal.

For more information, visit ifpug.org/certification/software-certification/.

Note that there has never been an application to date for Type 3 Software certification by IFPUG and, as such, the process has never been fully developed. Should there be such application in the future, the IFPUG Board will formalize and standardize this certification.

Chuck wrote a *MetricViews* article about Type 3 Software a few years back. It focused on functional sizing in the context of the CMMI requirements development process area. It identified what must be present in the requirements artifacts to perform an IFPUG Function Point count and spoke to the quality of the requirements necessary to count rather than estimate the functional size of the specified software. It closed with:

> *Some critical questions include: what artifacts, or "stored descriptions from software design and development tools," used by an organization contain this information? And, are the function point counts derived from these artifacts?*

Source: ifpug.org/Metric%20Views/MetricViewsFebruary2014.pdf

As a software measurement consultant, Carol has encountered resistance to Function Point adoption from clients whose main concerns are how to streamline the

continued on page 6

(IFPUG's Position on Automated Tools, continued from page 5)

Function Point counting process and reduce the labor intensity associated with counting monolithic software packages and systems. Certainly, automated tools offer promise to alleviate these concerns and, at the same time, Carol cautions that tools are not necessarily the panacea for those looking for a quick fix (see her article "A Fool with a Tool…" later in this issue). To date, the tools presented in the market still affirm that Function Point counting is an empirical process, that is, the Function Point results from a tool need to be reviewed by a qualified CFPS. Tools can and do automate the clerical aspects of Function Point counting, and the review process by a CFPS serves to validate (e.g., make sure that the results do not include duplicated functions, follow Function Point rules and make sense) and confirm the accuracy of Function Point results.

Having said this, it is exciting to have an entire issue of *MetricViews* dedicated to this subject. The included articles represent various industry views on automation of metrics collection and reporting irrespective of software sizing unit. These are individual/organizational experiences and not validated/endorsed by IFPUG.

We hope that everyone enjoys reading this issue as much as IFPUG enjoyed producing it. Comments and feedback from our readers are encouraged by emailing cmc@ifpug.org.

Happy reading! -- Chuck and Carol ■

Function Point Automation:

HUMANS ALWAYS MATTER?

By Eduardo Orefice and Sara Terrani

Software sizing automation was born in the 1960s with the automation of counting lines of code. When functional metrics were first introduced, they were preferred because the results were easier to link to software costs—the path to automation has since slowed down. The design of tools to automate software size has accompanied the evolution of functional metrics in the last 40 years. Despite the great steps forward, to date complete automation has not yet been reached. In fact, human intervention in the software evaluation process is still essential. The authors of this article, senior consultants in the governance of large IT organizations with 10 years of experience in the use of metrics and tools, offer their contribution on the topic by describing the criticalities and proposing suggestions for possible future developments.

The evolution of software metrics has generated a great level of user awareness and maturity and that has guaranteed the success of those methodologies that have been able to meet the needs of large organizations and to adapt to innovation. The diffusion of IFPUG Function Point Analysis is valid proof of this. The search for total automation of metrics goes on, but the efforts made so far—in the improvement of methodologies and support tools—are not yet enough to reach the goal: the complete elimination of human intervention, which commits resources in terms of time and people and inevitably leads to errors. In the remainder of this article we try to explain, according to our experience on functional metrics and measurement processes, the problems that prevent full automation and we offer some food for thought for possible future developments.

From LOCs to FP

The "Lines of Code," introduced as of 1960, initially proved to be quite effective given the low number of programming languages and the incidence of coding on the total project effort. Their rapid spread, due to simple application and total automation, was countered when the increase in the number of programming languages and ever larger applications (from less than 1,000 lines of code up to more than 10,000,000 lines of code). In addition, the introduction of new programming paradigms highlighted the limits of being able to estimate level of effort. The insight of Allan Albrecht, Function Point's inventor, was to size the software starting from the categorization of user requirements with an approach independent of the technology and the number of instructions made. On one hand, this has enabled organizations to better estimate software implementation costs, however, it has also caused the metrics automation process to slow down. Compared to the lines of code, the automation of Function Points is much more complicated.

The evaluation of functional requirements—based on the rules of the IFPUG manual that emphasize the user's point of view—requires analysis that often can lead to different evaluations on the same software product. In order to solve the problem, the efforts of the market operators have progressively focused on the production of tools that allow the automatic "extraction" of the Function Point number starting from the documentation instead of the code. Although conceptually consistent with the metric, this approach didn't produce significant results, mainly because functional requirements are mostly written in natural language and, even if structured, they are produced according to formats more driven by organizational processes and development methodologies than by measurement needs. Therefore, to complete Function Point automation, it would need to standardize the documentation and make it unique for all organizations. Inevitably this has, up to now, made it impossible or unproductive to pursue this avenue with the current market technologies. Indeed, the most advanced experiences of Function Point automation are still limited today to the support of measure compilation. There are products on the market that have proven to be valid supports for organizations that want to make reliable and safe data collection and dissemination of results.

Typically these products have the ability to:

- Manage counts
- Manage baselines
- Support audit process
- Enable productivity analysis
- Make internal and external benchmarking

Currently there are, however, no tools that can fully calculate Function Points due to the presence of decision-making processes that machines cannot reproduce from the documentation alone.

Automated Function Points

The CISQ OMG Automated Function Point (AFP) metric was introduced to calculate application Function Point through source code analysis. Unlike lines of code, the code inspection is aimed at identifying elementary processes and relevant logical files from a user's point of view. Indeed, the CISQ AFP standard is based on the IFPUG counting guidelines. Operatively, after appropriate calibration of a probe by an expert, the code is examined looking for transactions to the user and/or to other applications. The automatic evaluation exceeds the partial subjectivity of the IFPUG method, since the general problem about determining what the meaning of a written text (requirements document or software specifications) is solved by uniquely interpreting the examined code. However, a tool which implements the metric, regardless of the investments it may require, cannot renounce the intervention of the metric expert.

In fact, to guarantee reliable results, it is necessary, on a case-by-case basis according to the aims of the measure, to define the scope of the count and the boundary of the application that we know are elements that strongly influence the results. In addition, AFP counts may differ from the manual counts produced by IFPUG certified counters. Ultimately, although these tools are very useful for those who want

continued on page 8

(Function Point Automation, continued from page 7)

to measure the size of an application or an entire application portfolio in the absence of documentation, the inevitable ex-ante calibrations (in the setting phase of the counts) and ex-post (correction of the counts) does not yet allow to appreciate the cost/result benefits in preferring a count with AFP to the IFPUG Function Point manual count. It goes without saying that the use of automatic instruments guarantees a greater replicability of the measure than manual counts.

Conclusions

Although 40 years have passed since their introduction, IFPUG Function Points are still the most widely used metric for software measurement. The characteristic to be independent of technology and the analysis from the user's point of view make Function Point a versatile metric capable of overcoming even the challenges of new technological paradigms (Big Data, DevOps, etc.). In all likelihood it will be able to always withstand even in the future. But despite this, the need to automate the calculation process did not find equally complete answers. The AFP initiative brings us closer to the goal of an instrument that can "decide for itself." But, from our point of view, using the source code to get the measurement also creates dependencies by technology and forces the instruments to adapt continuously. Moreover, the new development paradigms can undermine this approach. Therefore, the efforts of market players should be focused on developing solutions that allow the interpretation of texts (user requirements and/or use cases), exploiting the wave of technological innovation of industry 4.0 (artificial intelligence, machine learning, etc.).

An automatic evaluation of user requirements combined with artificial intelligence can perhaps completely automate the process of calculating Function Points, and this is the area we should further explore over the next few years. We are spectators of an almost total computerization of all activities, including business, and also the measurement of software should soon achieve this result. ■

References:

- *ISO/IEC 14143-6:2012: Guide for use of ISO/IEC 14143 series and related International Standards iso.org/standard/60176.htmIISO/IEC 20926:2009 (IFPUG): Software and systems engineering—Software measurement—IFPUG functional size measurement method 2009 https://www.iso.org/standard/51717.html*
- *ABOUT THE AUTOMATED FUNCTION POINTS SPECIFICATION VERSION 1.0 omg.org/spec/AFP*
- *Jones, C. A Short History of Lines-of-Code Metrics; Capers Jones & Associates LLC; Narragansett, RI; December 2012, ifpug.org/Documents/Jones-LinesofCodeMetricV6.pdf*
- *Jones, C. The Origins of Function Point Metrics; Capers Jones & Associates LLC; Narragansett, RI; May 2016; ifpug.org/Documents/Jones-OriginsOfFunctionPointMetrics.pdf*
- *Problem Solving, Control Methods, and Search DISI - Department of Computer Science and Engineering University of Bologna*
- *Mitchell, T. (1997), Machine Learning, McGraw Hill*
- *George F. Luger (2009) Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th Edition)*
- *Albrecht, A. Software Function, Source Lines of Code, and Development Effort Estimation – A Software Science Validation. 1983*
- *Object Management Group, Inc., Automated Function Points (AFP) Version 1.0, January 2014, omg.org/spec/AFP*
- *Roger Heller, Automated Function Point Counting–A Fact Based Analysis, Q/P Management Group, 2013, qpmg.com/Library/confirm-download.php?id=49&*
- *Fischman, L. "Evolving Function Points" Crosstalk: The Journal of Defense Software Engineering. 2001. Volume 14, Issue 2. pp 24-27*
- *Morris P. Guidance on How to Choose a Functional Size Method—Total Metrics, https://www.totalmetrics.com/function-point-resources/downloads/R185_Why-use-Function-Points.pdf*

About the Authors:

Eduardo Orefice *(Business Integration Partners, Italy), graduate in management engineering, has more than 15 years of experience in Information Technology (IT), project management and metrics for private and public large IT companies. He is CFPS since 2009. He specializes in a variety of field disciplines including IT Software Sizing & Estimation, IT Contract Management, IT Performance Improvement Engineering, IT Process and IT Service Management.*

Sara Terrani *(Business Integration Partners, Italy) has almost 10 years of experience in IT. She is a CFPS since 2013 and she's working on a large-scale audit Function Point project in Italy's public administration. She has in-depth knowledge of IT processes as well as IT Program Management experience in diversified areas.*

Human Centric Automation

By Sujatha Sivaraman

Autonomation is the keyword that stands for Automation with a Human Touch. It also means stopping whenever an abnormal condition is detected, fixing the defect and then counter-measuring to prevent further occurrences.

The IT industry is undergoing a paradigm shift with the advent of artificial intelligence, machine learning, deep learning and other cognitive learning techniques on the one hand. On the other hand, it is still grappling with the basic challenges related to meaningful and standardized measurements. This article explains the challenges and possible solutions using analytical techniques that can be implemented so that the measurement becomes consistent and repeatable with less manual effort using autonomation.

The measurement challenges can broadly be classified under two areas:

Challenge #1: Plethora of Size Measures

Variety of size measures makes measurement and standardization more difficult, since functional size is the centerpiece of any benchmarking and productivity analysis.

In old economy industries, we don't have weight measures like iron, gold or platinum weight. Weight measure is the same whether we measure any type of material, whereas in the software industry we have variety of size measures such as Function Points, Story Points, Use Case Points, Complexity Points and SNAP points for non-functional requirement. Even when we do end up measuring one of them, there are no conversion factors which can be easily available to convert one from the other.

In Application Development and Maintenance (ADM) projects, it is a problem of having plenty of measurement choices, whereas in a Application Maintenance and Support

continued on page 10

(Human Centric Automation, continued from page 9)

(AMS) projects, they are virtually absent. Any estimation or planning in these projects is predominantly based on service delivery managers' hunches. Hence it becomes imperative to standardize the metrics.

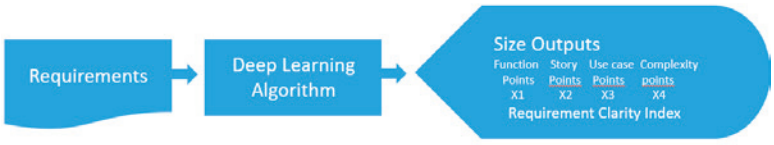
Challenge #2: Effort Required to Measure

The variegated-size measures make it more difficult first to understand the measure itself and then to expend effort to measure it. The effort required to report the metric is not unobtrusive. It requires manual effort and can lead to inconsistencies in measurement.

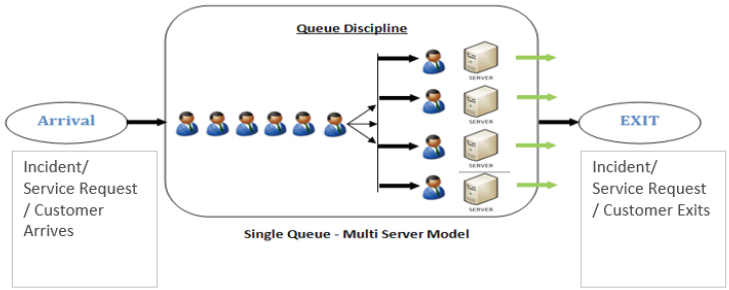
Suggested Solution #1: One Source of Input to Produce Various Size Measures as Output

For ADM Projects: Since requirements form the basis of size measurement, it should be possible to approximate or measure the size based on its requirement clarity and such measurement should be automated.

There are certain tools available on the market that allow enterprises to build their own solution leveraging deep learning algorithms, which will scan the requirement and indicate various size measures and indicate the level of clarity. The resulting sizing output should automatically indicate the Function Points, equivalent story points and any other sizing metric an organization uses. This should be automated in a way similar to how a diagnostic tool produces a diagnostic report when given a measurable parameter.



For AMS and Infrastructure and Support projects: System utilization and the number of servers required, response time and resolution time can be estimated using a queuing model1.For infrastructure support projects, the size of the infrastructure supported shall be measured using a sizing mechanism such as Syspoints2.



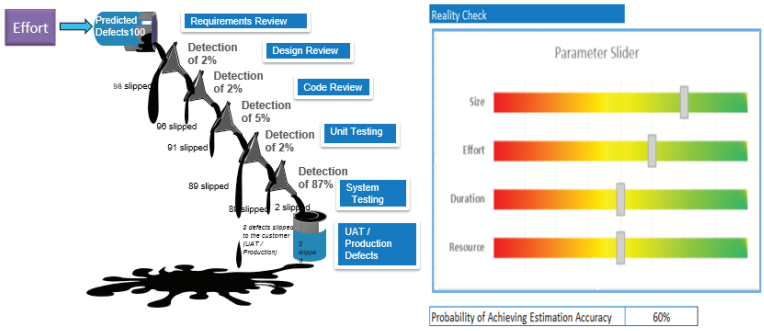
Probable Solution #2: Minimizing Effort Required to Measure Through Automation of Measurement and Instrumentation

There are critical sub-processes in measurement. They are data capture, validation and cleansing and reporting and analysis. In most enterprises, the reporting part is automated and the other three parts require large manual interventions. The analysis part is quite often left to the interpretations of the consumers.

Data Capture Improvisations: Data capture, in terms of effort and schedule, is largely automated through enterprise management tools whereas quality defects are often a manual entry process. In the case of projects that have client security restrictions which do not allow for sharing of defect details, provisions should be made for defect summaries. Projects without any restrictions should be programmed to automatically upload defect details to avoid potential errors.

Data Cleansing and Validation: A lot of manual time is spent on manually cleaning and validating data at the enterprise level, which could potentially be automated through scripts to minimize human error.

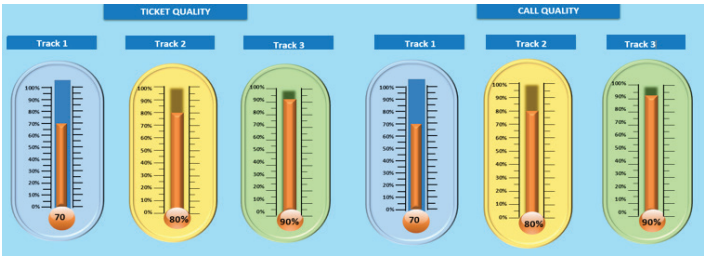
Analysis for ADM Projects: It is not enough to show the outcome in terms of size measure, but also visually represent the sub-processes, health, Key Performance Indicators (KPIs) of these sub-processes, planned effort and actual effort of the various phases, possible internal defects, defect leakages and test cases required for the estimated functional size. Detecting abnormalities should happen through predictive and prescriptive analytics.



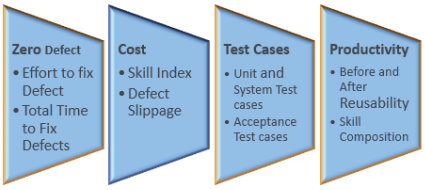
For AMS Projects: For monitoring systems or L1 support projects, the incoming volume would be high and monitoring the queue manually may take a lot of effort. A real-time dashboard, Kanban, or QBAN (See Figure 4) can show the various queue stages and a Red Amber Green (RAG) status indicator would show if certain thresholds are breached.

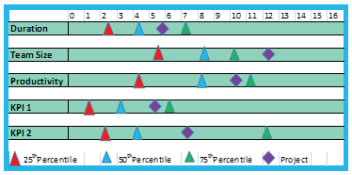
	SHIFT TURNOVER	Assigned	Investigation		L2 Contact			Journal Entry			DONE
			WIP	Done	WIP	Pending	Done	WIP	Pending	Done	
TRACK 1	10	10	2	8	3	2	5	8	8	8	8
TRACK 2	200	20	2	8	2	2	5	8	8	8	8
TRACK 3	300	30	2	8	2	2	5	8	8	8	8

To measure service quality, assign a scoring pattern to key parameters of their service and instead of manual scoring, automate it through a machine learning algorithm that produces a scale of one to 100.



Counter measures to fix and prevent defects should be enabled through recommender systems and diagnostic analysis at project level, business unit and organization level.

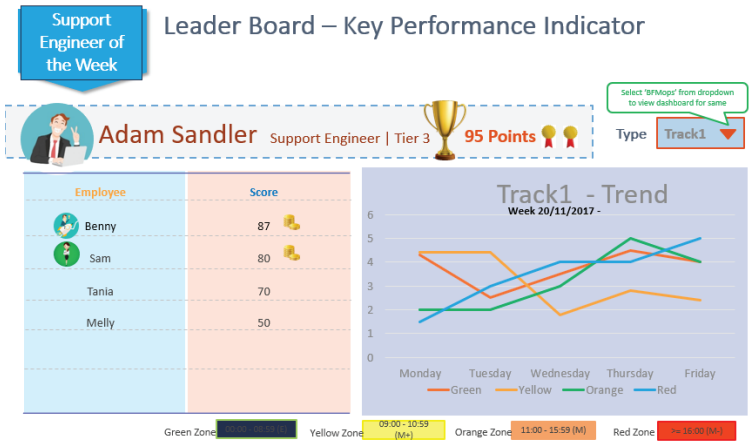




Recommended Risk to be Mitigated	Risk Description
Customer	Delay in Providing Requirements Clarification
Environment	Delay in Setting up of UAT environment
Requirements	Changes in Reporting requirements leading to Rewo
Team	Delay in onboarding the team
Solution	New and untried technology leading to solution risk

- Prescriptive Analytics helps to analyze the effort required to make a zero defect delivery , for a particular skill index and the defect metrics
- It also prescribes the number of test cases required to test the functionality and prescribes the productivity for various combinations
- Descriptive analytics helps the manager to compare where the project stands in terms of estimation parameters and other KPIs

To engage employees and foster productivity, use gamification with leaderboards of key performance indicators.



Conclusion:

It is just not enough to simply measure. It is essential to measure and automate sizing and other KPIs in a human-centric way while also enhancing employee participation through gamification. This will ensure that people are focused less on how to measure and the mechanics behind it and more focused on how to implement changes that improve user effectiveness, reduce manual errors and boost customer value.

References:

Queiuing Model using Little's Law: en.wikipedia.org/wiki/Little%27s_law

Syspoints: cit.fer.hr/index.php/CIT/article/view/1530/1234

Namcook Analytics: namcook.com

About the Author:

Sujatha Sivaraman *has more than 22 years of cross-functional experience in development, project management and quality assurance functions in the IT field, and eight years in the electronics and instrumentation field. She is currently holding the position of Director at Virtusa India. She holds certifications, such as CFPS, PMP, ISO 20k lead auditor, ITIL, Lean, Six Sigma black belt, design thinking, machine learning, python and Certified Scrum Master (CSM). She has trained more than 700 project leads and managers on Function Point counting and other estimation methodologies. She is also involved in the process improvement and transformation initiatives that catapult projects into improvement and value trajectories. She has published estimation articles and presented papers in International Software Estimation Colloquium (ISEC), Software Process Improvement Network (SPIN) and International Software and Measurement Analysis (ISMA) forums.*

Measurement Automation Using M.A.R.IN.E

By Kevin McKeel

Advances in technology have allowed time-consuming, manual and repetitive tasks to be automated through artificial intelligence. Law firm assocaites are utilizing E-Discovery software using syntactic analysis and keyword recognition. The combination of machine algorithms and Big Data has automated financial analytics that were once the domain of financial advisors and equity analysts. Thus, it stands to reason that software measurement can also be automated.

This article discusses a tool to improve and automate the requirements analysis and software sizing process. A cutting-edge application developed by Logapps, Machine Assisted Requirements Inspection and Evaluation (M.A.R.IN.E.), automates the review of software requirements for quality and consistency, and develops high-level software size estimates through Function Point automation. The good news for the *MetricViews* community is that software measurement professionals will not be replaced; some mundane tasks can be automated through artificial intelligence. Such innovations will help analysts understand requirements, identify duplication in both language and meaning and dramatically reduce the time and effort necessary to accurately analyze projects.

The Key to Successful Software Planning

The first step in most software projects is for business analysts to elicit requirements from stakeholders and document within a requirements management tool. Requirements can be requirement statements, user stories, use cases and other formats. Well-written requirements specify system behavior and determine what should be implemented. Poorly-written requirements often result in lower quality, ineffective design, additional rework and unnecessary test runs. Once functional requirements have been documented, they can be sized through Function Point analysis, though the accuracy of the functional size measurement is largely dependent on the quality of the requirements. Although M.A.R.IN.E is not a requirement management system, it can identify duplicate and similar requirements, non-functional requirements and improve requirement quality.

Poor requirements are a problem that has plagued the software industry since its earliest years. Similarly, software estimates in the early stages of a project are frequently inaccurate due to the cone of uncertainty, as changes in requirements and technology can have a huge impact on software development costs. Part of the reason is the difficulty finding resources with the talents and background needed to parse through hundreds and sometimes thousands of requirements in an effort to define system design needs, determine the expected software size and accurately estimate the software development cost. This detail-oriented and demanding process requires analysts who are often functional experts, but not necessarily software developers, to have a tremendous amount of tolerance for repetitive requirements parsing—and despite even the most meticulous analysis, mistakes are still inevitable.

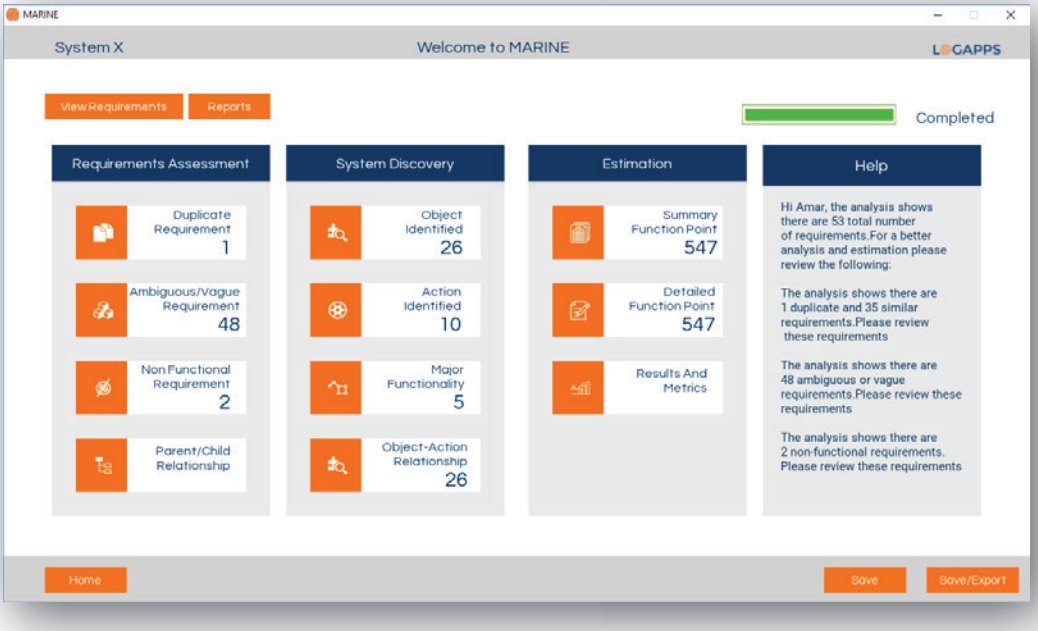
"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements...No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later".
Fred Brooks, American computer architect, computer scientist and software engineer.

From Good to Great: Machine Assisted Analysis and Automation

The capability to automate functional sizing has evolved due to changes in technology, such as Natural Language Processing (NLP) and Artificial Intelligence. In 2013, Object Management Group (OMG) adopted the Automated Function Point (AFP) specification using the IFPUG counting guidelines. This adoption was met very positively from the functional sizing community. Capers Jones, noted software industry subject matter expert, stated, "The arrival of automated high-speed Function Point counting...will elevate the importance of Function Point analysis from being a tool for mid-range applications to becoming a powerful tool for executive analysis of the largest and costliest software applications. Both software productivity and software quality data based on function points will expand rapidly, as will reliable software benchmarks" [1]. To date, automated functional sizing has been performed on developed code through static code analysis. The next breakthrough is automating Functional Size Measurement (FSM) through evaluation of project requirement statements.

Automation with the M.A.R.IN.E Tool

The M.A.R.IN.E desktop tool automates both requirements analysis and Function Point estimation processes with NLP and a robust rules engine. M.A.R.IN.E provides the requirements analyst with immediate feedback on the clarity of requirement statements and also removes duplicate requirements. The tool assists the cost estimator to produce software size, cost and schedule estimates from a given set of requirements. M.A.R.IN.E also allows the project manager to summarize system capabilities and verify that business needs align with the project requirements as well as provide Rough Order of Magnitude (ROM) costs by requirement to support trade-off decisions.



continued on page 14

(Measurement Automation Using M.A.R.IN.E., continued from page 13)

M.A.R.IN.E is built around a core NLP capability that processes each requirement. It separates the statements that make up each individual requirement into parts of speech that are important to the analyst: action word, action phrase, object, prepositional phrase and word pairings. Figure 1 below shows the main dashboard of the M.A.R.IN.E application after the requirements set has been uploaded..

Automating Software Measurement

FSM is based on IFPUG rules and counting practices, now an ISO standard (ISO/IEC 20926:2009). FSM has evolved since Alan Albrecht defined Function Points in Measuring Application Development Productivity in 1979 [3], but it remains a labor-intensive activity.

Trained, and frequently certified, practitioners go through artifacts such as user stories, use cases, logical data models, user guides and design specifications to develop robust software size estimates, and then iterate through discussions with the development organization's engineers or project managers. FSM practitioners apply and document counting rules at the requirement level. There are different methods of FSM, some of which require less rigor, such as fast Function Points. The challenge with manual FSM is the time involved in extracting, reviewing and evaluating each requirement. Many project managers have at best an elementary understanding of FSM, and there is a limited supply of trained Certified Function Point Specialists (CFPS). The opportunity with FSM automation is to expand the capability to create rough order of magnitude FSM and reach a broader audience.

M.A.R.IN.E automation involves FSM at the proposal or requirements phase. It should be noted that within M.A.R.IN.E, the analyst will need to refine the FSM, but the tasks of identifying transactions and data objects are simplified. Some of the intricacies of FSM within the IFPUG context, such as identifying File Types Referenced (FTRs), Data Element Types (DETs) and Record Element Types (RETs), are currently beyond the scope of M.A.R.IN.E. In many cases, not enough information is available early in the lifecycle to identify DETs in the requirements phase. However, fast Function Point counting at the proposal stage can be automated, in which case the analyst identifies data and transactional functions, and then makes assumptions regarding complexity. A key aspect of automation is exporting requirements into a manageable format, which can be a cumbersome process. The core of the automation involves cycling through functional requirements and identifying transactions and their associated objects (data) through keyword analysis. There is still a human factor, as the user can review M.A.R.IN.E's initial size evaluation and make adjustments at the requirements level, or export to a CSV or Microsoft Excel format for further evaluation. M.A.R.IN.E is also designed as a learning tool that will aid analysts with tips and automated suggestions that can be used as a training resource. Reports can be exported from M.A.R.IN.E in three formats: HTML, PDF and MS Word. Figure 2 displays a sample metrics summary report.

Alignment with IFPUG, Community Feedback and the High-Level Roadmap

M.A.R.IN.E has been designed based on IFPUG CPM 4.3 rules but is not IFPUG compliant in the sense that it follows the fast Function Point counting process at the requirements phase, and thus does not identify FTRs, DETs and RETs. While many within the IFPUG community may view this as a limitation, it may also be viewed as the evolving role of the analyst. It is reasonable to expect M.A.R.IN.E's automation capability to expand into ingesting images (such as entity relationship models) and design files (such as wireframes) and automate in a manner that is closely aligned with IFPUG rules.

Measure	Value
Requirements	53
Function Points	547
SLOC	28,991
Efforts(Person-Month)	109.4
Duration(Months)	9.1
Average Team Size	12.0

Measure	Value
Function Points/Requirement	10.3
Function Points/Month	60
SLOC/Month	265
Function Points/Person-Month	5

A 2017 study commissioned by Logapps identified variances between 2-60 percent, with an average variance of 20 percent between automated and manual size estimates. The variance can be reduced through analysis of requirements and review of redundant data functions and duplicate requirements. As with Function Point analysis in general, M.A.R.IN.E is well suited for transactional systems.

Logapps has used a crowd-sourcing model to identify desired future capabilities. The feedback has been positive and many users see great value in the efficiency of automated FSM. Many reviewers have requested integration with requirements and estimation tools, the ability to ingest images and access a database of project requirements and associated size metrics.

Conclusion: The Next Generation Role of Software Measurement

The manners in which software is developed, driven by Agile and Dev Ops, is rapidly evolving. Thus, it stands to reason that software analysis will also change. In the not-so-distant future, functional size measurement will rely more on automation. Delivered systems will be sized from static code analysis tools and early design sizing will rely on tools like M.A.R.IN.E. The shift in the automation of functional software measurement and requirements analysis will present opportunities for labor efficiency and change how software applications are analyzed. The role of CFPS and other functional sizing subject matter experts will likely focus on adjusting the dials of the automated models and then identifying DETs, RETs and FTRs for more precise functional sizing. M.A.R.IN.E brings this technology to the analyst's desktop.

While it is a human tendency to resist change, those who see value in evolving technology stand to benefit the most. In the words of the famed statistician who revolutionized quality control methods, W. Edwards Deming, "It is not necessary to change. Survival is not mandatory" [4]. ■

References

[1] "OMG Adopts Automated Function Point Specification," Object Management Group, 17-Jan-2013. [Online]. Available: omg.org/news/releases/pr2013/01-17-13.htm. [Accessed: 01-Jun-2018].

[2] F. Brooks, "No Silver Bullet–Essence and Accident in Software Engineering," p. 16, 1986.

[3] A. Albrecht, "Measuring Application Development Productivity," 1979.

[4] F. Voehl, Deming, "The Way We Knew Him." CRC Press, p. 125, 1995.

About the Author: Kevin McKeel *is the co-founder of Logapps LLC in the Washington, D.C. metro area. He has 21 years of experience in IT strategy, systems analysis and software cost estimation. He is a Certified Cost Estimator/Analyst, Function Point Specialist and subject-matter expert in software requirements analysis, Function Point sizing and parametric cost modeling. McKeel works with the Logapps team to deliver LCCEs, ICEs, Analyses of Alternatives (AoA) and business case analyses for client organizations across the federal government and commercial industry, including the IRS, FAA, NOAA and U.S. Navy. As an ICEAA and IFPUG member, he frequently speaks at industry conferences around the country.*

Automation Tools for Real-Time Embedded Software Functional Size Measurement

By Hassan Soubra

Applying Functional Size Measurement (FSM) procedures manually is tedious and time-consuming for organizations with a large number of projects to measure within a very short timeframe, both for project estimation purposes and for productivity studies [1] [2].

In addition, the manual application of FSM to a very large set of requirements requires specialized expertise when these requirements are complex. Automating requirements measurement is particularly challenging when the measurement inputs are expressed in an unconstrained format (textual descriptions of requirements, for example). However, when the requirements are described, or modeled using a specific notation, automation becomes feasible, and even more so when the requirements notation is computer-based.

FSM automation is an attractive approach for organizations using commercial modeling tools to document their software functional requirements. In such contexts, the models for their requirements are embedded in the tools used by engineers and can readily serve as inputs for automating the measurement of the functional size of the software. Of course, the accuracy of the automation tools implementing FSM should be verified objectively prior to their use [3].

This paper presents an automation prototype tool initially developed at Renault SA, in collaboration with the software engineering research teams at the École de Technologie Supérieure (University of Québec, Canada) and the University of Versailles at St-Quentin en Yvelines, France (UVSQ) [1]. In addition, two COSMIC-based automation prototype tools were developed at ESTACA: they can be downloaded for free from the ESTACA website [4]. While the first prototype tool was developed to measure the size of aerospace real-time embedded software modeled using the Safety-Critical Application Development Environment (SCADE) commercial tool [5], the second was developed to correctly measure the functional size of ECU application software designed following the Automotive Open System Architecture (AUTOSAR) standard [6]. AUTOSAR [7] is the new generation of ECU software design architecture, methodology and metamodel. It has become an important part of the production design criteria for many vehicle manufacturers, especially in the automotive electronics industry. The procedure automated by the prototype is based on the measurement guideline presented [8] and has a set of mapping rules to be applied to the system modeled in order to obtain its functional size.

At Renault SA

The automation prototype in this case study is a COSMIC automation prototype developed in collaboration with the software engineering research teams at the École de Technologie Supérieure (University of Québec, Canada) and the UVSQ [1] [3]. Here, Renault SA uses commercial modeling tools (such as Statemate and Simulink) to prepare the functional user requirements (FURs) allocated to software. The verification protocol was applied with a set of 77 distinct specification models (designed in Simulink) at Renault SA, where various sizes of specifications were chosen among a number of software functions that represent different engine control modules (ECMs)in the department where the automation prototype-tool was initially developed.

The use of the verification protocol [3] demonstrates that only 9 percent of the input specifications presented a variation between the manual measurement and the automated one (i.e. for seven of the 77 specifications), and that those differences varied from 1.7 percent to 12.67 percent for variations caused by limitations in the prototype.

Overall, the difference in the total size of the 76 correct requirement models obtained both manually (i.e. 1,729 CFP) and using the automation prototype (i.e. 1,739 CFP) is less than 1 percent (see Table I). Therefore, the accuracy of the automation prototype after testing is greater than 99 percent. The application of the proposed verification protocol made it possible to detect one incomplete requirement specification. In addition, it helped identify the limitations of the prototype-tool, stemming from the limitations inherent in the libraries used in the prototype-tool. Those limitations identified in the prototype-tool discussed in this paper were then corrected in the industrial automation tool that was developed based on

continued on page 18

(Automation Tools, continued from page 17)

the verified prototype: this automation tool is more robust, has a greater level of accuracy and is now used in a number of departments at Renault SA [2].

Table I: Difference between the total size obtained manually and using the prototype-tool (Expressed in COSMIC Function Points, CFP)

Total Number of Models	Total Size obtained manually (CFP)	Total Size obtained using the prototype-tool (CFP)	Difference (%)	Accuracy
76 fault-free models	1,729	1,739	Less than 1%	>99%
All 77 models	1,758	1,791	1.8%	>98%

At ESTACA

Two COSMIC-based automation prototype tools were developed at ESTACA [4]. While the first prototype tool was developed to measure the size of aircraft real-time embedded software modeled using the SCADE commercial tool [5], the second one was developed to correctly measure the functional size of ECU application software designed following the AUTOSAR standard [6].

Tool 1: SCADE

The FSM automation prototype tool presented in this section is based on SCADE and COSMIC. SCADE [9] is an industrial development tool used in many major companies developing safety-critical embedded systems.

The automation prototype tool identifies all the packages in a SCADE project. Next, for each package identified, it opens the XSCADE file of the same name. SCADE nodes and functions could be located in different files.

Figure 1 shows our prototype-tool's interface: it has a main window with three tabs. In the FSM tab, users select the file to-be-measured and the dumping files (PDF and XML) if the "save when measuring" checkbox is checked. The "Compare" is still under development and the "About Us" tab gives information about the tool. A "Measure" button appears so that the user triggers the measurement process. When clicked, a pop up window (see figure 2) shows the measurement results.

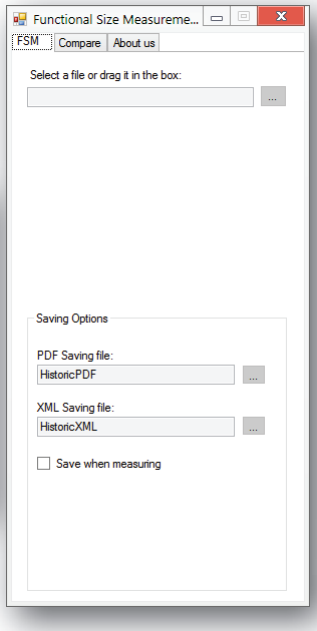


Figure 1: Interface of the prototype

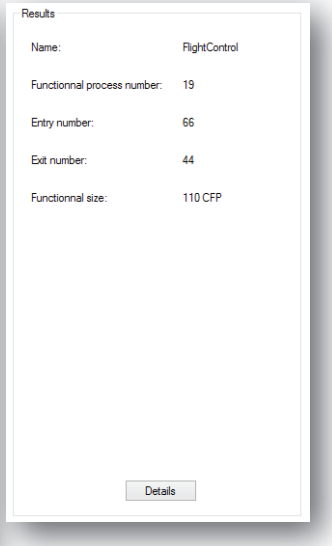


Figure 2: Measurement Results window

When clicked, the "Details" button shows the sizes in CFP of each Function Point identified (see figure 3) and opens a pdf file with the details of the data movements identified in the Function Point identified (see figure 4).

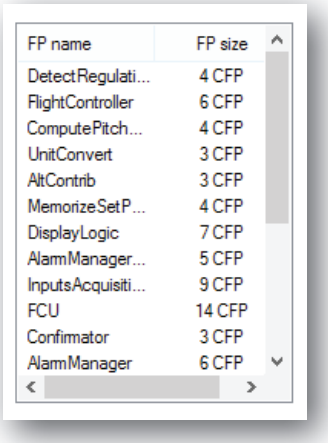
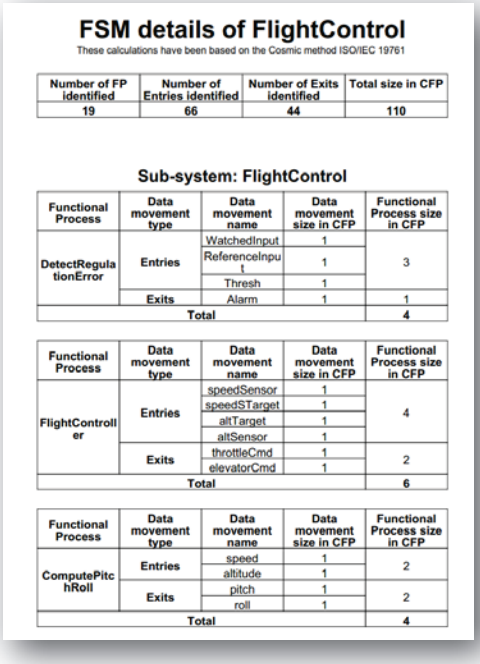


Figure 3: Functional size per FP identified

Figure 4 shows, as an example, a fragment of the pdf document produced when clicking the "Details" button. It gives the total Function Point number identified, in addition to the total number of entries and exits of the system. It also gives the size in CFP of the system. Next, it shows the measurement details per Function Point, entries and exits of data movements identified with the names of their corresponding flows per package.



FSM details of FlightControl

These calculations have been based on the Cosmic method ISO/IEC 19761

Number of FP identified	Number of Entries identified	Number of Exits identified	Total size in CFP
19	66	44	110

Sub-system: FlightControl

Functional Process	Data movement type	Data movement name	Data movement size in CFP	Functional Process size in CFP
DetectRegulationError	Entries	WatchedInput	1	3
		ReferenceInput	1	
		Thresh	1	
	Exits	Alarm	1	1
Total				4

Functional Process	Data movement type	Data movement name	Data movement size in CFP	Functional Process size in CFP
FlightController	Entries	speedSensor	1	4
		speedSTarget	1	
		altTarget	1	
		altSensor	1	
	Exits	throttleCmd	1	2
		elevatorCmd	1	
Total				6

Functional Process	Data movement type	Data movement name	Data movement size in CFP	Functional Process size in CFP
ComputePitchRoll	Entries	speed	1	2
		altitude	1	
	Exits	pitch	1	2
		roll	1	
Total				4

Figure 4: Fragment of the pdf details file produced

The automation prototype tool was applied on a set of six distinct systems designed using SCADE: pilot, flight control, digital stop watch, cruise control, ABC_N and roll control.

For verification purposes (see Table II), the systems have been measured manually using the measurement procedure implemented by the prototype tool. Thus, in addition to the functional size obtained, all the functional processes and data groups' movements are identified by the manual measurement. We have also kept track of the time needed to manually apply the measurement procedure and to document the measurement results obtained using Excel sheets. The time needed varies from 22 minutes to 151 minutes, according to the measured system's size. In contrast, the prototype tool produced the results almost instantly, including the detailed documentation, for all the systems measured.

Tool 2: AUTOSAR

The FSM automation prototype tool presented in this section is based on AUTOSAR, SYMTA/S and COSMIC. The SYMTA/S tool models and analyzes real-time embedded systems in order to measure system performance (e.g. Worst Case Execution Time (WCET), CPU load, end to end latencies, etc.) while taking into account scheduling constraints and differing execution scenarios. SYMTA/S is suitable for several system architectures including AUTOSAR.

The automation prototype tool is developed in JAVA. The inputs are SYMTA/S simulation files that include both AUTOSAR models and ECU processor load information. This tool makes it possible to measure automatically software functional size, in CFP, designed following AUTOSAR methodology and meta-model. This tool is also capable of measuring, simultaneously, a group of input specifications. The prototype tool's primary functionalities are:

A. Automatically measures COSMIC functional sizes of the input models.

B. Determines ECU processor load for each input model using processor load information in SYMTA/S simulation files.

C. Yields ECU load vs COSMIC functional size graphs using input files.

D. Estimates processor load for additional models using previously generated graphs by using ECU load vs COSMIC functional size graphs from step A to estimate ECU processor load for new input models.

The tool outputs the functional sizes, ECU load and curves plotting the relationship between functional size and ECU load. Figure 5 illustrates an example of output using the automation tool.

continued on page 20

Table II: Functional size of the systems measured

System	Number of FP identified	Number of Entries identified	Number of Exits identified	Total size in CFP	Total Manual Measurement time in minutes
Pilot	24	45	32	77	105
Flight Control	19	66	44	110	151
Digital Stop Watch	6	15	11	26	39
Cruise Control	11	45	20	65	92
ABC_N	3	7	6	13	22
Roll Control	5	12	9	21	31

(Automation Tools, continued from page 19)

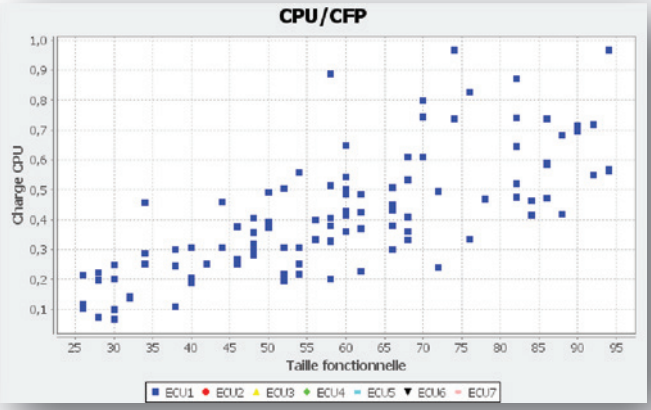


Figure 5: Functional Size and processor load

Conclusion

FSM may be a time-consuming task when performed manually and also because staff resources may not be available for measurement purposes within the required time frame. Measurement with automated tools eliminates possible variances caused by different measurers, which may lead to different measurement results for the same set of requirements. That is why a tool that automates the measurement procedure while ensuring the accuracy of the measurement results is useful and can benefit organizations in terms of reducing the workload of measurement specialists, as well as eliminating measurement delays.

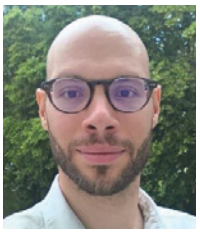
This paper presented an automation prototype tool that was initially developed at Renault SA in collaboration with the software engineering research teams at the École de Technologie Supérieure (University of Québec, Canada) and the UVSQ.

This paper also presented two COSMIC-based automation prototype tools developed at ESTACA. The first prototype tool was developed to measure the size of aircraft real-time embedded software modeled using the SCADE commercial tool. The second one was developed to correctly measure the functional size of ECU application software designed following the AUTOSAR standard.

References:

1. *Hassan Soubra, Alain Abran, Sophie Stern and Amar Ramdane -Cherif, "Design of a Functional Size Measurement Procedure for Real-Time Embedded Software Requirements Expressed using the Simulink Model," IWSM-MENSURA, Nara, Japan, IEEE CS Press, 2011, pp. 76-85.*
2. *Renault SA, "Design of a Functional Size Measurement Tool for Real-Time Embedded Software Requirements Expressed Using a Simulink Model," MathWorks Automotive Conference. Stuttgart, 2012*
3. *Hassan Soubra, Alain Abran, Amar Ramdane Cherif "Verifying the Accuracy of Automation Tools for the Measurement of Software with COSMIC–ISO 19761 Including an AUTOSAR-Based Example and a Case Study" Joint 24rd International Workshop on Software Measurement and 9th MENSURA Conference, Rotterdam (The Netherlands), Oct. 6- 8 2014, IEEE CS Press, pp. 23-31.*
4. *estaca.fr/hassan-soubra.html*
5. *H. Soubra, L. Jacot, S. Lemaire, "Manual and Automated Functional Size Measurement of an Aerospace Real-Time Embedded System: A Case Study Based on SCADE and on COSMIC ISO 19761," International Journal of Engineering Research and Science Technology, Vol. 4 n°2, pp. 79-100, 2015.*
6. *H. Soubra, A. Abran and M. Sehit, "Functional Size Measurement for Processor Load Estimation in AUTOSAR," Joint 25th International Workshop on Software Measurement and 10th MENSURA Conference, Kraków, Poland," Oct. 5-7, 2015, Lecture Notes in Business Information Processing-Springer, vol. 230, pp. 114-129, 2015.*
7. *autosar.org*
8. *H. Soubra, and K. Chaaban, "Functional Size Measurement of Electronic Control Units Software Designed Following the AUTOSAR Standard: A Measurement Guideline Based on the COSMIC ISO 19761 Standard" Joint 22nd International Workshop on Software Measurement and 7th MENSURA Conference, IEEE CS Press, 2012.*
9. *esterel-technologies.com/products/scade-suite/*

Note: IFPUG Function Point can be used for real-time embedded software. Refer to the white paper on the subject at ifpug.org/fssc-publishes-white-paper-on-counting-embedded-systems.).



About the Author:
Hassan Soubra *is currently an associate professor at ESTACA–Engineering School in France. He is a member of the Embedded Systems research team. His research is focused on metrics and measurements related to industry problems in the autonomous vehicle and smart city domains. He is a nominated member of the Measurement Practices Committee at COSMIC. His previous work experience includes working at Renault, LAAS CNRS and Technicolor.*

THINK YOU'RE READY FOR A MEASUREMENT TOOL? TOP 10 QUESTIONS TO ASK BEFORE GOING SHOPPING

by Carol Dekkers

Software tools are a mainstay in our lives today. Can you imagine an office without an MS Office® suite of Word and Excel, MS Project® and SharePoint®? The time they save us by streamlining document production, spreadsheets, project plans and other functions is something we take for granted without even thinking.

When it comes to IT measurement, emerging suites of commercial software tools promise us automation to help with what appear to be routine measurement functions:

a) Measurement collection/storage (repository software)

b) Project and program cost estimation

c) Function Point counting

d) Benchmarking

e) Overall program and project measurement

Shopping for tools can be fun… but remember to heed the warning: "A fool with a tool, is still a fool" (quote variously attributed to Grady Booch, R. Buckminster Fuller or Ron Weinstein) and do not start shopping for tools before you've answered these 10 top questions. A software tool should be seen as an investment to help you to streamline your work, not as a frivolous spend with the goal of having the tool do your work for you.

Top 10 Questions to Ask Before You Go Shopping for IT Measurement Tools

1. Do you understand the data and subject matter you want to support with a tool?

This may seem like an obvious question, but it's more fundamental than you might think. Here's a few quick examples to illustrate this:

a. If you want a Function Point repository tool to help maintain your Function Point portfolio of counts (i.e., apply project counts to update application baselines) and you do not have existing baseline counts, the tool will only be able to update counts for which you have existing data. One client assumed that a tool they purchased would be able to compare team productivities across projects (a simple numerical calculation based on Function Point and work effort) and found out that their collected data were at different levels (effort was summarized from department level timesheets, Function Point counts were at the project

continued on page 22

(Think You're Ready, continued from page 21)

level) and could not meet their needs without modification.

b. Another client assumed that an estimating tool would use their collected historical data as the basis for estimating future projects (without a facility to enter or use such data).

c. Knowing your data and what good results are is a pre-requisite for purchasing any support tool. Without this knowledge, a tool can produce flawed results (garbage in means garbage out) and you won't realize it. Consider the simple example of someone purchasing a multiplication calculator who doesn't know their basic "times tables." If the user enters the equation of five multiplied by four and the calculator returns a value of 15, it illustrates the fool with a tool (someone who has a calculator but does not know how to calculate) is still a fool.

2. What are the Five Ws and H for a Tool?

a. **Why** do you need a tool? (measurable objectives for the tool)

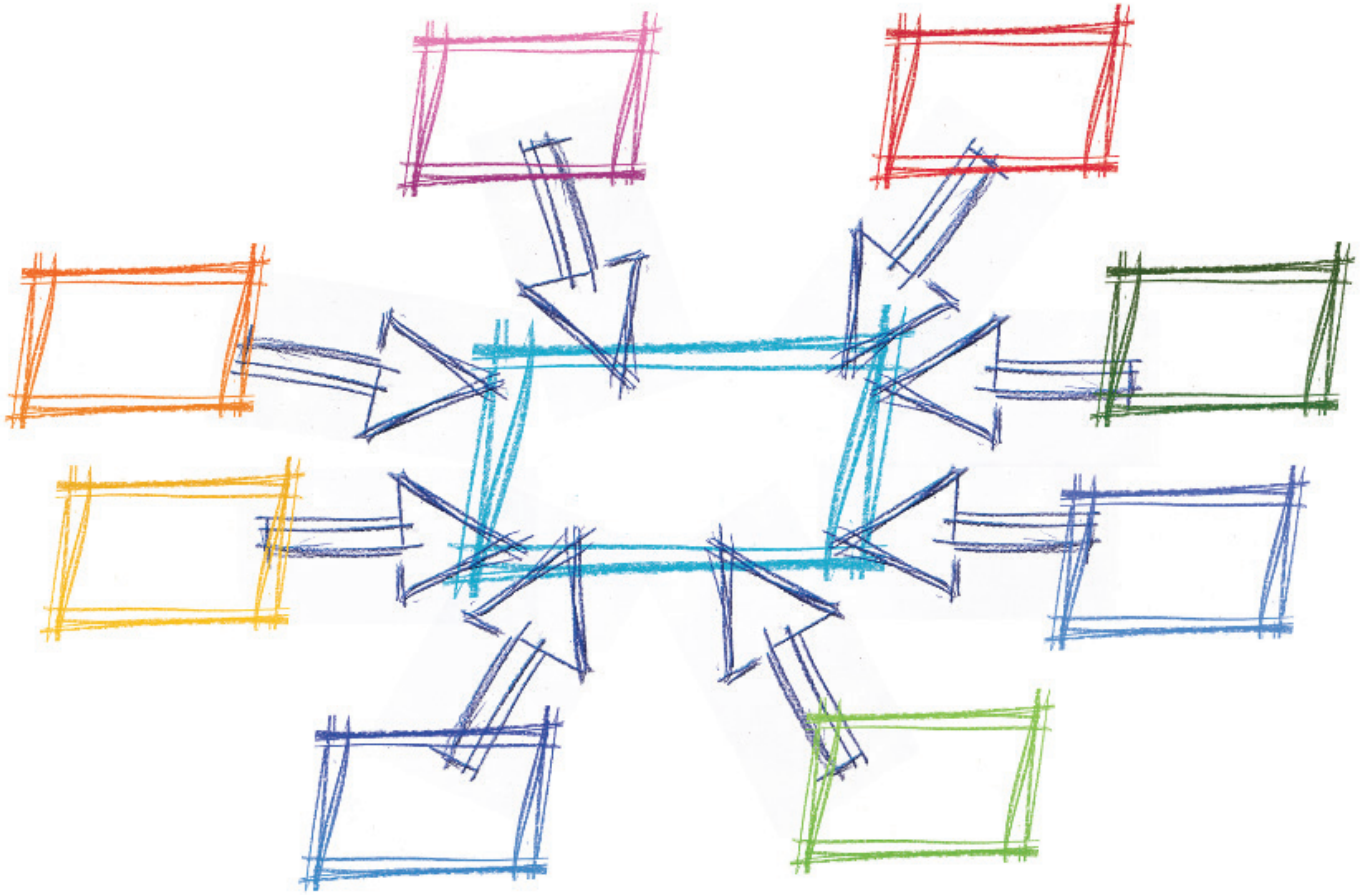
b. **What** are the (functional and non-functional) requirements for the tool? For example,, if you cannot do even a rough Function Point count and SNAP count of your requirements, you are not yet ready for a tool. Document your tool requirements, including data collection/importing, data entry, reporting, what-if-analysis, query, etc. functions and the non-functional requirements.

c. **What** does the tool need to measure? For example, do you need Function Points from existing code or store (e.g., project work effort in hours/person days/person months, type of project, project/application size, defects by defined category, duration, tasks, etc)? What units of measure do you need (e.g., RACI objects, Function Point, SNAP points, hours/days/months, etc? *As a pre-requisite to this question, it is suggested that you have established a solid measurement program based on the Goal-Question-Metric approach of Victor Basilli and have at least started collecting/analyzing/reporting/using the data.*

d. **Who** needs access to the tool? Do you plan to have different users entering data versus analyzing or reporting data?

e. **Where** do you plan to use the tool and where will it (and the data) reside? Will it be on a central server, multiple/single desktop or remote sites, simultaneous access, etc)?

f. **When** and **How** will the tool be used/needed? Will it be used at the beginning/end of a project, during



a project, whenever an estimate is needed, etc.?

3. What is the minimum viable product (minimum requirements) for the tool today?

a. What additional requirements are envisioned for the future and how will a tool support your existing (and future) requirements? Could a spreadsheet work today? Do you need to keep track of both project and application Function Point details or simply totals?

b. How will the tool fit/integrate (or not) with existing processes?

4. Are there tools on the market that are a good functional fit? Or should you invest in building your own tool?

5. What is your budget for a tool? Consider license(s) purchase, cost of installation at the site(s), vendor/self-configuration, additional hardware/software to be purchased, vendor support costs, customization (if needed), user training, annual maintenance, upgrades and expansion (multiple/single sites), etc.

6. What is included in the tool price? What is included as part of the base product/suite? Is there a historical data base, the ability to add your own data or a software trial option available? What are optional extras (e.g., automated interfaces, compatible importing/exporting interfaces, reporting/graphing capabilities, expansion/growth potential, training options (some vendors require training at their facilities as pre-requisites to a purchase.)

7. What is the track record and stability of the vendor? Lifetime upgrades may not mean anything if the vendor goes out of business. Are customer references available?

8. What is the plan for training? Consider the costs, schedule and roll-out training for the package, as well as subject matter training, if required.

9. What is the plan to audit/validate/confirm the results of data provided by the tool? If your tool will be used to count Function Point on installed applications, how will you verify the results or calibrate the tool? If the tool is an estimating tool, what are the plans to calibrate the data entry for the next project based on known results of completed projects? Tools rely on data and remember: garbage in means garbage out! To be successful, measurement must be an empirical process where human inspection and common sense ensure that reported data are valid.

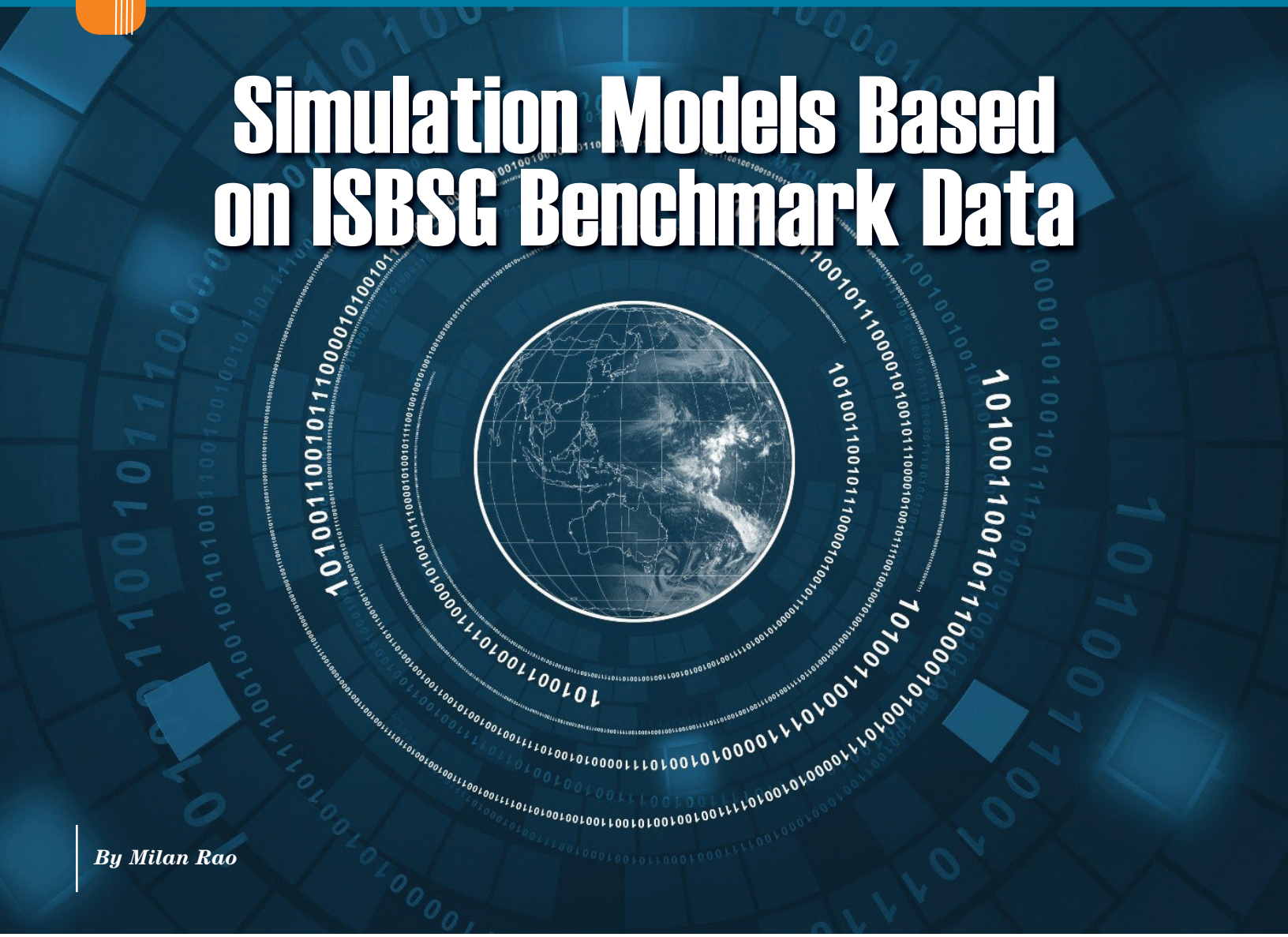
10. Is there a realistic return on investment (ROI) for the tool? Has there been a cost/benefit analysis done on the tool or is it going to be an outright spend? Too often software tools are a great idea that become shelf ware down the road.

This probably seems like more prep work than you might have envisioned before buying a software tool, but this checklist of 10 questions will help you focus and ensure that the purchase will be an investment that pays off for your company.

Don't be a fool with a tool. Be a genius whose measurement-related work was made easier by a tool! ■

About the Author: Carol Dekkers, *PMP, CFPS (Fellow), P.Eng., CSM, AEC, is founder of Quality Plus Technologies Inc. and consults with companies that want to succeed with software measurement and functional size measurement. She has published textbooks and articles and is a frequent speaker worldwide on technical subjects (estimating, measurement, ISO standards, project management and quality), as well as leadership and soft skills. Currently, she serves as the director of communications and marketing for IFPUG and is part of the U.S. delegation to ISO. Contact her at dekkers@qualityplustech.com.*

Simulation Models Based on ISBSG Benchmark Data



By Milan Rao

Projects are the building blocks of an organization. A typical project starts with the "Discovery" or "Study" phase in which the project team tries to understand customer business and the pain-points that technology can solve. Once there is clarity on a solution, various project attributes are estimated. Based on the estimates, plans are drawn to determine the schedule, resource-loading patterns, release strategy, etc. During the execution phase, plans are followed and actions taken when deviations occur. In the end, the project is either a success or a failure. The success or failure depends on the effectiveness of the estimates. If the estimates are flawed, the plans are flawed and eventually the execution is flawed.

What do we estimate?

Many things must be estimated accurately in order for a project to succeed. The major ones are listed below:

1. **Cost:** This includes estimated costs of people involved in the project, infrastructure, training, etc. In addition, the teams will have to estimate the variances possible at the end of the project.
2. **People:** This is the most important ingredient of any project. How many people will be required to complete the project? In addition to headcount, consider skill ratios, technical competencies, etc.
3. **Schedule:** How much time will be required to complete the project? Consider things like the best path and that dependencies are taken care of while creating the schedule.
4. **Defects:** Consider where the application might fail. Plan for adequate time and resources to handle any failures.

In reality, teams predict the risks that can occur in any of these elements and identify appropriate mitigation actions to reduce the probability of failure.

How to predict risks

Broadly, risks are predicted by:

- **Wisdom:** A team member's intellect and intuition will help to identify risks. Consider a heart surgeon who has been performing cardiac surgeries for 20 years. He/she would be able to diagnose potential cardiac problems in a patient just by reading a simple health report.
- **Experience:** While wisdom is based on a person's intuition, experience is based on past/similar experiences. Like wisdom, this is non-repeatable and an extremely person-centric approach.
- **Data:** Past information can help predict the future. Data collected over a period of time can help provide insight into what may go wrong in the system.

This article is about how International Software Benchmarking Standards Group (ISBSG) data is used in Mindtree to predict risks. We live in an age where data is used extensively in all forms of business. From a small shopkeeper who uses data to predict his customer's buying pattern to an oil exploration firm using data to predict potential oil wells, data is becoming the building blocks for all organizations.

This leads to the question: How can data help a project manager predict the risks better? More importantly, is there a data store that can be used to predict risks? Our answer to these questions is ISBSG.

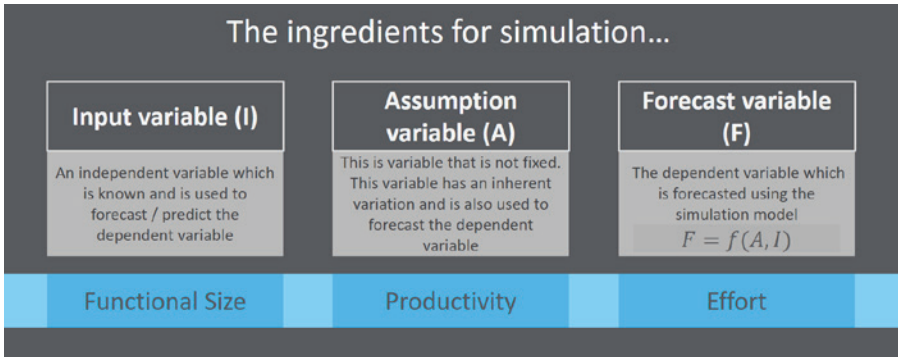
Why ISBSG?

ISBSG (isbsg.org) is a not-for-profit organization. It was founded in 1997 by a group of national software metrics associations whose aim was to promote the use of information technology (IT) industry data to improve software processes and products.
The ISBSG's mission is to help organizations improve the planning and management of IT software projects.

ISBSG's development and enhancement database has information and metrics data of around 7,500 projects collected over a period of 20 years. The demography of data spans various domains, technologies, platform, application types, etc.

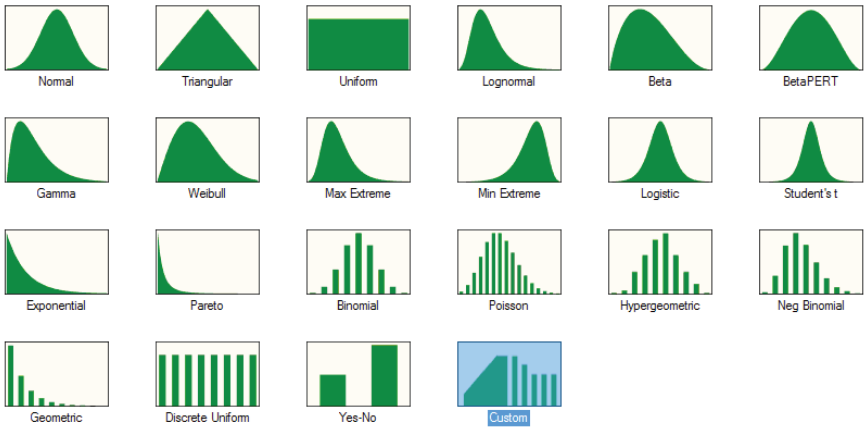
Traditionally ISBSG data has been used for benchmarking, setting targets and estimation. We attempted to build simulation models based on the data to predict the risks. Simulation is the process of emulating a particular behavior multiple times and is used to predict risks.

Ingredients for Simulation



A simulation model consists of three ingredients:

1. **Input Variables (I):** An independent variable is known and used to forecast/predict a dependent variable.
2. **Assumption Variables (A):** This variable is not fixed. This variable has an inherent variation and is used to forecast the dependent variable. The assumption variable is a distribution derived based on a past baseline. The distribution can either be continuous or discrete as shown in the below figure.



3. **Forecast Variable (F):** The dependent variable is forecasted using the simulation model. It is a factor of the input variable (I) and the assumption variable (A). The relationship can be a regression model or an arithmetic equation.

F=f(A,I)

continued on page 26

(Simulation Models, continued from page 25)

The Models

Four models are built using the ISBSG benchmark data. The input variables, assumption variables and the forecast variables are shown below.

Effort Forecast Model		
Industry Sector	Productivity (Effort per FP)	Total Effort (Phrs)
Development Platform		
Language Type		
Development Type		
Team Size		
Data Quality		
Count Approach		
Size		

Schedule Forecast Model		
Industry Sector	Delivery Rate (FP per Calendar Months)	Schedule (Calendar Months)
Development Platform		
Language Type		
Development Type		
Team Size		
Data Quality		
Count Approach		
Size		

Cost Forecast Model		
Industry Sector	Cost Rate (Cost per FP)	Cost (Currency)
Development Platform		
Language Type		
Development Type		
Team Size		
Data Quality		
Count Approach		
Cost Currency		
Size		

Defect Forecast Model		
Industry Sector	Defect Density (Defects per FP)	Total Defects
Development Platform		
Language Type		
Development Type		Phase-wise Defects
Team Size		
Data Quality		
Count Approach		
Size		

All the models are MS Excel-based utilities that filter, identify distribution, define model parameters and initiate the simulation. The user needs to select the appropriate values for the input variables and run the simulation to get the results as the distribution for the forecast variable. The distribution helps the team understand the risks involved in each parameter.

How Simulation Works

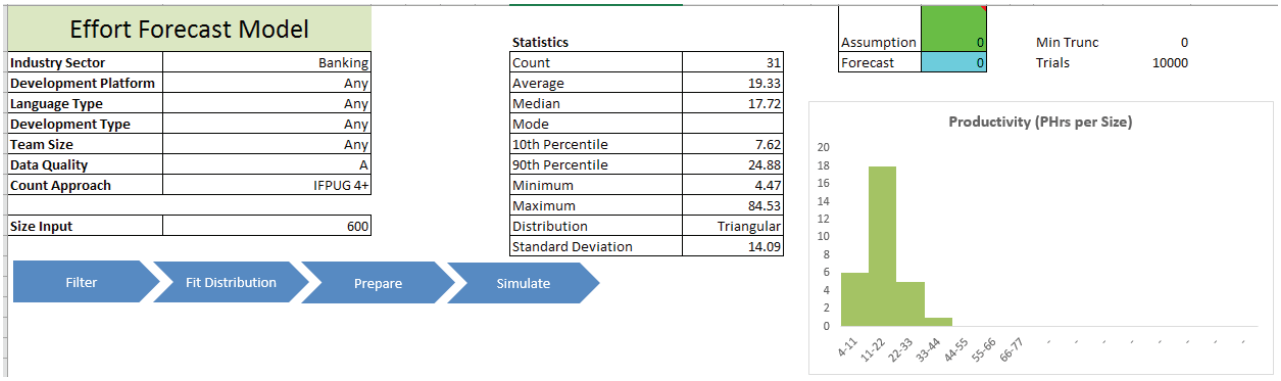
Oracle Crystal Ball, an MS Excel add-in, used to build and run simulation models. A major part of the model-building involves identification of the distribution of the assumption variable. Defining the assumption involves selecting the distribution and setting the distribution's height and spread parameters. For example, a normal distribution uses mean and standard deviation as parameters.

The next step is to define the forecast variable–a formula of the inputs and assumption variables. The user selects the appropriate values for the input variables, sets the number of trials and then runs the simulation. Based on the number of trials, the model randomly generates values for the assumption based on its distribution. The random number is equated into the forecast variable's equation. Once all the simulation trials are run, the distribution for the forecast variable is generated. This probability distribution of the forecast variable is used to assess the percentage probability or the percentage risk to the forecast variable.

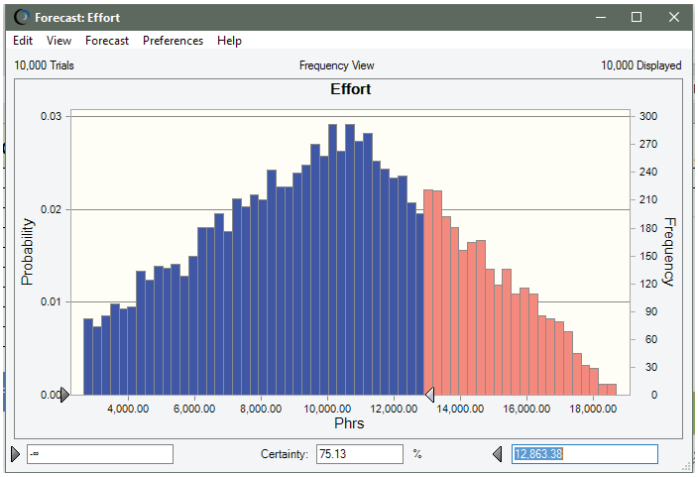
Example and Results

Below is a snapshot of effort forecast model. The "Industry Sector" selected is "Banking" and all the other input variable are set as "Any." In addition, the "Size in FP" is set to 600 Function Points. When the "Filter" button is clicked, the model runs through the 7,500 rows of ISBSG data and filters the rows of data based on the input values. Next, by clicking the "Fit Distribution" button, the model identifies the best-fit distribution for the filtered assumption data. In this case, "Triangular Distribution" has been identified. "Prepare" will configure the assumption variable as shown in the "Green" cell based on the distribution and its parameters. It also configures the forecast variable shown in "Blue." In this case, the forecast variable is "Effort," the input variable is "Size" and the assumption is the "Productivity."

Productivity=Effort/Size→Effort=Productivity*Size



Clicking on the "Simulate" button will run the simulation 10,000 times, the set number of trials. Once the simulation is complete, the probability distribution for "Effort" forecast variable is generated as shown below. The model shows that the confidence to complete the project worth 600 Function Points, with 12,000 person hours of effort, is estimated at 75 percent. Project teams can perform "what-ifs" by selecting the appropriate confidence level/risk cushion.



The model is used in multiple projects and the model forecasts and the actual performance numbers are close. Specifically, the effort, schedule and defect models have an accuracy of around 75 percent (the cost model has lower accuracy due to lack of enough data-points). Presales teams used the models to validate the estimates before committing to the customer.

Conclusion

With the ever-increasing complexities involved in delivering successful projects, it is important to provide project managers with tools and techniques to help identify and mitigate risks more effectively. Data-driven models like the ones listed in this article help project managers make better decisions early in a project's lifecycle.

The sustenance of these models lies in their relevance and accuracy. ISBSG is an internationally-acclaimed body for benchmarking, thus, the accuracy part is taken care of. At Mindtree, the data is constantly cleaned, and the models are frequently recalibrated to ensure that the forecasts are relevant.

About the Author:
Milan Rao *is a senior manager at Mindtree, where he handles metrics and customer satisfaction. He has more than 14 years of experience in the industry. His background spans the fields of software quality assurance, metrics, customer satisfaction and project management. He specializes in exploratory data analysis, process performance benchmarking and statistical modelling for software projects. Prior to joining Mindtree, he was a Scrum Master in a large product implementation project. Rao is a certified data scientist with specialization in R & Data Visualization and a Certified Scrum Master. He recently presented papers at the ISBSG IT Confidence Conference in Beijing and at the CMMI Capability Counts 2018 Conference in Reston, Virginia.*

Automation and Too Many Metrics Sharing the Word 'Points'

By Antonio Ferre Albero

A fascinating capability of information technology (IT) is freeing people from repetitive tasks that can be done by computers at a speed of millions of instructions per second. IT brings value by making life easier, helping organizations be more competitive and provide more value and having computers execute non-value add tasks in a cheaper and faster way. Can this automation replace persons regarding sizing applications and talking about Functional Size?

One of the secrets of the IFPUG Functional Size Method is that the size is determined based on the functionalities that the user receives and not on the physical software code, physical files or tables used: more lines of code cannot be synonymous with a higher product size. Another key point is that different CFPS experts will arrive at the same size, counting an application, enhancement or change request. Determining this size is not needed to know the programming language or to see technically how the application has been developed. IFPUG Function Points, defined in 1979 by Allan Albrecht, is a universal method to measure the product that the user receives; it has been improved along the way and is in continuing progress.

What sometimes creates confusion, and even misunderstandings, is that there are many metrics that share the word "Points" or even the words "Function Points" for naming completely different concepts. This can contribute

to a kind of concept contamination, because when someone sees a metric ending with "Points" or with "Function Points" they may think it is similar to the other one; for example, Function Points versus Story Points.

Automated Function Points (AFP) are Not IFPUG Function Points Automatically Counted by a Tool

I have heard many people defend manual counts and others arguing the automatic counting boundaries, but in any case both approaches might be delimited because they create different metrics. That they share the same name, "Function Points," does not help and, as mentioned, can create confusion. Why? Automated Function Points (AFP) are not IFPUG Function Points counted in an automatic way because the counting method is different and the results are different, too.

AFP is software code dependent and IFPUG Function Points are based on the functionalities that the user receives. In some cases, the percentage of deviation between both can be low, but in others can be very high.

AFP are supported by CISQ (Consortium for IT Software Quality), being as co-founders the Object Management Group (OMG) and the Software Engineering Institute (SEI). Its specifications are based on the main ideas of the IFPUG standard, trying to be as much as possible similar to the IFPUG method.

In spite of the fact that AFP try to work with regard to the IFPUG philosophy, to transform software lines of code and physical data models into IFPUG concepts such as elementary process, uniqueness, boundaries, temporary, internal logical file or external logical file (just as examples) is extremely complex. Trying to compare AFP with IFPUG Function Points can be tricky.

The objective of this article is not to praise the IFPUG method nor to be a critic of AFP. It is to remind that these are two different methods. Both can provide results more or less similar or completely different (a matter of luck), but AFP are not IFPUG Function Points counted in an automatic way.

The Best Metrics in Different Project Phases

IFPUG Function Points serve different purposes and at different times during a project. They can be used for estimation purposes (transforming size into effort, effort into cost or to directly convert size into cost based on a price per Function Point), for analyzing changes in requirements (or omitted ones) and for re-planning, making different snapshots to the size in different points in time, for sizing the product delivered and as a basis for productivity, quality, benchmarking, etc.

Determining the price of a project based on the IFPUG Functional Size (this is typical in countries such as Brazil and Italy) not only provides a benefit to pay based on the functionalities of the product, but puts more emphasis in the requirements activities, avoiding as much as possible missing or incorrect requirements that contribute negatively to the project's success (rework, project delays, less than desired quality due to lack of time or possibly an incomplete regression test) and in the fascinating "technical debt" concept.

Besides, size obtained from the code can only be used once the code is created, so it is not possible to use in the early stages of a project, such as the requirements, estimation or analysis phases. This behavior does not only apply to AFP but to any software size approach based on the software code created: the metrics only can be used once the software has been created.

Benchmarking, the Magic Word

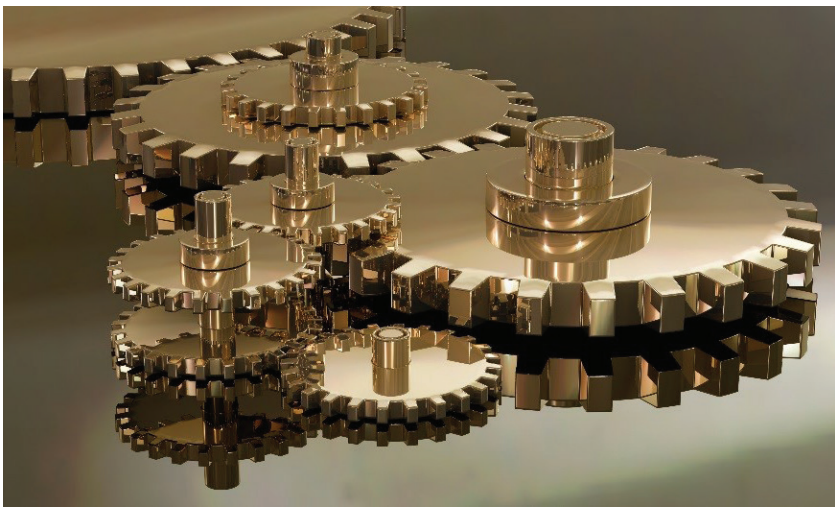
It is not news that benchmarking, internally or externally, is a magic word widely used in IT-mature organizations. This benchmarking activity, at least initially, might not aim to punish or to reward, but to have C-level strategic information, such as how competitive a company is versus competitors with regard to productivity, quality, time to market, price of the product, etc. Are we best in class? In the running race results, are we near the front, in the middle or near last? To have this accurate information is extremely fascinating because the results (numbers on hand) can or cannot be aligned to the ones that someone initially had in mind. If we are in an incorrect position or if we do not know our position, then actions might be taken without delay.

There are a variety of consulting companies that offer benchmarking services: Galorath, Leda, Namcook Analytics/Capers Jones, Premios (formerly David Consulting Group), Q/P Management Group, QSM, Reifer or TI Métricas (among others) have internal data repositories to help customers. Another interesting repository (IT software development and maintenance) is maintained by the International Software Benchmarking Standards Group (ISBSG), a nonprofit organization founded in 1997 that collects anonymized data from IT companies and IT customers.

A common denominator among benchmarking sources is that the techniques used to quantify software are mainly based on the standard Functional Size Measurement concept, even though different concrete methods can be used (IFPUG, NESMA, COSMIC, etc.) We can use ad-hoc methods to compare different aspects internally, but if we want to compare projects, products and companies externally (with our competitors), then we need to use the most widely used methods. If we use miles as a standard way to measure the distance between points, then we can compare the speed (miles per hour), the consumption (fuel per mile) and much more, but if we use an ad-hoc metric such as "xyz," then we can say that a car's consumption is "0.25 liters of fuel per xyz" but this will not be comparable to anything and nothing valid will come out of our boundaries.

continued on page 30

(Automation and Too Many Metrics, continued from page 29)



Therefore, Functional Standard Methods provide the greatest value when comparing productivity, quality and much more, internally (across projects) and externally with other companies, markets, etc. If a tool is used and it provides other metrics, it is not possible to have standard benchmarking, and results can only be compared with organizations using this same ad-hoc method.

Counting Big Software Legacy Applications

The Function Points method is extremely good for sizing actual projects and products, but can be difficult to apply for sizing big software legacy applications in a short period of time. It is typical in very big organizations to see backend or core systems with a legacy of more than two decades and with hundreds of thousands of programs.

In spite of that, in a theoretical world, all the system's functionalities might be correctly documented and up-to-date, theory is not always reality. When discussing applications with thousands of functionalities, sometimes with a high number of users and with functional knowledge scattered in many documents created at different times, counting the functional size of the installation, or even the applications, can be a challenging task. Perhaps we will have, as unique information, a set of production software libraries with thousands of programs and the task to create mechanisms that have automatic traceability to determine which software and functionalities are really used because sometimes there is a trend of not removing obsolete software from production.

If we need, for example, to size big applications in a short time, to size the product and its quality and to estimate (for example outsourcing maintenances) without inconveniencing users/customers much more than expected it would be almost impossible using a Functional Size Method. Theoretically, it would be possible, but not in practical application.

In those cases, to use a tool to automate metrics of the application, or even the whole organization, and to have results in a short time is essential. I invented one to analyze and measure a large number of software: size metrics, quality of the physical code, inter-relationships with other systems (sometimes maintained by third parties) and a lot of other information to a speed of more than 5,000 lines of code analyzed per second. In this case, using tools is the less bad option, focusing not on functionalities but in the physical code, and for example transforming statements in backfiring metrics: far away from standard Sizing Metrics is the most practical approach.

Perhaps the best automation approach regarding Functional Size might be applied in design phases by transforming, in an automatic way, design documents into Functional Size and comparing different document versions to resize and to correctly revise the projects. It cannot be unusual to discover that the documentation used to determine an application size, for example, is unrealistic. With those functional document systems, we can define tools to count the size in an accurate way and, at the same time, ensure the documentation is standard and not proprietary. So, different benefits can be obtained: sizing automatically from design documents, generating more standard documents and ensuring that the requirements and designs are precise and perfectly fit the customer needs. ■

About the Author: Antonio Ferre Albero *(Valencia, Spain) has more than 30 years of experience in information technology (IT), project management and metrics for private companies, government and large IT companies. He is CFPS accredited, has been member of different IFPUG committees for years and is currently the IFPUG CMC chair. Antonio is project manager at GFT, a European company with offices in 11 countries focused on innovative IT solutions. He specializes in a variety of disciplines including project management, quality and CMMI, metrics, functional size and Function Points, productivity, benchmarking, estimation processes, technology strategies, Db2, databases and big systems. As a senior technologist and project management passionate, he applies best practices to insure IT helps organizations and their employees. Antonio's technical articles have been published many times in newspapers and other print publications.*

LOGAPPS *PRESENTS*

M.A.R.IN.E

Requirements and Function Point Automation Tool

M.A.R.IN.E. (**M**achine **A**ssisted **R**equirements **In**spection and **E**valuation) is a cutting-edge desktop application that automates function software size measurement following the IFPUG standard, and also streamlines requirements analysis. Our innovation in Natural Language Processing (NLP) integrates with Office applications and develops customizable high level software size estimates through automation, while also producing effort and schedule estimates.

logapps.com/marine
marine@logapps.com
(703) 592-6362

Thanks ISMA15, Rome:

INSPIRED BY OTHERS, SHARING IDEAS AND SYNERGIES

By Antonio Ferre Albero

The ISMA15 conference took place in Rome, Italy from May 9-May 11. The three intensive days included four workshops with 45 attendees, an IFPUG exam that resulted in 18 new certified IFPUG CFPSs and CFPPs, many IFPUG board meetings and one day-long main conference for which 389 people registered.

The main conference, held on May 11, included 15 presentations and more than 20 speakers. Opening remarks were given by Mauricio Aguiar, IFPUG president. He shared the idea that "metrics are one of the best kept secrets in the IT world." He gave thanks to GUFPI ISMA association and he announced that the next ISMA (ISMA16) will take place in October 2018 in São Paulo, Brazil. Luigi Buglione (GUFPI ISMA president and IFPUG board member) and Filippo de Carli (GUFPI ISMA vice president and IFPUG education and conferences chair) shared in the opening of the conference by giving thanks to the ISMA15 sponsors and partners and presenting the agenda topics.

Thomas Fehlmann began the presentation portion of the conference with a discussion about consumer metrics for privacy and safety with Test, Privacy, Safety and Consumer Metrics and introducing interesting metrics models that determine privacy indexes based on privacy needs and protection using a graphical representation that could be similar to the EU Energy Consumption label.

Talmon Ben-Cnaan from Israel, one of the fathers of the IFPUG SNAP method and chair of the IFPUG Non-Functional Software Sizing Committee, talked about whether test estimation is a science or art. He asked how many scenarios are needed to test a concrete functionality, compared the concepts Scenario, Pairwise and three-way combinatorial and shared his powerful insights into testing estimation techniques, testing stages and streams, complexity factors and more.

Cecilie Thormodsrud presented in a practical and concise way that Function Points have provided a lot of benefits to Telenor Norway, a company with more than 30,000 employees, including strategy for IT decisions and productivity root cause analysis and improvements. She also discussed how to succeed using Function Points and what not to do while using them. The success stories were inspiring.

Rosangela Riccotta talked about innovation and metrics in corporate governance with a great deal of energy and enthusiasm. She discussed how companies deal with the speed of innovation, motivators of innovation and factors of innovation, including human, organizational culture company strategy and corporate governance, as well as how to innovate metrics.

Paolo Cecchini, Principal Project Management Expert at Ericsson Telecomunicazioni S.p.A. and active member of the PM, shared his vision with a presentation called "Tracking Project Performance: From Analytical to Strategic Results" and discussed the challenging topics of cost, time, scope, quality, priority and value as key words for project success. He also talked about the importance of having performance indicators (relevance indexes) about process adherence, project governance adherence and customer satisfaction.

Fabrizio Di Cola, Domenico Geluardi and Daniele Zottarel from Sogei, a company with 26 IFPUG-certified CFPSs/CFPPs and an IFPUG FPA Function Point Counts database of around 1,300 applications and 5,000 development and enhancement projects, focused on the challenging links between iterative and agile development and IFPUG Function Point analysis including applying elementary process and logical file definition in iterative development and the measurement difference between iterative software measurement and related development effort, giving examples on how to use IFPUG Function Point analysis with iteration and in effort and velocity, among others.

Carlo Capeccia and Alberto Leardi from Leonardo S.p.A., a high-tech company headquartered in Italy, focused on aerospace, defense and security, and with more than 45,600 employees, shared their vision about process improvement in software development. "Delivering Quality" was about the interesting and real experience of selecting the tool/platform to manage and to measure quality and having strategic metrics, as well as how projects are piloted and setup, roadmaps and lessons learned, among other topics.

continued on page 34

(Thanks ISMA15, continued from page 33)

In the CEP valid conference, Gianfranco Lanza took on the challenging topic of "How to Measure a CMS," demonstrating that a Content Management System (CMS) can be measured using IFPUG Function Points in a standard way, giving a set of highlights and counting tips, including files concept, elementary processes, CMS type object sizes and the importance of being careful in the productivity calculation.

In the afternoon, Christelle Delcourt and Anne-Lies Willemen inspired the audience with their success story, "Ten Years of Software Development Measurement in a Financial Market Infrastructure, Evolution Just Happens!," reflecting on the Euroclear group way of how Function Point Counting has been more than an estimation tool and how it is also a powerful management KPI framework tool, moving from reporting figures to reporting key messages, interpreting results and identifying and understanding root causes, providing outside the core of the delivery expert team objective management analysis on their software delivery performance and sharing that they obtained very interesting ideas from attending previous IFPUG ISMA conferences (we learned from ISMAx). They suggested that if they were inspired by others attending those events, maybe others can be inspired by their story and take away some ideas, even if they're small ones. We would like to thank them for sharing and spreading those ideas.

Thimoty Barbieri and Irene Rocca discussed an interesting topic, how to support the IFPUG FPA-SNAP bimodal estimation (Waterfall and Agile) using JIRA as an alternative to the story point estimation. They also talked about the strategic comparison of Agile (Story Points) to FSM (Function Points) and presented a demo of the free JIRA plugin they developed that manages the data functions, transactional functions in JIRA, resulting in Epic FPA estimation and SNAP estimations.

Eduardo Alves de Oliveira from Brazil dealt with using FPA to pay software development contracts and the SISP method (System of Administration of Information Technology Resources effort manual), which is supported by the Brazilian federal government (Brazilian standard), the 23 different counting types, how contracts are paid using effort FP (SISP FP), how SLA contracts are defined using SISP FP, examples of counting projects using SISP and other interesting topics.

Paola Billia and Maurizio Sapienza discussed "Waterfall Versus Agile: How Can We Compare Them?" and shared a set of points to compare measures of Agile projects against Waterfall projects using Function Points and SNAP Points, company historical data, measuring FP Size in Agile Frameworks and the interesting "Fit Indicator" concept in addition to the traditional "Productivity" concept that provides strategic information.

The "Balloon Effect: How (an Improper) Scope Management Can Impact from Size to Effort, Duration and Costs," which was presented by Luigi Buglione, analyzed the old and the new productivity paradox using the ABC+123 schemas, analyzed the value chain to verify the sizing units used for obtaining more affordable estimates and discussed the side effects that could take place due to the balloon effect and how only two of the sides can be accomplished. He demonstrated this idea with a real balloon.

Simon Wright from the United Kingdom focused on the interesting topic of measuring the requirement quality, as well as the effect of this quality in better estimates. He talked about accomplishing goals based on measuring the quality of the user requirements, how to improve the quality of those requirements and offered a set of very interesting guidelines and rules for writing measurable requirements complete with examples of high quality user requirements.

At the end of the day, Federico Maria Capo, on behalf of the IIBA Italy Chapter, discussed how the IIBA organization and IIBA Italy Chapter are uniting a community of professionals with the goal to be the world's leading association for business analysis professionals, as well as the concept to "keep it simple but valuable." The presentation titled, "Agile Business Analysis: Presenting IIBA Approach to Agile Delivery," also reviewed business analysis, Agile principles and applying the principles of Agile business analysis at Strategy and Initiative Horizon.

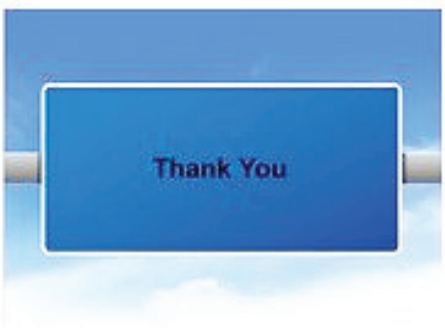
Domenico Natale delivered some interesting news about ISO and software standards, a group of Italian students and professors were recognized for their contribution to the metrics world (thanks), and the IFPUG board explained different topics related to IFPUG. The GUFPI-ISMA association was also featured, as well as much more in a nice and friendly atmosphere.

Before the main conference, several workshops took place including the IFPUG Software Non-Functional Assessment Process (SNAP). Steve Kitching discussed SNAP, which measures non-functional requirements and is complementary to FPA. In the afternoon, Philippe-Emmanuel Douziech and Michele Slocovich talked about Consortium for IT Software Quality (CISQ) with a focus on the automated from source code positioning.

On the morning of the second day, Massimo Canducci conducted a workshop on the Lego Serious Play methodology as a process to enhance innovation and business performance, as well as working with the measure and requirement concepts. In the afternoon, Roberto Meli discussed the approximation E&QFP technique and the Functional Size Measurement Method (FSMM) Simple Function Point (SiFP).

During the three days of the conference, many IFPUG board meetings took place as did talk about strategic guidelines for providing the best value. The IFPUG exam resulted in 18 new certified IFPUG CFPSs/CFPPs. On behalf of IFPUG, I would like to welcome this passionate group of people that trust that metrics in the IT world are not only necessary but essential.

Thanks to everyone who attended the conference in Italy for your active passion and knowledge in this IT metrics world. Thanks to GUFPI-ISMA, to Luigi, to Filippo and to all the people who have been working hard for others and for the IT metrics world. Thanks to the ISMA[15] sponsors and partners and, finally, thanks to all those from different countries (United Kingdom, France, Italy, Switzerland, Brazil, Belgium, Israel, Norway, Spain, United States, Denmark, Finland, India, Greece, etc.) who have been an inspiration to others during this ISMA event. Again, thanks Italy. We look forward to seeing everyone again at ISMA[16] in Brazil.



Managing Agile Activities Using Standardized Measures and Managing Agile at Scale: A Briefing for Software Executives and CIOs

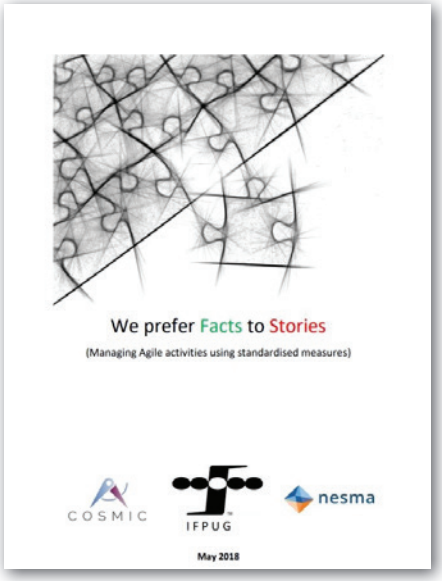
"We Prefer Facts to Stories (Managing Agile Activities Using Standardized Measures)" is a white paper produced by IFPUG, COSMIC and Nesma, all of which are international organizations that maintain the ISO/IEC software sizing standards used in industry for estimation, budgeting, contract and project management, supplier performance measurement, benchmarking and other management activities. This white paper shows how the ISO/IEC software sizing standards methods can be used to manage Agile-at-Scale activities, achieving this whilst leaving existing Agile processes unchanged at the team level.

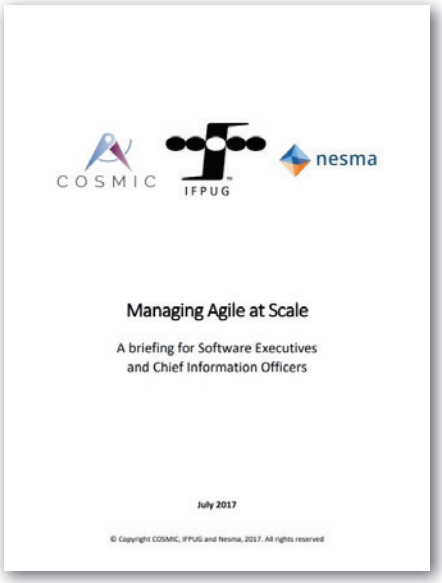
The essential processes that management relies on for budgeting, estimating and controlling Agile-at-Scale activities need a more reliable and objective measure of work output (i.e., software size) than story points. Any one of the ISO standard methods for estimating or measuring software size can meet this need. Senior management is responsible for setting budgets and allocating resources optimally so as to deliver the greatest value to the organization and for tracking progress against budgets across the organization.

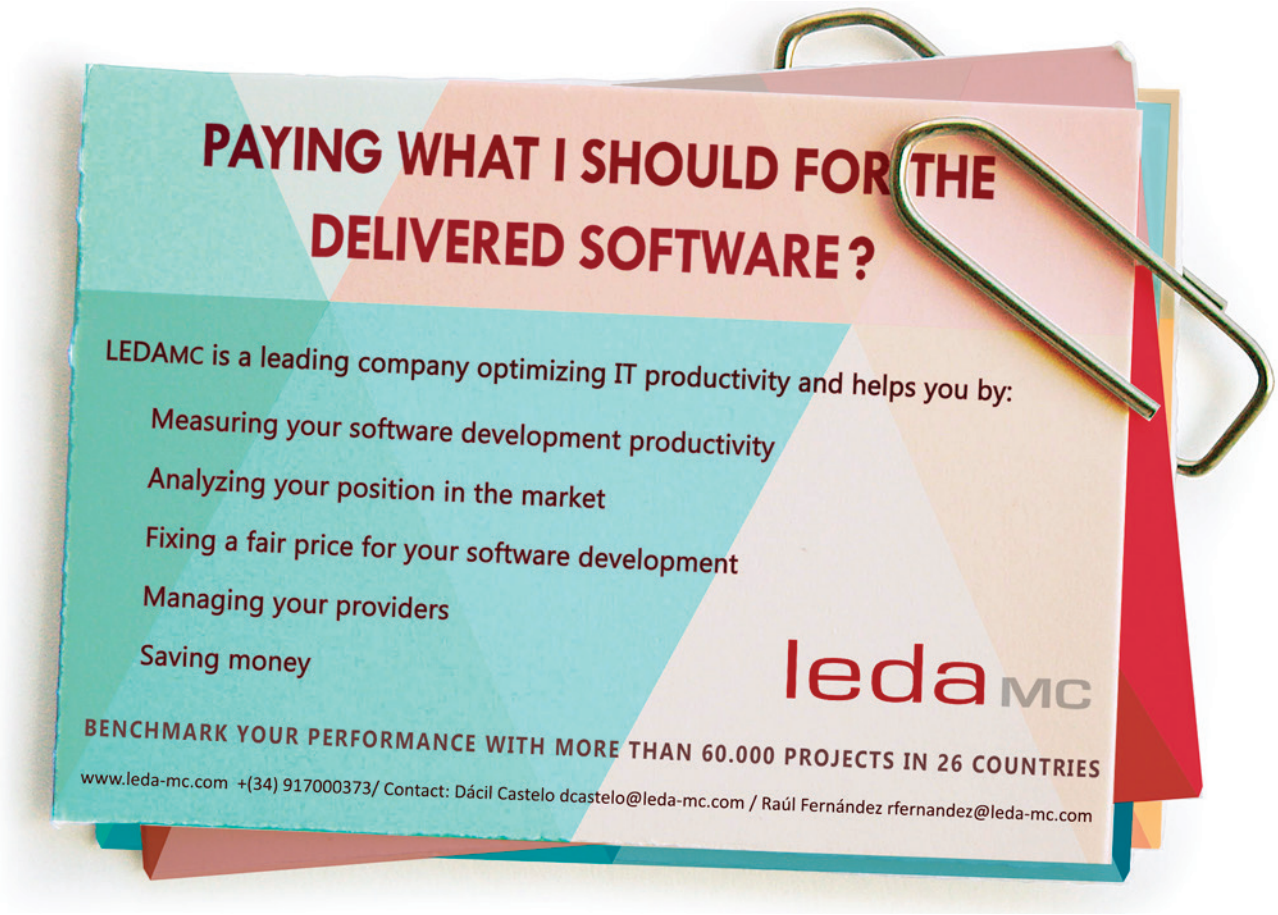
This cannot be done properly for a software group only using typical Agile processes where there are no common performance data across all the teams. These management tasks become even more difficult for an organization that has contracted out its software development to external suppliers that use Agile processes, but that do not use any standard performance measures.

This paper explains the challenges that management faces when confronted with the limitations of Agile metrics and shows how simple but effective and long-established ISO standard software measures can fit seamlessly into Agile processes to enable managers to estimate and control Agile delivery at scale.

This white paper complements the previous document "Managing Agile at Scale: A Briefing for Software Executives and CIOs" that offers simple but effective and long-established international standard solutions to manage Agile delivery at scale without risk of losing the speed and flexibility benefits of Agile processes.







IFPUG Board of Directors

Certification Committee

by Sheila Dennis, Committee Chair

I get feedback all the time from folks who don't think that they need certification, are convinced that their time involvement in IFPUG is wasted or think that we are just "old school." For those of you who think that we are an antiquated or dwindling profession, allow me to reassure you that we are not!

I had the privilege of manning the IFPUG booth at the International Cost Estimating and Analysis Association (ICEAA) held June 11-15 in Phoenix Arizona with Carol Dekkers. There was a great deal of interest and activity. It was interesting to see how many organizations are still using SLOC as their size measure. Also, while many organizations have well defined acquisition processes, their estimation processes are not. One of the keynote speakers came by to talk with us about the use of Function Points in estimation. In effect, he asked me if I knew how many "IT shops" there were out there that are struggling with consistency, reliability and accuracy in their size and estimation models? According to this distinguished gentleman, we have barely "scratched the surface" of the number of organizations that would benefit from our cumulative experience. He encouraged us to keep pushing functional (and non-functional) sizing methods but most of all–how to use these methods.

Promoting the use of our product through a variety of marketing venues, partnering with other organizations and encouraging certifications are several ways to keep the methodology relevant and moving forward. One of the partnerships is with ICEAA, which will be offering a new certification next year for software estimation. IFPUG Function Points are part of that body of knowledge. From this perspective, certifications that represent our knowledge and abilities are as important as they ever were.

The certification committee has updated all of its automated Function Point exams (CFPS/CFPP) and has made them available in English, Italian, Spanish and Portuguese. We are currently working on Korean and Japanese exams and are planning to start translation into French in the near future. The Certified SNAP Practitioner (CSP) exam for 2.4 is now available and is scheduled for translation into Portuguese as well. If you don't have your certification, there is no better time to get it!

Communications and Marketing Committee

by Antonio Ferre Albero, Committee Chair

One of the deliverables of this committee is *MetricViews*, the publication that you are reading right now, perhaps using a laptop, tablet or smartphone. We have tried in recent issues to provide more visuals and to improve the look and feel of the magazine. At the same time, we have included main articles and relevant IFPUG news. Thanks a lot to the *MetricViews* editor and to all the people working behind the scenes on these matters.

In the last few months, IFPUG has adapted to the General Data Protection Regulation (GDPR), ensuring that only people who want to receive IFPUG news are the ones receiving it. We have also switched to mailing systems with determined roles to ensure data privacy. If you previously received our news and now you are not receiving it, you may not have confirmed the "IPFUG opt-in to receive our communications, as well as consent

to our privacy policy" that we sent along in some emails. If you miss the IFPUG news, please contact the IFPUG office at ifpug@ifpug.org.

The Communications and Marketing Committee (CMC) works closely with other committees. For example, we work with the Conference and Education Committee to spread as much information as possible about events such as ISMA editions. In addition, from the technical point of view, the committee has upgraded different aspects of the website. For example, we have updated the PHP version and other aspects to improve security. At the same time, a set of metrics has been put in place, for example, to analyze periodically the countries that access IFPUG content in order to improve and/or translate concrete documents or PDFs to the languages of those countries. The initial and short overview is that IFPUG is really global: IFPUG content has been read in 130 countries in less than three months.

In the CMC, we have multi-country volunteers and members, and we want to expand with a few more people. If you are interested in volunteering, working for others in the IT metrics world, you can submit an IFPUG Volunteer Form to IFPUG at ifpug@ifpug.org.

Conference and Education Committee

by Filippo De Carli, Committee Chair

After the success of ISMA[15] in Rome, Italy in May, with more than 400 people attending the four-day event, the Conference and Education Committee (CEC) is going to spend the second half of 2018 supporting and organizing the new ISMA conferences for 2018 and 2019.

The first one will be ISMA[16], which will be organized and hosted by the Brazilian Function Point User Group (BFPUG), will be held in Sao Paulo, Brazil on Oct. 18 and has been recognized as CEP valid. You can find more info at ifpug.org/isma16.

The second conference, ISMA[17], will take us back to India for the second time but this time to Bangalore in early March 2019. At ISMA[17], we'll celebrate "40 Years of Function Points" since the first Albrecht paper creating our FSM movement and community was published in May 1979, just 40 years ago. Further information about ISMA[17] will be provided during the next few weeks. This new event will include conference day workshops and an automated exams session. Please check our website for more information by clicking on the "Events" menu.

A reminder: IFPUG members can access conference proceedings, at no charge, in the "Knowledge Base" within the "Members Services Area" of the IFPUG website.

As any IFPUG committee, we are delighted to work together with all of you interested in helping us. If you have comments, suggestions or feedback, please contact us at cec@ifpug.org!

Functional Sizing Standards Committee

by Dan French, Committee Chair

2018 has been a busy and productive year for the Functional Sizing Standards Committee (FSSC) so far. The committee has published the "FPA Applied to BPM-Based System Project" white paper, as well as an update to iTip #3 Logon. We also welcomed three new committee members: Esteban Sanchez, Sergio Brigido and David Lambert. Our committee is grateful to have these new members on board and looks forward to the contributions they will make. If you or anyone you know is interested in volunteering for the committee, please submit an IFPUG Volunteer Form to IFPUG at ifpug@ifpug.org.

In addition to our monthly committee meetings, the FSSC met for three days in June prior to the International Cost Estimating and Analysis Association held in Phoenix, Arizona from June 12-15. At the annual meeting, we will be finalizing some projects for publication by year-end, including the addendum to the "Data Warehouse" white paper and updates to the iTip #5 Real-Time Data Sharing and iTip #6 Shared Data. We will also be initiating a joint project with the Non-Functional Software Standards Committee (NFSSC) regarding the General System Characteristics (GSC).

The committee is always looking for new projects to work on and welcomes suggestions from members on topics of interest. You can submit your suggestions to dfrench@cobec.com.

The FSSC is looking forward to continuing a productive 2018 with our new members contributing to the FSSC and IFPUG's success.

continued on page 40

International Membership Committee

by Saurabh Saxena, Committee Chair

The International Membership Committee (IMC) firmly believes in enhancing IFPUG members' experiences by resolving all types of queries quickly. This not only brings value to existing members but also attracts new members who now have a one-stop solution for all their queries. Country representatives Marcio Silveira (Brazil), Gianfranco Lanza (Italy), Lionel Perrot (France), Cao Ji (China), Ivan Pinedo (Spain) and Saurabh Saxena (India) interact with people in their local language and act as first-level point of contacts for all IFPUG related queries.

During the past few months, based on discussions with members, the following actions were identified:

- **Membership Renewal Process**
  - o Grace Period: The earlier grace period of 90 days was provided to any person leaving the company covered by corporate membership. Now, the grace period benefit of 90 days has been extended to the employees wherein their employer did not renew the corporate membership.
  - o The international membership committee will work in tandem with CMA to not only smoothen the renewal process but also track the actual pain areas/benefits required by the members.
- **Enhancing Membership Experience**
  - o Create strong local chapters for networking and knowledge sharing, starting with creating a WhatsApp and Linkedin group for the India region.
  - o Included a country representative for the France region.
  - o Assist members with the new CFPS process (provide a smooth transition from Prometric to iSQi) and Certification Extension Program (CEP).

In the coming times, we look to:

- Continue the good work of providing quick and accurate response to all IFPUG related queries.
- Follow up with IFPUG members who wish to discontinue their membership, understand their pain areas and try to win them back.
- Build a strong IFPUG India chapter and later build similar ones in regions globally.
- Start the process for the nomination of a 2018 IFPUG Honorary Member. ■

Industry Standards Committee

by Steven Woodward, Committee Chair

Carol Dekkers and Steven Woodward continue to represent the United States and Canada, respectively, as part of ISO SC7º (Software and Systems Engineering) activities, keeping IFPUG visible as a valuable sizing method for the systems of today.

Mr. Woodward attended the ISO SC 38 (Cloud and Distributed Processing) meetings in Warsaw, Poland as a Standards Council of Canada representative, where cloud SLAs are a metrics subject of interest. He also presented at and met with OpenStack, OMG, itSMF, ISACA, ICEAA, NIST and Enterprise Architects, spreading the software metrics concepts and value proposition.

ISO/IEC Standards: The Object Management Group (OMG) will be submitting its specification for automated Function Point counts (with a basis in IFPUG FP) as a Fast-Path/PAS (Publicly Available Specification) to ISO/IEC JTC1 in the coming weeks. Currently, the IFPUG board is preparing a response (with the input of Dan French, Sheila Dennis, Carol Dekkers, Steve Woodward and others) because the specification is not completely conformant with our IFPUG FP standard.

Carol Dekkers, Dan French and others from IFPUG attended and presented at the ICEAA (Costing and Estimation) conference in Phoenix, Arizona providing further visibility for IFPUG in this Agile world! Talmon Ben-Cnaan, as chairperson of the IEEE Non-Functional Sizing Standardization activity, continues moving forward with IEEE to standardize non-functional measures.

International Cost Estimation and Analysis Association (ICEAA) is introducing a new Software Cost Estimation Body of Knowledge (SCEBOK) and will launch a new certification next May at its annual conference (slated for Tampa, Florida.) ISO/IEC functional size measurement standards (IFPUG, NESMA, COSMIC) are included as sizing units of measure in this new SCEBOK and Carol Dekkers, Sheila Dennis and Christine Green are involved in assisting with curriculum development.

Pierre Almen, as the liaison with International Software Benchmarking Standards Group (ISBSG) community, will be attending the ISBSG IT Confidence conference and meetings in Mexico City on Sept. 12.

2019 will by a busy year for industry standards, as requests for standardized directions in subject areas such as Artificial Intelligence and Machine Learning will be on executives' radars. The IFPUG method, in conjunction with other measures, helps provide a stable foundation to plan and deploy the complex and diverse solutions of 2020.

We welcome your participation in helping to increase software metrics competencies across multiple ICT standards communities.

Non-Functional Sizing Standards Committee

by Talmon Ben-Cnaan, Committee Chair

IFPUG is working with IEEE's Software and Systems Engineering Standards Committee to generalize SNAP as an IEEE standard. The standard was prepared, reviewed internally by the workgroup and has been sent to IEEE for review.

ISO 14143: "Information Technology—Software Measurement—Functional Size Measurement" is a standard that defines the requirements from a functional sizing method. It has a set of rules and conditions that every functional sizing method must meet. Since there is no similar standard for non-functional requirements, the Non-Functional Sizing Standards Committee (NFSSC) has converted the requirements of ISO 14143 into requirements for non-functional standard. As expected, SNAP meets all these requirements.

More companies have sent us SNAP data. The data is used to verify the correlation between SNAP and effort. The NFSSC is investigating an improvement in the definition of sub-category 1.5 based on the data we have. This is still to be verified with more industries and more companies before we publish our conclusions.

Committee Rosters

Certification Committee

- Sheila Dennis, Premios – ***Chair***
- Greg Allen, Pershing LLC
- Mahesh Ananthakrishnan, Cognizant Technology Solutions
- Donald Beckett, QSM
- Manuel Buitrago, LEDA Consulting, S.L.
- Teresa Cristina De Spagna Zenga Beraldo, BANCO BRADESCO S/A
- Francesco Gasparro, Capgemini Italia, SPA
- Jim McCauley
- Roopali Thapar – Board Liaison

Communications and Marketing Committee

- Antonio Ferre Albero, GFT IT Consulting – ***Chair***
- Stephen Neuendorf, NMC – ***Vice Chair***
- David Herron, David Herron & Associates
- Justin Keswick, Bank of Montreal
- Tamara De San Teodoro, LEDA Consulting, S.L.
- Carol Dekkers, Quality Plus Technologies, Inc., – Board Liaison

Conference and Education Committee

- Filippo De Carli, GUFPI-ISMA Gruppo Utenti Function Point Italia – ***Chair***
- Thiago Silva Da Conceicao, Synapsis
- Prof. Eduardo Alves De Oliveira, Servico Federal De Processamento De Dados (SERPRO)
- Antonio Ferre Albero, GFT IT Consulting
- Mr. Eduardo Alves Oliveira, Sr.
- Dr. Luigi Buglione, Engineering Ingegneria Informatica SpA – Board Liaison
- Saurabh Saxena, Zensar – Volunteer
- Sushmitha Anatha, Accenture – Volunteer
- Alfonso Gonzalez, LEDA mc – Volunteer

Functional Sizing Standards Committee

- Daniel Bradford French, Cobec Consulting – ***Chair***
- Bonnie Brown, DXC Technology – ***Vice Chair***
- Diana Baklizky, TI Metricas
- E. Jay Fischer, JRF Consulting
- Peter Thomas, Steria
- Adri Timp, Equens
- Tammy Preuss, AT&T
- Steve Keim, Premios
- Charles Wesolowski – Board Liaison

International Membership Committee

- Saurabh Saxena – ***Chair***
- Ivan Pinedo, Premios
- Dr. Lionel Y Perrot, Semantys
- Anjali Mogre, Atos Origin International SAS
- Cao Ji, Beijing Suiji Tech
- Gianfranco Lanza, CSI Piedmonte
- Sergio Brigido, TI Métricas
- Dacil Castelo, LEDA Consulting – Board Liaison

Industry Standards Committee

- Steven Woodward, Cloud Perspectives – ***Chair***
- Carol Dekkers, Quality Plus Technologies, Inc.
- Pierre Almen, ImproveIT
- Talmon Ben-Cnaan, Amdocs
- Charles Wesolowski – Board Liaison

Non-Functional Sizing Standards Committee

- Talmon Ben-Cnaan, Amdocs – ***Chair***
- Dr. Charley Tichenor, Marymount University – ***Vice Chair***
- Francisco Julian Gomez, LEDA Consulting, S.L.
- Kathy Lamoureaux
- Tomasz Marchel, Asseco Poland S.A.
- Roopali Thapar, IBM
- Saurabh Saxena, Amdocs Ltd – Volunteer
- Srinivasa Rao K, Mindtree Consulting – Volunteer
- Dr. Luigi Buglione, Engineering Ingegneria Informatica SpA – Volunteer
- Charles Wesolowski, – Board Liaison

CURRENT CONTACT INFORMATION?

To ensure you won't miss out on any IFPUG communications, please log in to your profile on the IFPUG Members Services Area and update your information.
Go to www.ifpug.org
Send emails to ifpug@ifpug.org, call 609-799-4900 or fax 609-799-7032
Write to:
IFPUG, 191 Clarksville Road,
Princeton Junction, NJ 08550

Congratulations to these NEW and Extended Certified Function Point Specialists!

Marisa Accacio

Alessandra Albinati
Almaviva SpA

Sandro Adanti

Stefano Alunni
Capgemini Italia SPA

Antonio Attadia
Almaviva SpA

Ermanno Bagnai
Almaviva SpA

Armando Baldinacci
Leonardo S.p.a.

Luisa Banco

Maria Barbino
Capgemini Italia SPA

Marco Bassano
Capgemini Italia SPA

Donald Beckett
Quantitative Software Management, Inc.

Elisabetta Belotti
Almaviva SpA

Alberto Bentivegna
Almaviva SpA

Andrea Bertanelli
NTT DATA Italia SpA

Rafał Bielicki
SAS Institute

Tiziana Borsini
IBM

Annamaria Bucci
NTT DATA Italia SpA

Giulio Buonomini
Leonardo S.p.a.

Lucas Calmon
Banco de Brasilia S.A.

Maria Cristina Calzetti
D.P.O. Srl

Mario Camilli
Capgemini Italia SPA

Gabriele Caramanica

Fiorenzo Nicola Carmenini
Almaviva SpA

Anna Carnera
Almaviva SpA

Giorgio Caporaletti
Leonardo S.p.a.

Marcello Ceccarelli
Almaviva SpA

Angelo Celani

Przemyslaw Chelstowski
SAS Institute

Andrea Cherici
Almaviva SpA

Ho Chuen Chan

Francesco Cincotti
Almaviva SpA

Fulvio Contaldi
Information Services Group Italy (ISG)

Laura Contino
Almaviva SpA

Elielton Costa Da Silva

Indra Brasil Soluções e Serviços Tecnológicos S/A

Fabio Costanzo
Capgemini Italia SPA

Simona D'Ambrosio Barbarisi
Almaviva SpA

Anitha Damodharan
IBM

Giambattista De Luca Carignani
IBM

Monica Del Buono
IBM

Isabela Del Corso
TI Metricas Ltda

Leon Denizard De Carvalho Gomes
TI Metricas

Chiara Di Bonito
Almaviva SpA

Nunzia Di Lecce
Capgemini Italia SPA

Giuseppe Di Iorio
Accenture

Diego d'Ippolito
Almaviva SpA

Emma Di Pasquale
NTT DATA Italia SpA

Santa Di Salvo
Leonardo S.p.a.

Jlenya Di Sanza
Leonardo S.p.a.

Orietta D'Olimpio
IBM

Jonatas dos Santos

Laura Fabbri
NTT DATA Italia SpA

Simona Farina
Leonardo S.p.a.

Roberto Fenaroli

Maria Cristina Ferrari
Almaviva SpA

Claudia Filiberti
Leonardo S.p.a.

Ferruccio Foti
Capgemini Italia SPA

Alessandra Gallozzi
Almaviva SpA

Andrea Garavini
NTT DATA Italia SpA

Tomas Gasiorowski
Orange Polska

Stefano Gavassuti
NTT DATA Italia SpA

Vittorio Giacchetti
Almaviva SpA

Rosalba Giunta
Almaviva SpA

Gianluca Rocco Greco
Almaviva SpA

Dora Grillo
Almaviva SpA

Francesca Grossi
Almaviva SpA

Daniel Guinda
LEDA Consulting, S.L.

Annunziata Italiano
Almaviva SpA

Krzysztof Klosowski
Orange Polska

Rajesh Koduru
MPHASIS

Tetyana Komarova
NTT DATA Italia SpA

Rosangela La Fiandra
Capgemini Italia SPA

Daniele Lagatta
Almaviva SpA

Aldo Langianese
Almaviva SpA

Roberto Laurita
NTT DATA Italia SpA

Michal Lazicki
Accenture

Maria Lonetti
NTT DATA Italia SpA

Jose Lopez
Soriano

Annunziata Luciano
Sirti S.p.A.

Valeria Maiorino
Almaviva SpA

Gianluca Marchetti
Almaviva SpA

Paolo Marchi
Leonardo S.p.a.

Sauro Marini
Leonardo S.p.a.

Laura Massucci
TIM S.P.A

Livio Maurizi
Sirti S.p.A.

Maurizio Menghini
Almaviva SpA

Marcia Morais
CTIS TECNOLOGIA S/A

Gelsomina Mori
Eustema S.p.A

Paolo Mori
Almaviva SpA

Barbara Nardis
NTT Data Italia SpA

Roberta Nasti
Leonardo S.p.a.

Mauricio Neves

Fernando Oliveira
Banco de Brasilia S.A.

Daniele Papa
Capgemini Italia SPA

Nicola Marcello Parisi
Sirti S.p.A.

continued on page 44

Antonella Patrignani
IBM

Rosana Paulino

Antonio Pellicanò
Ericsson It Solutions & Services S.P.A.

Daniela Petrucci
Almaviva SpA

Vincenzo Pinto
Almaviva SpA

Paolo Pioli
Capgemini Italia SPA

Irma Policicchio
Capgemini Italia SPA

Emanuela Polito
Almaviva SpA

Aldo Pondaco Neto
Accenture

Veronica Porta

Filippo Raciti
Leonardo S.p.a.

Rafael Ramos
Passaro Prime Informatica Alpha Ltda

Nicolò Rebella
Leonardo S.p.a.

Leandro Renno Siniscalchi
Cast Informatica S.A.

Paolo Ricotta
Leonardo S.p.a.

Maria Romano
Almaviva SpA

Luca Santillo
Agile Metrics

Guido Servili

Capgemini Italia SPA

Marcello Sgamma
NTT DATA Italia SpA

Barbara Sgaragli
Leonardo S.p.a.

Damian Sierajewski
Accenture

Francesco Sorace
NTT DATA Italia SpA

Claudio Strazzullo
Almaviva SpA

Luca Tamburini
Almaviva SpA

Adriana Trentini
TI Metricas Ltda

Alberto Livio Troisi
Almaviva SpA

Olga Uberti
NTT DATA Italia SpA

Alessandro Venturi
Leonardo S.p.a.

Antonina Vicario
Almaviva SpA

Congratulations to these NEW Certified Function Point Practitioners!

Tiziana Angelini
Engineering Ingegneria Informatica SpA

Flavio Barsotti
NTT DATA Italia SpA

Samuel Batagliao
Accenture

Mariacristina Cadolini
TIM S.P.A

Eleonora Cesaretti
Eustema S.p.A

Luca Cilento
Almaviva SpA

Filomena Chiariello
Accenture

Ciro Coppola
Almaviva SpA

Eduardo de Albuquerque Gomes Pereira
Plennus TI

Severino Di Cola
Leonardo S.p.a.

Sonia Fraioli
Eustema S.p.A

Maria Grazia Di Pasquale
NTT DATA Italia SpA

Maria Di Russo
NTT DATA Italia SpA

Stefano Ferrari
Ericsson It Solutions & Services S.P.A.

Massimo Fratini
Leonardo S.p.a.

Luana Guimaraes Silva
Banco Bradesco S/A

Massimiliano Loconte
Sopra Steria Group SpA

Marina Mancini
Almaviva SpA

Fabricia Marques Magna
Sistemas e Consultoria S/A

Luca Mele
Accenture

Julio Cesar Montoya M
IBM

Claudio Occoffer
Pride S.p.A.

Tamara Papa
Leonardo S.p.a.

Bianca Rodrigues De Faria
Banco Bradesco S/A

Davide Sammartino
Accenture

Flavia Soares
SISTRAN Informática

Joao Paulo Souza Alves De Souza
Servico Federal De Processamento De Dados (SERPRO)

Jussania Souza
Cast Informatica S.A.

Alan Tomé de Souza
ATSNET Informática e Desenvolvimento de Sistemas LTDA - ME

Imma Volpe
Almaviva SpA

Patrizia Volpi
Almaviva SpA

Sabina Wolski
TIM S.P.A

Congratulations to this NEW Certified Function Point Specialists Fellow!

Greg Allen
Pershing LLC