

MetricViews

March 2019

Vol. 13 • Issue 1

inside

Artificial Intelligence Rationalization through Function Points

When Strategic Software Metrics are Even More Important than Financial Metrics

Nucleon: How to Better Assess Your Delivery Capability in Development Teams

Nucleon and Function Point Analysis

High-Quality Definition of Non-Functional Sizing Method

Function Points and Agile Models, How to Estimate

MEASURING FOR SUCCESS



In this Edition.....	2
Message from the President.....	3
From the Editor's Desk.....	5
Artificial Intelligence Rationalization through Function Points.....	6
When Strategic Software Metrics are Even More Important than Financial Metrics.....	11
Nucleon: How to Better Assess Your Delivery Capability in Development Teams.....	16
Nucleon and Function Point Analysis.....	21
High-Quality Definition of Non-Functional Sizing Method.....	26
Function Points and Agile Models, How to Estimate.....	30
ISMA ¹⁶ Brazil 'Métricas 2018:' São Paulo Prides Itself on Hosting the IFPUG Conference Once Again!.....	36
IFPUG 2018 Annual Meeting.....	40
Committee Reports	
• Certification Committee.....	41
• Communications and Marketing Committee.....	41
• Conference and Education Committee.....	42
• Functional Sizing Standards Committee.....	42
• International Membership Committee.....	42
• Industry Standards Committee.....	43
• Non-Functional Sizing Standards Committee.....	43
Committee Rosters.....	44
CFPS Certification Listing.....	45
CFPP Certification Listing.....	45
Stay Connected.....	46

IN THIS EDITION

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

- **Artificial Intelligence Rationalization through Function Points** (By AmolKumar Keote and Deepti Patil). In this article on Artificial Intelligence (AI), the authors discuss how AI can be rationalized through Function Point Analysis. This is achieved by mapping software functional requirements with the AI components. Learn what the authors have to say about this interesting topic.
- **When Strategic Software Metrics are Even More Important than Financial Metrics** (By Antonio Ferre Albero). The author makes the case for why companies must consider the “economics” of function points and related software measures. Information Technology (IT) is more than a service, it is a vital part of a company’s success and therefore product value, price and quality need to be measured.
- **Nucleon: How to Better Assess Your Delivery Capability in Development Teams** (By Jeppe Hedaa). Is your IT department viewed as a “black box?” Mr. Hedaa shares a formula for success, Nucleon. Based on research, his team has identified individual factors that lead to high performance in software development. Learn how you can apply this formula and improve your development capacity.
- **Nucleon and Function Point Analysis** (By Christine Green). The author suggests that team efficiency and scope management are key to successful and cost-effective IT projects. We learn how using Function Point Analysis to illuminate the scope and then linking the scope directly to the problems business’ are trying to solve will achieve success.
- **High-Quality Definition of Non-Functional Sizing Method** (By Srinivasa Rao Kanneganti and Talmon Ben-Cnaan). Considering using SNAP? This article recommends guidelines for evaluating a non-functional sizing method and evaluates the level of compliance of SNAP to the functional characteristics as defined by ISO/IEC 14143.
- **Function Points and Agile Models, How to Estimate** (Cristiane Baccarin). The author reminds us that there is a lack of consensus as to the effectiveness of using function points when developing software using agile methods and techniques. The article suggests how functional sizing could be used in combination with agile techniques. ■



Mauricio Aguiar

Message from the President

We Prefer Facts to Stories

Approximately one year ago, the three leading software measurement organizations—IFPUG, COSMIC and Nesma—got together and wrote a paper titled “We Prefer Facts to Stories.” This paper discussed the application of Functional Size Measurement to Agile development. For several reasons, the paper did not get as promoted as it deserved. As one of the authors, I think many IFPUG members would benefit from reading the paper. I have included some excerpts from the original article below.

Agile processes have brought major benefits to many businesses, such as faster delivery of software that better meets evolving customer needs. However, the freedom given to individual teams to manage their own processes has made it difficult to manage the activities across Agile teams—what we call managing “Agile-at-scale.”

To be specific, Agile metrics, such as Story Points, may be used by individual teams to manage their own affairs, but are very little help for the tasks of planning and monitoring progress across teams, for understanding performance and whether it is improving or not and for estimating future investments.

Senior management is responsible for setting budgets and allocating resources optimally 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 that only uses typical Agile processes in which 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.

One needs to understand the challenges that management faces when confronted with the limitations of Agile metrics. There are simple but effective and long-established International Standards Organization (ISO) standard software measures that can fit seamlessly into Agile processes to enable managers to estimate and control Agile delivery at scale. This can be achieved without the need to change any of the underlying Agile processes, and while continuing to obtain the benefits that Agile teams can bring in the speed and flexibility of delivering business value.

The idea of measuring the size of the functional requirements of software originated more than three decades ago—Alan Albrecht was the pioneer. Nowadays, three ISO standard software sizing methods—the IFPUG, COSMIC and Nesma methods—are used around the world to measure the requirements of all types of software, developed in all types of environments. The IFPUG method is the most widely-used method by organizations worldwide.

Any software sizing method requires actual functional requirements to be mapped to their respective model of software, which can then be measured. Requirements must be identified and measured to obtain the functional size.

As the full article states, long-established standard software sizing methods satisfy all Agile-at-scale needs. These methods measure software sizes that:

- Depend only on the software requirements, e.g. as expressed in user stories.
- May be approximately estimated from early requirements through precise measurements of delivered requirements, i.e. for code that is “done.”
- Enable measurements of real productivity (i.e. size delivered/actual effort) that can be used for future effort estimation.
- Enable objective comparisons of productivity across different teams using different technologies, etc. to monitor performance trends, etc.

MetricViews

Published twice a year by the International Function Point Users Group (IFPUG), headquartered in Princeton Junction, New Jersey, United States

MetricViews March 2019

Editor

David Herron

IFPUG Board of Directors

President

Mauricio Aguiar

Vice President

Christine Green

Treasurer

Kriste Lawrence

Immediate Past President

Thomas Cagley

Director of Certifications

Roopali Thapar

Director of

Communications & Marketing

Diana Baklizky

Director of Sizing Standards

Charles Wesolowski

Director of

International Membership

Dácil Castelo

Director of Education & Conference Services

Luigi Buglione

IFPUG Office

Interim Executive Director

Roland Pinto

Views and opinions presented in *MetricViews* articles may not represent those of the International Function Point Users Group (IFPUG).

Please submit all articles, news releases and advertising to:



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

When there is a contractual relationship between a customer and supplier, it is common for suppliers of Agile development capacity to argue for a contract based on time and materials (T&M). A supplier's reasoning would be that if the customer wants to start an Agile development before much is known of the requirements, it is impossible to bid on any other basis than to offer day rates for staff with various levels of promised skill.

However, pure T&M contracts are extremely unbalanced. A T&M contract is 100 percent safe for the supplier—s/he gets paid no matter what s/he delivers—but leaves the customer with the total budget as the only control on cost. There is no mechanism for the customer to judge whether progress payments can be justified, whether the supplier's performance is improving or deteriorating over time or whether the supplier is providing value for the money spent and for the delivered functionality.

A solution that helps balance the negotiating strengths between the customer and supplier is to contract on the basis of "price/unit size" for which the size of delivered (or developed) functionality is measured by a standard software sizing method. The customer then takes the risk on the total size of her/his requirements and the supplier takes the risk on the offered unit price. However, even for this arrangement to work, a customer will have to define some of his requirements before a supplier can reasonably bid a unit price and will remain responsible for ensuring that delivered functionality translates into business value (which is possible only when the functionality is used). Both parties would need to agree when "done" functionality can be measured and invoiced.

There is evidently a "clash of cultures" between the values of individual self-organizing Agile teams and the values and justified control needs of higher management. So, if management wants to introduce a standard software sizing method into an existing Agile development group, they must carefully take into account the Agile culture so they do not disrupt teams and risk losing the benefits of Agile processes.

You may find more information on the application of Function Size Measurement methods to Agile development in the original paper by Aguiar, Symons and van der Vliet that can be downloaded from *We Prefer Facts to Stories* on the IFPUG website. ■

Mauricio Aguiar
IFPUG President



David Herron

From the Editor's Desk

For this issue of *MetricViews*, we wanted to help raise industry awareness and promote software measurement by sharing your success stories with functional sizing. We received a number of quality articles that address how organizations are succeeding in their use of functional sizing. Thank you for your contributions.

In this issue, you can read about individuals and organizations that have created inventive ways to measure performance along with articles on the more practical application of performance metrics. There is an article that provides insight into the use of Function Points with Artificial Intelligence components. You can also learn a new formula to help you improve your development capacity or read about companies that use Function Points for greater team efficiency and scope management. These and other articles reflect the advancements

made and those continuing to be made with software measurement initiatives.

Why were we so focused on gathering and telling success stories? There are a number of reasons. Success breeds success. Success adds to credibility. Success is marketable. Success sells! We want to share our success stories with a broad audience. We have made great strides over the past two years toward improving the quality and presentation of *MetricViews* content. Our hope and desire is that you, the measurement expert, will share these articles and this publication with your peers and management. Spread the word and share the goodness. ■

David Herron
Communications and Marketing Committee



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 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 United States

Artificial Intelligence

RATIONALIZATION THROUGH FUNCTION POINTS

By AmolKumar Keote and Deepti Patil



Executive Summary

The scope of Artificial Intelligence (AI) in the world of software solutions has increased, as they enable new functionalities and often allow interactive solutions for users with problem solving skills, by automating activities that currently require human intelligence. This work is typified by recent advances in search-based software solutions, and by

long-established work in probabilistic reasoning and machine learning for software engineering. The software solution is quantified in terms of Function Point, which is a “unit of measurement” to express the amount of business functionality the software provides to a user. In this paper, we review how the AI can be rationalized through Function Point. This is achieved by mapping software functional requirements to the AI components, such as learning, reasoning, problem solving, perception

and language-understanding. Furthermore, Function Point for the mapped requirements are converted into Natural Language Processing (NLP) for AI implementation. AI solutions can be optimized and potentially reduce the cost/efforts of the software development using this mapping.

Introduction

In the everchanging IT industry, software systems continue to grow in terms of size and complexity. They are becoming more and more difficult to understand. Improvement of coding tools allows software developers to produce large amounts of software to meet an ever-expanding need from users, but these continuous changes also lead us to technical debts.

Technical debt is the amount of capital it takes a company to fix the defects and warnings to make them fit for today's business environment. It is also the effort needed to maintain software quality at acceptable standards of risk, performance and agility. This also leads to an increase in the use of AI as machine-learning algorithms make predictions or decisions based on data. These learning algorithms can be embedded within applications to provide automated AI features or be used in an AI platform to build brand new applications. With incredible advances made in data collection, processing and computation power, intelligent systems can now be deployed to take over a variety of tasks, enable connectivity, enhance productivity and fix the vulnerabilities.

The software effort estimation is done on the basis of Function Point methodology, which measures the amount of business functionality delivered to the user. Further, cost/effort for the software is estimated using this Function Point. Now that AI is being introduced in the software, the Function Point and effort estimation approach needs to be updated to incorporate the flexibility and components provided by the AI.

“These learning algorithms can be embedded within applications to provide automated AI features or be used in an AI platform to build brand new applications.”

Functional Size and Technical Debt

Technical debt is a concept in software development that reflects the implied cost of additional rework caused by choosing an easy solution now instead of using a better approach that would take longer. Technical debt can be

compared to monetary debt and it can be caused by multiple factors, like due to deadline crunch, and sometimes the proper delivery approach is not followed for quick delivery of product to the market.

The amount and cost of technical debt keeps expanding due to its progressive nature:

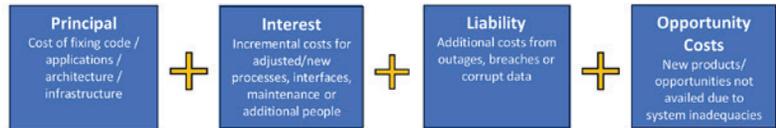


Figure 1: Technical debt nature

Software requirements represent a product's business value and quality goals. Different phases of software development can be potential sources of technical debt due to various ambiguities, such as the requirement phase, design phase, build phase, etc. One of the problems arising due to an increase in technical debt is that the scalability of the software is affected. Modifications in the software requirements are also not easy due to an increase in technical debt. Hence, we can define the relation between the requirements and technical debt as:

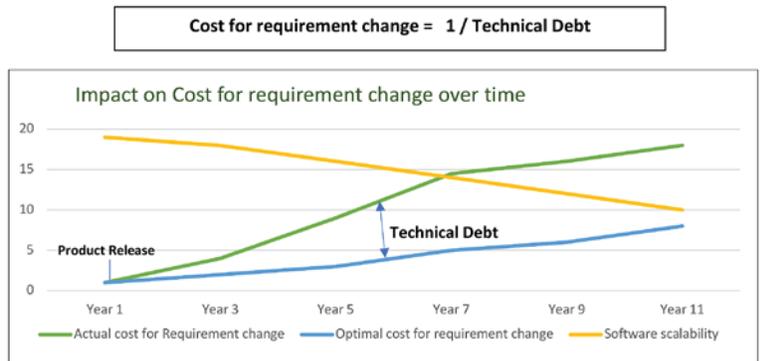


Figure 2: Impact on Cost due to change in requirement over time

AI for Reducing Technical Debt

Technical debt can be reduced by fixing issues such as bad design, incorrect requirements, poor coding standards, etc. by manually using human intelligence or AI. Additionally, multiple tasks can be automated to fix the software vulnerability and repayment of technical debt using AI.

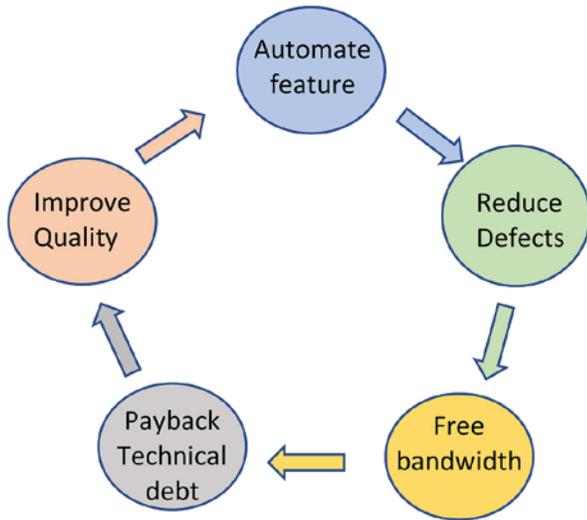


Figure 3: AI used for reducing Technical debt

As shown in Figure 3, by automating some of the component using AI, we can reduce the number of defects and save efforts spent on the maintenance. The additional bandwidth saved on the resources can be utilized to repay some of the technical debt and thereby improve the overall quality of the software. An AI script was written in Java and mainframe to validate the above hypothesis by fixing some of the common critical violations in a software. Here is the impact analysis:

For Mainframe:

	Total Quality Index	Robustness	Security
Before AI fix (script)	2.19	1.89	1.73
After AI fix (script)	2.4	2.26	2.4
% improvement	9.5%	19.5%	38.7%

After running the AI fix script, the total quality of the software increased along with the robustness and security, as per the CAST (software quality measurement) tool.

For Java project 1:

	Number of Defects	Total Quality Index	Technical Debt
Before AI fix (script)	1636	2.18	\$ 3.3 Million
After AI fix (script)	1167	2.51	\$ 3 Million
% change	29% reduction	15.1% Improvement	9% reduction

For Java project 2:

	Number of Defects	Total Quality Index	Technical Debt
Before AI fix (script)	122	2.75	\$ 1.2 Million
After AI fix (script)	95	2.76	\$ 1.1 Million
% change	22% reduction	1% Improvement	8% reduction

For Java project 3:

	Number of Defects	Total Quality Index	Technical Debt
Before AI fix (script)	257	3.33	\$ 511 Thousand
After AI fix (script)	23	2.94	\$ 277 Thousand
% change	91% reduction	12% Improvement	45% reduction

After running the AI fix script, the total quality of the software increased and technical debt along with the number of defects reduced, as per the CAST tool.

Please note that the data used above is for representation purposes only.

Function Point in AI

Function Point analysis is a structured technique of classifying components of a system by breaking it into smaller components for better analysis, understanding and problem solving. In Function Point analysis, systems are divided into five large classes and general system characteristics. Now that the AI has been introduced in the software system, the requirements are further divided into AI components before carrying out Function Point analysis.

Following are the AI components:

1. Learning: It is distinguished by several forms and one of them is by remembering the successful move.
2. Reasoning: To draw inferences appropriate to the situation in hand.
3. Problem Solving: This is divided into two categories, special-purpose and general-purpose.
4. Perception: Surroundings are scanned by means of various sensory organs.
5. Language Understanding: Mapping the given inputs in natural language into useful representation.

The functional requirement for the software system can be categorized under AI components as follows:

- Break down the requirements in natural language by understanding and converting them into less ambiguous requirements.



- Create a module to develop a knowledge base for the system using knowledge-based requirements and domain problems.
- Use Function Point methodology to size the requirements that are incomplete and incorrectly prioritized.
- Characterize the requirements that have formal specifications such as:
 - o Expressiveness: The requirements semantics which cover enough details to map real-world descriptions.
 - o Readability: Requirements which have specifications and validation details.
 - o Structuring: Requirements which contain an end-to-end mechanism for successful completion of tasks.

Function Point sizing for the updated requirements can be performed after the requirements are converted into AI components. As the number of functional requirements are reduced due to the introduction of AI, the Function Point size for the software will also reduce, leading to cost savings in terms of software development and maintenance.

Following is the comparison of the Function Point size between projects before introduction of AI versus the projects after the introduction of AI:

Project Name	No. of Requirements	Function Point Without AI	Function point With AI	Changes (%)
Project 1	100	528	496	6%
Project 2	120	667	600	10%
Project 3	150	794	661	17%
Project 4	180	1,162	906	22%
Project 5	200	1,225	858	30%

“AI is a constellation of technologies that enables machines to act with higher levels of intelligence and emulate the human capabilities of sense, comprehension and action.”

Please note that the Function Point size depends on the complexity of the requirements; hence there is no relation between the number of requirements and Function Point size.

Based on the above sample data, it is safe to say that the relation between the Function Point before AI is introduced to the system and Function Point after introduction of AI is as follows:

Function Point size before AI > Function Point size after AI implementation

AI and Function Point Metrics

The main premise of redefining the AI metrics is because of the AI approach. AI is a constellation of technologies that enable machines to act with higher levels of intelligence and emulate the human capabilities of sense, comprehension and action. Thus, software can actively perceive the solution approach by continuous learning and problem solving. The natural language processing and inference engines can enable AI systems to analyse and understand the information collected. An AI system can also learn through technologies such as expert systems and inference engines or undertake actions of users. These human capabilities are augmented by the ability to learn from experience and keep adapting over time. Irrespective of the type of AI being used, however, every application begins with large amounts of training data. In the past, this kind of performance was driven by rules-based data analytics programs, statistical regressions and early “expert systems.”

Software effort estimations follow traditional Function Point metrics containing the parameter to define multiple aspects of the system solution:

- a. Development hours/FP (productivity)
- b. Defect ratio
- c. Maintenance hours/FP
- d. Total effort hours

Following is the sample traditional metrics (average) for the software solution based on Python:

FP	Total Hours	Development Hours/FP	Defect Ratio	Maintenance Hours/FP
100	1,368	13.68	6.08	8.61
250	3,875	15.50	3.37	6.95
500	8,655	17.31	3.52	7.44
1000	20,277	20.28	3.69	8.15
2500	70,454	28.18	3.94	10.05

Now that we are implementing the software using AI, our traditional Function Point matrix needs to be modified. Based on the process we defined previously to calculate the Function Point sizing for the AI project, following are the impacts on the traditional software metrics.

- a. Total effort hours for AI implementation
- b. AI development hours/FP (productivity)
- c. Defect ratio
- d. AI maintenance hours/FP

AI FP	Effort Hours	AI Development Hours/FP	AI Defect Ratio	AI Maintenance Hours/FP
94	1,432	15.23	5.27	7.72
237	3,950	16.67	3.35	6.63
446	8,554	19.18	3.5	7.34
834	19,106	22.91	3.66	8.30
2,368	66,733	28.18	3.89	8.35

We can also call the above next-gen FP metrics as the entire effort and software solution is designed using AI.

If we compare the data from both, the metrics, i.e. traditional metrics and next-gen FP metrics, we can see that there is significant reduction in the Function Point size, total

effort hours, hours per FP, defect ratio and maintenance per Function Point, which lead to the cost and effort savings with the software system.

Conclusion:

Software engineering remains a highly-skilled human-intensive activity and relies on problem-solving skills of human knowledge and experiences. Therefore, AI, expert system and knowledge engineering will continue to play a major role in automating numerous software development activities. The interplay between AI and software engineering is significant and it makes sense to take advantage of their mutual strengths. There is much scope for exploring and evaluating the scope on rationalization of AI using Function Point in the software development lifecycle in the future. Also, there is scope in refining the metrics for the AI based on updated Function Point approach for saving the effort and cost in the software development along with increased scalability and quality of the software. ■

References:

- *International Journal of Advanced Computer Engineering and Communication Technology*

About the Authors:



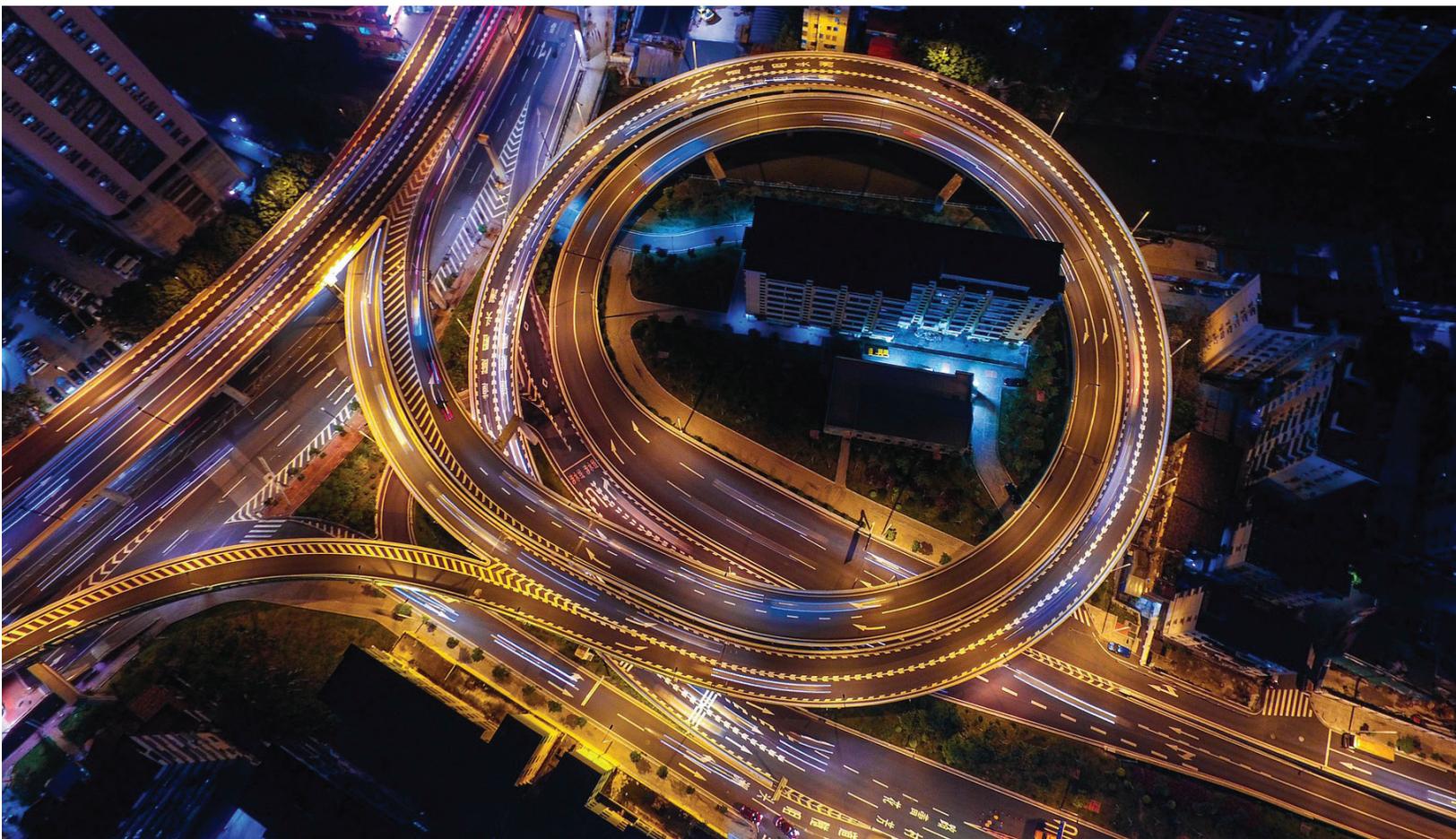
Amolkumar Keote has 15+ years of post-qualification experience in RFP/RFI and deal support, System Analysis, Solution Design, Functional Design, Application Development, Project Management, Consultation and implementation. He is a Certified Function Point Specialist (CFPS) and Agile Certified Scrum Master. He is ITIL V3 Certified.



Deepti Patil has 9+ years of experience in Function Point Analysis, RFP/RFI and deal support, System Analysis, Functional Design, Business Analysis, Product Management and Data Analysis. She is a Certified Function Point Practitioner (CFPP) and experienced in Software effort estimations and applications including: IFPUG Function Point analysis, Gartner Function Point analysis, Parametric Estimation & Reconciliation and Baseline and productivity measurement using Function Point for projects.

When Strategic Software Metrics are Even More Important than Financial Metrics

By Antonio Ferre Albero



Perhaps too many times companies (small and not-so-small) focus too much on financial metrics and financial numbers. EBITDA (Earnings Before Interest, Taxes, Depreciation and Amortization) is probably one of the most sought after pieces of information by CEOs, company owners and shareholders. In short, they want to know if the company is making a profit or experiencing losses. We could also provide other interesting information, such as cash flow, share price evolution and historical trends.

It is not news that financial metrics are vital information

for any company. We can use the aforementioned EBITDA, cash flow or 10-Q/10-K forms (in the United States) in the case of a large company to create a 400-page quarterly financial report or to deliver a few-page report in the case of not-so-big companies. With all of this information on hand, we will have an interesting overview of the company's financial health and evolution. All companies manage these financial metrics with standard concepts and without complex ways of creating the information. Even though accounting systems can be different depending on the countries where they are located, the main concepts will be the same.

“The main focus might be put on the product—or service—delivered; the process helps on that mission, but it is not the mission itself.”

Financial results, just consequences

Nevertheless, how does this link with measuring software size, functional sizing or functional sizing success stories?

Financial results are just consequences of other things, with strong dependencies on the value of the product received, the product innovation or the price. Obviously, financial results are also dependent on other factors: company service, brand reputation, company management and expenses, market competitors, commercial networks, marketing and even the CO₂ impacts in the production process.

In a factory, the raw material and production costs are perhaps the most important expenditures -obviously not the only ones- and the final product is more tangible. To reduce the personnel costs by including machinery and robotics (if possible) is essential to produce a more competitive product.

In the case of the IT industry, the personnel costs are key expenditures, and the product is very dependent on people,

being that it is not easy to create software in an automatic way. It is a manual and artisan industry, despite that different software components or pieces can be reused, not having to reinvent things already established (such as Open Source Software (OSS) components) or creating software that can be reused.

Software as a product is much more, apparently, intangible than other products, such as cars, tables, houses or laptops.

Obviously, it is not needed to remark about the differences between products, processes and projects, or that a customer (internal if you are an IT department or external if you are a software company) receives a software product but not a project or a process. Sometimes, in the IT industry there seems to be a trend to put more focus on process (or even projects) concepts to build a product than on the product itself. The main focus might be put on the product—or service—delivered; the process helps on that mission, but it is not the mission itself.

Sometimes, companies manage dozens and dozens of metrics, charts and fascinating presentations based on the project and on the process concepts, and just a few of them based on the product, on the value it offers, on the size, on the quality or on the productivity, which has a strong relationship with the time to market or with the competitiveness, amongst others. It is essential to measure the software product with standard methods.



“It is essential to have always in mind the differences of being good at managing projects and following processes and being good at creating IT products.”

It is important to mention that models, such as CMMI (Capability Maturity Model Integration), are mainly focused on the process and on the project level, and less on the tangible product. The Measurement and Analysis (MA) CMMI process area initially has this focus, despite including business and organizational levels: “Measurement activities should support information needs at multiple levels, including the business, organizational unit and project to minimize rework as the organization matures.”

However, when creating software it is important to put the focus on a set of aspects:

1. Software size as an isolate measure does not make sense.
2. Software size is the base for the most strategic metrics.
3. The customer receives a product, not a task finished.
4. The customer receives a product or service, not a project.
5. Too many times KPI concepts are used with fragile information or sometimes are more focused on creating information, accomplishment of SLAs, nice charts or evolutions, than on strategic conclusions.
6. Taking base “project” concepts (such as effort or time) instead of “software product” concepts (such as size or quality) provide incorrect conclusions from the point of view of the product.
7. To associate “competitiveness” with the fact that the company is earning (or not earning) money with projects can be a fragile conclusion and short-term vision.
8. Accurate, key and sincere strategic information are needed, much more than traditional accounting information.
9. A company or department can be extremely good in the process, managing projects, having projects with profits, but the products cannot be as good as desired or expected. It is essential to have always in mind the differences of being good at managing projects and following processes and being good at creating IT products.
10. To provide high-quality software products with coherent and strong relationships among quality, price and value for customers might be core objectives.

Different articles can be created to focus on the above-mentioned ideas, despite that almost all of them are covered by the “common sense” umbrella, and might be taken into account day by day.

Answering strategic questions

While that theory is already well known, I propose a kind of self-assessment. Taking into account that your activity (and product) is to create software solutions (because you are an IT company or an IT department having your own company as an internal customer), have you recorded and do you have—numbers on hand—metrics that answer the following questions?

“Are you better, and by how much, with regard to quality and productivity than your competitors?”

- What is your software development productivity?
- What is your development productivity compared with the market?
- Has this productivity improved or is it worse than previous years, and by how much?
- What are your historical trends and future projections?
- What is your software quality?
- How is your software quality compared with the market?
- How does multi-site development affect your productivity and your quality?
- How does project size and product size affect the productivity and the quality?
- How does the team capability and experience affect the final product?
- What is your rework ratio?
- What are the historical trends of this rework ratio?
- How is this productivity and quality different for different customers, domain areas, geographical areas and technologies?
- How have initiatives such as CMMI, Lean or Agile, contributed to the IT products, especially to creating better products and faster?
- How have those methods contributed to being more competitive in the market?
- What has the ROI been after implementing them?
- Are you better, and by how much, with regard to quality and productivity than your competitors?

We might take into account that almost all of those questions might be answered based on numbers stored and compared (not on words), and not based on perceptions or just with financial numbers as conclusions. To conclude that we are a



brilliant company just because we have a positive EBITDA is not enough, and can be volatile short-term results.

On the other hand, being honest with the numbers and avoiding any kind of “cooking the books” might be “a must” to detect the reality and improve the day-to-day operations.

With those above questions answered, it would be possible to feed regularly the estimation process with multi-level axis information based on recent projects, developments and experiences, bringing the company’s estimation process to a more mature level.

We might take into account too that almost all of the questions mentioned above need the software size measure, and it will be the main basis for the conclusions. As in the case of a company that produces shoes, it is essential to have recorded metrics of the number of shoes (or amount) produced yearly and even the profit by unit. In the case of software, we need to have the amount of software produced, not the hours invested in producing this software. Easy but not always applied.

“Amount of software” is the key concept of this article. Do you know the amount of software created in the last year by the IT company or by the IT department? And, take into account that “amount of software” does not mean person

hours, person days or financial incomes. Amount of software means “amount of software,” and it is as easy as that!

Harmonising quality, productivity and product costs: metrics, knowledge and answering why

Software size is the basis used to calculate the effort and schedule, and in some cases or countries this size is transformed directly into price, despite that a set of drivers might be applied, such as if the software is highly critical, possible time constraints or reused ratios and drivers.

For the amount of software, or software size, it is needed to measure the software, and here is when the “Functional Size” comes into the picture. What is the size of the software received by the customer?

And yes, it is true that the Functional Size is not the only method to measure the software (others techniques include Story Points, Use Cases, T-Shirt, Statements, Backfiring, based on the number of software elements, or ad-hoc techniques), but we can say that it is the most universal, ISO/IEC standard, and that fits the higher number of purposes without being effort dependent. Therefore, it is not dependent if the solution has been created in a more clever way or with more-than-

necessary physical software code, in one project or divided in 12 projects or using a concrete method.

In intangible software products, to size the product is even more necessary than in the tangible products, such as the “shoes” example mentioned above. With this measure in place, and combining many different numbers, we will have a set of strategic and fascinating metrics to fulfil objectives such as a harmonization between quality, productivity and product cost, measuring and knowing how competitive the company (or IT department) is in the market and the answer to the most important question: why?.

“How can we answer those apparently easy but vital questions (such as product/value, product/quality or product/price) if we do not measure the product itself?”

To answer this “Why” question, again numbers on hand, can derive in strategic conclusions, and sometimes even in urgent needed actions, perhaps far away from the ones that different people had in the mind.

Even if yours is not an IT company or an IT department, it is not news that Information Technology in this century is key in all of the industries, having the CEO strong dependencies with the CIO (Chief Information Officer), sometimes addressed towards a “Chief Efficiency Officer” role. The role of Information Technology has changed completely in recent years, no longer being considered a service but an engine of many economic sectors.

To measure this IT engine might not be an option but

something essential. To measure the IT product created might be for IT companies and departments one of the highest strategies, as mentioned, to provide the best product and the best value at the best price. However, how can we answer those apparently easy but vital questions (such as product/value, product/quality or product/price) if we do not measure the product itself? As easy as that!

Sizing the product and having concrete metrics and answers to the questions raised in this article is essential. After that, financial metrics, financial numbers and EBITDA will be seen in a different perspective, just a consequence of a set of the mentioned strategic metrics that have almost all of them the size of the IT product as a basis. Again, as easy as that! ■

About the Author:



Antonio Ferre Alberó (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 a 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.



NUCLEON:

HOW TO BETTER ASSESS YOUR DELIVERY CAPABILITY IN DEVELOPMENT TEAMS

By Jeppe Hedaa



To most people, their company's IT department is a black box. This is especially true of complex systems development settings in large corporations, where hundreds or thousands of IT specialists work together to serve the organization.

“What is needed is to remove the mystery and open up the dialogue in a way that strategically aligns the business and IT organizations.”

It is a black box in terms of mystery and complexity. Those outside the department probably don't know what happens within IT's walls or have a clear understanding of how it impacts their day-to-day work. What they do know is that the product they so desperately need is often delivered past its deadline and that IT continually asks for additional resources to manage this mystery and complexity.

This collision of demands results in frustration, internal hostility and an unproductive environment. What is needed is to remove the mystery and open up the dialogue in a way that strategically aligns the business and IT organizations, and which creates a common focus on prioritizing efforts and improving performance.

Over the years, our team researched hundreds of IT disciplines, reviewed academic research, best practices and applied empirical studies in causal thinking. We worked hard to identify the individual factors that lead to high performance in IT development, and that research has led to the creation of Nucleon™, a formula that gives you a task-independent performance number and shows the optimal way to improve your IT development capacity. It is essentially to an IT development team what horsepower is to a car.

Nucleon illuminates the black box of IT. It provides a way to focus on the causal drivers of performance in application development.

The Nucleon formula was developed because there was a need for a consistent tool that would indicate how much development power a team or a company has. For me, the Nucleon calculation is just as relevant as the price-to-earnings ratio (P/E) or return on investment (ROI) and perhaps even more so, as the IT department determines most companies' future success.

The Nucleon formula is based on two key principles:

- Great people on your development team increase your performance score
- Poor organization and/or a highly complex IT environment reduces performance

“Top performers make a remarkable difference in an IT department’s performance.”

The People Factor

When considering IT performance, the most important aspect is people. The human element is by far the most performance-generating factor in your entire setup.

Nucleon assesses people by assigning a numerical category from 1 to 10: the lowest-level performers are 1s and the top performers are 10s. Our research in performance revealed, conservatively, that 10s, who are the top 1 percent of all IT professionals in a normal distribution, perform at a level that is almost 2x greater than that of the next level’s performers who we call 9s and who typically represent the top 3 percent of a population.

Going further down the line, 10s perform at a level that is 20x greater than that of an average IT specialist. 10s are at a level that is 100x greater than that of individuals classified as 1s. The performance curve is exponential. Allow me to substantiate further:

- Twitter’s former CEO Dick Costolo said that he believed the top 1 percent performs 50-100x better than the average developer
- Apple believes the top 1 percent performs 25x above the average specialist
- Google beats them all by claiming that the top 1 percent performs at a level 300x above the average developer

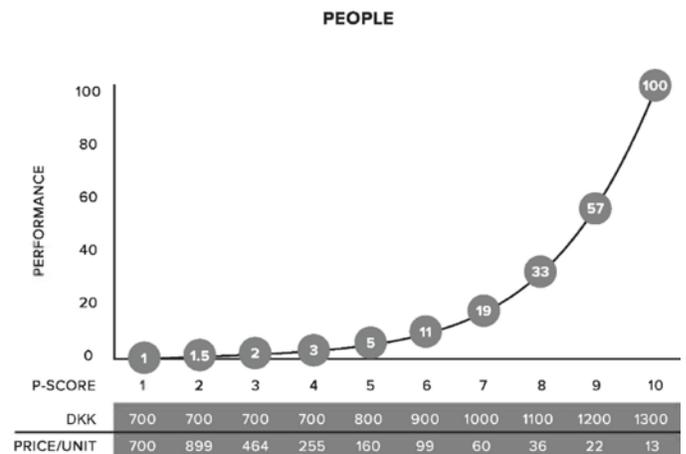
There are many opinions about this. Though the numbers may differ, the bottom line remains the same: regardless of the multiplier you choose, top performers make a remarkable difference in an IT department’s performance. In the Nucleon formula, we use a more conservative multiplier that suggests a 10 performs 20x better than an average employee.

By changing the focus from input cost (salaries or hourly rates) to output cost (cost per delivery unit) and striving to have as many 10s on your development teams as possible, you will make a huge difference in helping IT succeed.

For example, the chart below reflects actual rates for freelance project managers in the Danish market, combining their fee with how much they delivered using the 20x performance factor that our studies confirm as a conservative reality.

$$N = \sum_{i=1}^k P_i \left(\frac{1}{f(O) + f(C)} \right)$$

DEVELOPMENT EFFECTIVITY
THE PEOPLE FACTOR
THE ORGANIZATIONAL FACTOR
THE COMPLEXITY FACTOR



The highest paid project managers (earning 1300 DKK (Danish Krone) per hour) actually cost around 13 DKK per

delivery unit, while an average project manager with a rate around 800 DKK per hour in the Danish market would cost 160 DKK per delivery unit. A project manager who is truly a poor fit but who represents himself as an IT professional would never charge less than 700 DKK per hour—and in this example, would cost 700 DKK per delivery unit.

This means it is 54x more expensive per delivery unit to pay the lowest salary than to pay for a top performer. Acquiring as many 10s as possible will change your focus from cost-incurred to value-delivered and will not only lower your total cost of ownership, but also increase your total performance.

The numbers we assign do not indicate a person's potential, and they are not an indication of an individual's overall value. They are a snapshot of an individual's current IT performance. An employee might perform much better in a different function or environment.

The Organizational Factor

Strong performers are a necessary start toward improved IT performance, but they are only as effective as the organization in which they work. Poor organization drags down even top performers.

Nucleon defines four universal areas related to organization that greatly impact IT performance:

- Team Size
- Bureaucracy
- Decision-Maker Proximity
- Spillover

Team Size: Through studies of thousands of comparable projects, the conclusion is that larger teams have up to 48 percent lower performance than smaller ones.

One study analyzed 564 comparable projects in which 100,000 lines of equivalent code were written. The projects were divided into two groups of either small teams with under five people or large teams with more than 20 people. Though the larger teams finished the job six days earlier than the smaller teams, they cost 7.3x as much.

Bureaucracy: Power struggles, jungles of paperwork, and overly-complicated approval processes from administrative staff can cause teams to lose as much as 20 percent of their potential performance.

There's no question that all organizations need administrative staff, but as soon as these support teams take on a life and agenda of their own, things start to go bad. The core organization loses motivation, and teams lose productivity.

That, in turn, can frustrate IT teams to such a degree that they'll eventually leave for a better environment.

IT bureaucracy should be optimized with only a few, simple rules as guidelines:

- **Autonomous:** Each team decides how the group will work, and responsibilities are clearly defined.
- **End-to-end knowledge:** The team should possess all knowledge required to deliver a working product.
- **Communicating:** Development is all about close collaboration. Ideally, the entire team would be sitting in the same room so that there are no barriers in communication.

- **Dedicated:** Every member of the team should be assigned to the project full-time because any distraction will only delay work.

Decision-Maker Proximity: Not having the decision maker (the person who calls the shots regarding final product features, layout, etc.) close to the team significantly hurts performance by up to 32 percent. This is due to a rise in idle time, confusion and communication lag.

One of the worst side effects of an absent idea-owner is that the IT development team experiences too many waiting cycles. I call these cycles “black holes” because they suck all of the energy out of the team. Even though the team can normally make non-problematic decisions on their own, there are many issues in a development project that require the business owner’s attention. For instance, the initial design phase of a project is extremely demanding and requires lots of dialogue; it is of the utmost importance that the team is working toward the goal that the business owner had in mind.

The business owner doesn’t have to be engaged full time, but they do need to be readily available.

Spillover: The measure of the fact that great people lift those around them, and vice versa.

Everyone in a given team is affected by their teammates and can produce up to 3x their individual potential or 1/3x depending on their teammates’ performance levels.

I was trained to think in talent pyramids: put a 10 at the top of a team and then staff it with a lot of less-talented performers. This is the way it has always been done in big IT organizations, as well as most other industries. The belief is that you can lift an entire team of mediocre people to a higher level by placing a 10 in their midst to lead them.

Our experience and research reveal that great performers deliver at a completely different and far superior level when they are put together with other great performers.

When placed on a team, a person who is a 10 will lift another 10 to far beyond what could have been achieved individually or when placed among those of lesser abilities. Though it’s true that the same 10 can be placed amongst 5s and make them more efficient, the 10 will be negatively affected by working with a group that has a lower overall rating.

The Complexity Factor

An organization’s complexity must be minimized to give team members the greatest opportunity for success.

There are four IT-related complexity areas that Nucleon identifies as being critical to performance:

- Culture

“The IT organization is always the target of critique when the systems don’t reflect the solution that business had in mind.”

- Method Maturity
- Legacy
- Architecture

Culture: The environment in which your team operates and interacts. Poor team culture can reduce performance by up to 17 percent.

Culture is inherently difficult to measure, as it is often considered an intangible phenomenon, but many scholars and business professionals have created tools for assessing and measuring it, such as the HBS Culture Profile and Vega Factors Total Motivation (TOMO).

Two important patterns have emerged: cultural alignment accelerates positive culture, and positive culture improves performance. Cultural alignment is measured as the share of employees and departments that have the same overall purpose. This is also a great indicator of overall performance because it correlates with levels of employee engagement and customer orientation.

Nucleon examines six empirically proven cultural performance indicators:

- Play – the feeling of having fun doing your job.
- Purpose – when your work contributes toward something meaningful to you.
- Potential – when the job is an active asset towards achieving your goals.
- Emotional Pressure – when emotions are forcing you to work.
- Economic Pressure – when you work solely for financial reasons.
- Inertia – when there is no good reason why you work.

Method Maturity: There are three main method-related performance effects; agility, practical experience and team consistency. Not being agile and not having the practical experience within the team can reduce your team performance up to 51 percent.

Methodology is a rigid area of competence, and that’s probably because IT development is so difficult. In the end, the IT organization is always the target of critique when the systems don’t reflect the solution that business had in mind. As a result, IT people are very conservative as to what the best development approach may be.



In general, the choice of methodology should be the same throughout your organization, but the most important thing is to have the same knowledge and methodology within an individual team.

Legacy: A team's share of systems that are not of strategic value to the company's IT infrastructure have a performance drag of up to 18 percent.

Generally speaking, existing systems are not altogether terrible. Businesses live and thrive using these systems, but they may be taking up far more resources than you realize.

Old legacy systems affect teams through a maintenance burden and the difficulty and expense needed to implement new systems. If your business functionality or data are not easy to access, it will drag performance out of an otherwise straightforward integration project.

Architecture: The architecture score of a team is a product of the quality of their APIs, the degree to which the architecture allows for reuse, and how well the teams are future-proofing their applications in development. Poor architecture reduces a team's performance by up to 12 percent.

Architecture is the most basic and important function of all the support staff areas in IT. If the architecture department doesn't work well, all projects suffer.

Architecture should be easy for developers to navigate, and so too should choosing approved technology stacks for projects. Requirements for interfaces should be simple to understand and the environment should encourage reuse of existing stubs and code. It should also discourage development of proprietary interfaces. A strong enterprise architecture will help you avoid redundancy and improve quality, availability and sharing of data. It will save significant expense.

This performance view does not isolate architecture to a measure of documentation or process rigidity, but instead looks at how well the architecture is enabling fast production and ensuring that all development carried out will have an accelerating effect on future work.

Conclusion

Let us derive realistic performance numbers so that it becomes clear when things have gone badly or if your team is

performing brilliantly. That's why we need tools to assist us in getting a realistic and valuable picture of our collective performance level. That's the idea behind Nucleon.

People are the core of anything you want to do in IT. You need to identify the individuals who have a realistic chance of delivering high performance, and you need to match them perfectly to the jobs they are best suited for. Focusing on identifying the best people and attracting them for the right position will guarantee you enormous success in your development efforts, project after project. I promise you.

Organization provides a foundation upon which your people can deliver. Small teams, little to no bureaucracy, and a decision maker that is close to the team make the best possible foundation.

Complexity can slow down your team's time to delivery and the quality of their work. Good culture, agile methods, few legacy systems and intelligent architecture reduce the barriers to your team's success.

The idea is to shift the focus by looking at what delivers results.

Nucleon forces organizations to look at the areas that they are not thinking about and mathematically assess how they affect their performance. To not operate on "gut feelings," but to agree on real facts that determine our reality. To know that these things are right, let's calculate the numbers and either verify or disprove what we might have intuitively thought.

When hard numbers enter the equation, you can have robust discussions on what is right and wrong. This is healthy for any organization. "How much does this mean to us? How important is this? Are there other areas that should be more important for us to address?" It's not "if we should address it," it's "what to address, and how important is it?" With Nucleon, we can prioritize and plan rationally.

You can learn more about Nucleon, or take the Nucleon assessment, at nucleonformula.com. ■

About the Author:



Jeppe Hedaa has been working with complex systems development for more than 30 years, serving the largest IT development departments. He is the CEO and owner of 7N, an agent for top 3 percent IT specialists. 7N has departments in the United States, Switzerland, Finland, Sweden, Norway, Poland,

India and Denmark. In September 2018 he published the book "Nucleon: The missing formula that measures your IT department's performance," in which he describes how to calculate a hard number for an IT team's performance that could best be compared to that of horsepower in a car.

NUCLEON AND FUNCTION POINT ANALYSIS

By Christine Green

Editor's Note:
This article is a companion piece to Jeppe Hedaa's article.

When Jeppe Hedaa states “that the product they so desperately need is often delivered past its deadline and that IT continually asks for additional resources to manage this mystery and complexity,” I can totally relate.

The book *Nucleon*¹ was published a couple of months ago, and in the book the author, Jeppe Hedaa, has presented a formula that can calculate how well a development team is performing. By adding up the individual developers and by correcting for how well they are organized and how complex their setting is, he manages to calculate a hard number for each team—like horsepower in a car. Jeppe's goal when talking about Nucleons is to open the door of visibility to the team's performance towards management—and assess the impact to productivity. A key factor in reaching this goal is to help the management in large organizations by making sure that they understand the performance level within their operation.

My dream too is to open a door of visibility to management as well, but my focus is related to the scope of our software projects and portfolio. Throughout my professional career, I have seen more projects be late or even cancelled than projects succeed within the time and budget initially set out. This pattern, perhaps counterintuitively, seems to be completely independent of the competence and effectiveness of the team. It seems that regardless of how effective your team is—if the scope is poor, uncontrolled and untraceable, the project will still be a mess. If they work toward unclear or wrong requirements, they will most likely fail—not because they did a bad job, but because the scope was not controlled and managed correctly. I believe the combination of a strong and efficient team measured by Nucleon and good scope management using Function Point Analysis are both paths to future successful and cost-effective IT projects and among the greatest challenges for strategic alignment with IT management.

Talking with Jeppe about his approach when developing Nucleon gave me a key insight into how to address this challenge. Jeppe has an outstanding understanding of how the

minds of management work. Function Point Programs have often failed due to the lack of understanding from management. I have experienced that the only thing that management sees in Function Point Analysis is the end result—the “magic” number. This singular focus combined with failed projects that do not deliver on time or budget often leads to a very critical view of Function Point Analysis—what is the use, and how is the investment justified?

When reading about Nucleon the first time I got very excited. First of all, Nucleon addresses a key topic we all talk about but usually avoid measuring—the team’s performance and ability to deliver high quality with high speed. Secondly, Nucleon operates through the lens of management and provides tangible measures with great simplicity. The way that Nucleon targets the management of an organization rather than the team itself, prompted me to ask myself: could we do the same with Function Points? Could we define an approach that illuminates the understanding of the scope in any given application with the same simplicity and ease? The answer, I believe, is yes!

By design, we have the illumination of scope in Function Point Analysis (FPA), but we seldom use it to educate and enlighten the same audience as Nucleon targets—management. Perhaps, to get the attention and time required to utilize IFPUG Function Point Analysis², we need to elevate our focus when sharing it—making the information relevant for decision makers.

“First of all, Nucleon addresses a key topic we all talk about but usually avoid measuring—the team’s performance and ability to deliver high quality with high speed.”

Doing so, also requires that we incorporate the perspective of the management in our analysis—linking the scope directly to the business problems. For this purpose, we don’t need every single detail, or screen—we need just enough information to determine scope, define the cross-operational trade-offs and barriers and enable strategic dialogue between the business and IT. So... one could ask “What is just enough information?”

For the purpose of this article, let’s use the feet analogy; the higher you go, the less detail and more assumptions you need to make in order to understand what you see. It is important to remember that zooming out might mean a loss in detail but delivers another perspective than when you are close to the ground. If you imagine you do this with Function Points Analysis, you can always drop down to the lower level to verify individual assumptions. And you can always spin up in the air and look at it from a distance to see if you are going in the right direction.

“For this purpose, we don’t need every single detail, or screen—we need just enough information to determine scope, define the cross-operational trade-offs and barriers and enable strategic dialogue between the business and IT.”

Flying high with Function Points Analysis. I am nerdy enough to love it.

10 Feet – you can see all the details

The full Counting Practices Manual³ describes the rules, the definitions and all of the details about performing Function Point Analysis⁴. Function Point Analysis includes identification of DETs, RETs and FTRs—things like number of fields, files etc. When you do that you are “10 Feet above the ground” and very nerdy.

This is a level of detail that is required, when we want to analyze the scope down to the level of +5 percent of accuracy. Unfortunately, I have often seen this 10-foot perspective used when the size measure was purely used for the ballpark estimation, benchmark productivity etc. That is in my mind not an effective use of Function Point Analysis. We should actively use the detail for test management and requirement traceability during execution of critical projects. This granularity is the information that the project manager and test manager should be using actively in their planning, tracking and execution of both project development and test.

“Lack of good scope management has for years been called out for being the biggest issue in failed projects.”

I completely agree with the input from many who learn about Function Points for the first time. It is time consuming, it is costly and all of the other comments about why Function Points are not used. What I do not agree with is that it is useless. It should be used to ensure that a project never fails on scope quality, verification, validation and delivery. At this level, it is in my opinion that you have the best Functional Requirement review process—the ultimate scope management process. Lack of good scope management has for years been called out for being the biggest issue in failed projects. Requirement traceability could easily start with the Function Point Analysis. It would be objective, clear and follow an agreed process. It would break the business requirements

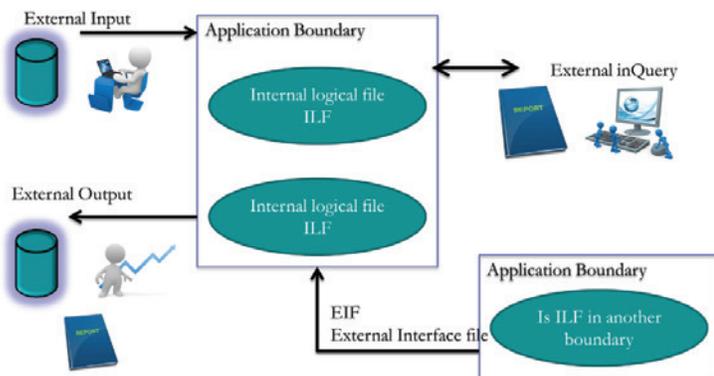
down to transactions and data elements that could clarify what needs to be done, what has been done and even if you choose to do so—what has not been implemented in the software that is undergoing the Function Point Analysis.

Most projects, if they have a scope change of 5 percent, would require an action to estimate cost, price and resources. Most projects need to have an early warning when the scope is changing. It could be a re-planning of the scope—decrease in the original scope in order to meet and include the new and more important scope that has been identified. Regardless, actions need to be taken. Unfortunately, most projects today do not have this level of understanding of the size of their scope. Function Point Analysis can provide that in a simple, easy and trackable way. Scope management will be objective, measurable and consistent with Function Point Analysis.

100-500 Feet - just enough to understand

The 100-500 Feet perspective is when we simplify the Function Point Analysis just a little bit to give it a twist. We make assumptions that can be verified if needed and required by the project. Assumptions can be things like average complexity on all functions, correlating to a specific function point size. These techniques are referred to as Early Function Points⁵.

The boundary drawing from Function Point Analysis is a good example of a 100-500 feet drawing.



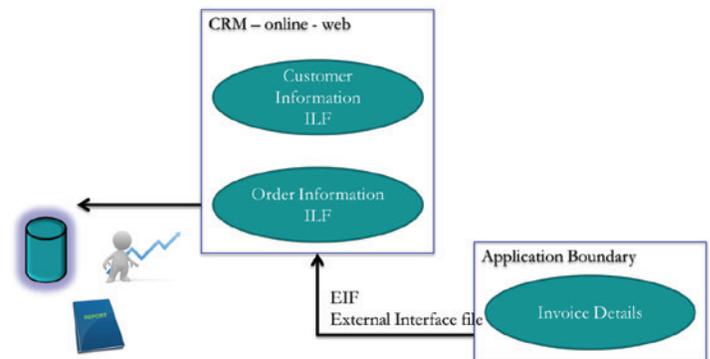
IFPUG Function Point Boundary Drawing – 100 Feet

It shows graphically the boundaries. Boundaries are always from the user perspective. A boundary diagram will include the logical boundaries—seen as the business problem to be solved by software. It will include functionality—what is read, added, deleted, what data are maintained, what data is referenced to the outside of the boundary. The data referenced or updated outside the boundary are the interfaces and dependencies to other logical boundaries—or other business areas. The boundary from a user perspective is how they see the problem solved and what functionality the user believes they need.

A boundary diagram will therefore be excellent for the project execution; discussion input with users, test input for test cases. Whenever something changes—the boundary diagram needs to reflect the change. It will therefore visually show how the users and developers get wiser and find or improve the requirement during the lifecycle. This will unlikely be of interest for management. It's too much information—unless the decision is on a specific application.

Even for specific applications—in most cases the 500 feet view would be of more value since it only highlights the top level from a business perspective. A boundary diagram from 500 feet might only have the ILFs and the interfaces drawn. In this case, use the same drawing technique:

- If it is only sending from one boundary to another—**one arrow**
- If it maintains and sends data—**arrow both ways**
- Add the most important output reports—from a business perspective



500 Feet Boundary Drawing

The 500 feet diagram only looks at the main components of the ILFs. It might not even get all the ILFs. This perspective is excellent for ball park estimation or first assessment of scope. This level, I have often seen used for Request for Proposals, but unfortunately without a measurable and traceable size. If all requests for proposals included a size measure, a lot of the disputes and debates would have been avoided during the lifecycle of the project since it can be verified at a lower level of detail. You can minimize the risk of uncertainty by making assumptions using Function Point Analysis.

Remember, the size measure does not need to be accurate down to the level of 5 percent. It needs to be a ball park figure. It needs to be based on assumptions, sizing experience and knowledge about the application. The size measure can be related to Function Points—so if there is a need for verification of the size measure with greater accuracy—it can be done “flying” down to 100 feet or even 10 feet to have a closer

look. You can decide to have a closer look at only part of the application—using a sampling technique to verify your assumption—or you can have a closer look at the entire application and confirm or change your assumptions.

1,000 Feet – The landscape perspective

When you are 1,000 feet up in the air, you can see the landscape. You can see the most important features, but not all the details. You can see the big changes in the landscape and the most important infrastructure, but not all the small roads. This is the right level of detail for management.

The 1,000 feet diagram would draw the applications (high-level boundaries) and interfaces. It will describe only the high-level business functions. Some of these business functions might be boundaries from an IFPUG Function Point Analysis perspective, but the idea with this drawing is to align the Nucleon perspective of productivity for a team to the business perspective of Function Point Analysis with just enough information to align scope and productivity of the team. I believe the 1,000 feet perspective is the right management level of perspective of the business processes and the size of a project or the size of an application from a portfolio perspective.

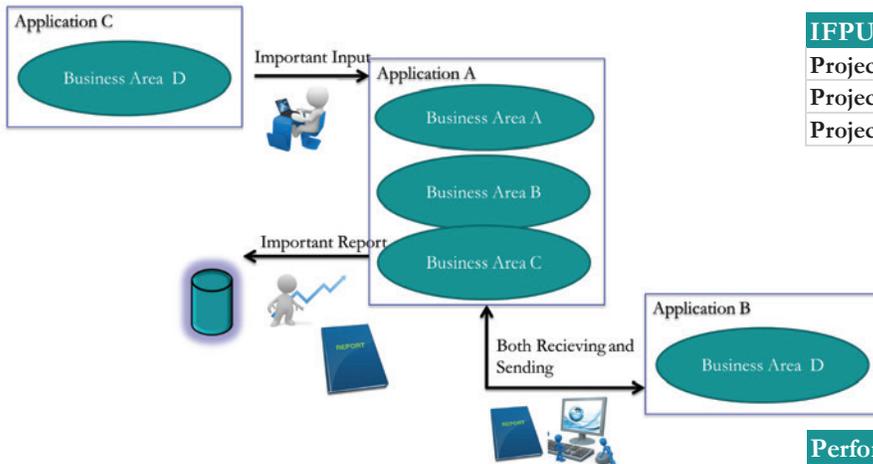


Figure 1

1,000 Feet Diagram - business perspective

In order to get management’s attention, we need to move ourselves to present Function Point Analysis in a way that management understands and has interest in. I believe that this way is the 1,000 feet perspective with its business perspective and overview of the landscape that can be aligned with the strategy rather than the details.

Performance Indicators – Nucleon and Function Points

Using the boundary diagram together with Nucleon for performance will provide a new perspective for management that will enable the strategic decision making.

In the tables below are examples of measures from 3-1,000 feet boundaries and three teams.

The best team is in this case Team A, since it has the highest Nucleon score.

Nucleon	Nucleon Result
Team A	815
Team B	390
Team C	420

In order to show why there is a difference, I have added three additional measures Total Cost of Ownership (TCO), average skills and team size.

Other Measures	Monthly TCO	Team Size	Average Skill
Team A	100	3	10
Team B	140	7	8
Team C	333	20	5.7

The size for three projects is below:

IFPUG FPA	Size in FP
Project 1	800
Project 2	1850
Project 3	2500

Combining this data into performance indicators can give the visibility needed for management for easy decision making. Below are simple illustrations to show this.

Just using performance and size (Nucleon and Function Point) together give an indication of ability of delivery.

Performance Indicator	Team A	Team B	Team C
Project 1	1.0	2.1	1.9
Project 2	2.3	4.7	4.4
Project 3	3.1	6.4	6.0

As you can see, Team B and Team C can deliver with almost the same performance, whereas Team A can deliver the scope (measured in Function Point) twice as effectively as the other teams.

If you include the Total Cost of Ownership (TCO), the result is different due to the amount of staff needed for Team C to



perform as Team A. In this example, I have just multiplied the TCO to the previous performance indicator.

Performance Indicator	Team A	Team B	Team C
Project 1	98	287	635
Project 2	227	664	1468
Project 3	307	897	1984

This makes it evident that Team A is able to deliver faster and more cost effectively than any of the other teams. Such simple analysis is highly valuable for management and enables management to take control of their strategic prioritization of their development and usage of their staff.

You can combine the measures in any performance indicator you like. However, I highly recommend using some performance indicators that are aligned with the strategy of the management. If management is mostly interested in time to market, make sure to include measures for duration and indicators that include cost, Function Point and Nucleons, etc.

I will end this with the statement that IFPUG Function Point Analysis would be a great companion for the Nucleon performance measure and can potentially give IT professionals a valuable tool for estimating, planning, scope management and executing IT projects with an unseen insight and predictability. By taking the steps needed to estimate, organizations and professionals adopting this approach will be the ones that emerges successfully—having a better compass to navigate unknown waters. I look forward to seeing how the combination of Nucleon and IFPUG Function Point Analysis will be used at a management level. ■

References:

- ¹ You can learn more about Nucleon, or take the Nucleon assessment at nucleonformula.com.
- ² IFPUG.org for more information
- ³ IFPUG Function Point Analysis, Counting Practices Manual v4.3.1
- ⁴ IFPUG has also released iTips and uTips on how to utilize and count function points when dealing with special technologies or areas
- ⁵ uTip - Early Function Point Analysis and Consistent Cost Estimating—see <http://www.ifpug.org/uTips/uTip003EarlyFPAandConsistentCost-Estimating.pdf>

About the Author:



Christine Green is vice president of IFPUG and owner of IPbyGreen. Christine's focus as an IFPUG board member is to improve the method and services that IFPUG is providing to its members. One of the ways is to focus on the broader perspective of benchmarking and measurement, such as the benefits of using Function Point Analysis for more cost-effective software services from both a purchasing and delivery perspective. In the past, Christine has been a very active volunteer for IFPUG within IT performance and the development of SNAP. Outside of her IFPUG work, Christine is an IT consultant and owner of IPbyGreen (a Danish Consultant Company). With 20 years of experience within IT outsourcing, Christine is considered an expert within process improvement with special focus on Metric, Measurement, Benchmarking, Dashboard and Estimating and moving troubled projects to a successful go live.

High-Quality Definition of Non-Functional Sizing Method

DOES SNAP COMPLY WITH THE REQUIREMENTS FOR A HIGH-QUALITY SIZING METHOD?

By Srinivasa Rao Kanneganti and Talmon Ben-Cnaan

Introduction

Any Functional Sizing Method (FSM) should comply with fundamental concepts that ensure that the FSM is well-defined, formulated, repeatable, evolved and consistent.

ISO/IEC 14143 defines these fundamental concepts. Users may select an appropriate FSM based on the guidelines of the ISO/IEC 14143. However, there is no guideline to create or select a non-functional sizing method.

This article recommends guidelines for evaluating or selecting a Non-functional Sizing Method (NFSM), based on the functional characteristics as defined by ISO/IEC 14143. Furthermore, it evaluates the level of compliance of SNAP to these requirements.

This article is intended for use by those persons who consider using SNAP, persons who are associated with software sizing, software acquisition, benchmarking or controlling software development or maintenance projects.

Characteristics and requirements of non-functional sizing

ISO/IEC 14143 defines a set of requirements, which any FSM should meet. Although these requirements are specific to functional sizing, similar requirements for a non-functional sizing method can be deduced.

Based on ISO/IEC 14143, a parallel set of characteristics and requirements was created by the authors for non-functional

sizing method. These characteristics are described below:

1. A Non-functional Sizing Measurement Method (NFSM) shall have the following characteristics:
 - a. It is based on a representation of the non-functional user requirements from the perspective of the users.
 - b. It can be applied as soon as any non-functional user requirements have been defined and while they are available.
 - c. It derives a non-functional size.
2. An NFSM method should be as independent as possible of particular software development methods or technologies.
3. A non-functional counting unit is defined and:
 - a. It expresses only non-functional user requirements.
 - b. It is classified as one, and only one unit type.
4. The non-functional size shall have the following characteristics:
 - a. It is not derived from the effort required to develop the software being measured¹.
 - b. It is not derived from the effort required to support the software being measured.
 - c. It is independent of the methods used to develop the software being measured.
 - d. It is independent of the methods used to support the software being measured.
 - e. It is independent of the physical components of the software being measured.
 - f. It is independent of the technological components of the software being measured.
5. A Non-functional Sizing Method shall:
 - a. Define the attributes of the counting unit.
 - b. Define the rules used to assess the counting unit.
 - c. Define the units in which non-functional size is expressed (for example, "SNAP Points").
 - d. Describe the non-functional domain(s) to which the NFSM method can be applied,
 - e. Describe the kind of information necessary to enable the NFSM method to be applied.
 - f. Provide guidelines on how to document a specific NFSM instance.
 - g. Describe the purposes for which the NFSM method can best be used such that the users of the NFSM method can judge its suitability for their purpose.
 - h. State its degree of convertibility to other sizing methods.

6. The requirements from the counting unit.

An NFSM method shall:

- a. Define the counting unit types.
- b. Describe how to identify which non-functional user requirements will be included within the scope of the NFSM.
- c. Describe how to identify the counting units within the non-functional user requirements.
- d. Define how to classify the counting units into unit types.
- e. Define how to assign a numeric value to a counting unit according to its unit type.
- f. Define the relationship, if any, between the SCU type and the boundary.
- g. Define the relationships, if any, between the counting unit types.

7. Designation of non-functional size

The NFSM method shall state the conventions to be adopted when reporting non-functional size such that it is qualified with:

- a. The units of the NFSM method.
- b. The name of the NFSM method.
- c. An indicator that a local customization of a particular NFSM method has been used, where applicable.

Example: SNAP Points APM version 2.4.

8. The process for applying an NFSM method

An NFSM method shall include the following activities in order to derive non-functional size:

- a. Determine the scope of the NFSM.
- b. Identify the non-functional user requirements within the scope of the NFSM.
- c. Identify the counting unit within the non-functional user requirements.
- d. Classify the counting units into unit types, if applicable.
- e. Assign the appropriate numeric value to each counting unit.
- f. Calculate non-functional size.

9. NFSM method labeling conventions

An NFSM method shall:

- a. Use a name that will distinguish it from all other existing NFSM methods (for example, method name = XYZ).
- b. Where applicable, include a version number appended to the method name, which will distinguish it from all other versions of the method.

SNAP compliance with the requirements from a NFSM

After defining the requirements and the characteristics of an NFSM, we can assess how SNAP complies with the requirements.

The following table describes assessment statements that were used to check the compliance of SNAP with NFSM requirements and the level of compliance of SNAP.

Description	SNAP Compliance
The NFSM is based on a representation of the Non-Functional User Requirements from the perspective of the users	Yes. User perspective is a key aspect in SNAP definitions
The NFSM can be applied as soon as any Non-Functional User Requirements have been defined and while they are available	Yes. The APM describes how SNAP size is applied per the maturity of the NFR
The NFSM derives a quantified Non-Functional Size	Yes.



Description	SNAP Compliance
The NFSM Method exclude functional User Requirements from the Non-functional User Requirements when deriving Non-functional size?	Yes. Moreover, the APM contains guidelines how to count function points and SNAP without overlap or duplicated size
The Non-functional User Requirements used by the NFSM Method is a defined sub-set of the user requirements	Yes. The APM refers to the Functional user requirements and the non-functional requirements as complementary sub-sets of the requirements
The identification of which Non-functional User Requirements are to be included within the Scope of the NFSM is an activity required to derive Size?	Yes.
The NFSM Method describes how to identify which User Requirements will be included within the Scope of the NFSM?	Yes.
The NFSM Method should be as independent as possible of particular software development methods or technologies	Yes. SNAP is not affected by the development method and by technologies
The derivation of Non-functional Size is independent of the effort required to develop the software being measured	Yes. SNAP size definition is independent of the effort to develop the software
The derivation of Non-functional Size is independent of the methods used to support the software being measured	Yes. SNAP size definition is independent of the effort to support the software
The derivation of Non-functional Size is independent of any physical components of the software being measured	Yes.
The derivation of Non-functional Size is independent of any technological components of the software being measured	Yes.
The NFSM expresses only Non-Functional User Requirements	Yes. SNAP contains guidelines to count only non-functional user requirements. Moreover, the APM includes guidelines for dividing mixed requirements into their FUR and NFR
The boundary corresponds to the conceptual interface between the software under study and its users is defined	Yes. APM defines the boundary and how to determine it
The determination of the Scope of the NFSM is an activity required to derive Size	Yes. The APM defines scoping as part of the sizing process
The NFSM Method have, or refer to, a definition for the concept of a BNFC (Basic Non-functional Component)	Yes. The concept of a counting unit (SCU) is defined
The NFSM defines a basic counting unit	Yes. The APM defines an SCU, which is the base counting unit
The definition for a BNFC correspond to being an elementary unit of Non-functional User Requirements?	Yes.

Description	SNAP Compliance
The NFSM Method use these elementary units of for measurement purposes	Yes. The SCU is used as an elementary unit to calculate SNAP size
The NFSM defines counting unit types	Yes. The APM defines the SCU per each sub--category
The NFSM Method define the attributes of BNFCs?	Yes. Each sub-category defines the attributes of the SCU (called complexity parameters)
The NFSM Method defines rules used to assess the BNFCs?	Yes. Rules are defined to size the SCU
There is a description of how to identify the BNFCs within the Non-Functional User Requirements?	Yes.
The characteristics of a BNFC only express Non-Functional User Requirements?	Yes. Although an SCU may be common to size FUR and NFR, its non-functional aspects are defined irrespective of the functional size
There is a definition of how to assign a numeric value to a BNFC according to its BNFC Type?	Yes. The APM presents formulas and tables to assign a numeric value (SNAP points) to the SCU
The NFSM Method defines how to calculate the Non-functional Size?	Yes. Formulas are presented in the APM
There a definition of the units in which the Non-functional Size is expressed?	Yes. – SNAP Points
When reporting the Non-functional Size, the user is required to qualify it with the units specified by the NFSM Method	Yes. Size is expressed in terms of SNAP points
When reporting the Non-functional Size, the user is required to qualify it with the name specified by the NFSM Method	Yes. The APM will add the requirement to have the APM version as part of the size, for example: 500 SNAP points (APM 2.4)
There a description of the kind of information necessary to enable the NFSM Method to be applied?	Yes. The APM describes, as part of the sizing process, which information should be gathered
There are guidelines provided on how to document a specific instance of NFSM?	Yes. IFPUG also provided a counting tool with the documentation of the SNAP size
There is a description of the purposes for which the NFSM Method can best be used, such that the users of the NFSM can judge its suitability for their purpose?	Yes. This is part of the sizing process
The NFSM Method use the concept of Non-Functional Domain which corresponds to a class of software based on the characteristics of Non-Functional User Requirements which are pertinent to NFSM?	No, and not needed. There is no need to define domains based on non-functional requirements – the functional domains should be used to classify projects or software products

Description	SNAP Compliance
The identification of the BNFCs within the Non-functional User Requirements is an activity required to derive Size?	Yes. Identifying the SCU as part of the SNAP assessment process
Does the NFSM Method use a name that distinguishes it from all other existing FSM Methods	Yes.
If the NFSM Method implies that there are other versions of the Method, then does it also include the current version number that it appends to its name?	Yes.

Conclusions

In this article we have tried to provide guidelines for selecting an NFSM and assess SNAP in light of such requirements. The table above proves that SNAP well fulfills the requirements of NFSM. Hence, the adoption of SNAP methodology will be a good practice for sizing/measuring non-functional requirements. Some recommendations to further improve SNAP are evaluated by the NFSSC and will be added to the next APM version. ■

References:

- ¹ The ISO/IEEE standard does not allow to define size in terms of effort. However, it is expected that, in projects, the size correlates with the effort required to build this size.

About the Authors:



Srinivasa Rao Kanneganti (Srini) is a manager at Mindtree Ltd. Srini has more than 11 years of professional experience in Information Technology, project management and metrics. He is a Certified Function Point Specialist (CFPS) and active IFPUG Non-Functional Sizing Standards Committee (NFSSC) volunteer. He specializes in a variety of field disciplines including estimation processes, function points, SNAP, metrics, productivity and benchmarking.



Talmon Ben-Cnaan is a quality manager at Amdocs Testing and the chair of IFPUG Non-Functional Sizing Standards Committee. He led the quality measurements in his company. The author was also responsible for implementing Function Points in his organization. The author manages quality operations in Amdocs Testing. Amdocs Testing division comprises of 3,000 testing experts, located at 30+ sites worldwide, and specializes in testing for Telecommunication Service Providers.

FUNCTION POINTS AND AGILE MODELS, HOW TO ESTIMATE

By Cristiane Baccarin



Summary

Agile development has become increasingly common in the software development environment, however estimating the size of the software projects while using agile methodologies is done differently than those undertaken in traditional projects. There is no consensus among experts and users whether estimating function points is possible within projects developed under such a project management technique.

“Quick and short iterations and quick feedback are some features that distinguish agile from traditional projects, thus traditional techniques of effort estimation may not be suitable for agile methodologies.”

1. Introduction

In software projects using agile methodologies, estimation is performed differently to those undertaken in traditional projects. Quick and short iterations and quick feedback are some features that distinguish agile from traditional projects, thus traditional techniques of effort estimation may not be suitable for agile methodologies. The first step in the process of measurement in Function Point Analysis (FPA), for example, is to get the documentation available on the system and/or project that will be measured. Proper documentation can include requirements, data/models, class diagrams, data flow diagrams, use cases, processes descriptions, layout of reports and screens, user manuals and other artifacts of software development [IFPUG, 2010]. None of these documents are common in agile methods. Thus, techniques for agile methodologies have been created and used in software projects [Fuqua, 2003].

2. Function point analysis

FPA is a measurement technique of the functionality provided by a software artifact from the user's point of view. Function point (FP) is the unit of measure that aims to make the measurement independent of the technology used to build the software.

Function points do not directly measure effort, productivity or cost. It is a measure of functional size of software.

3. Agile model

Agile methods have been singled out as an alternative to traditional approaches to software development. Traditional

methods should be applied only in situations where the system requirements are stable and when future requirements are predictable. However, when this is not possible—for example, when there are a lot of change projects, the requirements are liable to changes, software delivery dates are short and rapid development is crucial—use agile methods (BOEHM; TURNER, 2005 and FUQUA, 2003).

3.1. Scrum

Scrum is an agile methodology for the management, planning and development of software projects. It does not define nor does it impose tools and/or systems development techniques, but shows how teams must work in environments with frequent changes and with the emergence of new requirements. This will create a margin for error that can avoid conflicts in time limits with the client.

3.2. Sprint

In the Scrum, a project starts with a simple vision of the product that will be developed. The vision can be vague at first and come clear gradually. The Product Owner (PO) then turns this vision into a list of functional and non-functional requirements that, when they are developed, reflect this vision. This list, called a Product Backlog, is prioritized by the PO so that items that generate more value to the product have highest priority. The starting point is to divide the Product Backlog into releases and it is expected that the content, the priority and the grouping of the Product Backlog will require changes from the time the project starts. These changes reflect changes in business rules and requirements and how quickly the team can turn it into product. All the work is done in Sprints that are two- to four-week iterations. Scrum requires that the Scrum Team develop a product increment every Sprint.

Each increment should be well structured, encoded and tested. Each Sprint begins with a meeting called a Sprint Planning Meeting in which the PO and the team decide what will be developed in this Sprint and its complexity.

4. Perform FPA estimates for Agile models

This article deals with the theory and practice of the function point count and agile models. It presents in a clear and succinct way the theoretical part of the function point count taking into consideration the user's vision and involving the use of the Scrum, where it could be estimated, taking into account the complexity estimates made by developers.

The estimate of the Scrum basically depends on the definition of the amount of work informed by the developer and may vary from one to another. That way, its use cannot be a basis for definition of deadlines and/or cost for management.

The focus of the FP count is the vision of the user, so the only counted functional requirements are those perceived and recognized by him. They describe what the software should do in terms of tasks and services. The focus of the Scrum estimation takes into account how long the professional will take to implement, document and test certain functionality. Thus, it is possible to combine both techniques.

The estimate using FPA provides the base measurement of various metrics; so it can be held for a Sprint or an entire project.

If counts are performed for each Sprint, it may be possible to add the results together to the full size, which is the same as the count of the project. This allows all interested parties to be aware of the size of the project, as well as how much effort the team is spending.

“The focus of the FP count is the vision of the user, so the only counted functional requirements are those perceived and recognized by him.”

4.1. Sprint Backlog

The Sprint Backlog is a list of tasks that the Scrum Team is committed to implement within a given Sprint. The Sprint Backlog items are chosen from the Product Backlog by the team, according to the priorities set by the Product Owner (PO) and the estimation performed taking into consideration the user’s vision.

5. Practice Count

This section presents an example that illustrates our approach: integrating the Scrum with software metrics.

Example – Client registration report Count using agile methodology

For this example, we will use a Sprint (defined with the term of one month) to implement the functionality and deliver it with test completed.

The PO, or client, is a specialist who will describe and prioritize the requirements and rules to be developed. The PO, together with the stakeholder, defined 10 requirements and the Scrum Master (SM) along with the PO made the priority list, thus generating the Product Backlog.

The Ideal Day metric forecast was made for each activity using a scale (0.25, 0.5, 0.75, 1, 1.5 and 1.25, etc.), so the definition is to make the functionality in only one Sprint. We have

coded 14.5 days. The remainder will be used for documentation, tests and to fix any bug found by the test team.

For the calculation of the Ideal Day, we considered 90 percent productivity on a journey of eight hours per day. With these values set, the formula presented in the section software metrics and the Scrum was applied and it was obtained:

$$= 14.5/(1-90)$$

$$= 16.29 \text{ days}$$

The metric of Planning Poker was applied and, after several discussions, resulted in a total of 125 story points. For each point, it was estimated a value in time with a total of 220 hours. The PO and the SM started the Sprint and initiated an analysis of the information discussed at the meeting of Sprint Planning Meeting to perform the function point count. It is very important to remember that the analysis, the count and development are carried out in parallel.

Count using FPA

Count of Transaction

- Send Client report CE.
- Total ALR-1.
- Total DER-8.

PF unadjusted count

To extract the complexity and the total points of both of the functions (ALIs and/or AIEs) and transaction functions (SE, CE and/or EEs), you need to calculate the function points not adjusted. For example, client enrollment. Note: ALI = ILF, AIE = EIF, EE = EI, SE = EO, CE = EQ, ALR = FTR and DER = DET.

Function type	Functional Complexity			Total Complexity	Total function type
ALI		Low	x 7	0	
		Average	x 10		
		High	x 15		
AIE		Low	x 5		
		Average	x 7		
		High	x 10		
EE		Low	x 3	0	
		Average	x 4		
		High	x 6		
SE		Low	x 4		
		Average	x 5		
		High	x 7		
CE	1	Low	x 3	3	3
		Average	x 4		
		High	x 6		
Total:					3

TOTAL POINTS OF FUNCTION DOES NOT SET

Calculation of adjusted function points

After the calculations of adjusted function and adjustment factor will be adjusted function point calculated. For our example:

$$\text{Set PF} = (3 + 0 + 0 - 0 - 0) \times 1.01$$

Set PF = 3.03

That way, the size of functionality to be implemented would be 3.03 points.

6. Examples of estimates using Scrum and FPA

This table exemplifies the Product Backlog.

Point count worksheet function							
Backlog Item	PF = 184						
Description	Type	(I/A/E)	TD	AR/TR	Complex.	PF	Local PF
1 cycle (sprint backlog 1)							
Authenticate User (Login)	EE	/	3	1	Low	3	3,000
I forgot my password, reset password	SE	/	3	1	Low	4	4,00
Change (sw ap) password	EE	/	4	1	Low	3	3,00
Sending payment gateway	SE	/	100	2	High	7	7,00
Return payment gateway	EE	/	100	2	High	6	6,00
Distributor							
Keep Distributor	ALI	/	24	2	Average	10	10,00
Include Distributor	EE	/	26	2	High	6	6,00
Edit Distributor	EE	/	25	2	High	6	6,00
Consult Distribuidor	SE	/	26	2	High	7	7,00
Send accepted terms and conditions, Send quality policy	SE	/	2	1	Low	4	4,00
Products							
keep products	ALI	/	20	3	Average	10	10,00
Active/assessment products-distributor	CE	/	5	3	Low	3	3,00
Import products-distributor	EE	/	22	3	High	6	6,00
Edit product	EE	/	21	3	High	6	6,00
Include product	EE	/	22	3	Low	6	6,00
Search product	SE	/	22	3	High	7	7,00
Buyer							
Keep Buyer	ALI	/	22	3	Average	10	10,00
Include Buyer	EE	/	24	3	High	6	6,00
Edit Buyer	EE	/	23	3	High	6	6,00
Search Buyer	SE	/	24	3	High	7	7,00
2 cycle (sprint baclog 2)							
Manufacturer							
Keep Manufacturer	ALI	/	22	2	Average	10	10,00
Include Manufacturer	EE	/	24	2	High	6	6,00
Change fabricate	EE	/	23	2	High	6	6,00
Search Manufacturer	SE	/	24	2	High	7	7,00
Thread							
Keep Thread						3	3,00
Include Thread	EE	/	6	1	Low	3	3,00
Edit Thread	EE	/	6	1	Low	3	3,00
Delete Thread	EE	/	1	1	Low	3	3,00
Search Thread	SE	/	6	1	Low	4	4,00
Administrator							
Audit log	ALI	I	6	2	Low	7	7,00
Export	CE	I	8	2	Average	4	4,00
Show	CE	I	6	2	Average	4	4,00
Include	EE	I	8	2	Average	4	4,00

TABLE 1 – EXAMPLE OF PRODUCT BACKLOG

During a Sprint, the SM keeps updating the Sprint Backlog to reflect tasks that are completed and how long will it take to complete those that are not yet ready. A Sprint Backlog example is shown in Table 2:

Description	Complex	Estimate (PF)	Productivi	Owner	Status	Sprin	Total hou	day 1	day 2	day 3	day 4	day 5	day 6
Ciclo 1 (sprint backlog 1)													
Authenticate User (Login)	Low	3	2	Paula	completed	1	6	6					
I forgot my password, reset password	Low	4	2	Marcos	completed	1	8	8					
Change (swap) password	Low	3	2	Marcos	completed	1	6		6				
Sending payment gateway	High	7	2	Lucas	completed	1	14	8	6	2			
Gateway de pagamento retorno	High	6	2	Lucas	completed	1	12			6	6		
Distributor													
Manter Distribuidor	Average	10	2	Paula	completed	1	20		8	8	4		
Include Distribuidor	High	6	2	Paula	completed	1	12				4	8	
Edit Distribuidor	High	6	2	Paula	In Progress	1	12						8
Consult Distribuidor	High	7	2	Paula	To Do	1	14						
Send accepted terms and conditions, Send quality policy	Low	4	2	Paula	To Do	1	8						
Products													
Keep products	Average	10	2	Marcos	completed	1	20			2	8	8	2
Active/assessment products-distributor	Low	3	2	Marcos	In Progress	1	6					4	1
Import products-distributor	High	6	2	Marcos	To Do	1	12						
Edit product	High	6	2	Marcos	To Do	1	12						
Include product	Low	6	2	Marcos	To Do	1	12						
Search product	High	7	2	Marcos	To Do	1	14						
Buyer													
Keep Buyer	Average	10	2	Lucas	In Progress	1	20				2	8	8
Include Buyer	High	6	2	Lucas	To Do	1	12						
Edit Buyer	High	6	2	Lucas	To Do	1	12						
Search Buyer	High	7	2	Lucas	To Do	1	14						
2 cycle (sprint backlog 2)													
Manufacturer													
Keep Manufacturer	Average	10	2	Paula	To Do	2	20						
Include Manufacturer	High	6	2	Paula	To Do	2	12						
Change Fabricate	High	6	2	Paula	To Do	2	12						
Search Manufacturer	High	7	2	Paula	To Do	2	14						
Thread													
Keep Thread	Low	7	2	Marcos	To Do	2	14						
Incluir Segmento	Low	3	2	Marcos	To Do	2	6						
Edit Thread	Low	3	2	Marcos	To Do	2	6						
Delete Thread	Low	3	2	Marcos	To Do	2	6						
Search Thread	Low	4	2	Marcos	To Do	2	8						
Administrator													
Audit log	Low	7	2	Lucas	To Do	2	14						
Export	Average	4	2	Lucas	To Do	2	8						
Show	Average	4	2	Lucas	To Do	2	8						
Include	Average	4	2	Lucas	To Do	2	8						
							0						
							382	360	336	312	288	266	249

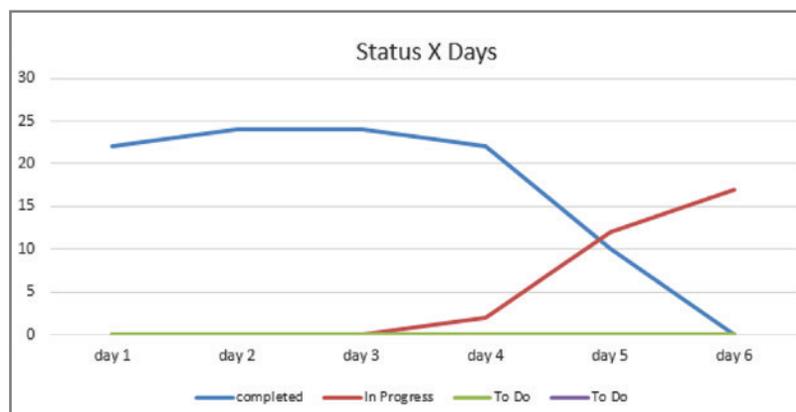
TABLE 2 – EXAMPLE OF SPRINT BACKLOG

Through the results obtained according Table 2.

1) X Day Status:

Status	day 1	day 2	day 3	day 4	day 5	day 6
completed	22	24	24	22	10	0
In Progress	0	0	0	2	12	17
To Do	0	0	0	0	0	0

STATUS X DAY



STATUS X DAY

7. Conclusion

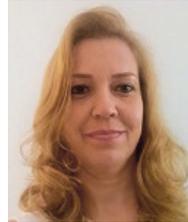
Companies in software development always seek to improve their services in order to beat the competition.

We can conclude that the function point counts are quite different from those obtained through the technique presented by Scrum because the Section 5 Example 1 – Client Registration Report states the estimate of the Ideal Day is 125 PF, being higher than that calculated by the APF with PF 3.03 using the same example. However, it is possible to perform a combination of the three techniques, taking into account the estimates made by developers, and the size, taking into consideration the user's vision. ■

References:

- BOEHM; TURNER 2005, FUQUA 2003
- CMMI® para Desenvolvimento Versão 1.3
- Fuqua, a. m., 2003; "Using Function Points in XP-Considerations". In: *Extreme Programming and Agile Processes in Software Engineering*, LNCS vol. 2675/2003, pp. 1013
- International Function Point Users Group, ifpug.org

About the Author:



Cristiane Baccarin has more than 20 years of experience in information technology (IT), project management with CMMI 3, systems analysis for data warehouse, systems analysis and metrics for private and public IT companies. She is a Certified Function Point Specialist (CFPS) since 2016. She also concluded different specialization courses such as engineering and project management - Pontificia Catholic University of Paraná "A Model for Risk Management" Information Technoly - Paraná Federal University "Federal Databases," IT Production Manager, Positivo University "The Power of IT/Project in Data Warehouse..."



ISMA¹⁶ Brazil ‘Métricas 2018:’

São Paulo Prides Itself on Hosting the IFPUG Conference Once Again!

IFPUG held ISMA¹⁶ in São Paulo, Brazil. This is the fourth time that the event was held in Brazil.

Brazilian companies, including industrial and government entities, have been using IFPUG Function Points in contracts for more than 20 years. During this long journey, Brazil has faced many challenges. That is the price of innovation. Today, Brazil can say it was worth it. Companies have recognized the need to identify and assess whether the value paid for the software product is aligned with cost, effort, productivity, speed, quality and expectations. We will continue to work to ensure Function Points are current despite new trends, new

methodologies and new technologies.

The conference, organized by BFPUG (Brazilian Function Point Users Group) and held on Oct. 18, 2018, featured nine excellent presentations, diving into different shades of measurement, addressing technical and business aspects and recognizing how Brazilian companies are dealing with new development trends (for example, scaled Agile, digital transformation and mobile architecture). IFPUG members will be able to access these presentations in the Members’ Services Area, Knowledge Base Page.



There was live translation from English to Portuguese/Portuguese to English during the entire conference. This conference has received IFPUG approval for CFPS certification extension. CFPS attendees would be eligible for a one-year extension on their certification, provided they comply with all related IFPUG rules and requirements. There was also a raffle of Donald Reifer's book, *Agile Software Quality: Fundamentals*, and the winner was attendee Antonia Dutra.

The welcome and closing remarks were delivered by Mauricio Aguiar, Christine Green and Tom Cagley, who are IFPUG's president, vice president and past president, respectively. During the Annual IFPUG Meeting, the election results for an opening on the Board of Directors were announced with Diana Baklizky winning the seat. Mrs. Baklizky will serve as Director of Communications and Marketing.

Tom Cagley presented **Product Owners In Agile - The Really Hard Role!** Product owners are a part of the leadership structure in Agile teams. Leadership in Agile teams includes some combination of Scrum master, product owner, team leader, coach and/or team members. The product owner has a special level of power and leadership as the voice of business and often as the conduit to the sponsor's wallet. Product owners, when the right people fill the role, are perfectly situated to shape a team's culture and are critical when adopting Agile.

Dr. Luigi Buglione discussed **The 'Sync' Effect - How to Apply FPA in Mobile Architecture Environments with Synchronization Requirements** (valid for CFPS extension).

When dealing with Non-Functional Requirements (NFRs), people often tend to think that 100 percent should be sized/evaluated with non-functional techniques, while a requirement can be split into (at least) three parts, the so-called 'ABC Schema': Product-Functional (A), Product-Non Functional (B), Project-Related (C). Connectivity is one of the possible NFR attributes for a software solution that is listed under the ISO/IEC 25010 taxonomy. But, is it possible that having (or not) the connection on could also be measured (and sized) with Functional Size Measurement (FSM) methods such as IFPUG FPA? In this presentation, Dr. Buglione showed a mobile architecture environment scenario where synchronization requirements could be required to be deployed into a software solution, discussing the way a layered architecture with Front-Ends and Back-Ends dialogues with related impacts on project estimates.

Christine Green presented **Create 'Agile' Contracts Using Size Measures.** It was explained issues regarding what originally seemed to be a small contractual statement about the scope of the project turned out to be a dispute during the execution of the contract. A lot of the time the dispute is due to lack of knowledge of the scope at the time of the creation of the contract. Agile approaches can solve this issue, but it can be hard to go all the way and be agile in contractual terms. Highlights of this presentation included using size measure in contracts where some of these types of disputes can be avoided, making the contract a lot more agile for all parties. Contract statements can be flexible with regard to the scope



Flávio Marietto (Banco Santander)



Felipe Barbalho (Pitang Agile IT)

if you use size measures in the right places at the right times without losing the ability to have a contract with the right level of control both legally and financially.

Dácil Castelo (author) and **Christine Green** (presenter) partnered for **Software Rates vs Function Point Price: Productivity and FP Price in SW Portfolio Management**. Companies all over the world usually manage their project portfolio based on man hours, negotiating the rates on a per-contract basis and other indicators such as time to market, timely delivery, deviations (effort, time or cost) or project backlog, but not focusing on the delivered product and, due to this, neither on the unit product price. This issue presents very interesting situations in terms of expenses and unit prices. This presentation talked about some of the most important concepts of IT software development management with a simple and easy approach based, as always, on real scenarios.

Charles Wesolowski introduced the topic **Quality Software Requirements - IFPUG Function Points and Model-Based Systems Engineering** addressing the production of quality software requirements using Model Based Systems Engineering (MBSE) techniques. MBSE uses Open Standard Modeling Languages, such as SysML and UML, to formally describe system requirements, including Software Functional Requirements. Highlights of this presentation were the IFPUG meta-model,

CMMI Specific Goals for the Requirements Development and Measurement and Analysis Process Areas, as well as the SysML/UML expressions necessary to effectively express software functional requirements in a manner that is useful for customers, managers and engineers.

Carol Dekkers presented **Three Cs to (FSM) Measurement Success: Create, Confirm, Convince**. To practitioners of FSM methods it may seem obvious that function points and related metrics should be part of every software development organization. However, the majority of technology organizations are unaware of FSM even though they are software centric agile and use project management methods. Highlights of this presentation suggested a 3C approach to growing the acceptance and spread of FSM organically, one company at a time.

Diana Baklizky talked about **Agile & Function Points Go Well Together!** It was an update of her presentation last year that featured actual cases of FP measurement in agile contexts (scaled or not), how clients have benefited from the results, how the role of the measurement analyst changed, what was measured, where the measurement inputs were found, the main challenges faced, as well as lessons learned. Highlights of this presentation were the updates included, new cases, as well as progress made on the preceding ones.



Mauricio Aguiar (IFPUG President)

Felipe Barbalho conducted **Measuring Agile Projects with Function Point-Based Compensation**. As a software supplier using agile approaches to development, he shared his experience in measuring projects where supplier compensation is based on function points. The following topics were addressed: satisfying contract requirements, the interaction between developers and the metrics team, measurement inputs, how to deal with refactoring and scope changes in different contracts, challenges and lessons learned. Highlights of this presentation were the approach used in how to manage agile contracts and how to identify the refactoring rate in supplier point of view.

Flavio Marietto presented **No Waterfall at Santander – Agile is Productive!** In 2016, Santander Brazil adopted the Scaled Agile Framework. Since then, the bank has been consistently measuring software project productivity, having

adopted Function Point Analysis in 2017 because it is a highly-used technique with many specialized professionals in the market. During this period, productivity measurement grew and became one of the major elements supporting the planning process involving all agile projects. The bank currently has 170 squads with 1,427 participants distributed in 15 tribes. The big motivation for measuring the productivity of agile projects at Santander Brazil was the need to plan and assess squad capability, as well as to identify deviations and make comparisons using tools, DevOps and agile practices maturity. As a result, work processes and measurement methods were adjusted for the agile approach, and a productivity measurement guide was defined with rules to complement FPA.

Thanks to the ISMA São Paulo sponsors TI Métricas and Obrigado Brazil! ■



IFPUG 2018 ANNUAL MEETING



IFPUG held its annual meeting on Oct. 18, 2018 in Sao Paulo, Brazil to share information with the membership and to present our finances, election results and to request comments from the membership.

Financial results: Kriste Lawrence reported that IFPUG is in good financial shape. However, we did experience an operating expense loss for the 2017/18 fiscal year.

Revenues: \$225,583.87
Expenses: \$261,707.48
Net Income: (\$36,123.61)

Our balance sheet is still in solid

territory due to the diligence of our treasurers over the years.

Election results: In a close election for the one open position, Diana Baklizky was elected to the IFPUG Board of Directors. This year, 115 votes were tallied from 15 countries. Please welcome Diana. She will replace Carol Dekkers.

The 2018/19 Board of Directors is comprised of:

Mauricio Aguiar, President
Christine Green, Vice President
Kriste Lawrence, Treasurer
Tom Cagley, Immediate Past President
Luigi Buglione

Dácil Castelo
Chuck Wesolowski
Roopali Thapar
Diana Baklizky

The board recognized Carol Dekkers' selfless efforts as a member of the board. We will miss her deep knowledge and wisdom. We look forward to Carol's participation on international standards issues. We all hope that Carol considers running for the board again in the future.

Mauricio Aguiar concluded the annual meeting by asking for comments from the floor. Minutes will be published after approval at the next annual meeting. ■

IFPUG Board of Directors

Mauricio Aguiar, President

TI Métricas
mauricio@metricas.com.br

Christine Green, Vice President

IP by Green
Info@ipbygreen.com

Kriste Lawrence, Treasurer

Perspecta
Krista.Lawrence@perspecta.com

Thomas M. Cagley, Immediate Past President

Tom Cagley and Associates
tcagley@tomcagley.com

Charles Wesolowski, Director of Sizing Standards

SAIC
charles_wesolowki@bellsouth.net

Dr. Luigi Buglione, Director of Education & Conference Services

Engineering Ingegneria Informatica SpA
luigi.buglione@eng.it

Dácil Castelo, Director of International Membership

LEDA Consulting, S.L.
dcastelo@leda-mc.com

Diana Baklizky, Director of Communications & Marketing

TI Métricas
diana@metricas.com.br

Roopali Thapar, Director of Certifications

IBM
roopali.thapar@in.ibm.com

Certification Committee

By Gregory Allen, Committee Chair

The Certification Committee has been working on making the Certified Function Point Specialist (CFPS) exam available in more languages worldwide. The translation of the CFPS exam into Spanish and Korean is complete and has been sent to International Software Quality Institute (ISQI) for publication. We expect the CFPS exam to be available in these two languages in the first quarter of 2019. We will also be working on making the CFPS exam available in Japanese later in 2019. Other languages are also being considered as the number of IFPUG members continues to expand in countries like France and China. ■

Communications and Marketing Committee

by Antonio Ferre Albero, Committee Chair

The IFPUG CMC's (Communications and Marketing Committee) mission is to serve, in a cross way, IFPUG as an organization, and help IFPUG and the committees to spread the importance of applying metrics related with the Information Technology and with the different IFPUG methods. Volunteers from different countries and cultures comprise IFPUG. To volunteer means to give part of your free time without any kind of economic return; usually is just a consequence of having passion for a thing. The passion for the IT metrics is perhaps the aspect of the people involved in IFPUG. *MetricViews*, the publication that you have in your hands, is one of the CMC deliverables, with content written by people with passion and knowledge from different parts of the world in a nonprofit way.

Linked with the above ideas of volunteering, news since the previous *MetricViews* edition is the change of the CMC IFPUG board liaison. IFPUG board members, following the IFPUG bylaws, are elected and have a duration of three years. In Q3 2018, Diana Baklizky from Brazil was elected as a new board member with the role of CMC liaison, a role held by Carol Dekkers from the United States in the last three years. The announcement was unveiled during the IFPUG annual meeting (held in ISMA event, in São Paulo, Brazil), in a truly emotional moment thanking Carol for the services rendered. Carol has been a person who has worked closely with IFPUG along the history and was IFPUG president from 1998-1999. Thanks Carol and welcome Diana. Diana is a person with strong metrics commitment and knowledge who comes from Brazil where metrics and functional size are a must, with solid testimonials and live strategies around the software size and IT metrics concepts.

Regarding other aspects, after the incorporation of the GDPR law we are detecting that the number of email addresses of people to which IFPUG sends information has been reduced, perhaps due to opt-in reasons or perhaps just because you didn't confirm your email address. So, if you were receiving our news and now are not receiving it, please contact ifpug@ifpug.org. ■

Conference and Education Committee

by *Filippo De Carli, Committee Chair*

For 2019, the Conference & Education Committee (CEC) is co-organizing two new IFPUG conferences around the world: we'll be back in India on March 6-8, this time in Bangalore, for ISMA¹⁷ (www.ifpug.org/isma17) after the ISMA¹³ experience two years ago and then we're planning the fall conference (ISMA¹⁸), possibly returning to Europe in September (for more information, stay tuned to the IFPUG communication channels) after the two ISMA events in Rome, Italy in 2016 and 2018.

A new CEC initiative that will take place this year will be to produce a series of webinars for our members that will also be recorded as podcasts to access in the member-only section of the IFPUG website with your credentials. By doing so, many interesting presentations and discussions could be shared outside of the formal conference "space" and be used for better understanding or to clarify some concepts from the FPA and SNAP guides. For instance, some terms, such as "baseline" or "functional reuse" are familiar to many of you but could be better explained. Let's discuss this together in the next #MetricsWebinars that will come in the following months!

Thus, #StayTuned and if you have comments, suggestions or feedbacks, please contact us at cec@ifpug.org! ■

Functional Sizing Standards Committee

by *Dan French, Committee Chair*

The Functional Sizing Standards Committee (FSSC) closed out 2018 with the publication of the addendum to the Data Warehouse white paper, the FPA Applied to BPM-Based Systems white paper, and updates to the iTip #5 Real-Time Data Sharing and iTip #6 Shared Data Real-time Responses. We also had one of our members, Diana Baklizky, elected to the IFPUG Board of Directors! The committee appreciates all of the work Diana has done in support of the FSSC and we wish her great success while serving on the board.

The FSSC has initiated new projects to develop white papers on elementary processes, mobile application counting and determining boundaries. The first two are in the process of

initial draft review by the FSSC. The boundary paper outline has been approved and an initial draft is being developed. The XML white paper is almost ready for publication and should be published in the coming months.

In addition to our monthly committee meetings, the FSSC held its annual meeting in June at the International Cost Estimating and Analysis Association in Phoenix, Arizona in the United States. During the annual meeting, we reviewed all ongoing projects, completed a few of them as well and started planning for 2019. We also decided that the FSSC and Non-Functional Software Standards Committee (NFSSC) will not work on the proposed project to study either changing or eliminating the General System Characteristics (GSC).

Our committee is always looking for new projects and welcomes members to suggest topics of interest. If you have any topics that you would like the FSSC to address or if you or anyone you know would like to volunteer for the FSSC, please contact Dan French, FSSC chairman, at dfrench@cobec.com.

FSSC looks forward to a successful 2019 and we appreciate the support of the IFPUG Board of Directors and the IFPUG membership. ■

International Membership Committee

by *Saurabh Saxena, Committee Chair*

The International Membership Committee is working to enhance the member experience, enhance membership value and increase the IFPUG footprint around the globe. Our team interacts and resolves all kind of IFPUG-related queries with high priority. We have dedicated country representatives looking after the India, Brazil, China, Italy, France and United Kingdom regions.

Apart from quick resolutions on a vast range of queries (e.g. how to register for CFPS/CFPP/CSP, how to apply for CEP, benefits of IFPUG membership, etc.), the following actions took place:

- Collaborated with CMA to smoothen the membership renewal process. Helped bring several members back to IFPUG.
- Reviewed and updated IMC-related webpages on the IFPUG website (e.g. added information on "Grace Period for IFPUG members").
- IFPUG India Function Point Chapter created (WhatsApp Group and LinkedIn).

- Collaborated with CEC in planning for ISMA¹⁷, scheduled in India (by interacting with Indian members and encouraging them to participate).
- Added Sergio (country representative for Brazil) and Rajesh (India) to the committee.
- Works in progress:
 - ✓ SNAP manual and IFPUG F&Qs Italian Translation.
 - ✓ Review to improve the CPM French-translated manual.
 - ✓ Start process for “IFPUG Honorary Member” selection for 2018.

We look to continue the good work of providing quick and accurate responses to all IFPUG-related queries and assist to further enhance membership experience. ■

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 SC7 (Software and Systems Engineering) activities, keeping IFPUG visible as a valuable sizing method for the systems of today.

Steven Woodward provided recommendations to the ISO/IEC 25000 series of documents to reference IFPUG and other ISO/IEC software sizing standards, when performing defect density and other comparative quality analysis.

Talmon Ben-Cnaan is chairperson of the IEEE Non-Functional Sizing Standardization activity, where the IEEE voting was just completed and the results, comments and recommendations are now being analyzed.

Pierre Almen, as the liaison with International Software Benchmarking Standards Group (ISBSG) community, is encouraging more collaboration and joint activities that can benefit IFPUG and ISBSG.

Steven Woodward returned from presenting at QA Days in Austria on Metrics and is hoping to coordinate further with the Object Management Group (OMG) to clarify the benefits from consistent software sizing using the IFPUG method.

Steven Woodward plans to attend the ISO Plenary meetings in Helsinki Finland as part of the Canadian delegation after attending GUFPI in Rome.

Functional size provides valuable perspectives in 2019 for IoT, Blockchain, AI and cloud computing, where the Industry Standards Committee will gain visibility in several ICT organizations and standards communities.

We welcome your participation to help increase software metrics competencies across multiple ICT standards communities ■

Non-Functional Sizing Standards Committee

by Talmon Ben-Cnaan, Committee Chair

SNAP as an IEEE Standard

IFPUG is working with IEEE’s Software and Systems Engineering Standards Committee to generalize SNAP as an IEEE standard. The standard draft has successfully completed its public review, and will be published once the review comments will be implemented.

SNAP and GSCs - A New Research on General System Characteristics (GSCs)

In Allan Albrecht’s original 1977 paper on function point analysis (Measuring Application Development Productivity), he included 10 “complexity factors” which were weighted from zero to five depending on their degree of influence toward the application being developed. He updated these in his 1983 publication (Software Function, Source Lines of Code, and Development Effort Prediction: a Size Validation) into 14 complexity factors, which are the foundations of the GSCs published today in the Counting Practices Manual. These 14 GSC have been relatively unchanged since 1983 although additional clarification has been published.

Marymount University, together with IFPUG FSSC and NFSSC, has initiated research to investigate whether the GSCs can be improved in structure, and/or whether their use can be improved. This is to take into account the influence of SNAP and other new technologies. The research may conclude that the GSCs should not be changed, or it may recommend changes of some form, or it may recommend changes at some conditions only. ■

Committee Rosters

Certification Committee

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

Communications and Marketing Committee

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

Conference and Education Committee

- Filippo De Carli, GUFPI-ISMA Gruppo Utenti Function Point Italia – **Chair**
- Antonio Ferre Albero, GFT IT Consulting
- Eduardo Alves Oliveira Sr.
- Saurabh Saxena, Amdocs Development Centre India Pvt Ltd.
- Sushmitha Anatha, Accenture
- Dr. Luigi Buglione, Engineering Ingegneria Informatica SpA – Board Liaison

Finance Committee

- Mauricio Aguiar, TI Métricas
- Thomas M. Cagley, Tom Cagley & Associates
- Christine Green, IP By Green
- Kriste Lawrence, Perspecta

Functional Sizing Standards Committee

- Daniel Bradford French, Cobec Consulting – **Chair**
- Bonnie Brown, Perspecta – **Vice Chair**
- Diana Baklizky, TI Métricas
- Sergio de Quintal Brigido, TI Métricas
- E. Jay Fischer, JRF Consulting
- Steve Keim
- David Lambert, David Consulting Group
- Tammy Preiss, AT&T
- Esteban Sanchez, Galorath
- Peter Thomas
- Adri Timp, equensWorldwide SE
- Charles Wesolowski – Board Liaison

Industry Standards Committee

- Steven Woodward, Cloud Perspectives – **Chair**
- Pierre Almen, CFPS, ImproveIT – ISBSG Liaison
- Talmon Ben-Cnaan, Amdocs – SNAP IEEE Working Group
- Carol Dekkers, Quality Plus Technologies – US TAG/ISO Representative
- Charles Wesolowski – Board Liaison

International Membership Committee

- Saurabh Saxena, Amdocs Development Centre India Pvt. Ltd. – **Chair**
- Sergio de Quintal Brigido, TI Métricas – **Brazil Country Representative**
- Cao Ji, Beijing Suiji Tech – **China Country Representative**
- Gianfranco Lanza, CSI Piedmonte – **Italy Country Representative**
- Dr. Lionel Y Perrot, Semantys – **France Country Representative**
- Ivan Pinedo, LEDA Consulting, S.L. – **Spain Country Representative**
- Dacil Castelo, LEDA Consulting – 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 Lamoureux
- Tomasz Marchel, Asseco Poland S.A.
- Roopali Thapar, IBM
- Saurabh Saxena, Amdocs Development Centra India Pvt Ltd.
- Srinivasa Rao K, Mindtree Consulting
- Dr. Luigi Buglione, Engineering Ingegneria Informatica SpA
- Sachin Thakur, Amdocs Development Centra India Pvt Ltd.
- Charles Wesolowski, SAIC – Board Liaison

Congratulations to these NEW and Extended Certified Function Point Specialists!

Enea Aganetti
Engineering Ingegneria Informatica SpA

Vincenzo Maria Aleo
P.R.S. - Planning Ricerche E Studi

Riccardo Bisini
SOGEI

Ian Brown
Galorath Incorporated

Daniele Carlacci
Business Integration Partners SPA

Elisabetta Cauzillo
Eustema S.p.A

Donatella Cola
Almaviva SpA

Paola Concolino

Cezar Costa

Francesca Fedeli
SOGEI

Cinzia Ferrero
CSI PIEMONTE VAT. N.
IT01995120019

Tommaso Galiano
Business Integration Partners SPA

Roberta Generali
Almaviva SpA

Pasquale Giugliano
Engineering Ingegneria Informatica SpA

Jayalakshmi Gurumoorthy
IBM

Nandhini Gurunathan
IBM

Yasuharu Honda
JFPUG-Japan Function Point Users Group

Jiro Imai
JFPUG-Japan Function Point Users Group

Nobutake Iwai
JFPUG-Japan Function Point Users Group

Miyako Iwakiri
JFPUG-Japan Function Point Users Group

Bo Liang
AIA Information Technology (Guangzhou) Company, Limited

Ananthavalli Manivannan
IBM

Lucas Mariano

Vincenzo Mauro
Business Integration Partners SPA

Ismael Melo
Abrantes Solucoes Ltda.

Harish Mishra
Vodafone Consumer Products & Services

Akira Miyata
JFPUG-Japan Function Point Users Group

Marta Monteleone

Ercilia Maria Muxagata Conrado
Caixa Economica Federal

Masako Niwa
JFPUG-Japan Function Point Users Group

Sawako Oiwa
JFPUG-Japan Function Point Users Group

Ryo Ooya
JFPUG-Japan Function Point Users Group

Valeria Parlagreco
Almaviva SpA

Yoshitaka Sakamoto
JFPUG-Japan Function Point Users Group

João Vitor Silva
FATTO Consultoria e Sistemas

Paulo Silva Chaves

Ryo Takahashi
JFPUG-Japan Function Point Users Group

Congratulations to these NEW Certified Function Point Practitioners!

Alessandra Aliperta
SOGEI

Eleonora Ambrosiano
Engineering Ingegneria Informatica SpA

Fabrizia Andrenacci
SOGEI

Domenico Aprile
ISTAT

Bianca Maria Bruno
Engineering Ingegneria Informatica SpA

Anna Carla Bumma
Almaviva SpA

Marilena Capoluongo
Engineering Ingegneria Informatica SpA
Bledar Cili
Accenture

Fausta Cosenza
SOGEI

Stefania Crusco
Almaviva SpA

Antonio Falcone
Almaviva SpA

Keiji Funaki
JFPUG-Japan Function Point Users Group

Irene Gattuso
Almaviva SpA

Chiaki Hosoya
JFPUG-Japan Function Point Users Group

Roberto Ianni
SOGEI

Raffaella Angela Gabriella Izzo
Almaviva SpA

Kazukuni Koseki
JFPUG-Japan Function Point Users Group

Antonia Laviola
Almaviva SpA

Antonia Laviola
Almaviva SpA

Donato Magnolo
P.R.S. - Planning Ricerche E Studi

Martina Munno
Engineering Ingegneria Informatica SpA

Marina Pirone
Almaviva SpA

Priscila Silva
FATTO Consultoria e Sistemas

Daniela Trubbas
Almaviva SpA



STAY CONNECTED

As an IFPUG member, you are part of an international association dedicated to improving the quality and future of the information technology industry. Are you taking full advantage of all that your membership offers?

Benefits include:

- Access to education and professional growth through semi-annual IFPUG Workshops and the annual IFPUG Conference at special member rates.
- Opportunity to join a local IFPUG Chapter, where you can exchange ideas, share experiences, and learn about new techniques on an ongoing basis, in your area.
- Participation in IFPUG communities to advance state-of-the-art software measurement and professional networking with colleagues from around the world.
- Professional certifications, which establish your credentials as a specialist in the growing field of software metrics.
- Access to state-of-the-art products and services at vendor showcases during the annual conference.
- Special member rates on IFPUG materials (e.g., the Function Point Counting Practices Manual and the International Software Benchmarking Standards Group publications).

IFPUG's social media channels allow you to stay connected to your fellow IFPUG colleagues and the HQ staff.



Be Informed! Stay Connected!

Advertise Around the World with IFPUG

Argentina
Belgium
Brazil
Canada
Chile

China
Colombia
Croatia
Czech Republic
Denmark

France
Ghana
Hong Kong
India
Israel

Italy
Japan
South Korea
Luxembourg
Malaysia

Netherlands
New Zealand
Peru
Poland
Singapore

Spain
Sweden
Switzerland
Taiwan
Thailand

United Kingdom
United States



**Spread your message to a global audience in 32 countries across 6 continents.
Promote your product or service by placing a highly visible ad in the September 2019 issue of *MetricViews!***

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