

August 2016

INTERNATIONAL
FUNCTION POINT
USERS GROUP

191 Clarkville Road
Princeton Junction, NJ 08550
USA



Vol. 10 • Issue 2

MetricViews www.ifug.org



SHOW ME THE METRICS

Inside...

In This Edition.....	2
Letter from the President.....	2
From the Editor.....	3
Finding the Right Service Level Measure in a Changing Outsourcing Landscape...	4
Future for Metrics and Quality in an IoT Cloud World.....	6
ISMA12, an amazing event: thanks Italy.....	7
Variations in Software Development by Function Point Size.....	9
Datawarehouse FP Counting Guide used in Brazilian Government Contracts.....	11
Industry Best Practices v. Moral Hazard in Software Development.....	15
SNAP for Sizing Corrective Maintenance.....	17
Functional User Requirements: Level of Granularity and FPA.....	19
Use Relevant Industry Data as a Valuable Outside View for Your Analysis.....	22
Committee Reports	
Communications and Marketing Committee.....	24
Functional Sizing Standards Committee.....	24
Non-Functional Sizing Standards Committee.....	25
International Membership Committee.....	25
ISO Standards Committee.....	25
Committee Rosters.....	26
Board of Directors.....	27
Behind the Scenes.....	27
CFPS Certification Listing.....	28
CFPP Certification Listing.....	28
CSP Certification Listing.....	29
CFPS Fellows.....	29

IN THIS EDITION

ISMA¹² in Rome sounds like a great experience. The breadth of topics and confluence of metrics achieved represents a milestone for IFPUG. Antonio Fereal gives us a quick overview of this experience in this *Metrics Views* – if you are interested in metrics, some of these papers will interest you.

News from the other committees is also interesting – there is a lot going on right now. Worth a look. And, of course, we have some interesting articles for you. Capers Jones outlines the profile of activities in small projects compared to large projects and demonstrates one of the key reasons for the very different productivity rates achieved. David Herron reminds us of the key business value and imperatives that metrics support – and Charlie Tichenor reminds us of some core imperatives when establishing metrics goals.

On top of that, we have some strong advice and discussion on how and when to apply SNAP. Also, some strong technical articles on counting data warehouses, on defining an appropriate level of detail when sizing ...and much more.

Entrée, main meal and dessert – it's all in this *Metrics Views*.



Tom Cagley

Message from the President

Fellow IFPUGers

The past few months have been an exciting time within IFPUG. In May, the Board met at the GUFPI-ISMA Conference in Rome. Italy has the largest CFPS and CFPP communities. Approximately 350 people attended and the Board enjoyed meeting many of the Italian members.

A few of the highlights from accomplishments include:

- The expansion of Country Representative role within IFPUG. Country “Reps” help represent IFPUG in their specific country and assist the Board of Directors, the International Membership Committee Chair and the IFPUG Office in any IFPUG-related actions executed in their respective country/geographical area.

- The Non-functional Sizing Standards Committee submitted a SNAP article to Elsevier’s Special Issue on software measurement.

A few future things coming include:

- An “Un-conference” on September 16th near Baltimore, Maryland. I am sure the Conference and Education Committee will have a lot more to say about that in the near future.
- The Functional Sizing Standards Committee (FSSC) has multiple white papers to be approved in the near future.

All of us on the Board recognize that IFPUG is an association and that your needs and ideas should drive the association. Getting involved with ANY of IFPUG’s committees is one avenue to help steer the organization. If committee membership isn’t your style consider attending one of the monthly virtual coffees I am holding. I am rotating through the time zones with significant IFPUG

(continued on next page)

(President's Message, continued from page 2)

membership. The Virtual Coffees are a platform to talk about IFPUG and just plain SNAP or function points.

During the Board's recent meeting we spent a significant portion of our time identifying several steps needed to speed up decision-making and to make the organization more responsive to the functional and non-functional sizing community across the globe. We are currently exploring options to speed up the decision-making process and flow of information from the Board. I would like to entertain your ideas on how IFPUG can serve you better. Come to the Virtual Coffee and feel free to send me an email at president@ifpug.org.

Tom Cagley
IFPUG President

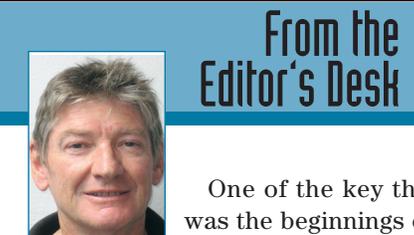
data impacted. Three potential levels are set and variances in FUR size are limited. COSMIC, on the other hand, has forever been worried about very large functions and very small functions having much the same size "benefit". Consequently, many individuals have spent a lot of time and effort in defining a methodology which enables one to break down a function into logical sub-particles which can be then summed to a size for each identified function. Going the other direction, Simple Function Points asserts that, on average, most applications will end up ...well, average. In other words, whilst some functions are "under-sized" and others "over-sized", the end result is the same. And achieved with far less cost and time.

Then there are the "automated" function point counting tools. The fact is, they don't work. Not to count function points as we know them. But isn't automated sizing a useful goal? Should we be working to make this a possibility, to review and refine our rules to encompass extended capabilities for the method?

IFPUG is currently "treading water" in relation to all of these aspects. But, eventually, one has to strike out for the shore – and it is time for IFPUG to choose a direction or the weight of those 360 pages of existing rules will simply drag us down. Where are we going? Are we looking to make IFPUG more practicable, more accessible? Do we need a method that is more repeatable, more consistent? Do we need greater detailed definition, such as COSMIC promotes? Do we see an eventual synching up with other measures, or even a range of methods applicable to different problems?

For IFPUG to be relevant in the longer term, it has to start providing some leadership. Now would be a good time.

Paul Radford
Communications and Marketing Committee



One of the key things arising in ISMA¹² was the beginnings of confluence between IFPUG and other related measurement organisations. All the decent software metrics are based around the function point analysis paradigm. However, the optimum method of applying this basic approach has not always been agreed.

Most agree, largely, on the definition of a function, or FUR. But from there, pathways split. IFPUG has never changed from initial settings, with levels of complexity set according to the

	Q/P MANAGEMENT GROUP, INC.	<i>Providing Industry Leading Software Measurement Consulting since 1990</i>
Software Benchmarking including Agile		
<ul style="list-style-type: none">• Worlds Largest and most accurate FP based database• Measures productivity, effort, quality, schedule and more• New Development, Enhancement and Maintenance		<ul style="list-style-type: none">• Certified FPA and SNAP Training• Measurement and Estimation• Productivity, Quality and Cost
www.qpmg.com moreinfo@qpmg.com		North America: +1 781.438.2692 Europe: +44 20.3287.921

MetricViews

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

MetricViews August 2016

Editor

Paul Radford

Assistant Editor

David Herron

IFPUG Board of Directors

President

Tom Cagley

Vice President

Mauricio Aguiar

Secretary and Director of

Certifications

Christine Green

Treasurer

Debra Maschino

Immediate Past President

Kriste Lawrence

Director of

Communications & Marketing

Carol Dekkers

Director of Sizing Standards

Dácil Castelo

Director of International

& Organizational Affairs

Pierre Almén

Director of Education

& Conference Services

Luigi Buglione

IFPUG Office

Executive Director

Gabrielle Copperwheat

Association Coordinator

Michele Giovine

Membership Coordinator

Nicole Lauzon

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

(609) 799-4900

ifpug@ifpug.org

Finding the Right Service Level Measure in a Changing Outsourcing Landscape

by David Herron

A number of recent articles and reports on the state of outsourcing point to a trend towards smaller/ shorter term deals and an increase in using multiple vendors; usually referred to as multi-sourcing. As a point of reference, multi-sourcing typically includes the providing of IT services using an optimal combination of domestic and offshore providers.

But, for as much as outsourcing arrangements may be changing and as customers continue to look for greater efficiencies, there is still much that remains the same in regard to contract governance challenges. The ideal governance model seeks to find a balance between price and value; therefore, service levels must establish measures that will key in on those two variables.

The Changing Landscape

Several recent articles in CIO magazine make note of a changing landscape. Among some of the changes reported were the following –

- Shorter and smaller contracts are increasing. Three to five year deals have steadily increased. 26% in 2010 to 31% in 2013 according to Everest Group.
- IT outsourcing contract counts in the Americas were at an all time high in 2014. However there has been a decade long decline in the size of outsourcing deals. Sub 100-million dollar deals make up about 70% of the contracting activity.
- Sourcing to multiple providers has become a predominant trend according to the Information Services Group (ISG).
- Mature buyers show greater adoption of multi-sourcing and less mature markets that typically did sole-sourcing are now moving to multi-sourcing.
- There is a strong anti-incumbency sentiment among IT service buyers. Companies terminated approximately 27% of their existing IT infrastructure deals. Factors included disappointment with service delivery and desire to unbundle services.
- Trending towards IT service providers specializing in specific industry or technology capabilities.

Governance Issues

John Keppel, from the Information Service Group, points out that several dominant issues still plague multi-sourcing arrangements, including -

- Unclear delineation of responsibilities
- Lack of supplier Collaboration
- Contracting issues
- Impotent Governance

These are not new to the list of outsourcing contract problems. Clearly defined roles and responsibilities, identified risk mitigation practices and proper service level measures are key components to successful multisourcing governance.

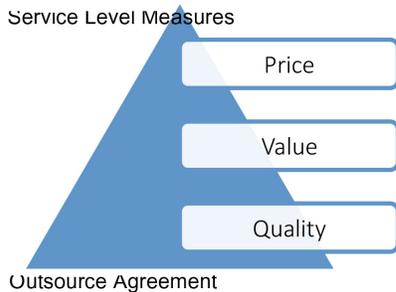
A Proper Service Level Measure

Application Development and Maintenance (ADM) outsourcing arrangements and contracts are commonly priced on the basis of labor cost. However, price (cost of labor) is just one of several dimensions that need to be considered when establishing

(continued on next page)

(Finding the Right Service Level, continued from page 4)

service level measures that serve to guide a successful outsourcing engagement. The other two key measures, in addition to price, are value delivered and quality of the deliverable.



A successful outsourcing arrangement is one that delivers a high quality product that meets the needs of the business (value) for a reasonable price. Service level agreements and measures need to be established to take these three dimensions into account. The following discussion focuses on a measurement technique that has been successfully used on numerous ADM outsourcing arrangements to measure these three dimensions.

Price and value; these two measures serve as equal partners in the equation. Simply stated, you want to measure your outsource providers level of service to ensure the delivery of real value at a reasonable price. An ideal scenario would be to have one single measure that would serve to measure both cost and value. For example, in manufacturing, output is often measured as a cost per unit of work; a unit of work representing a high quality finished product. Wouldn't it be nice to have a cost per unit of work for software?

How might a unit of work for software be defined? Software delivers value to the customer in the form of business functionality. A customer typically wants the ability to input data, send data outbound, make inquiries into their system, store data in their system for future use and to interface or communicate with other systems. Simply stated, these are the key functions that deliver and provide business value to the user. Therefore, we want a measurement technique that would size the functionality (value) we are delivering to the end user.

Function Point analysis is an industry accepted software sizing method that measures the functionality of delivered software. In brief, the methodology calls for the identification of five key components that make up the software deliverable; input data, output data, system inquiries, internal storage of data and interface files. Each of the five components are evaluated based on their individual complexity and then weighted and added together to derive a total function point size which can be used as a measure of the total functionality being delivered by the software. This sizing mechanism could be considered a unit of work measure; one function point equals one unit of work.

Before selecting an outsourcing vendor, an organizational function point baseline should be established. This is accomplished by sizing a representative sampling of recently completed projects. The resulting size of each project can be coupled with the total labor cost of each project and a cost per function point level of performance can be established thereby creating a cost per unit of work measure!

This size and cost information should be used to develop a pricing matrix. Not all projects are of equal value nor is it accurate to assume a single price point for all projects. A simple pricing matrix would be developed that showed price ranges based on technology and project size. In the simple example below, the technology column would include the various technology configurations which may include some combination of development platform and language complexity. For each configuration and FP size, a cost per Function Point would be determined based on the results of the function point baseline.

	Functional Size Range			
Technology	150 - 350	351 - 750	751 - 1250	> 1250
Configuration 1	\$550	\$750	\$1050	\$1200
Configuration 2	\$800	\$925		
Configuration 3	\$600		\$770	
Configuration 4	\$1100		\$750	
Configuration 5	\$490	\$510	\$590	\$650

Each cell represents a cost per Function Point based on size and technology

Armed with this information, an organization now has the basis for a sound negotiating position with a third party provider. Once negotiated, similar measures can be part of the service level measures established for proper governance of the successful engagement. Vendor work product deliverables can be based on this cost per unit of work. You also now have the added advantage of a sizing technique that allows you to size the resulting deliverable to ensure you received all the features and functions you required.

The function point size metric can also be used to effectively measure the quality of the software deliverable. There are several common function point based quality metrics, the most notable one being defect density. This is often calculated as the number of delivered defects per 1000 FPs. During the baselining process described above, the organization can collect defect data, along with function point size and cost, for each project in the baseline.

In summary, whether sole-sourcing or multisourcing, outsourcing arrangements should include service level agreements that measure both cost and value. These contracts are more likely to be successful and well governed. It is to both parties advantage to consider the metrics discussed above. The value for the buyer is an assurance that business value deliverables are right priced. The upside for the provider is to be able to demonstrate that what was promised was delivered within budget.

Future for Metrics and Quality in an IoT Cloud World

by Steven Woodward, CEO Cloud Perspectives

Your company has just completed development of a great application solution that leverages new Internet of Things (IoT) components.

The application has been rigorously tested in a comprehensive manner, meeting and exceeding all functionality, performance and security tests. The associated connectivity points are based on “state of the art” Software Defined Networking (SDN) and Network Function Virtualization (NFV) frameworks.

It gets rolled into production and, voila! Multiple outages and unsatisfactory customer experiences. Does this sound familiar?

When new technologies are embraced and operationalized, they usually fail sooner rather than later.

The IoT, SDN, NFV and Cloud systems are developing into perfect storms that will take time to develop, and to then move onward to maturity, eventually providing calm, consistent conditions.

Based on historical data (metrics) we know that as new technology is introduced, learning curves are required, mistakes are made and defects appear out of nowhere. Now, let’s think about additional security risks and threats that are more often the major targets on the radar. Moving forward, the risks intensify further. Fortunately, technical staff are inherent optimists (something that usually surprises the business), therefore everything is fine, until things go wrong.

So does this mean we should never use new technology?

Of course not! However, it does mean that the developers and testers must be allocated adequate budget, schedule, and tools to plan for failure so that risks can be mitigated and benefits realized.

Remember, I said the technical staff are optimists, so the reality is that they will underestimate what is really required.

The technical community must clearly communicate the risks (financial, legal and credibility) and the multiple points of potential failure and/or security vulnerabilities so stakeholders can make informed decisions.

Metrics provide critical pieces of information in this new world of IoT, Cloud and Software Defined Everything. Decision making, monitoring, management and overall governance will be tremendously more challenging without solid repeatable metrics that are align with the specific objectives, addressing specific opportunities.

About the author:

Steven Woodward, CFPS, CSQA
Cloud Perspectives, Ottawa, Canada

Steven helps companies and governments provide, plan and leverage cloud services for competitive advantage. He co-leads the Cloud Audit and Cloud Carrier sub-groups with the National Institute of Standards for Technology (NIST). As a Cloud Security Alliance Canadian Chapter Director, he fosters collaboration and cooperation to clarify realistic cloud controls that build trust. His other contributions and leadership include ITU-T (under the United Nations), IEEE, TM Forum, Object Management Group (OMG), Canadian Cloud Council, International Institute of Business Analysts (IIBA), ISO/IEC JTC1 as part of the Standards Council of Canada and Chair of the IFPUG ISO Committee. Steven is also serving as a Canadian Advanced Technology Alliance (CATA) lead on the Shared Services Canada (SSC) Architecture Framework Advisory Committee.

The logo for DVG (Data Value Group) consists of the letters 'D', 'V', and 'G' in a bold, serif font. The 'V' is positioned between the 'D' and 'G', and all three letters are interconnected.

Est. 1994

Software Value

www.softwarevalue.com

- The leading independent provider of software sizing.
- Certified Function Point Counters and Certified SNAP Practitioners.
- Training, consulting and customized services, available in North America and Europe.

ISMA¹², an amazing event: thanks Italy

by Antonio Ferre Albero

“Amazing” could be the word that summarizes a very successful ISMA¹² (the 12th IFPUG International Software Measurement & Analysis Conference) titled as “Creating Value from Measurement” and celebrated on May 3-5, 2016 in Rome, Italy.

Multiple factors contributed to this success. One of them is that the last ISMA conference celebrated in Italy was 20 years ago: we enjoyed meeting the people that were there then, looking backward and looking forward - at the same time - in metrics matters.

A second key success factor was the selection of interesting topics discussed during those three days, and the senior expertise level of the speakers.

On the morning of May 3, the **workshop** about Automated Function Points was presented by Bill Curtis, Executive Director of Consortium for IT Software Quality CISQ, and by Philippe-Emmanuel Douziech, Principal Research Scientist of CAST. On the afternoon Thomas Fehlmann, from Euro Project Office AG, conducted a **workshop on** “Sizing Software for Various Characteristics such as Non-Functional Requirements, Safety or Security: A Generic Excel tool for Assessing Software based on ISO/IEC 20926 and ISO/IEC 19761”.

The second day, May 4, Roberto Meli, CEO of DPO, led the **third workshop** about a New Unified Model of Custom Software Costs Determination in Contracts. This same day other interesting events took place: the **IFPUG CSP Exam**, the **IFPUG Board meeting with the Italian Community** in an welcoming way, the **MAIN** (Metrics Associations International Network) **meeting**, and a **social dinner** where in an interesting and relaxing atmosphere we shared experiences, points of view and synergies.

On the third day, May 5, many different topics were presented at the **ISMA¹² Conference**. Christine Green and Luca Santillo, from IFPUG and from COSMIC respectively, presented the topic “Accounting for Non-functional and Project requirements: COSMIC and IFPUG developments.” Although COSMIC and IFPUG have tackled how to measure the non-functional requirements in different ways, both have created a glossary of terms for non-functional requirements and for project requirements and constraints. The collaboration and synergy between COSMIC and IFPUG for sure brings great success to the IT metrics community.

Gianfranco Lanza, from CSI Piemonte, presented the topic “Functional User Requirements (FUR) and Non-Functional Requirements (NFR): two sides of the same coin,” answering a set of challenging questions such as how to distinguish between FUR and NFR; how Snap Points coexist can with Function Points; whether or not all software development activities can be measured; and many other interesting questions.

Thimoty Barbieri discussed “Leveraging Enterprise Architectural Standards for Automated Function Point Analysis in Agile/ADM Processes”, dealing with motivating topics such as mapping Counting Practices between IFPUG CPM FPA and Java Enterprise JPA 2.1 and JSF 2.1 that could allow automatic sizing and provide metrics for Service Level Agreements, and even a plug-in for SonarQube that can analyze code and reverse-engineer a functional sizing measurement.

Raúl David Fernández Rodríguez, LEDA-mc Research & Development Manager, introduced and gave the interesting presentation “Software Rates vs Cost per Function Point, Productivity and Quality.” The company LEDA-mc has been analyzing the relationship between the Software Rates and the cost per Function Point (using its own database of more than 18.000 software development projects) showing the relationship of Software Quality with Productivity in real scenarios.

(continued on next page)



(ISMA¹², an Amazing Event, continued from page 7)

Then Piergiacomo Ferrari, IT Quality Manager at Allianz Italia, discussed the stimulating case “Risk and AFP Measurement in a digital transformation program, Allianz Italia use case,” explaining his experience in preventing operational risks, and how they set up the contextual measurement of the functional size with AFP, in order to measure the baseline size and the vendors productivity.

Luigi Lavazza, Professor at Università dell’Insubria, and Sandro Morasca presented the outstanding topic “Some considerations on Function Points (as a measure),” showing that the concepts and practices used in Function Point Analysis are commonly used in other areas and disciplines, illustrating with examples such as the value of an apartment: it is shown that the value and sale price are affected by both characteristics of the apartment and factors that depend on the environment.

Caterina Trovato presented the case study (and CEC-eligible conference) “Translating information flows in the Electricity/Gas market with FPA,” applying IFPUG FPA in the Energy and Utilities domain using multi-component architectures, discussing interesting

aspects and guidelines applied such as counting web services, and developing a set of different lessons learned.

The last presentation was given by Roberto Meli, SiFPA president and CEO of DPO, that discussed the management viewpoint of Software Measurement, dealing with motivating topics such as approximate approaches are often preferred to rigorous approaches, and “better” is preferred to “best”.

At the same time, and during the ISMA¹² Conference, the IFPUG Board recognized the new CFPS (Certified Function Point Specialist) Fellows who have 20 or more years of continuous certification. Thanks for your loyalty/fidelity, and becoming CFPS Fellows:

- Anna Battistata
- Massimo Beretta
- Steven Keim
- Makato Kurashige
- James Mayes
- Maurizio Menghini
- Franco Perna
- Janet Russac
- Luca Santillo
- Walter David Thompson
- Daniele Zottarel



Congratulations too, to Mattia Pignalosa and to Michele Pasquale for the GUFPI-ISMA Students’ Awards received.

The large number of attendees at the ISMA¹² conference on May 5 was another success: more than 320 “first players” metrics experts from 16 countries (Italy, Argentina, Belgium, Brazil, Canada, China, Denmark, France, Germany, Korea, Netherlands, Poland, Spain, Sweden, Switzerland, and USA) attended. At the same time, this gave testimony that Italy is a mature and very active country regarding IT metrics and functional size metrics, among others.

And another success factor, in the technical as well as the human aspect: was the “savoir faire” of the GUFPI-ISMA (Gruppo Utenti Function Point Italia Italian Software Metrics Association) members who made us feel like at home. Thanks to Luigi Buglione, Tommaso Lorio, Filippo De Carli, Gianfranco Lanza, Guido Moretto, Luca Santillo, and thanks even more to those of you that worked behind the scenes to contribute to the success of the conference.

The 12rd IFPUG International Software Measurement & Analysis Conference (ISMA¹²), celebrated in Centro Congressi Frentani, Rome, less than 10 minutes by foot from the Rome Termini train station, was organized by GUFPI ISMA on behalf of IFPUG, sponsored by the companies CAST, DPO (Data Processing Organization), Engineering Ingegneria Informatica, Euro Project Office, ESTIMANCY, LEDAmc, and ti MÉTRICAS, and with more than a dozen of partners.

Thanks to all of you who made this ISMA¹² such an amazing event, congratulations and ... **thanks Italy!**



ISMA 12 dinner

Datawarehouse FP Counting Guide Used In Brazilian Government Contracts

by Luiz Flavio Riberio

There are growing demands for IT services in government agencies in Brazil. The need to outsource these services to software factories has made it complex to comply with legal directives by government auditors. The use of metrics that would enable the objective examination of size, time, cost and quality has become necessary. By normative regulation by TCU (a tribunal that scrutinizes federal government budgets), the Ministry of Planning drew up a specific standard, which resulted in the publication of Instruction #4, in 2010. It recommended the use of Function Point Analysis Unadjusted in order to limit the use of Man / Hour metrics. The Man/hour metrics can't be audited so it can't provide transparency to government IT contracts.

The IT government services are solicited from the market and the bid with the lowest price of the FP delivered is hired. That price can be set either to hire the whole project including management or to hire only one software project cycle; e.g., the construction phase. But, as the FPA does not measure the effort, customers and suppliers have begun to question the costs of services that either were too high to do a little work, or too low for a great effort. The impact caused by the non-functional requirements must be negotiated to adjust the price paid for a service.

In 2010, under the coordination of the Ministry of Planning, a group of representatives from various government agencies (customers and IT vendors) was formed. The objective was to achieve an agreement that would minimize the financial impact caused by the use of FPA. This agreement would

propose a way to balance the remuneration for executed services considering the effort. This work resulted in the publication of the 1st version of the SISP Guide (now in version 2.1), an official counting guide that tracks IT contracts in Brazilian government agencies.

This Guide transcribes the agreements set between the parts, customers and IT vendors. Generally, indexes (or deflators) are set to be applied in an FP count to balance the cost/effort relation in each functionality. That count is converted into currency through the price of the FP hired.

For example, a maintenance service in an application a 50% deflator is applied in the FP count of the impacted functionality. It's because changing a feature already built requires a lesser effort than to develop it from the beginning. This percentage considers the requirements gathering, changes in code, tests, documentation and approval.

Issues that impact the software development productivity were addressed. Changes in user interfaces, static web pages development, language and database version updates, services reuse etc, for all these issues the effort was considered in the development. Scope creep was also addressed contemplating the phase in the software life cycle, at the time of the changes. The count for agile projects was the theme of the latest version of the Guide, but it is beyond the scope of this article.

Only transactional systems were addressed in the first versions of this Guide and it was necessary to treat issues inherent in the analytical systems. To open discussions, again under the coordination of the Ministry of Planning,

a group of government agencies representatives was formed. I attended it as a representative of Dataprev, a state-owned enterprise that provides IT services to Brazilian Social Security. This group developed a guide to the FP counting of Data Warehousing environments. It was verified that DW FP counting guides which were attending these institutions individually, were mostly based on the DW NESMA method (2014). So, the group decided this method would be the foundation for the development of the Guide Count SISP Function Points for DW Projects-V1.0.

Briefly, the method NESMA establishes that:

- The DW boundary is unique;
- Data Staging Area (DSA) and Operation Data Store (ODS) are disregarded in the count, unless there are specific requirements for data searching in these areas;
- Fact tables are identified and counted as ILFs;
- Dimensions that are directly linked to the Fact tables are also identified and counted as ILFs;
- Dimensions Static are related to code data and are then disregarded in the count;
- Hierarchical dimensions that qualify the main dimensions are considered logical records of these dimensions;
- ETL processes (Extraction, Transformation and Load) are counted for each ILF identified as well as for the tables that comprise the Logical Records of the hierarchical dimensions.
- The generation of the Cube is

(continued on next page)

(Datawarehouse FP Counting Guide, continued from page 9)

counted as an EO. The cubes are three-dimensional tables used for consumption by the Datamart

Also according to NESMA, data extraction from a source transactional system in order to load the Data Warehouse, through the ETL process, happens because of two scenarios:

1) Flat files generation on the border of the source system: an EQ or EO is identified to generate the flat file on the source system. After transforming and cleaning, data are loaded (EI) into DW tables;

2) Direct interface between the source system database and the DSA: In this case, these data are not stored in DW tables, but they are available to the application.

In the Guide, based on the analysis of the counting rules described in the CPM (chapter “Sharing of Data”), the following data mining scenarios were extended presented in NESMA to 3 scenarios, as described below:

Scenario 1: A copy of the source system data. The flat file is a copy or image of a source system ILF, without additional processing for its generation. In the ETL process only the DET's used in the functions identified to load DW tables are considered.

Scenario 2: The flat files are recorded in the source system boundary. When there is a business need to transfer data from the source system to DW, and the DET's group is different in the two systems, only one EQ or EO to generate that flat file in the source system boundary is identified. Later in the ETL process, the flat file data are loaded into the DSA for processing transformations and data cleansing. After that they are loaded (EI) in the DW ILF tables.

Scenario 3: direct interface between the source system database and the DSA. No data transference from the source system is performed into DW, but they are available for reading (Example: EQ by Web Service). For counting purposes,

it is considered as described in Scenario 1: the source system data are made available by Web Service and are DET's in the EI function to load DW tables.

Still in the ETL process, the extraction and initial loads (historical basis) and incremental (periodic) are considered, respecting the criteria of the elementary process uniqueness.

In enhancement projects, many common issues relating to transactional and DW environments are treated in the same way. For example, if there are requirements for creating new metrics and/or adding new attributes in ILF, the 50% deflator will be applied in the count of the affected data and transaction functions, such as those presented in the ETL process and in the generation of the Cube. Also the following issues were addressed, among others:

- Data Updates on Static Dimensions;
- Tier of benches reorganization (repositioning items in the user interface);
- Report filters creation;
- Deletion of the expired data.

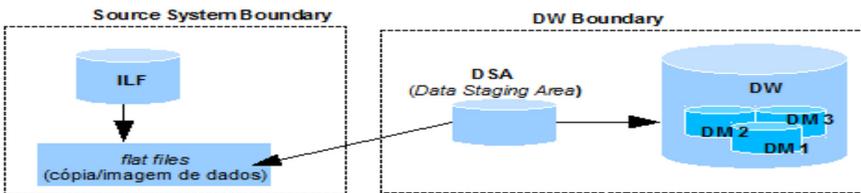
New versions of this Guide are periodically published to announce new negotiations, including those that arise with the advents of new technologies. Developed to attend the government agencies, it is often mentioned in private sector contracts. The Guide objectively provides a fairer remuneration when it is used in addition to the CPM rules, and has effectively collaborated in the implementation of the FPA in Brazilian software contracts.

References

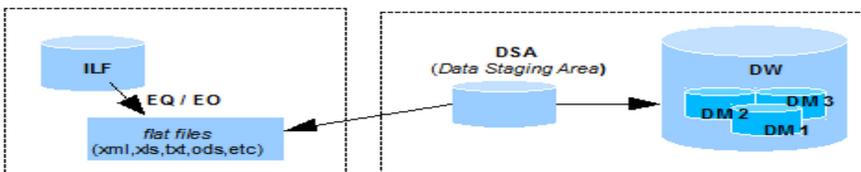
Guide Function Point Count Data Warehousing V1.0 projects

FPA applied to Data Warehousing (NESMA 2014 - V1.2)

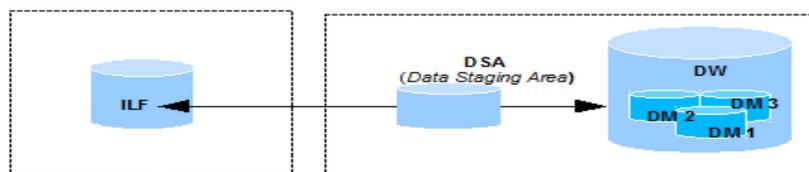
Scenario 1: flat files are a copy of the source system tables



Scenario 2: flat files are recorded into the source system boundary



Scenario 3: direct interface between the source system data and the DSA



Variations In Software Development By Function Point Size

by Capers Jones

Abstract

Building small applications and building large systems are very different. Small software projects of a 100 function points can be built by a few developers and a few other personnel. Large systems in the 10,000 function point size range require many skilled specialists such as business analysts, architects, and data base analysts who may not be needed for small projects.

This paper shows typical results for three size ranges: 100, 1,000, and 10,000 function points. The data comes from benchmark studies.

The differences in function point size leads to very different kinds of development practices and to very different productivity rates at the low end compared to the high end. For example for some large systems finding and fixing bugs and creating paper documents cost more than the code itself.

Capers Jones

Vice President and CTO, Namcook Analytics LLC

Email: Capers.Jones3@gmail.com

Blog: <http://Namcookanalytics.com>

Web www.Namcook.com

Copyright 2016 by Capers Jones.

All Rights Reserved.

Variations in Software Development by Function Point Size

In many industries building large products is not the same as building small products. Consider the differences in specialization and methods required to build a rowboat versus building an 80,000 ton cruise ship.

A rowboat can be constructed by a single individual using only hand tools. But a large modern cruise ship requires more than 350 workers including many specialists such as pipe fitters, electricians, steel workers, painters, and even interior decorators and a few fine artists.

Software follows a similar pattern: Building large system in the 10,000 to 100,000 function point range is more or less equivalent to building other large structures such as ships, office buildings, or bridges. Many kinds of specialists are utilized and the development activities are quite extensive compared to smaller applications.

Table 1 illustrates the variations in development activities noted for the six size plateaus using the author's 25-activity checklist for development projects:

Table 1: Development Activities for Six Project Size Plateaus

Activities Performed	1 Function Point	10 Function Points	100 Function Points	1000 Function Points	10,000 Function Points	100,000 Function Points
01 Requirements	X	X	X	X	X	X
02 Prototyping				X	X	X
03 Architecture					X	X
04 Project plans				X	X	X
05 Initial design		X	X	X	X	X
06 Detail design			X	X	X	X
07 Design reviews					X	X
08 Coding	X	X	X	X	X	X
09 Reuse acquisition	X	X	X	X	X	X
10 Package purchase					X	X
11 Code inspections				X	X	X
12 Ind. Verif. & Valid.						
13 Change control				X	X	X
14 Formal integration				X	X	X
15 User documentation			X	X	X	X
16 Unit testing	X	X	X	X	X	X
17 Function testing			X	X	X	X
18 Integration testing				X	X	X
19 System testing				X	X	X
20 Beta testing					X	X
21 Acceptance testing				X	X	X
22 Independent testing						
23 Quality assurance						X
24 Installation/training				X	X	X
25 Project management			X	X	X	X
Activities	4	5	9	18	22	23

Below the plateau of 1000 function points (which is roughly equivalent to 100,000 source code statements in a procedural language such as COBOL) less than half of the 25 activities are normally performed. But large systems in the 10,000 to 100,000 function point range perform more than 20 of these activities.

To illustrate these points table 2 shows quantitative variations in results for three size plateaus, 100, 1,000, and 10,000 function points:

(continued on next page)

(Variations In Software, continued from page 11)

Table 2: Powers of Ten for 100 ,1,000 and 10,000 Function Points

Size in Function Points	100	1,000	10,000
Examples	Medium update	Smart Phone app	Local System
Team experience	Average	Average	Average
Methodology	Agile	Iterative	Hybrid
Sample size for this table	150	450	50
CMMI levels (0 = CMMI not used)	0	1	1
Monthly burdened costs	\$10,000	\$10,000	\$10,000
Major Cost Drivers (rank order)			
1	Coding	Bug repairs	Bug repairs
2	Bug repairs	Coding	Paperwork
3	Management	Paperwork	Coding
4	Meetings	Management	Creep
5	Paperwork	Meetings	Meetings
6		Integration	Integration
7		Creep	Management
Programming language	Java	Java	Java
Source statements per function point	53.00	53.00	53.00
Size in logical code statements (author's default for LOC)	5,300	53,000	530,000
Size in logical KLOC (author's default for KLOC)	5.30	53.00	530.00
Size in physical LOC (not recommended)	19,345	193,450	1,934,500
Size in physical KLOC (not recommended)	19.35	193.45	1,934.50
Client planned schedule in calendar months	5.25	12.50	28.00
Actual Schedule in calendar months	5.75	13.80	33.11
Plan/actual schedule difference	0.50	1.30	5.11
Schedule slip percent	9.61%	10.43%	18.26%
Staff size (technical + management)	1.25	6.50	66.67
Effort in staff months	7.19	89.72	2,207.54
Work hours per month (U.S. value)	132	132	132
Unpaid overtime per month (software norms)	0	8	16
Effort in staff hours	949.48	11,843.70	291,395.39
IFPUG Function points per month	13.90	11.15	4.53
Work hours per function point	9.49	11.84	29.14
Logical Lines of code (LOC) per month (Includes executable statements and data definitions)	736.83	590.69	240.09

Physical lines of code (LOC) per month (Includes blank lines, comments, headers, etc.)	2,689.42	2,156.0	876.31
Requirements creep (total percent growth)	1.00%	6.00%	15.00%
Requirements creep (function points)	1	60	1,500
Probable deferred features to release 2	0.00	0.00	2,500
Client planned project cost:	\$65,625	\$812,500	\$18,667,600
Actual total project cost	\$71,930	\$897,250	\$22,075,408
Plan/Actual cost difference	\$6,305	\$84,750	\$3,407,808
Plan/Actual percent difference	8.77%	9.45%	15.44%
Planned cost per function point	\$656.25	\$812.50	\$1,866.76
Actual cost per function point	\$719.30	\$897.25	\$2,207.54
Defect Potentials and Removal %			
Defect Potentials	Defects	Defects	Defects
Requirements defects	5	445	6,750
Architecture defects	0	1	27
Design defects	25	995	14,700
Code defects	175	2,150	30,500
Document defects	11	160	1,650
Bad fix defects	15	336	3,900
TOTAL DEFECTS	231	4,087	57,527
Defects per function point	2.31	4.09	5.75
Defect removal efficiency (DRE)	97.50%	96.00%	92.50%
Delivered Defects	6	163	4,313
High-severity defects	1	20	539
Security flaws	0	3	81
Delivered Defects per Function Point	0.06	0.16	0.43
Delivered defects per KLOC	1.09	3.08	8.14
Test Cases for Selected Tests	Test Cases	Test Cases	Test Cases
Unit test	101	1,026	10,461
Function test	112	1,137	11,592
Regression test	50	512	5,216
Component test	67	682	6,955
Performance test	33	341	3,477
System test	106	1,080	11,012
Acceptance test	23	237	2,413
TOTAL	492	5,016	51,126

(continued on next page)

(Variations In Software, continued from page 12)

Test cases per function point	4.92	5.02	5.11
Probable test coverage	95.00%	92.00%	87.00%
Probable peak cyclomatic complexity	12.00	15.00	> 25.00

Document Sizing

Document Sizes	Pages	Pages	Pages
Requirements	40	275	2,126
Architecture	17	76	376
Initial design	45	325	2,625
Detail design	70	574	5,118
Test plans	23	145	1,158
Development Plans	6	55	550
Cost estimates	17	76	376
User manuals	38	267	2,111
HELP text	19	191	1,964
Courses	15	145	1,450
Status reports	20	119	1,249
Change requests	18	191	2,067
Bug reports	97	1,048	11,467
TOTAL	423	3,486	32,638

Document set completeness	96.96%	91.21%	78.24%
Document pages per function point	4.23	3.49	3.26

Project Risks	Risk %	Risk %	Risk %
Cancellation	8.80%	14.23%	26.47%
Negative ROI	11.15%	18.02%	33.53%
Cost overrun	9.68%	15.65%	34.00%
Schedule slip	10.74%	18.97%	38.00%
Unhappy customers	7.04%	11.38%	34.00%
Litigation	3.87%	6.26%	11.65%
Technical debt/high COQ	5.00%	16.00%	26.21%

Cyber attacks	7.00%	9.75%	15.30%
Financial Risk	9.00%	21.00%	41.00%
High warranty repairs/low maintainability	6.00%	14.75%	32.00%
RISK AVERAGE	7.83%	14.60%	29.22%

Project Staffing by Occupation Group	100	1,000	10,000
Programmers	1.91	6.23	43.53
Testers	1.85	5.66	38.58
Designers	0.51	2.13	18.00
Business analysts	0.00	2.13	9.00
Technical writers	0.44	1.05	7.00
Quality assurance	0.46	0.98	5.00
1st line managers	1.21	1.85	7.13
Data base administration	0.00	0.00	3.68
Project Office staff	0.00	0.00	3.19
Administrative support	0.00	0.00	3.68
Configuration control	0.00	0.00	2.08
Project librarians	0.00	0.00	1.72
2nd line managers	0.00	0.00	1.43
Estimating specialists	0.00	0.00	1.23
Architects	0.00	0.00	0.86
Security specialists	0.00	0.00	0.49
Performance specialists	0.00	0.00	0.49
Function point counters	0.00	0.07	0.49
Human factors specialists	0.00	0.00	0.49
3rd line managers	0.00	0.00	0.36
TOTAL STAFF	6.37	20.11	148.42

As can be seen from Table 2 what happens for a small project of 100 function points can be very different from what happens for a large system of 10,000 function points. Note the presence of many kinds of software specialists at the large 10,000 function point size and their absence for the smaller

(continued on next page)

CURRENT CONTACT INFORMATION?

To ensure you won't miss out on any IFPUG communications, please log in to your profile on the IFPUG Members Services Area and update your information.

Go to www.ifpug.org

Send emails to ifpug@ifpug.org, call 609-799-4900 or fax 609-799-7032

Write to: IFPUG, 191 Clarksville Road, Princeton Junction, NJ 08550

(Variations In Software, continued from page 13)

sizes. As application size in function points goes up a number of problems get worse:

Table 3: Problems of Large Software Applications

1. Requirements completeness declines
2. Requirements changes increase
3. Document volumes grow rapidly
4. Document completeness declines
5. Defect potentials increase
6. Defect removal efficiency (DRE) declines
7. Numbers of test cases increase
8. Test coverage declines
9. Cyclomatic complexity goes up
10. Risks of cancellation and delays go up alarmingly
11. Function point counting costs go up
12. Many large applications don't use function points

The software industry has done well for small projects but not for large systems. Function point metrics have been widely used for small applications but are seldom used above 10,000 function points due to the high cost and lengthy time interval required. There are several forms of high-speed function points such as pattern matching for new projects and automated counts for legacy applications, but manual counts by certified function point personnel remain the most common.

Summary and Conclusions

There are major differences in software development methods, software staffing, software quality, and software productivity between small applications of 100 function points and large systems of 10,000 function points or more. Small projects are generally successful and have fairly good quality and productivity. Large systems fail more often than they succeed and seldom have good quality and productivity.

REFERENCES AND READINGS

Abran, A. and Robillard, P.N.; "Function Point Analysis, An Empirical Study of its Measurement Processes"; IEEE Transactions on Software Engineering, Vol 22, No. 12; Dec. 1996; pp. 895-909.

Bogan, Christopher E. and English, Michael J.; Benchmarking for Best Practices; McGraw Hill, New York, NY; ISBN 0-07-006375-3; 1994; 312 pages.

Gack, Gary; Managing the Black Hole: The Executives Guide to Software Project Risk; Business Expert Publishing, Thomson, GA; 2010; ISBN10: 1-935602-01-9.

Humphrey, Watts S.; Managing the Software Process; Addison Wesley Longman, Reading, MA; 1989.

IFPUG Counting Practices Manual, Release 6, International Function Point Users Group, Westerville, OH; April 2015; 105 pages.

Jones, Capers; The Technical and Social History of Software Engineering, Addison Wesley, 2014.

Jones, Capers; Estimating Software Costs; 2nd edition; McGraw Hill; New York, NY; 2007.

Jones, Capers and Bonsignour, Olivier; The Economics of Software Quality; Addison Wesley, Boston, MA; 2011; ISBN 978-0-13-258220-9; 587 pages.

Jones, Capers; "A Ten-Year Retrospective of the ITT Programming Technology Center"; Software Productivity Research, Burlington, MA; 1988.

Jones, Capers; Applied Software Measurement; McGraw Hill, 3rd edition 2008.

Jones, Capers; Software Engineering Best Practices; McGraw Hill, 1st edition 2010.

Jones, Capers; Assessment and Control of Software Risks; Prentice Hall, 1994; ISBN 0-13-741406-4; 711 pages.

Jones, Capers; Patterns of Software System Failure and Success; International Thomson Computer Press, Boston, MA; December 1995; 250 pages; ISBN 1-850-32804-8; 292 pages.

Jones, Capers; Software Assessments, Benchmarks, and Best Practices; Addison Wesley Longman, Boston, MA; 2000 (due in May of 2000); 600 pages.

Jones, Capers; Software Quality – Analysis and Guidelines for Success; International Thomson Computer Press, Boston, MA; ISBN 1-85032-876-6; 1997; 492 pages.

Jones, Capers; The Economics of Object-Oriented Software; Software Productivity Research, Burlington, MA; April 1997; 22 pages.

Jones, Capers; Becoming Best in Class; Software Productivity Research, Burlington, MA; January 1998; 40 pages.

Kan, Stephen H.; Metrics and Models in Software Quality Engineering; 2nd edition; Addison Wesley Longman, Boston, MA; ISBN 0-201-72915-6; 2003; 528 pages.

McMahon, Paul; 15 Fundamentals for Higher Performance in Software Development; PEM Systems 2014.

Radice, Ronald A.; High Quality Low Cost Software Inspections; Paradoxicon Publishing Andover, MA; ISBN 0-9645913-1-6; 2002; 479 pages.

Wieggers, Karl A; Creating a Software Engineering Culture; Dorset House Press, New York, NY; 1996; ISBN 0-932633-33-1; 358 pages.

Yourdon, Ed; Death March - The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects; Prentice Hall PTR, Upper Saddle River, NJ; ISBN 0-13-748310-4; 1997; 218 pages.

Industry Best Practices V. Moral Hazard In Software Development

By Charley Tichenor

Introduction

I think that professionals should be aware of the industry best practices in their areas of expertise, and consider using them as situations arise. In my opinion, one best practice that applies to probably most fields is the reduction of the effects of “moral hazard” we might face.

Moral hazard is a term from economics. It occurs when a person performs an activity of some kind, but is not responsible for many or all of its consequences; this can influence how the activity is performed. For example, a new teen driver may not be able to afford automobile insurance, so the parents pay the insurance. Because the teen does not pay the insurance, the teen may be tempted to take more chances while driving. If the teen gets into an accident, it is then the parents who experience increased insurance rate expenses. The teen presents a moral hazard to the parents. When a manager hires a new employee, there is some degree of moral hazard in that it is not certain at that time that the new employee will choose to be productive.

The potential for moral hazard also occurs during the process of software development. The following is my opinion of two potential moral hazard situations, and is based on my experience. The paper’s focus is on software development cost, although moral hazards could be identified for other aspects of software development.

Theory

I am a relatively new Assistant Professor of Management Science. Part of the course requirements for my students during a typical semester is writing a paper. When I go over the paper requirements (to include a grading rubric!), somebody in the class invariably asks “What is the grading criteria?” Students often ask if I grade –

- By the number of words.
- By the number of sentences.
- By the number of pages.

I answer, “No, I grade by the content.”

A very similar question is posed when a value must be placed on new software, i.e., how much will it cost to build? Or, if a project has completed, was the amount of cost expended reasonable?

One of the first attempts at trying to quantitatively determine a value for software was by examining the program coding. Each line of code is basically a “sentence,” and the value of

software was determined by counting the number of its sentences. It seemed logical (to some, rather) that software with more sentences would take longer to write, and therefore cost more to deliver. Numerous contracts were priced back then in terms of number of sentences and are still done today. They are priced by some form of the metric, “dollars / line of code.”

As software metrics technology improved, many issues arose with using counts of program sentences as the measure of software value. Here are four of many:

- If we require a student to write a paper, imagine grading it by “number of sentences.” What do we incentivize the student to do? Should the grading standard be something like “900 sentences gets an “A,” 800 sentences gets a “B,” 700 sentences gets a “C,” and so on? No – number of sentences does not measure content. Determining software value by “number of lines of code” turns out to be much like grading by “number of sentences” – at least to me.
- Different new programming languages required different numbers of lines of code to program the same software content. Newer languages may require fewer lines than older languages.
- Different programmers will use a different number of lines of code to generate the same content.
- Lines of code metrics are easily gamed. To increase productivity by 10%, simply increase the number of lines of code written by 10% – but the content does not change. I actually measured this effect with a software development team!

An alternative to lines of code was developed in the late 1970’s at IBM by Dr. Allan J. Albrecht and his research team. This approach was a successful attempt to actually measure the “content” of the software – its data processing capacity. In a nutshell, the data processing capacity of software is called its “functionality,” and one individual unit of data processing capacity is the “function point.” Today’s improved function points are an ISO standard.

Using function points, it does not matter what language is used, or the skill of the programmer, or how many lines of code are used. The only thing that matters is the size of the content. Also, the size of the content cannot be “gamed.”

When we can buy “by the piece of content,” it is easy to forecast how much an order will cost. For example –

- Price per gallon of gas = \$2.30
- Number of gallons to buy = 10.
- Total forecast cost = \$23.00

(continued on next page)

(Industry Best Practices, continued from page 15)

The function point metric family solves the problem of measuring the value of the functionality of software. This value is the \$dollars / function point.

In my experience, good software development teams want to use function points. There is an economic demand, and the following questions are readily answerable.

- What is my cost per function point?
- How many days is it taking to build a function point?
- Are my costs and schedule forecasts within industry benchmarks?
- Can I document my value to the organization in terms of cost control?

The same general approach applies to SNAP.

Moral Hazards

Ethical problems arise in software development when the software team is presented with some form of economic incentive to “behave” in a way which is contrary to the goals of the organization. The goals of an organization’s software development program include delivering software at the forecast cost. However, team members may be incentivized not to reach those goals; many either experience or self-generate just the opposite. This presents a moral hazard to their senior management.

First Moral Hazard: the Software Team Which *Refuses* to Use Software Metrics. Some software development managers and/or team members I have known refuse to use software metrics for any one or more of the following reasons.

- They don’t want to be measured based on content.
- They don’t want to be measured against industry benchmarks.
- They don’t want to be held accountable for meeting measurable budgets.
- They are concerned about being measured inaccurately.

The moral hazard presented is that the team is being paid to deliver software in such a way to meet organizational goals, but there is no way senior management can determine how well the goals are being met until, basically, “after the money is spent.” If the team overspends, then the cost overrun responsibility shifts to senior management since no metric measures the effectiveness of the team.

Second Moral Hazard: Deliberately Using Lines of Code Instead of Function Points. Many organizations have decided to implement a software metrics program to try to quantitatively forecast and control cost. However, they choose the “lines of code metric” that is known not to work adequately.

Sometimes the choice to use lines of code is made with the knowledge of the huge inherent margins of error in that

metric. This amounts, to me, as another form of the first moral hazard example where software teams try to avoid full accountability.

Sometimes the choice reflects lack of qualified metrics analysts. Management wants good metrics, but their metrics analysts are under qualified and cannot provide the level of expertise needed.

Some developers appear to “welcome” lines of code metrics, which allows them to overcharge clients who are not aware of any other way.

For whatever reason, the outcome of such refusal is that numerous software projects go seriously over budget. And their uninformed top management may “have no idea why.” Again, the development action performed is by the project team, but the responsibility for the cost overrun is shifted to senior management.

Conclusion

Moral hazard may arise in software development when software development project teams are presented with some form of economic incentive or reward to “behave” in a way which is contrary to the goals of the organization. The goals of an organization’s software development program include delivering software at the forecast cost. For whatever reason, not all in software development are incentivized to try to reach those goals; many either experience or self-generate just the opposite.

One way to reduce these moral hazards is to have senior management ensure economic incentives point in the same direction as the ethical incentives. In my opinion, an organization may need one or more “honest brokers” who can educate and sell senior management on a function point-based metrics program, and ask senior management to install the discipline and resources to maintain it. Here are some suggestions that seem to work for me.

- The boss must understand the basics of the methodology and direct its institution. It would be good to put this decision in writing and place metrics procedures into the organization’s operating manual.
- The metrics team must either directly report or have an unobstructed avenue to either the boss or the deputy. They must not be placed deep into the organization so that their work product and requests for support are filtered by intervening layers of management.
- The software development organization needs to be trained in the function point-based metrics methodology.
- The metrics team must generate success after success in metrics program implementation.

SNAP for Sizing Corrective Maintenance

by Amit Arun Javadekar and Sarika Dandawate

Abstract

Corrective maintenance is defined as reactive modification of a software product performed after delivery to correct discovered problems. These reactive modifications (also known as defect fixes) are mostly combined with software enhancement requests to form a software release. While planning for a software release it is necessary to estimate the effort and duration required to deliver that release. While Function Points can be used to size the enhancements, there is no standard sizing method available for sizing defect fixes. Consequently estimating the effort and time required for defect fixes is person dependent and often subjective. Similarly other size dependent metrics like productivity cannot be tracked for defect fixes. This paper describes the authors' attempts to use SNAP to size defect fixes. It concludes by highlighting the challenges encountered in this experiment.

The concept

The application maintenance service has suffered from the lack of a standard sizing model for defect fixes for a long time. Consequently the standard estimation process of **Size -> Effort -> Duration -> Cost** is difficult to implement. Individual organizations have evolved effort estimation models for defect fixes most of which are complexity-based models, organization specific and whose output is difficult to compare with those of other models. The idea of using SNAP to size defect fixes is based on the following aspects of defect fixes:

- 1) They deal with **how** the software will meet the user's requirements
- 2) They are **technical** in nature in which FP is not applicable

Scope of the experiment

While defects can be of many different types, the scope of this experiment was limited to frequently occurring defects encountered in applications of the telecom domain. The experiment was limited to defects of the following types:

- data validation issues
- data formatting issues
- incorrect or incomplete configuration settings
- restrictions on user interface elements
- defects resulting in only code change with no database and/or UI changes

Using SNAP to size defects

Given below are examples of how SNAP was used to size the various types of defects.

1. Defect related to Data Validation

Problem: User reported that in a telecom application for a particular transaction, the output obtained was not as per expectations/requirements

Defect Analysis: The transaction had 20 input fields where trailing blank spaces were being allowed as part of the user provided input. These trailing blank spaces were the cause of incorrect output being generated

Solution: It was decided by the application maintenance team to validate the user input and disallow any trailing blank spaces.

SNAP Sizing:

Category: Data Operations

Sub-category: 1.1 Data Entry Validations

SCU Description	#DETs	# Nesting Levels	Complexity	Formula	SNAP Points
The identified transaction (1 Transaction)	20	1	Low	2* #DETs	2*20 = 40

2. Data Formatting related defect

Problem: In one of the applications, data synchronization issues were detected in a particular transaction.

Defect Analysis: It was found that the transaction had an input field where format for valid Month, Year and Quarter was incorrect. This was creating inconsistency in data synchronization.

Solution: It was decided to change the Data Formats required for Month, Year and Quarter to fix this defect.

SNAP Sizing:

Category: Data Operations

Sub-category: 1.3 Data Formatting

SCU Description	#DETs	Type of Data Formatting	Complexity	Formula	SNAP Points
The identified transaction (1 Transaction)	3	Date Format change	Low	2* #DETs	2*3 = 6

(continued on next page)

(SNAP for Sizing Corrective Maintenance, continued from page 17)

3. Defect due to incorrect configuration settings

Problem: The Sales application has a Forecast Engine that generates sales forecasts on a periodic basis. This Forecast Engine was not generating the expected results.

Defect Analysis: Data configured in a table referenced by the Forecast Engine is incorrect. Also the time at which the Engine should run as a batch process (Java Jar file) is not properly configured.

Solution: This defect can be fixed with configuration changes in the database as well as a configuration file.

Data base configuration changes: 1 column in a database table needs to be updated. This table has 6 records and one column in all the 6 rows needs to be updated.

Configuration file changes: Jar file needs to be triggered at 2:00 AM. This is done by specifying the value "2 am" in a configuration file.

SNAP Sizing:

Category: Data Operations

Sub-category: 1.5 Delivering Added Value to Users by Data Configuration

SCU Description	Number of Records	# Attributes	Transformation Complexity	Formula	SNAP Points
Forecast Engine transaction	6	1	Low	6*#Attributes	6*1 = 6

Category: Technical Environment

Sub-category: 3.3 Batch Processes

SCU Description	Number of FTR	# DETs	Transformation Complexity	Formula	SNAP Points
Forecast Engine Batch job	1	1	Low	4*# DE I S	4*1 = 4

4. Defect caused by restrictions on User Interface elements

Problem: In a web application, for one transaction user is unable to enter values with more than 30 characters length

Defect Analysis: User is unable to enter values with more than 30 characters length because current UI field length is set to 30

Solution: This defect can be fixed with change in UI field length to 50 characters. This means only 1 property needs to be set for 1 UI element in one screen.

Also fix needs altering the database column length in a table for one DET.

SNAP Sizing:

Category: Interface Design

Sub-category: 2.1 User Interfaces

SCU Description	# of Unique UI elements	The sum of the number of unique properties	Complexity	Formula	SNAP Points
UI screen of the transaction (1 Transaction)	1	1	Low	2*# unique UI elements	2*1 = 2

Category: Technical Environment

Sub-category: 3.2 Database Technology: Database Capacity Change

SCU Description	# DETs	# RETs	FTR complexity	# Database Changes	Formula	SNAP Points
The identified transaction (1 Transaction)	1	1	Low	1	6*# of Changes	6*1 = 6

Observed limitation of using SNAP

In the experiment there was one instance where SNAP could not be used to size the defect. In this scenario, a new database connection was being created before any database operation was carried out. As a result a single program would open up multiple database connections and these gave rise to performance issues. To fix this problem, the maintenance team opened a single connection at application startup. All database operations were carried out using this connection and it was closed only when the application was closed. In this scenario there were only code changes without any change to the User Interface or the Database. No appropriate SNAP sub category could be found to categorize this type of defect fix.

Conclusion

From an estimation point of view a standard mechanism to size defect fixes is required in order to estimate the effort, duration and cost of fixing these defects. SNAP was found to be useful for sizing the types of defects included in the scope of the experiment. Once size was available, other metrics like productivity for defect fixes could also be measured and used in the estimation process. Having a standard sizing method made it possible to compare estimates and other metrics between projects and accounts. However the limitation observed and described above indicates that there might be other types of defects where SNAP applicability may be limited. It is recommended that more experiments and pilots be carried out to further explore the suitability of using SNAP for sizing defect fixes in order to arrive at accurate cost and time estimates for fixing them.

(continued on next page)

(SNAP for Sizing Corrective Maintenance, continued from page 18)

References

- 1) SNAP Assessment Practices Manual Release 2.2
- 2) "SNAP: Going Beyond Sizing Non Functional Requirements" by Amit Javadekar (MetricViews Aug 2015)

About the author:

Sarika Dandawate is part of Estimation Centre of Excellence at Infosys Limited (sarika_dandawate@infosys.com)

Amit Arun Javadekar leads the Estimation Centre of Excellence at Infosys Limited (amit_javadekar@infosys.com)

Functional User Requirements: Level of Granularity and FPA

by Carlos Eduardo Vazquez, CFPS and Guilherme Siqueira Simões, CFPS

Introduction

Function Point Analysis (FPA) is a functional size software measurement method and therefore considers only one dimension of the requirements: the functional dimension. Functional User Requirements (FUR) describe the behavior of the software to provide tasks and services for its users. FUR are relative to 'the what' the software shall do. The non-functional requirements are relative to 'the how' functionalities will be delivered to the user. It may include quality, technical, environmental and organizational aspects.

In the measurement process described by the IFPUG manual (i.e. CPM), the two most important parts of the analysis are:

- Distinguish FUR from Non-Functional User Requirements
- Compose/decompose the FUR at the appropriate level for the measurement

The goal of this paper is to present some information to help the analyst to compose/decompose the FUR.

Examples of FUR in software requirement specification of a banking ATM system may include descriptions like:

1. Perform operations with the current account.
2. Transfer a value from a current account to another current account.
3. Validate customer card and pin password.
4. The sum of all customer operations shall not exceed \$5,000 per day.

Although these examples are valid as FUR, it can be noticed that they come in different levels of detail (or granularity). And it is very common for a specification to present the requirements at different levels of granularity.

The level of granularity is the greater or lesser extent in the description of behavior expected by the software in a functional specification. This is related to the type of goal

associated with the requirement. Figure 1 illustrates the relationship between these goals and the level of granularity. It will use a classification with three levels proposed by Cockburn (2000) for use cases and generalized by the authors to functional requirements.

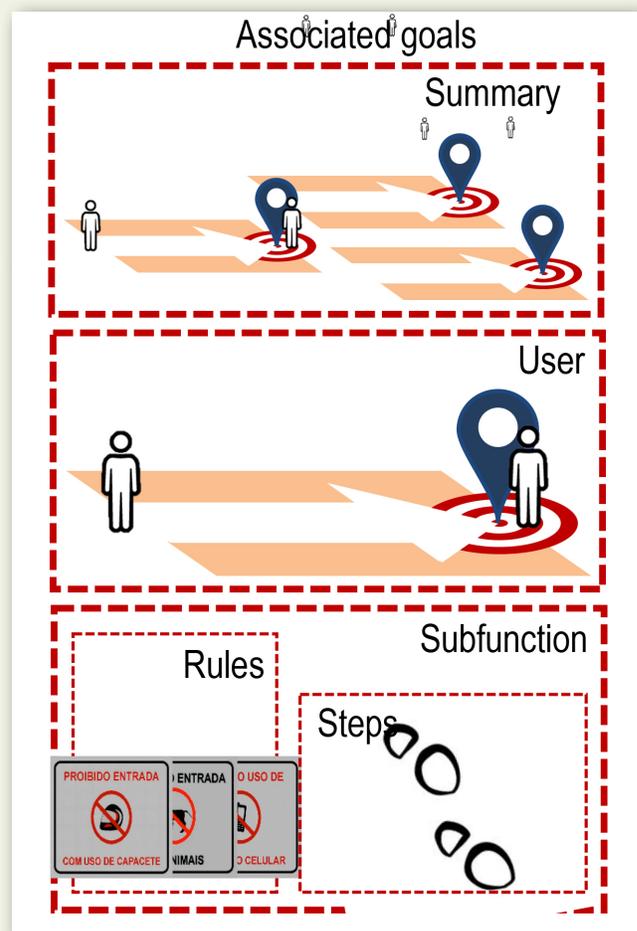


Figure 1: FUR's level of granularity.

(continued on next page)

(Functional User Requirements, continued from page 19)

Much of the function point analysis job is to refine more comprehensive requirements (example 1), or consolidate fragments (example 3 and 4), until they reach the appropriate level of measurement of a transaction function type (example 2) which is equivalent to the concept of IFPUG (2010) elementary process.

User goal

The user-goal requirement is the functional requirement that:

- Covers a single task under the responsibility of a single individual;
- It is carried out at a certain time in which the user has everything he/she needs for the task to be completed in a complete manner to the extent of their responsibility in the operational flow in which it operates.

At the end of the task, the user fulfills its purpose, it is satisfied and there is nothing else to do. If a job needs more than one individual, then there is more than one task involved. Once that requirement has been completed, all that should be done in response to an external event was done.

This description basically has the same practical effect of the IFPUG definition of elementary process: “the smallest unit of activity that is meaningful to the user that constitutes a complete transaction, is self-contained and leaves the business of the application being counted in a consistent state”.

This is the case of example 2 (“Transfer a value from a current account to another bank account.”). It is a single task (certainly composed of steps and rules), under the responsibility of a single individual, that at the end of all steps is satisfied with the goal achieved: a value was transferred to another account.

The functional requirement at this level of granularity is the only one that provides an outline of the software scope unequivocally; there is no doubt to what is comprised in it. This is the only level of processes description that can be standardized and is therefore used for all the measurement methods of functional size standardized by ISO / IEC 14143 standard.

If it were possible to find a specification described completely with this level of granularity, the function point analysis would be much easier: each transaction function type would correspond directly to a requirement.

Summary goal

These are requirements that aggregate several users-goals requirements in a single high-level specification. The higher the level, the more general its goals are, and for a higher level goal to be achieved, other lower level goals must be achieved first.

This type of requirement is related to more general goals and is at a level of coverage associated with collaborative goals; it is associated to high-level business processes. It does not refer to a single task or service. It summarizes a set of tasks of one or more users.

What specifically are the tasks associated with these requirements? Maybe it is obvious to the readers (i.e. the stakeholders) or they are not known yet. However in the latter case, it is known that there are activities to be done to elicit this requirement, it was decided that they are part of the scope of the software under development.

In early phases of software projects, perhaps most of the requirements elicited are summary-goal FURs. This is because there are still several decisions pending on the scope of the project.

However, some FURs have patterns that needs no more detail to be provided. An example is the simple registration form, i.e. CRUD (Create, Read, Update, Delete) forms. This type of requirement is usually expressed as: “The system must register products.” And in this case it is agreed that the verb “register” performs the CRUD. Therefore it is clear that the system will provide the user the following tasks: add, change, delete, and inquire product data.

This is the case of example 1 (“Perform operations with the current account “). It is possible that the entire set of tasks to operate current account from an ATM are obvious to the readers of the specification. If the function point analyst is in this group, he/she will know how to decompose this requirement in the equivalent elementary processes. If it is not so obvious what tasks are covered by the requirement, then he/she should look for a subject matter expert to provide this additional information.

However, if the requirement is at this level because there wasn't yet an opportunity in the project to elicit its details, the analyst will be able only to estimate the functional size, but not to measure it. In this case it must be clear to all who will use the results of the analysis that this is a size approximation. The approach to estimate this type of requirement may vary depending on the project context. An example would be to try to identify patterns in similar requirements that can be

(continued on next page)

(Functional User Requirements, continued from page 20)

applied to this requirement. Certainly the analyst must make assumptions (and document them) to follow on the analysis.

Subfunctions

These are fragments of user-goal requirements; they can be related to a set of steps or to rules that are part of others user-goal requirements.

The subfunction requirement that represents a set of steps describes the exchange of data in both directions between the user and the software; and between the latter and the data requirements. This is the case of example 3 (“Validate customer card and pin password.”). Each type of operation that affects the customer’s current account (ex.: withdraw money, transfer values, pay expenses, etc) requires the same set of steps described by example 3, which could be:

- Check if the card is valid.
- Check if the desired operation is compatible with card’s type.
- Check if password pin informed is correct.
- Increase password pin errors if pin password informed is incorrect.
- Reset password pin errors if the pin password informed is correct.

Validate customer card and pin password is not a goal for the user of an ATM system, but are necessary and intermediary steps to achieve its goal (for example, withdraw money). The definition of a requirement at this level is justified only when there is a behavior shared by several other functional requirements. This makes the requirements documents easily modifiable to changes because it reduces redundancy, avoiding describing the same set of steps more than once in the project.

The subfunction requirement related to rules is usually linked to laws that govern the business and describe in a complementary way the business processes. It is so often called business rules. It can describe corporate policies, government regulations and industry standards to which the software must be subordinated.

This is the case of example 4 (“The sum of all customer operations shall not exceed \$ 5,000 per day.”). Business rules are usually shared among different functional requirements,

including allocated to different software products. Therefore, it is a good practice to specify these type of FURs independently. It will help to better management requirements (ease of modification and ease of reuse).

The function point analyst, when faced with this kind of requirement, should never consider it an elementary process. He/she must investigate what tasks these subfunctions are part of. These will be the elementary process to consider.

Although this type of requirement does not influence the number of transactions function type in the analysis, their description usually has useful information to determine the functional complexity of the transactions in which they are part of; either by describing fields entering and leaving the boundary (data element type) or validations that needs to reference logical files (file type referenced).

Conclusion

Although the concept of these three levels of granularity is simple, in practice it is observed that several function point users do not pay attention to it and often make mistakes like:

- Undersizing the software, for example, counting a single elementary process for a summary-goal requirement and / or;
- Oversizing the software, for example, counting an elementary process for a subfunction requirement.

References

- COCKBURN, A. Writing Effective Use Cases. Boston, MA: Addison-Wesley, 2000.
- COSMIC. COSMIC Measurement Manual: 4.0.1. Common Software Measurement International Consortium, 2015.
- INTERNATIONAL FUNCTION POINT USERS GROUP. Function Point Counting Practices Manual. Release 4.3.1. New Jersey, NJ: IFPUG, 2010.
- VAZQUEZ, C.; SIMÕES, G.; ALBERT, R. Análise de Pontos de Função, Medição, Estimativas e Gerenciamento de Projetos de Software. São Paulo: Saraiva, 2013.
- VAZQUEZ, C.; SIMÕES, G. Engenharia de Requisitos: Software Orientado ao Negócio. Rio de Janeiro: Brasport, 2016.

Use Relevant Industry Data as a Valuable Outside View for Your Analysis

by *Harold van Heeringen, ISBSG president*

In the IFPUG community, as well as in other functional size measurement communities like Nesma or COSMIC, there is a lot of focus on measuring the functional size of software applications and software projects as accurately as possible. Of course this is important, as functional size is used in a lot of important areas, like software project estimation, IT supplier selection, benchmarking and supplier performance measurement. These are important disciplines for many organizations, or at least they should be as nowadays the cost efficient and productive development of new functionality is key in many business areas. So, now the question is... How can you carry out these activities once you have determined the functional size in an accurate way?

For software estimation for instance, you need an accurate Project Delivery Rate (PDR) expressed in hours per function point to estimate the number of effort hours needed for the various project activities in scope of the project. If the organization has a professional Estimation & Performance Measurement (E&PM) process in place for application development projects, there may be some relevant historical data available. Accurate historical data of completed projects is just as important for project estimation as determining the accurate size.

However, in many organizations the metrics teams are struggling to get all the relevant data from completed projects. In practice for instance, a lot of effort hours are not booked correctly (wrong project, wrong activity, overtime not recorded, etcetera). Also the actual project size delivered may be different from the size measured, as you need to be able to take into account the changes in scope during the project, and these are not always clear, even when the 'E&PM process' measures the actual size after project completion.

So, to base your estimates on incorrect historical data may still be just as dangerous for your project as to rely on immature estimation processes, like for instance asking subject matter experts to come up with an estimate. That's why I always recommend also to have an extra opinion 'from the industry', just to have a better understanding of the productivity that the industry peers have realized in the past in comparable projects.

Using the historical project data of the International Software Benchmarking Standards Group (ISBSG) helps organizations to better understand the reality value of their analysis. The data

is provided in Excel, therefore easy to filter and analyze. An example of presenting the data that I always find useful, for instance filtering on:

- Primary Programming Language: Java
- Size between 500 and 1000 FP
- Project type: Enhancement (release)
- Count approach: IFPUG 4+ or NESMA (the methods are almost the same nowadays)

Metric	PDR (hours/FP)
Number of projects: 31	
Minimum	1.1
Percentile 10%	2.7
Percentile 25%	4.0
Median	9.9
Percentile 75%	15.3
Percentile 90%	21.8
Maximum	43.6
Average	11.2

This results in 31 projects. Just to understand the spread in the data, it's better to show some descriptive statistics, in this case for the Project Delivery Rate (hours per FP).

So let's assume you have made a project estimate for a Java enhancement project of 700 function points and used a PDR of 7.2 hours per function point. The quick analysis of the industry data shows that this means that your organization would develop the project between the P25 and the median of the selected industry data set. This could be realistic if you understand your organizations capabilities are usually in this zone compared to the industry. However, if your projects are usually developed with a productivity much better or much worse than industry average, this would possibly raise a red flag and you may want to question your estimate. Maybe your PDR of 7.2 was calculated by using data that was not correctly collected?

The same type of analysis can be done for the other activities where you need industry data to assess whether productivity

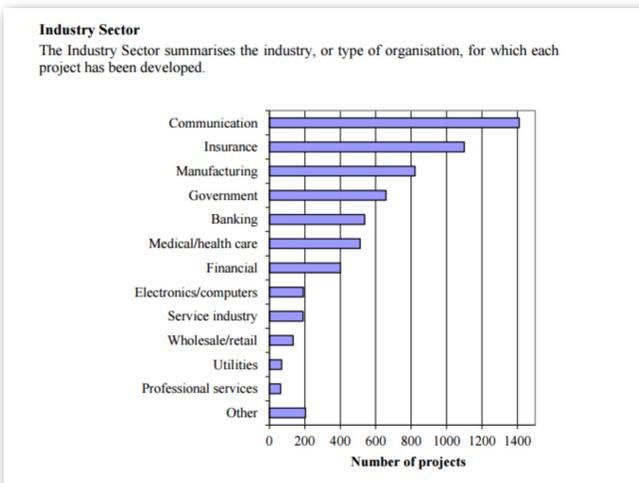
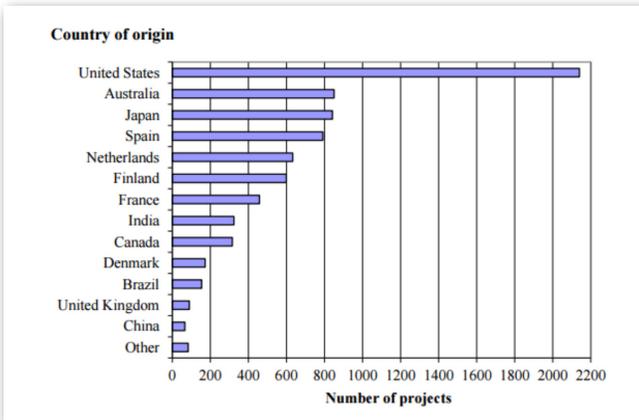
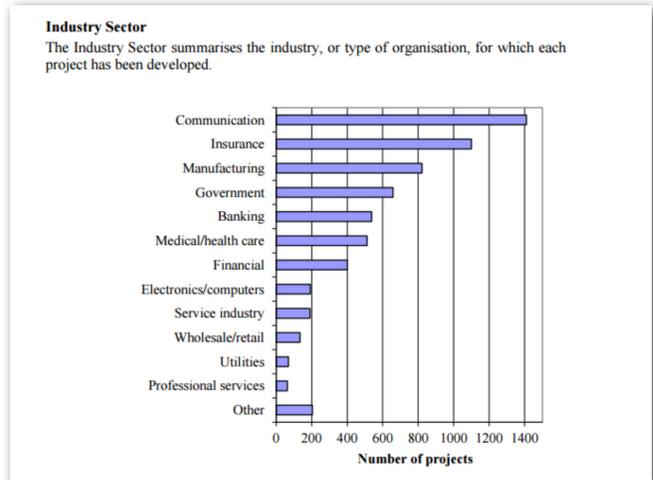
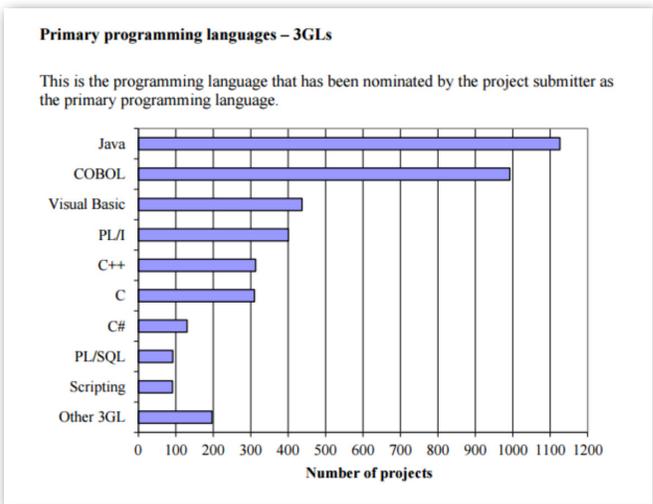
(continued on next page)

(Use Relevant Industry Data, continued from page 22)

for a specific (set of) projects or sprints is below or above market average. The data helps you to set the right peer group for benchmarking purposes. Also the data can help to set realistic targets with regard to metrics like productivity, cost efficiency and process quality for suppliers to reach in a specific period of time. Furthermore, it becomes possible to understand if the bidders on your RFP are trying to buy the deal, or if they may not have professional software cost estimating processes in place, both resulting in metrics that are too good to be true compared to the industry.

The industry data of ISBSG is therefore a cheap but valuable way to get an outside view on your analysis. As IFPUG is a Gold partner of ISBSG, IFPUG members are eligible for a 20% discount on all ISBSG products, including the data repositories! Use the code `ifpug-member-2016` at checkout.

Just to give you an idea of what is in the latest version of the 'Development & Enhancement' repository: >7500 projects, a few demographics.



For a complete overview of the demographics, check <http://isbsg.org/wp-content/uploads/2016/04/ISBSG-DE-Demographics-2016-R1.pdf>

Communications and Marketing Committee

By David Thompson, Chair

IFPUG marketing plan; Updates to the website FAQs page; Promotion of ISMA12; a simplified volunteer recruitment process; and other activities

During the first six months of 2016, the Communication and Marketing Committee (CMC) followed through on action items from the IFPUG marketing plan, moving to the website a new page on the Uses and Benefits of Function Point Analysis that includes ten case studies, and a shareable slide show on how to sell function point analysis to your manager.

A revised website FAQs page was installed in production in January, and subsequently additional questions and answers were added to the FAQs page via a new update mechanism. Additionally, there is a plan in place for the International Marketing Committee to provide hand-translated FAQ pages, in four languages, that will be linked from the website.

In January through April the CMC, through the website, the weekly eBlasts, and social media, engaged in a marketing campaign to publicize the ISMA12 conference, classes, and exams held at the Centro Congressi Frentani in Rome, Italy on May 3 - 5.

Working with the International Marketing Committee we developed and implemented a simplified process for applying to become an IFPUG Volunteer. Check it out on the web page, Get Involved! accessed from the Membership and Certification menu.

During the 6 month period, The CMC processed 26 web update requests and sent 20 eBlasts, on diverse topics, including those specifically targeted for the ISMA12 conference. And we continued archiving eBlast copies on the website. Recognizing that more than 50% of our messages are read on mobile devices, we continued using a new eBlast format "mobile-friendly" template.

And finally, we should mention the work to plan and produce the July 2016 edition of MetricViews.

Looking to the future, the committee will be investigating a concept called the Sales Funnel, that will lead to increased enrollment of new IFPUG members.

Look for more on this in the second half of 2016!

Functional Sizing Standards Committee Report

by Dan French, Chair

The first half of 2016 has been productive for the FSSC. We continue to meet monthly to discuss our current and future projects as well as counting related topics that have come to our attention. We have completed the Shared Data project and the Data Analytics addendum to the data warehouse paper will be published shortly. In addition, we've nearly completed the part one of the first joint FSSC/NFSSC white paper on Integrating procedures for FPA and SNAP.

Currently the FSSC is also working on a number for projects slated for completion 2nd half 2016 including white papers on counting workflow applications, Universal Markup Language (UML) modeling. The committee is also reviewing and updating previously released case studies to bring them into alignment with the current version of the Counting Practices Manual (CPM). The Counting Integrated EOs/EQs iTip include is also due to be finished this year.

We will be holding our annual FSSC meeting in conjunction with the upcoming

(continued on next page)

**Want to see your Company's Name here!
Advertise with IFPUG!**

Advertising spots are available for the 2017 *MetricViews*. If you are interested in placing an ad please contact IFPUG headquarters at ifpug@ifpug.org

(Committee Reports, continued from page 24)

Unconference and Board of Directors meeting this September in Baltimore, MD and look forward to seeing everyone there.

The committee would like to thank Karl Jentzsch, and Carlos Eduardo Vazquez for their work in support of FSSC projects and look forward to working with other dedicated IFPUG volunteers on future projects.

Non-Functional Sizing Standards Committee Report

by Talmon Ben-Cnaan, Chair

SNAP method of non-functional sizing continues to evolve

A Joint process to count Function Points and SNAP

The NFSSC and the FSSC have collaborated to create two guidelines (As a two-part white paper).

Part one provides guidance on an integrated process for Function Point Analysis (FPA) and Software Non-functional Assessment Process (SNAP), to size both the functional and non-functional user requirements of a project.

Part 2 describes how to use the two size measurements for estimation, process performance metrics and benchmarking.

The Digital Transformation and IoT

Companies in the Telecommunication and Banking industries are moving to the digital revolution, adding more functionality to their web applications, and allowing users to operate without the need to call a CSR or go to the bank. Usability, accessibility, compatibility and security become a major part of the design and testing of these web applications. We recommend using SNAP to size these non-functional requirements.

NFSSC is looking for use cases and success stories

The NFSSC is looking for case studies and success stories, showing the joint counting process of function points and SNAP point, and the use of SNAP for better estimation. Case studies, which can enrich the software measurement community, will be published by the NFSSC.

International Membership Committee Report

by Ivan Pinedo, Chair

The International Membership Committee has focused its efforts during the first half of 2016 on initiatives to improve the status and experience for members (individual and corporate) of the IFPUG community. In order to do so we have launched several initiatives, such as: a renewed proposition for the corporate membership, an improved volunteer process, a new set of rules for the country representatives, and a tool to help the country representatives to register all the queries that they receive from members. We are also working with the Indian members in the interest of having a future IFPUG conference there in the near future.

The country representatives have successfully answered over 100 requests from the users during this first six months and they are also working on a new dedicated translation for the IFPUG webpage FAQs.

Last but not least we have also added a new member to the Committee: welcome Bram Meyerson! Bram will help us expand the reach of IFPUG within the South African IFPUG community.

ISO Committee Report

by Steven Woodward, Chair

In 2016, the IFPUG ISO Conference Committee is looking at several options to expand participation and awareness of IFPUG and SNAP metrics approaches.

The ISO/IEC JTC1 SC 7 “Systems and Software Engineering” sub-committee had meetings in China this May. IFPUG did not have representatives attend, however, the IFPUG ISO committee did track the activities and tentative plans for the future. Steven Woodward did get approval from the Standards Council of Canada for participation within ISO/IEC JTC1.

The ISO/IEC JTC1 SC7 community wants to accelerate standards development and address core 2016 industry subject areas including: cloud computing, autonomous computing, mobile devices and Internet of Things. The IFPUG ISO committee is reflecting on its’ mandate (perhaps expand collaboration with more standards organizations) and will consider approaching ISO/IEC JTC1 SC 7 at the right time with SNAP recommendations regarding non-functional standardization.

It certainly appears to be a busy year, and we hope that the IFPUG community will participate and help expand membership and activities within the ISO Committee.

Committee Rosters

Certification Committee

- Mahesh Ananthakrishnan, Cognizant Technology Solutions – **Chair**
- Gregory Allen, Pershing – **Vice Chair**
- Jim McCauley
- Michel Ryan, BMO Financial Group / Bank of Montreal ON
- Teresa Cristina De Spagna Zenga Beraldo, BANCO BRADESCO S/A
- Christine Green, Hewlett Packard Enterprise – Board Liaison

Communications and Marketing Committee

- Walter David Thompson, Blue Pine Solution Centre – **Chair**
- Stephen Neuendorf, David Consulting Group
- Paul Radford, Charismatek Software Metrics
- David Herron, David Consulting Group
- Antonio Ferre Albero, GFT IT Consulting
- Justin Keswick, Bank of Montreal
- Carol Dekkers – Board Liaison
- Linda Hughes, Accenture – VOLUNTEER

Conference and Education Committee

- Peter Thomas, Steria – **Chair**
- Prof. Eduardo Alves De Oliveira, Servico Federal De Processamento De Dados (SERPRO)
- Thiago Silva Da Conceicao, Synapsis
- Antonio Ferre Albero, GFT IT Consulting

- Toni Ramos, David Consulting Group
- Dr. Luigi Buglione, Engineering Ingegneria Informatica SpA – Board Liaison

Functional Sizing Standards Committee

- Daniel Bradford French, Cobec Consulting – **Chair**
- Bonnie Brown, Hewlett-Packard – **Vice Chair**
- Diana Baklizky, TI Metrics
- E. Jay Fischer, JRF Consulting
- Peter Thomas, Steria
- Adri Timp, Equens
- Roopali Thapar, IBM
- Tammy Preuss, AT&T
- Steve Keim, David Consulting Group
- Charles Wesolowski
- Dacil Castelo, LEDA Consulting, S.L. – Board Liaison

International Membership Committee

- Ivan Pinedo, DCG Software Value – **Chair**
- Bram Meyerson, Quantimetrix
- Anjali Mogre, Atos Origin International SAS
- Cao Ji, Beijing Suiji Tech
- Gianfranco Lanza, CSI Piedmonte
- Saurabh Saxena, Zensar Technologies Ltd
- Marcio Silveira, HP
- Pierre Almén, ImproveIT – Board Liaison

ISO Committee

- Steven Woodward, Cloud Perspectives – **Chair**
- Carol Dekkers, Quality Plus Technologies, Inc.
- Christine Green, Hewlett Packard Enterprise – Board Liaison

Non-Functional Sizing Standards Committee

- Talmon Ben-Cnaan, Amdocs – **Chair**
- Kathy Lamoureaux, HP – **Vice Chair**
- Mousa George Mitwasi, Optum
- Dr. Charley Tichenor, Marymount University
- Francisco Julian Gomez, LEDA Consulting, S.L.
- Tomasz Marchel, Asseco Poland S.A.
- Roopali Thapar, IBM
- Jalaja Venkat, iGATE Global Solutions
- Saurabh Saxena, Zensar Technologies Ltd – VOLUNTEER
- Mauricio Aguiar, TI Metrics – VOLUNTEER
- Dr. Luigi Buglione, Engineering Ingegneria Informatica SpA – VOLUNTEER
- Dacil Castelo, LEDA Consulting, S.L. – Board Liaison

IFPUG Board of Directors

Tom Cagley, President

DCG Software Value
tcagley@softwarevalue.com

Mauricio Aguiar, Vice President

TI Metricas
mauricio@metricas.com.br

Christine Green, Secretary, Director of Certifications

Hewlett-Packard
christine.green@hpe.com

Debra Maschino, Treasurer

NASCO
debra.maschino@nasco.com

Kriste Lawrence, Immediate Past President

Hewlett-Packard
kriste.lawrence@hpe.com

Carol Dekkers, Director of Communications & Marketing

Quality Plus Technologies Inc
dekkers@qualityplustech.com

Dácil Castelo, Director of Sizing Standards

Leda-mc
dcastelo@leda-mc.com

Pierre Almén, Director of International & Organizational Affairs

ImproveIT
Pierrea@coolmail.se

Luigi Buglione, Director of Education & Conference Services

Engineering.IT SpA
luigi.buglione@eng.it

Behind the Scenes

by Nicole Lauzon, Membership Coordinator

We at Headquarters are excited for the new IFPUG membership year. An IFPUG membership provides many benefits including access to Counting Practices Manuals, Case Studies at a discounted rate, and the ability to sit for and receive certifications including CFPS, CFPP and CSP. We also hope that you have accessed the Members' Service Area of the website to take advantage of the resources in the Knowledge Base as well as to check your personal information.

In order to provide the best member services possible Headquarters needs your help with updating your personal information. On your profile you have the ability to list two email addresses, business and home. It is extremely important to list a personal email in case your employment changes. Emails are the main source of correspondence between IFPUG and its members and we want to ensure you receive notifications (such as certification expiration reminders) as well as news of events. With the upcoming elections, we would also ask that you update your mailing address since ballots will be sent out by mail.

The 2016 grace period for renewals ends on August 31, 2016 and a valid membership is required to be eligible to vote, so don't forget to renew. A valid IFPUG membership is also required to maintain any certifications you currently hold. Let us know if you need any assistance.

Please also visit the IFPUG Insights area and let us know what you think of the articles in this edition or previous editions of MetricViews. You will find this an excellent place to ask questions and discuss what you have read. This is YOUR IFPUG community, so get involved!

We're very excited about this membership year, and we hope you're equally excited to be a member. As always, we look forward to hearing from you at ifpug@ifpug.org.

*Best regards,
Nicole Lauzon*



Yuri Francis Araujo Ferreira DBA Engenharia De Sistemas Ltda.	Yasuharu Honda JFPUG-Japan Function Point Users Group	Stefano Mazzara Ministero Interno - DC Servizi Elettorali	Stefano Rondina IBM
Thiago Araujo PD Case Informatica Ltda	Alessandra Inzitari SOGEI	Flávio Millioli	Krzysztof Rozkowicz
Marisa Balbi	Tomokazu Kambayashi JFPUG-Japan Function Point Users Group	Akira Miyata JFPUG-Japan Function Point Users Group	Michele Carlo Rutigliani Accenture
Luca Bontempi	John Stephen Kitching David Consulting Group	Carmela Nacarolo	Joao Gabriel Santos Eficácia Organização
Priscilla Brito	Pedro Legaspe MStech Educação e Tecnologia SA	Masao Nagai JFPUG-Japan Function Point Users Group	Sherif Sayed
Libia Cunha	Roberto Lima Seguradora Líder - DPVAT	Saurabh Narendra Wani Accenture	Julice Silva
Andrea Da Silva	Paolo Lisco TBS IT	Sawako Oiwa JFPUG-Japan Function Point Users Group	Rashmi Sinha UnitedHealth Group IT
Aarti Dahiya	Brandon Lowden	Ryo Ooya JFPUG-Japan Function Point Users Group	Domenico Sorace SOGEI
Leandro De Albuquerque Eteg	Alessia Luchetta SOGEI	Fernanda Peres	Ciro Sorrentino Indra Italia Spa
Darsana Dinesh IBM	Mario Luiz Varella TI Metricas Ltda	Dominik Pociask	João Sousa
Leila Karita Espirito Santo	Leonardo Magazzini Infogroup SCpA	Patrizia Prioreshi Present S.p.A.	Daive Spadoni SOGEI
Matteo Falcone IBM	Fabricia Marques Magna Sistemas e Consultoria S/A	Ana Lucia Regino	Elaine Sundfeld Banco de Brasilia S.A.
Anna Fiorucci Indra Italia Spa	Fabricia Marques CPM Braxis S.A.	Ercole Riccitelli Ministero Interno - DC Servizi Elettorali	Francesco Tassone Present S.p.A.
Daive Fipaldini Accenture	Eduardo Martins	Luana Rinaldi Indra Italia Spa	Laura Tomassini Ministero Giustizia
Francesco Galiano Ministero dell'Interno Dip.Pubblica Sicurezza			Bartolomeo Turco Accenture
Francisco Julian Gomez LEDA Consulting, S.L.			
Gabriel Gori			

Congratulations to these NEW Certified Function Point Practitioners!

Sabdhami Babu	Syuzou Ishitani JFPUG-Japan Function Point Users Group	Daphine Moura TI Metricas Ltda	Fulvio Romani Ministero Giustizia
Giuseppe Barrile Ministero Giustizia	Miyako Iwakiri JFPUG-Japan Function Point Users Group	Hanae Nakamura JFPUG-Japan Function Point Users Group	Margherita Romaniello Ministero Interno - DC Servizi Elettorali
Corrado Belfiore SOGEI	Kesavulu Kalimidi Optum	Massimiliano Natalizia Accenture	Cleidiane Silva Ribeiro
Fabrizio Bonanni SOGEI	Tooru Kaneko JFPUG-Japan Function Point Users Group	Tomoki Oshino JFPUG-Japan Function Point Users Group	Bartosz Sredniak Accenture
Eduardo de Albuquerque Gomes Pereira Plennus TI	Manjusha Misra MPHASIS	Marcelo Paiva MStech Educação e Tecnologia SA	Terry Vogt Booz Allen Hamilton
Marcos Grigoletto			

Congratulations to these NEW Certified SNAP Practitioners!

Paola Billia
NTT DATA Italia SpA

Matteo Falcone
IBM

Emanuele Richiusa
Business Integration Partners
SPA

Simona Stefanelli
Business Integration Partners
SPA

Marco Buglielli
Finmeccanica - Società per
Azioni

Roberto Fenaroli

Vincenzo Mauro
Business Integration Partners
SPA

Marcello Sgamma
NTT DATA Italia SpA

Roberta Russo
Hewlett Packard Enterprise

Congratulations for 20 Years of CFPS Certified Function Point Specialist Fellows!

Spring 2016



Anna Battistata
Massimo Beretta
Steven Keim
Makato Kurashige
James Mayes
Maurizio Menghini



Franco Perna
Janet Russac
Luca Santillo
Walter David Thompson
Daniele Zottarel

IFPUG SEPTEMBER MEETING AND GREAT UNCONFERENCE

Save the date: Friday, September 16, 2016
at The Hyatt Place Hotel, Baltimore/BWI
Airport.

This event won't be a typical conference but an unconference; we invite members, who will be able to attend, and would like to facilitate a 'Birds of a Feather' session or sit on an 'Expert Panel', to e-mail cec@ifpug.org.

It's a 1-day, interactive, facilitated session where we'll collaborate as measurement practitioners and talk about things that matter with metrics today. The unConference is designed using agile concepts (high quality, iterative,... adaptive,

flexible,) with facilitator-guided sessions on topics of YOUR choice, (the topics and agenda are "scrumped" by attendees at the beginning of the day.

It's a coming together of like-minded metrics people, to share and discuss burning issues important to YOU... topics could range from making SNAP work for your company to convincing management about metrics. They could cover better estimating or how to make SNAP and FPA work together – it's whatever you make it. Innovation and new ideas are hallmarks of an unConference, and, we need you!

Will you join us on Sept 16?

See you in Baltimore!

CERTIFICATION MATTERS!

CFPS Matters!



“Personally becoming a CFPS gave me the opportunity to increase my logical thinking, intellectual ability and systems knowledge. Professionally I can impart to the company the functional sizing competency which is very important to achieve better negotiations.”

Jviana Mantecon Pereira de Souza, Brazil



“By becoming a CFPS I obtained more quality and professional competence in software measurement using function point analysis. This brought my company and team more efficiency and effectiveness in our area.”

Kleber Batista Soares De Oliveira, Brazil



“Undoubtedly, being a CFPS helped me to spread the usage of FPA in my organization, an IT business-leader company. It is helpful for sizing automatic procedures for taxation, allowing me to gain more knowledge about such complex systems. ‘Sizing’ means ‘knowing’: a good opportunity for exchanging experiences of those software systems.”

Daniele Zottarel, Italy