Using Function Points in Agile Projects

Célio Santana[1], Fabiana Leoneo[1], Alexandre Vasconcelos[1], Cristine Gusmão[2]

[1]Universidade Federal de Pernambuco, Centro de Informática, Prof Luiz Freire Avenue. S/N,50740-540 Recife, Pernambuco, Brazil

[2] Universidade Federal de Pernambuco, Centro de Telesaúda, Hospital das Clínicas | 2º andar | Av. Prof. Moraes Rego s/n , Recife , Pernambuco, Brazil CEP 50.670-420

{casj, fls2, amlv}@cin.ufpe.br, cristine.gusmao@nutes.ufpe.br

Abstract. Agile development has become increasingly common in the in an organizational software development environment, this paper, examines whether function points would be compatible with story points on agile projects. Specifically, it addresses the question of whether function points would be a relevant measure of velocity. Though any unit of measure can be used, this paper contrasts theoretical concepts about Story Points (SP) and function points (FP) as units for measuring size. Also, was realized a statistical correlation between FP and SP using 2191 stories and 18 iterations in a Brazilian public agency. The conclusion drawn from this study is that function points, in that particular case, could be related with the initial value of the Story Points found after the planning poker.

Keywords: Function Point, Function Point Analysis, Story Points.

1 Introduction

The software industry is almost 60 years old, which makes it a fairly maturity industry. One would think that after six decades the software industry would have well established methods for measuring productivity and quality, and also collected a large volume of accurate benchmark data of thousand of measured projects. However, this is not quite the case [1].

Initially to measure productivity and quality used one unit called lines of code (LOC). At the time, circa 1950, that metric was fairly effective once that coding took about 50% of the effort to build an application [1].

Between 1957 and 1967 the situation changed dramatically. Low level assembly languages started to be replaced by more powerful procedural languages such as COBOL and FORTRAM. Applications sizes grew from 1.000 lines of code past 100.000 lines of code raising problems when using LOC metrics [1].

These economics problems are what caused IBM to assign Allan Albretch to develop a useful metric that was independent of code volumes, and which could both economic productivity and quality without distortion. In 1979 Allan Albrecht was the first to publicly release a method for functionally sizing software called function point analysis (FPA) [2].

The use of FPA, as a measure of the functional size of software, has grown since the mid 1970s from a few interested organizations to an impressive list of organizations worldwide. The successful adoption of FPA was ratified with the creation of the ISO/IEC 14143:2007 [3].

In 2001 was presented the Agile Manifesto that proposed new values and principles based in responding to changes quickly and light documentation. This vision seems antagonist than that proposed by traditional engineering [4].

Schuh [5] defines agile development as a counter movement to 30 years of increasingly heavy-handed processes meant to refashion computer programming into software engineering, rendering it as manageable and predictable as any other engineering discipline.

Mnkandla [4] the agile movement could mark the emergence of a new engineering discipline that has shifted the values of the software development process from the mechanistic (i.e., driven by process and rules of science) to the organic (i.e., driven by softer issues of people and their interactions). Boehm [5] view believes agile methodologies as a challenge to the mainstream software development community that presents a counter-culture movement, which addresses change from a radically different perspective.

To an agile team, a plan is one view of the future but many alternatives are possible. As a team gains knowledge and experience they will count these into the plan [6]. A measurement system which support this kind of experience is considered "special" and some metrics used in this context is Story Points (SP) and Ideal Days [1].

Jones [1] states that one of the agile weaknesses is the widespread failure to measure projects using standards metrics such as function points. Based in this statement an ad-hoc search were conducted in the following sources: ACM Digital Library [1], CiteSeerX[2], IEEE Xplorer[3], Scopus[4] and SpringerLink[5].

Just one source was published and presents relevant work presenting scientific evidence about the function point analysis and story points running in agile software development environments. The work was published by Fuqua [7] that conducted a study by using function points in agile projects and tried to correlate with story points in that projects.

Other relevant work about this subject, not found in the ad-hoc search, presented by Jones [1] in a book, where he states that according to its empirical basis[6], it is noted that two function points is equal to one story point on average. But it is worth noting that this measure is an average of its empirical database.

In this light, this paper presents conceptual differences between function points and story points. The goal of this paper is to present the theoretical relationship between story points and function points as well as providing empirical data from a real life case study where one project was measured using these two approaches.

After this introductory section, section 2 explores function points analysis. The section 3 presents the concept of story points. Section 4 shows the size measurement concept which are related with FPA and SP. The section 5 shows theoretical differences between both techniques. Section 6 presents the case study in a Brazilian public agency. And section 7 shows the summary, related works and threats of validity of the work.

2 Function Point Analysis

Once the growth in the use of function points, there has been wider application and use of the measure. Since its formation in 1986 the International Function Point Users Group (IFPUG) has continuously enhanced the original Albrecht method for functionally sizing software.

This International Standard is the latest release in the continually improvement IFPUG method. This aims to promote the consistent interpretation of functional size measurement in conformance with ISO/IEC 14143-1:2007. The IFPUG functional size measurement method is known as function point analysis and its units of functional size are called Function Points. The IFPUG version of Function Points is published in The Counting Practices Manual in its actual version 4.3 [8].

---

[1] http://portal.acm.org/

[2] http://citeseer.ist.psu.edu/

[3] http://ieeexplore.ieee.org/

[4] http://www.scopus.com/

[5] http://www.springerlink.com/

[6] www.isbsg.org

IFPUG's method for function point analysis is an ISO standard and must be conformant to ISO/IEC 14143-1:2007. The method can measure "functional size" nether "non-functional size". This does not mean that the nonfunctional size cannot, or should not, be measured, instead it must be clearly stated as a separate measure [8]. The process diagram of IFPUG FPA counting is shown in Figure 1.

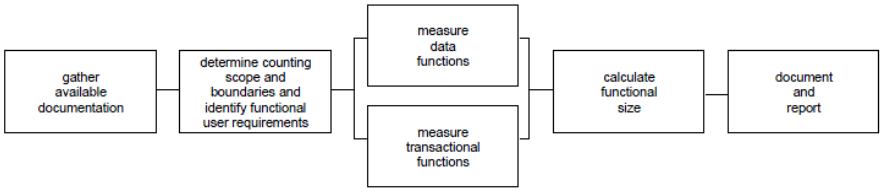

Figure 1. FPA Procedure Diagram [8].

The first stage in the function point counting procedure is to gather the available documentation. To support a functional size measurement, it shall describe the functionality delivered by the software or the functionality that is impacted by the software project that is being measured.

Suitable documentation may include requirements, data/object models, class diagrams, data flow diagrams, use cases, procedural descriptions, report layouts, screen layouts, user manuals and other software development artifacts. If sufficient documentation is not available, it is important to access experts who are able to provide additional information to address any gaps in the documentation. The ideal requirements are called perfect requirements by the FPA practitioners.

The next stage is responsible for counting scope which defines the set of Functional User Requirements to be included in the function point count. Also in this stage it is necessary determine the boundary which is a conceptual interface between the software under construction and its users.

Identifying the functional requirements are related with the concept of elementary process which is the smallest unit of activity that is meaningful to the user. To identify each elementary process, the following activities shall be performed:

- Is meaningful to the user;
- Constitutes a complete transaction;
- Is self-contained;
- Leaves the business of the application being counted in a consistent state,

Following these rules it is necessary identify all unique elementary processes.

After that it is necessary to measure data functions which represent functionality provided to the user to meet internal and external data storage requirements. A data function is either an internal logical file or an external interface file.

An internal logical file (ILF) is a user recognizable group of logically related data or control information maintained within the boundary of the application being measured.

An external interface file (EIF) is a user recognizable group of logically related data or control information, which is referenced by the application being measured, but which is maintained within the boundary of another application.

In parallel, could be performed the measuring of transactional functions. A transactional function is an elementary process that provides functionality to the user for processing data. A transactional function is an external input, external output, or external inquiry.

An external input (EI) is an elementary process that processes data or control information sent from outside the boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.

An external output (EO) is an elementary process that sends data or control information outside the application's boundary and includes additional processing beyond that of an external inquiry. The primary intent of an external output is to present information to a user through processing logic other than or in addition to the retrieval of data or control information. The processing logic must contain at least one

mathematical formula or calculation, create derived data, maintain one or more ILFs, and/or alter the behavior of the system.

An external inquiry (EQ) is an elementary process that sends data or control information outside the boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information. The processing logic contains no mathematical formula or calculation, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered.

On systems which present perfect requirements, different measurements performed by different people must have the same final result. Imperfect requirements lead the measurement performer take assumptions about that requirement and this kind of assumption could lead different results in counting the same sample of requirements. Because of this property is considered that FPA is an objective method for measuring software.

3 Story Points

Story points are a unit of measure for expressing the overall size of a user story, feature, or other piece of work. When we estimate with story points we assign a point value to each item. The raw value we assign is unimportant. What matters are the relative values [6].

The number of story points associated with a story represents the overall size of the story. There is no set formula for defining the size of a story. Rather, a story point estimate is an amalgamation of the amount of effort involved in developing the feature, the complexity of developing it, the risk inherent in it, and so on [6].

There are two common ways to get started. The first approach is to select a story that you expect to be one of the smallest stories you'll work with and say that story is estimated at 1 story point. The second approach is instead to select a story that seems somewhat medium-sized and give it a number somewhere in the middle of the range you expect to use. A story that is assigned a two should be twice as much as a story that is assigned a one [6].

Story estimates need to be owned collectively by the team. A story comprises multiple tasks and that a task estimate is owned by the individual who will perform the task. Story estimates, however, are owned by the team for two reasons: First, since the team doesn't yet know who will work on the story, ownership of the story cannot be more precisely assigned than to the team collectively. Second, estimates derived by the team, rather than a single individual, are probably more useful [9].

At the end of an iteration the team counts the number of story points they completed. They then use that as a forecast of how many story points they'll complete in upcoming iterations of the same length. The term velocity to refer to the number of story points a team completes, or expects to complete, in an iteration [9].

4 Size Measurement

A software measurement is a quantifiable dimension, attribute, or amount of any aspect of a software program, product, or process. It is the raw data which are associated with various elements of the software process and product. A typical set of metrics might include [10]:

- Quality.
- Size. (target of the study)
- Complexity.
- Effort
- Productivity,
- Cost.
- Schedule.
- Rework.

Two ways for measuring software size were catalogued in 1992 [11]. The first consider the physical source lines and logical source statements. Counts of physical lines described size in terms of the physical length of the code as it appears when printed for people to read.

The other way counts of logical statements, on the other hand, attempt to characterize size in terms of the number of software instructions, irrespective of their relationship to the physical formats in which they appear.

Both, function points as story points measure software size and are based in counts logical statements. Function points address functional size [8] while story points represent the business value of one user story [6].

In Fact, Agile teams separate estimates of size from estimates of duration [9] while function points are complemented by other methods when it comes to effort and cost estimate such as COCOMO II [12].

5 Function Points x Story Points

Although FP and SP estimate the size of the software to be delivered, some particularities make measures applied by both techniques to the same product have different sizes, variations and deviant behavior at the end of the measurement. Some of the strongest of these particularities are detailed in the following subsections.

5.1 Team expertise X Standardized Methods.

A nice feature of story points is that each team defines them as they see fit. One team may decide to define a story point as an ideal day of work and another team may define a story point as a measure of the complexity of the story [9].

In the last statement, Cohn suggest that the story points can vary from several teams based in their experience to assess the effort, complexity and risk associated with certain stories.

Any assumptions made in function points are considered a counting interpretation. A specification bringing perfect requirements, where no assumptions are made, must present the same final result. Any assumptions regarding primary intent must be documented for helps in next counting. Thus function point leaves no space for using expertise.

For Example, considering function point a small function for including one email address in a virtual schedule may be the same “size” of a function with perform a complex integral calculus with receive one equation as parameter and return the string with the result. The same example in story points should present very different results and these results could be different among different teams.

In the other hand, considering function points, store one formulary containing fifteen fields may be different for including one containing sixteen, while in story points this kind of difference is rare.

So this aspect are seeing in a different way in both techniques.

5.2 Functional Size x Product Size.

According ISO 9126 [13] non-functional requirements are that specifies criteria which can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

Considering the IFPUG definition of functional size is the size of the software derived by quantifying the Functional User Requirements we should assume that non-functional requirements are not covered in function points [8].

The IFPUG Framework for Functional Sizing [14] defines some kind of “sizes” present in software development such as functional size, technical size and quality size which are related to:

- **Functional User Requirements**: a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures

that the software must perform to fulfil the users' needs. They exclude Quality Requirements and any Technical Requirements

- **Quality Requirements**: any requirements relating to software quality as defined in ISO 9126:1991
- **Technical Requirements**: requirements relating to the technology and environment, for the development, maintenance, support and execution of the software.

The combination of the functional size, technical size and quality size represents de *Product Size*. But, this concept is not detailed by the IFPUG.

Considering the statement that story point estimate is an amalgamation of the amount of effort involved in developing the feature, the complexity of developing it, the risk inherent in it, and so on [6]. So it is look like the story point is concerned define a product size since the agile team considers any kind of risk and complexity to determine de size of the story, and this assumption are related to Quality and Technical requirements.

Nowadays, IFPUG is building a metric called Software Non-Functional Assessment Process (SNAP). The SNAP Project Team expects to develop a project assessment method that will use a series of questions grouped by category to measure the impact of non-functional requirements on the development and delivery (size) of the software product. The resulting size will be the size of the non-functional requirements, just as the functional size is the size of the functional requirements [15].

In a simple way, we still cannot consider the theoretical concept that SNAP size + FP Size = SP Size because the agile method considers the environment of the project and not just the product.

For example in function points, a bookstore which have no requirements for security, available, performance and its access are made in a local machine, will have the same FP size of this same bookstore considering the same restrictions of the amazon.com for example. In story points the amazon.com will be much larger than its offline, unsecure, slow and unstable version.

5.3 Small Pieces x Whole Product.

In the Agile Manifesto[7] were defined 12 principles which one of them states: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale". This statement reinforces the adoption of interactive life cycle largely adopted in agile projects.

This continuous delivery in small "timeboxes" reduces the total of points delivered in one iteration. Sometimes one big story, called epic, must be disaggregated for fit in one cycle. In fact split stories does not to be a simple task in agile projects.

There are a number of times when it may be necessary to split a user story into multiple, smaller parts [6]. First, a user story should be split when it is too large to fit within a single iteration. Sometimes a user story won't fit in an iteration because it is bigger than a full iteration.

Alternatively, a story may be small enough to fit within an iteration but it won't fit within the iteration being planned because there isn't enough room left. The team may feel they will have time to develop a portion of a story in the iteration, but not the entire story.

Second, it can be useful to split a large user story (an epic) if a more accurate estimate is necessary.

But the question about splitting stories raises from another Cohn's statement [9]: When a story, possibly an epic, is disaggregated into its constituent stories, the sum of the estimates for the individual stories does not need to equal the estimate of the initial story or epic. Similarly, a story may be disaggregated into constituent tasks. The sum of the estimates for the tasks does not need to equal the estimate of the initial story.

[7] www.agilemanifesto.org

Thus, splitting stories seems to be a team decision and there are no rules about how to split and how distribute the points, making this disaggregation a particular process which works only for that team in that environment.

Looking for function points splitting does not to be a problem. No data function or transactional function should be broken because they must follow the elementary process definition: “smallest unit of activity that is meaningful to the user”.

Even if a function must be broken for a technical reason, it only will be considered complete when all of the functionality is completely developed, which means or is delivered zero function points or all function points to the user.

But an anomalous behavior can be seen in the use of function points in interactive and incremental projects if the boundary of the counting just considers what is delivered in each iteration. In this case, the sum of the parts are bigger than the whole.

For example, a particular product is being built in an interactive and incremental whose two iterations have already been completed. In first iteration four features were delivered totalizing fifty function points. In the second iteration another four features were delivered but one feature delivered (and already counted with 10 function points) in first iteration was updated for technical reasons totalizing sixty function points delivered in second iteration.

But A + B totalize a hundred points, but that function which was built in first iteration and updated in the second was counted twice and this just happens because the boundary of the counting is not the whole product and which are delivered in each iteration. In function points the sum of the parts could be bigger than the whole (never smaller).

This problem does not occur in story points because the cost, in points, for updating one feature is embedded in the original story.

5.4 Maintenance and Changes

According ISO/IEC 14764 identified four categories of maintenance [16]:

- **Corrective maintenance:** Reactive modification of a software product performed after delivery to correct discovered problems.
- **Adaptive maintenance**: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
- **Perfective maintenance**: Modification of a software product after delivery to improve performance or maintainability.
- **Preventive maintenance**: Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

Agile software development considers the corrective maintenance as a bug and this kind of problem must not be managed but solved. Which means that must not be sized.

But adaptive maintenance (evolutive maintenance), perfective maintenance and preventive maintenance (refactoring), are considered and evaluated in agile projects as new stories. When maintenance needs to be performed, a new history is written for that specific demand.

The functional size measurement quantifies the size of business requirements. In an enhancement environment, it measures the effects of changes to those business requirements. Therefore, functional size measurement is applicable to a subset of adaptive maintenance. This includes the software functionality added, changed or deleted as well as the software functionality provided to convert data and meet other implementation requirements [8].

Function points clearly do not fit the types of corrective, perfective and preventive maintenance, fitting only a few cases of adaptive maintenance. A project that has undergone many changes may have enhanced the difference in scores between the two approaches

5.5 One Requirement X Many Requirements

Measuring a single feature using the two techniques and compare their variation may be the most logical path to be taken when attempting to evaluate the relationship between the two methods. And repeat this process for the all others features of the project in an attempt to increase the historical basis would be the next step in this comparison.

In terms of story points for this idea may not be the best. Cohn [9] states that the central limit theorem tells us that the sum of a number of independent samples from any distribution is approximately normally distributed.

For our purposes, this means that a team's story point estimates can be skewed way toward underestimation, way toward overestimation, or distributed in any other way. But when we grab an iteration's worth of stories from any of those distributions, the stories we grab will be normally distributed. This means that we can use the measured velocity of one iteration to predict the velocity of future iterations.

Naturally, the velocity of only one iteration is not a perfect predictor. Similarly, velocity may change as a team learns new technology, a new domain, or becomes accustomed to new team members or new ways of working. This means that to predict the behavior of the score, or the velocity, of a team is best to consider all stories delivered than each one individually.

Was also seen in the previous section to consider the score of a feature function points can be misleading. The measure of the sum of the parts can be greater than the whole, thus sizing all features is more accurate than sizing one by one as well.

The last reason to evaluate all the features is that there may not be a perfect match between a story and a requirement. So could be difficult assign exactly which stories are equivalent to what requirements, evaluating all iteration brings a greater reliance to the comparison.

6 Case Study

The *Agência Estadual de Tecnologia da Informação do Estado de Pernambuco*[8], hereafter called ATI, is following the Brazilian Federal Government instruction known as the: *Instrução Normativa 04 de Maio de 2008*, hereafter called IN04, which came into force on 2 January 2009 the Department of Logistics and Technology of the Planning Ministry [17].

This instruction in its article 14 states that the outsourcing strategy must define the understanding of the task to establish procedures and criteria for measurement of services provided, including metrics, indicators and values. With this the technique Function Point Analysis has been adopted as currency in the local authority outsourcing contracts of software products.

ATI was forced to be adherent to this instruction in early 2010. Before that, since January 2009, ATI has been using Scrum as tool for contract management [17]. After the adoption of this instruction ATI continued to manage its suppliers through the scrum, but the payment of invoices should be measured based on the product delivered sized using function points.

ATI and its supplier held a planning poker meeting where it is sized in story points all demands of that sprint. But now it was necessary to conduct an estimative counting in function points required by IN04 for project planning. At the end of the sprint is still necessary to perform a counting in function points to determine the size of the product delivered and thus pay the suppliers.

The estimative counting is needed only for the allocation of project resources, which does not demand an accurate count but only an approximation of reality. But while ATI can count about 5000 story points per day, the ability to count function points is reduced to 600 function points in a day. And this estimative counting is “bureaucracy” being unnecessary in most cases.

[8] www.ati.pe.gov.br

The record of the demands is held by ATI supported by a tool called Mantis as shown in Figure 2. The functional size and story points size of each demand is stored in this tool and can be recovered directly from the MYSql database which store the Mantis database.

Figure 2. Recorded data about one demand showing SP and FP in Mantis adopted by ATI.

ATI intended to reduce its work performing the estimative counting at the beginning of the Sprint. Based on the idea that the functional size (FP) is a part of the product size (SP), was cogitated the possibility of creating a method of conversion between the two metrics.

The basic idea was realize a statistical correlation between the two counting results (intention of this work), and if the correlations prove strong enough, will be performed a linear regression between the two (not finished in the present moment).

To the kick off project was a selected sample of 18 sprints from February 2009 until August 2010 because this is the all period of historical basis of story points and function points contained in the database. This implies a total of 18 results (Feb 2009 - Aug 2010) for each data sample containing 2191 demands recorded.

First, will be presented the variables and their total values within a Sprint in Table 1. PH and PF represents the amount of story points and function points collected in each month respectively. The statement Fev/09 until Ago/10 represents the sprints performed (February 2009 until August 2010). The statements *Média* and *Desvio Padrão* represent the average and standard deviation respectively.

Table 1. Data from two variables in the sample

	fev/09	mar/09	abr/09	mai/09	jun/09	jul/09	ago/09	set/09	out/09	nov/09	dez/09	jan/10	fev/10	mar/10	abr/10	mai/10	jun/10	jul/10	ago/10	Média	Desvio Padrão
PH	540	437	787	593	474	648	787	758	535	480	262	373	312	506	358	819	742	469	652	554,3158	171,0065147
PF	64	41	67	51	65	130	156	159	106	91	54	45	43	71	49	90	74	66	71	78,57895	35,75583214

The first step to perform the statistical correlation should be to test the normality of the variables, SP and FP, involved in the correlation. The two variables had their normality evaluated using the Shapiro-Wilk test to determine if the correlation method, next step, must be parametric or non-parametric [19]. The statistical tool used in this work was the R software. The results of the normality test are found in Frame 1.

```
        Shapiro-Wilk normality test

data:  PH
W = 0.9519, p-value = 0.426
```

```
        Shapiro-Wilk normality test

data:  PF
W = 0.8438, p-value = 0.005318
```

Frame 1. Shapiro-Wilk normality test result.

The normality test to the PH variable (Story Points) was considered **normal** while the variable PF (Function Points) were considered **not normal**, hence the method of statistical correlation must be a non-parametric. The chosen one was the Spearman rank correlation [19].

The Result of the Spearman's rank correlation is shown on Frame 2.

```
        Spearman's rank correlation rho

data:  PF and PH
S = 326.2862, p-value = 0.0005989
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.713784
```

Frame 2. Spearman's rank correlation result.

The result of the Spearman test rho (ρ) indicates the degree of linear correlation between the two variables. The value of ρ can range from -1 (negative correlation) to 1 (positive correlation) where |ρ| close to zero indicates a weak correlation and |ρ| close to 1 indicates strong correlation.

The value of $\rho \approx 0.7137$ means a strong positive correlation. The p-value indicates the confidence interval of the test, which is much lower value 0.05, thus indicates a large confidence interval. For visual verification of the strength of linear correlation, we have constructed a scatter plot which is shown in Figure 3.

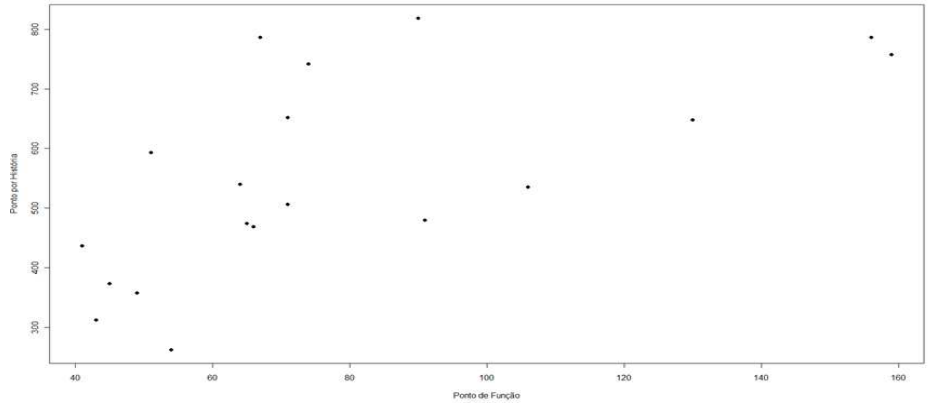

Figure 3. Scattered Plot of Variables PH (Story Points) and PF (Function Point).

The results shown in the scatter plot presenting points growing on a linear pattern, which support the Spearman's test on the correlation.

7 Conclusions

Despite the strong differences of size definition presented in function points and story points, were presented empirical evidence for a real life project realized by a Brazilian public agency, showing the correlation between functional size and number of story points which are delivered at the end of each sprint.

The strength of this correlation suggests a further distancing between the two variables studied which may come from the differences presented in section 4 of this work.

The result cannot be generalized, but it supports an idea that Product Size = Functional Size + Non-Functional Size + Environments Variables Size, ie Story Points = Function Points + Non-Functional Size + Environments Variables Size. This "formula" is not intended to be shown mathematically correct, but that represent the functional size is part of the product size and find a correlation between the whole (product size) and the piece (functional size) represents a valid proportion.

Obviously is necessary to respect the units of measurement and the reality of each organization, so the result itself is not valuable, but the method of assessment, if replicated in more environments, may prove useful for a particular company.

7.1 Discussion of results

Even being used to the same goal, function points and story points presents strong theoretical differences. Whereas the results of this study it is still surprising. Seeing a correlation between the functional size that is obtained accurately with impersonal method of sizing and story points obtained purely from the experience of the team.

Especially if we evaluate this short history about this subject, starting with the Fuqua's work [7] where he performed a correlation between function points and a set of indicators used by your company, then performed the same correlation using story points and found no significant correlation between these two variables.

Although this study [7] has been used another function points, now known as mark II [20] which is different form official function points provided by the IFPUG. The basis of impartiality in Mark II counting method remains. The work of Jones [1] only presents the statement of the relationship between FP \ SP = 2 without more information on how the result was obtained.

It is obvious which the intention of this work was not to generalize their results, but was expected to find the same results than Fuqua [7] and close it as more empirical evidence strengthening the argument toward to "not fit".

Facing the expectations and the results, we believe that the statement raised in the Framework for Functional Sizing of IFPUG [14] where it states that product size is a combination of (quality size, technical size and functional size) or Product Size = Non-Functional Size + Environments Variables Size + Functional Size.

Of course that different companies presents different "sizes" for their story points and different proportion of the impact of functional size into the product size, but the goal of this paper is to motivate of how these companies can find their ratio between FP and SP.

7.2 Implications for Research and Practice

The implications of this study for practice in first place concerns to own ATI, and the possibility for perform a linear correlation for find a conversion method between story points and function points.

Another practical implication is the description of a method that can be used by companies that are facing the same problem of the ATI and need a solution to how to assess the relationship between FP and SP within their organization. Remembering that the values found in this work will be only valid for that ATI project, but organizations can use this method in its own database and so finding their own correlation.

Those that are successful, including own ATI, can perform a linear regression and find a first degree equation ($y = Ax + B$) where y refers to the number of function points, x is the amount of story points and A,B are constants. From this equation, companies can predict with a certain margin of error, which is the value of these variables from one another.

The first implication of this work in research is to present other empirical results joining a small base of scientific information about the subject and the first empirical study, not considering Jones' work [1], adopting IFPUG method.

Other implication is formally presents the main differences between the two approaches in section four. Surely, there is plenty of theoretical information compiled about the subject. But still is the possibility for gathering current data from systematic reviews or systematic mapping as well as creating data from new experiments and case studies that will enrich the knowledge of the academy.

In order to present the idea that story points are related to product size and function points, or functional size, is only part of the product size. What seems clear is that the proportion between functional size and product size is different in every environment and can even be irregular within the same project.

7.3 Threats to Validity

The first group of threats to validity stems from a lack of theoretical concepts consolidated about a possible correlation between the approaches. This fact may have contributed to weaken several factors in this study such as the wrong selection of the method or the pooling of demands. It is an "exploratory" study which portrays a more specific need than a company that intends to conduct scientific research. Indeed, this threat not touches the section 5 of this work that could bring contributions if we were free of the limitations that are in the following section.

Another threat comes from this factor is the lack of information about a demand that could help in their treatment. For example there is no way of knowing whether a demand is perfective or corrective maintenance (which could be dropped from the study because function points do not support them) or whether it is adaptive maintenance. Another problem with this group is that the number of samples (18) is still small to reach any definitive conclusion on this study.

Finally the latest threats come from the validity of the statistical method used in this work. The lack of knowledge prevents to determine which type of method is most appropriate for the conduct of case studies and experiment. For example, the statistical method used in this work and the work Fuqua's [7] were different. In this method the assumption of measure the set of demands instead one individual (section 5.5) can bring bias to this study.

6.4 Limitation

The main limitation is the small amount of professionals who knows well the two techniques involved in this study. The impact of this work is the small number of sprints that could be counted because it was none the counting of function points from September until December, which would be four more sprints for data collection.

The second limitation was not performing the linear regression to support with more strength the results of the work, although with a rank correlation of 0.71 and a high confidence interval is very difficult that there is not a valid linear regression for this correlation.

6.5 Future Work

In industry, one future work is suggested that is the discovery the first degree equation FP (x*SP + y) where FP is equal to the total function points delivered after Sprint, SP represents the estimate given by points in history, x and y are constants. This time it included a regression analysis to identify the function conversion between the variables.

To academy we present as future work the attainment of studies using formal secondary collect data method such as systematic review or systematic mapping on the relationship between function points and story points.

Another future work is providing more empirical information about the relationship between FP and SP, to confirm the relationship Product Size = Functional Size + Non-Functional Size + Environments Variables Size.

Acknowledgments

Célio Santana is a doctoral student at the Center of Informatics of the Federal University of Pernambuco where he receives the funding from the Brazilian National Research Council (CNPq), process #141156/2010-4. The authors thank CNPq for partially funding the participation in the XP'2011 Conference.

References

1. Jones, C.: Applied Software Measurement, MCGraw Hill. (2008)
2. Albrecht, A. J.: Measuring Application Development Productivity. In Proc. IBM

Applications Development Symposium. GUIDE Int and Share Inc., IBM Corp., Monterey, pp. 83. (1979)
3. ISO/IEC 14143-1:2007, Information technology — Software measurement — Functional size measurement (2007)
4. Mnkandla, E., Dwolatzky, B.: Balancing the human and the engineering factors in software development. Proceeding of the IEEE AFRICON 2004 Conference, pp. 1207--1210, (2004).
5. Boehm, B., Turner, R.: Balancing agility and discipline: A guide for the perplexed 1st ed. pp. pp. 165-194, Addison Wesley (2004).
6. Cohn, M.: Agile Estimating and Planning. Addison-Wesley. (2005)
7. Fuqua, A.: Using Function Points in XP – Considerations. In Proceedings of Extreme Programming and Agile Processes in Software Engineering, Springerlink.
8. IFPUG: Counting Practices Manual V. 4.3. Available in www.ifpug.org. (2009)
9. Cohn, M.: User Stories Applied. Addison-Wesley. (2004)
10. Glass, R. L.: Building Quality Software. Prentice-Hall. (1992)
11. Park, R. E.: Software Size Measurement. A Framework for Counting Source Statements. SEI technical report available in: http://www.sei.cmu.edu/reports/92tr020.pdf. Last Access in 03/01/2010.
12. Boehm, B., Abts, C., Brown, A. W., Chulani, S., Bradford, K. C., Horowitz, E. Madachy, R., Reifer, J., Steece, B.: Software cost estimation with COCOMO II. Englewood Cliffs, NJ:Prentice-Hall. (2000)
13. ISO/IEC 9126-1: Software engineering — Product quality — Part 1: Quality model. (2001)
14. IFPUG: Framework for Functional Sizing. Available in www.ifpug.org. (2003)
15. IFPUG: Software Non-functional Assessment Process. Available in http://www.ifpug.org/about/SWNon-FunctionalAssessmentProcess_Final20-2009.pdf . Last Visit in 03/01/2011 (2009)
16. ISO/IEC 14764: Software Engineering — Software Life Cycle Processes — Maintenance (2006)
17. Department of Logistics and Technology.: *INSTRUÇÃO NORMATIVA Nº 4, 19 de maio de 2008*. Available in: http://www.governoeletronico.gov.br/anexos/instrucao-normativa-n-04. Last Access in 30/08/2010. (2008)
18. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall. (2004).
19. DeGroot, M., Schervish M. J.: Probability and Statistics 3 edition, Addison Wesley. (2001)
20. United Kingdom Software Metrics Association (UKSMA): "MKII Function Point Analysis Counting Practices Manual", Version 1.31 (Mk II FPA). Online at http://www.uksma.co.uk (1998)