Function point analysis by an SAP application

Katalin Nátz*, and Tamás Orosz**
* Szent István University, Budapest, Hungary
** Óbuda University/AMK, Székesfehérvár, Hungary
* natz.kata@gmail.com
**orosz.tamas@amk.uni-obuda.hu

***Abstract*— Nowadays from the beginning of the 21st century, the buying interest in SAP technology and Function Points Analysis (FPA) was very high. On the one hand, the software provides the IT landscape for business processes and enables the reusability of the software, which means most of all a big profit for customer. In addition, various solutions allow for the user.**

Of course, one can ask the question whether it pays to count the function points in the SAP systems. The answer is clearly yes. Above all, there are several reasons to measure the size of the SAP systems. For example, im-prove productivity of enhancing SAP. You can determine the purchase decision for purchased packages. You can determine the financial impact of an ERP implementation or improve your financial planning. This is why Func-tion Points (FP) has become the standard measure of software controlling and planning at many companies. There are, of course, big challenges to be overcome: time constraints, broad functional scope, highly integrated into the application, which requires the exact definition of application limits, etc.

***Keywords*: Function point, SAP technologies, Software measurement, Business processes, Financial planning**

I. INTRODUCTION

Applications and large systems like ERPs and the popular cloud platform systems are growing very fast. They are so complex and use such a huge amount of big data that they are becoming extremely difficult for the users to understand. The new coding tools already allow software developers to produce larger volumes of software to meet the ever-growing and complex user requirements. A clear and comprehensible method must be used for understanding, estimating and reviewing the projects.

FSM[1] methods are meant to measure the software size by quantifying the functionality provided to the users. In particular, the Function Point Analysis (FPA) was the first FSM method [1] and was originally introduced in the mid-1970s. Nowadays it is used by organizations worldwide. IBM's Allan Albrecht was the first to publish a method for functional sizing of software called Function Point Analysis (Albrecht, 1979, 1981) [3][9]. His name is inextricably linked with Function Point analysis. As an IBM employee, he developed Function Point analysis in the 1970s to assess the productivity of software projects [4]. Since its foundation in 1986, the International Function Point Users Group (IFPUG) has continuously maintained and developed the original Albrecht methodology for functional software (IFPUG CPM) [3][4]. In 2009, the new version CPM 4.3.1 of the IFPUG Function Point standard was published which has been in use since the beginning of 2010. [4]

The "function point" defines the number and functional size of the functions which are implemented or changed during the considered development, including the functions supported by this IT system. Function point analysis (FPA) can measure entire applications, even projects or releases. FPA is independent of the type of application or the type of implementation. It is a structured problem-solving solution, which breaks down systems into smaller components with the purpose of better understanding and analyzing them.

The appearance of the functional point technique has enabled the ICT community to increase the practice of software measurement with respect to the use of the "lines of code" (LOC) [5] approach. The FP count requires a clear, complete and detailed description level, e.g. design and application documents [2].

Meli and Santillo mention two possible cases where an alternative estimation meth-od compatible with the standard rules for FP could be crucial. The first case occurs when the development or enhancement project is in such an early stage that it is simply not possible to perform an FP count in accordance with IFPUG standards. The second case occurs when a survey of the existing software asset is required, but the required documentations and resources are not available for the detailed FP calculation. [2]

Because of these and other similar situations, the need for methods to estimate function points has been increased for organizations involved in the software business. This need is even more urgent for small businesses, which need usable and cost effective software.

II. FUNCTION POINT ANALYSIS (FPA)

To conduct a Function Point Analysis, we need the functional requirements provided by the project team. We differentiate between transactional functions and data func-tions. After identifying the transactional functions and data functions we have to determine the complexity and the functional size for each. Regardless of the imple-mentation technique used, it is possible to determine the size of the application. With the help of the results the average productivity can be estimated. If you know the experience of productivity, you can roughly calculate the project hours and the associated budget. [4]

A. Basics of FPA

FPA is in fact an evaluation process that defines the functional size of IT systems provided to users. During the measurement we do not evaluate the system itself, but to

[1] FSM: Functional Size Measurement

what extent it supports the implementation of business processes. In the course of the process, the developments implemented within the scope of the given project will be analyzed. Thus, the result of the FP analysis shows the functional size of the development.

Function points are the quantity of the number and scope of functional transactions supported by an IT system. In order to determine the function point value of an IT application or IT requirement, one investigates or determines the number of data functions managed thereby, as well as the input, output and inquiry functions available for the users. [4]

If one wants to determine the function points, the user view is decisive for the de-termination of the functional size. There are detailed rules in the standard for assigning the point value to transactions and databases. According to the number of elementary functions and the standardized values, the assigned FPs is accumulated.

In the FP analysis, we should also talk about the elementary processes. This is the smallest step that is meaningful to the user and constitutes a complete transaction. An elementary process is counted as the same compared with another one if it re-quires the same set of Data Element Types requires the same set of File Type Referenced and requires the same set of processing logic to complete the elementary process. [4][8]

B. Why or why not should we use the FPA method?

FPA offers many advantages over other software sizing methods [4][8][9]:

- It is independent of the programming language, development technology, plat-form or the knowledge of the project members.
- It connects directly with user requirements and features.
- Those who engage in FPA analysis play a consultative role in this case. The function points can help to create effective communication between the developers and users, and to give a better understanding to the non-technical user.
- It can be used on both "Waterfall" and "Agile" projects.
- It is applicable throughout the software development lifecycle.
- It provides a way to determine productivity and estimate the cost.
- FPA provides consistent, documented and repeatable measurement methods.
- FPA can highlight gaps in functional requirements, thus avoiding the early introduction of errors in the application.
- A goal is also to minimize the effort and severity of the measurement process.

C. FPA has also contras [1][7][11]:

- If there is no information about the complexity, it can be only use the rapid meth-od, which can corrupt the result. On the other hand, I have information on complexity, but it cannot be clearly counted, and hardly appreciated.
- Albrecht gives many advices , to identify logical internal files, but it can be very difficult to identify after that. IFPUG publishes many examples, but those are also very open.
- The applications have features and properties that can be almost the same for much software (such as SAP). Therefore, we can also make a big mistake in the survey, because the functions for the similar software are not the same in other companies.

D. Transactional functions

The transactions represent the functionality provided to the user for processing data by application. There are three transactions - external inputs (EIs), external outputs (EOs) and external inquires (EQs).

External Inputs: The main purpose of an input is to maintain one of several internal databases or to change system behavior. An input contains processing logic for processing technical data or control information that enters the application be-yond the limit [4][9].

External Outputs: The output is the presentation of information to the user. It contains at least one of the following forms of processing logic: mathematic computations, maintenance of datasets, generation of derived data, change of system behavior [4][9].

External Inquires: An inquiry is the presentation of information to the user, and it references a dataset to read technical data or control information, and it does not meet the requirements for an EO. It can be also a list box. It is important to mention that sorting and arrangement of data cannot be evaluated at all. [4][9].

In practice, the difference between EO and EQ is often difficult to impossible. A query should only be evaluated if it is clearly evident that the requirements for an output are not met, e.g. in a "simple" list box or multiple search.

E. Data Functions

Data functions show logical data groupings and databases that the user needs for his work. Data functions have two types - Internal Logical Files (internal data) and External Interface Files (external data).

Internal Logical Files (ILF): A dataset that is maintained within the considered application is classified as internal dataset: user-recognizable, logical groups of functional data, maintained by elementary processes of the application [4][7].

External Interface Files EIF): A dataset that is read-only but not maintained within the considered application and that is classified as ILF in at least one other application is classified as an External Interface File: user-identifiable, logical group of functional data that is not changed in the application / to be cared for. It is maintained by elementary processes of another application [4][7].

F. Non-functional categories [12][13]

Regarding to the analyzed sprints and the references we have specified a non-functional categories for the customer services company.

TABLE 1. NON-FUNCTIONAL CATEGORIES

Criteria	Comments/Examples for Collection
	• Security & Compliance: Effort to comply with the relevant requirements regarding security, compliance, legal, revision, etc. /Security requirements (confidentiality, in-formation security, data integrity, availability) • Stability: Effort for stability, availability, robustness (this also includes expenses for HWR measures) Reliability • Performance: Effort due to higher data volume, number of users, access and response times / performance and efficiency (response times, resource requirements, cost-effectiveness) scalability • Usability: Efforts to improve the usability and accessibility of apps, appearance and handling (look and feel), usability (comprehensibility, learnability) • Quality: Flexibility (support for standards), maintainability, changeability (analyzability, stability, testability, extensibility), portability (adaptability, installability, conformity, interchangeability) • ITSD: Effort to comply with the IT standards (patches, etc.), consolidation (code, documentation) and quality assurance, to comply with the maintenance contracts mandatory upgrades required • Incidents: Efforts for avoidance (business) errors / incidents, correctness • Migration: Migration effort (data migration and cleanup) • Support: Effort for support in requirement specification, user test and user training, process test and active operation validation • Launch: Effort and cost for launch • Operational Shares: Efforts to ensure optimized commissioning and efficient operation • Configuration: Efforts for the implementation of short-term application and product configurations • Retire: Effort for the shutdown of applications that are not functional (dismantling / conversion) can be assessed • HWR: Hardware Refresh • Other: Other non-functional efforts

G. Complexity of the transactions and data functions

Important for the determinations of the complexity are: for a transaction the number of used fields and datasets and for a dataset the number of included fields and field groups.

TABLE 2. COMPLEXITY FOR TRANSACTIONS AND DATASETS

Elements	Complexity		
	Low	Average	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

Both of these files include Data Element Type and / or Record Element Type. A data element type (DET) is a unique user recognizable, non-repeated field or attribute of ILF or EIF. A record element type (RET) is a user recognizable subgroup of data elements in ILF or EIF. When ILF / EIF are referred to as Transactional Functions for Processing Information, they are termed as File Type Referenced (FTR) [9].

TABLE 3. COMPLEXITY OF EI AND EQ

FTR	DET		
	1-4	5-15	> 15
0-1	Low	Low	Average
2	Low	Average	High
> 2	Average	High	High

TABLE 4. COMPLEXITY OF EO

FTR	DET		
	1-5	6-19	> 19
0-1	Low	Low	Average
2-3	Low	Average	High
> 3	Average	High	High

TABLE 5. COMPLEXITY OF ILF

RET	DET		
	1-19	20-50	> 50
1	Low	Low	Average
2-5	Low	Average	High
> 5	Average	High	High

TABLE 6. COMPLEXITY OF EIF

RET	DET		
	1-19	20-50	> 50
1	Low	Low	Average
2-5	Low	Average	High
> 5	Average	High	High

III. A CASE STUDY – FUNCTION POINT ANALYSIS OF A SCP APPLICATION

A. Function point analysis of the customer module of a SCP application

The analyzed company is a service company that deals with customer care and in-formation technology. Since 2016, the company has been developing an SCP application to analyze whether it is worth introducing and fulfilling customer expectations.

The company has been developing this software agile for 3 years so that flexible development is possible and unnecessary documentation is spared. In the development

should be justified, if the software was developed according to the expectations and can be continued. In addition, a possible KPI range should be determined if the data can be meaningfully derived from the data.

B. Steps by the counting [6]

According to IFPUG, following are the steps for Function Point counting:

- Determine the type of Function Point count and the application boundary
- Identify data functions and their complexity
- Identify transactional function and their complexity and non-functional part
- Compare the results in each year, then calculate a possible KPI for each category using budget

C. Counting of function points and non-functional categories

If a greenfield project is to be measured, that is a grateful task. You can get to know the new application very well. But there is also the challenge of finding all the functions and measuring them according to the regulations and to form a KPI with the data to be used.

After completing the application, the following information should be mentioned: The application is designed for customer service to store customer data and support product sales. The purpose of the application is to create a customer database, ar-chive the data and enrich it in a system that can be searched both offline and online. The enriched information should be exported at any time. The user should be in-formed about the activities at all times. The individual fields should be edited, maintained and deleted, but the history should always be available.

As far as the agile software projects are concerned, it can be said that it is a greefield area, you can find less analysis in this case. During the measurement, more than 260 sprints were analyzed and categorized regarding the functional parts.

TABLE 7. SPRINTS OF TRANSACTIONS AND DATABASES

	2016	2017	2018
EI	9	42	49
EO	11	26	21
EQ	9	37	35
ILF	11	6	3
EIF	2	4	2
WA	42	115	110

It is important to mention that it was the first time that the company developed agile. The experiences are still collected.

TABLE 8. COUNTING OF TRANSACTIONS AND DATABASES

	2016	2017	2018
EI	54	4	6
EO	77	5	7
EQ	54	4	6
ILF	165	10	15
EIF	20	7	10
SUMMARY	370	786	716

If we look at the results of the measurement we can see that all transactions and datasets belong to the category "high" regarding to the counting of RET, DET and FTR of functions and databases. In point 2.6 we can find the details for the complex-ity and the calculation of the elements.

TABLE 9. CALCULATION OF FP / EUR

	2016	2017	2018
EI	772 €	698 €	655 €
EO	972 €	999 €	954 €
EQ	949 €	912 €	955 €
ILF	461 €	846 €	848 €
EIF	643 €	738 €	821 €
SUMMARY	694 €	847 €	821 €

If the results are looked at in the FP / Eur calculation, it can be stated that the values are almost between 500 and 1000 FP / Eur. Compared to the FP table, it can be seen that the FP / Eur value is also higher with the increased FP count, which is probably related to the higher and functionality and complexity of SAP.

TABLE 10. SPRINTS OF NON-FUNCTIONAL CATEGORIES

	2016	2017	2018
Support	89	83	121
Configuration	226	95	201
Usability	26	27	68
Quality	4	4	4
Security	5	5	7
Incidents	12	9	32
Stability	8	2	2
Performance	4	5	5
Other	2	2	17
Summary	376	232	457

According to the non-functional list in paragraph 2.5, we identified 9 categories during the measurement. As it is a greenfield application, it is no surprise that the highest number belongs to the configuration.

TABLE 11. COUNTING OF NON-FUNCTIONAL CATEGORIES

	2016	2017	2018
Support	567	433	572
Configuration	1439	495	950
Usability	166	141	322
Quality	25	21	19
Security	32	26	33
Incidents	76	47	151
Stability	51	10	9
Performance	25	26	24
Other	13	10	80
Summary	2394	1209	2161

The valuation of non-functional parts without automation is a developing area. The nFP-s are rated here by the budget and the hours. Several evaluations could be used to deduct clear conclusions and make suggestions for the calculation.

TABLE 12. CALCULATION OF NFP / EUR

	Non-functional categories		
	2016	2017	2018
Support	4 005 €	4 178 €	3 387 €
Configuration	4 761 €	4 689 €	4 479 €
Usability	4 088 €	3 937 €	3 956 €
Quality	3 601 €	3 882 €	3 632 €
Security	3 850 €	3 739 €	4 168 €
Incidents	3 631 €	5 582 €	3 070 €
Stability	3 165 €	3 814 €	3 382 €
Performance	4 614 €	3 882 €	4 082 €
Other	981 €	7 128 €	1 530 €
WA	4 420 €	4 415 €	3 882 €

According to the table, it can be seen that most categories changes in a certain range. The exceptions are incidents and others; where in this case is about bug fixes or unreacted requirements.

IV. CONCLUSION

This article presents results of a case study conducted to analyze existing FPA meth-od in an SAP application where function points are used as a unit of measure to determine a KPI range. We have analyzed the application for 3 years in the user view.

All studies advise to use COSMIC FPA for size estimation. But as the documentation is often incomplete in an agile development, it should conduct many interviews, select the information and identify the data movements. In this case, there are such assumptions that no automated estimates can be made. [15]

This study does not directly provide a cost estimation refinement for the SCP pro-jects for the business scenarios. Instead, it provides an estimation matrix for the analyzed company that represents estimation strategies based on the situation, historical data availability, and knowledge base [14].

These studies show us that business processes are valuable resources for effort estimation. FPA method in the software projects could be a good candidate for the size estimation of SCP projects.

Since the SCP ideas are developing rapidly in addition to the SAP solutions, the FPA method in this area should also be objectively applicable to an FPA user. However, it is not easy to get accurate rule of thumb estimates, especially for new custom requirements. In the case study, the rule for the measurement was in a wide range, which could lead to estimation errors.

These conclusions cannot be generalized to all types of estimation scenarios for ERP systems due to the limitations of the case study. The main limitation of our study is that we only applied the function point-based estimation method to a specific type of ERP system, SCP. Another limitation is the sample. We only applied the method to one project. Further studies are needed, which also cover other cloud ERP systems and different types of ERP projects.

In the study, we found that FPA method is a good option for SCP projects. Critical parameters for this are the incomplete documentation, the decisive change and extinguishing scenarios that are not marked and the extra functionality that should be precisely defined and calculated with the COCOMO model. Further FPA-based studies should be carried out considering this model.

REFERENCES

[1] A.J. Albrecht: Measuring application development productivity, in: Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, 1979, pp. 83–91.

[2] R. Meli, L. Santillo: Function Point Estimation Methods: A Comparative Overview (1999), https://www.researchgate.net/publication/2462027_Function_Point_Estimation_Methods_A_Comparative_Overview, last accessed 2018/03/24

[3] CISQ Homepage, http://it-cisq.org/wp-content/uploads/2014/11/Automated-Function-Points-Specification-OMG-Formal-January-2014.pdf, last accessed 2018/03/24

[4] B. Pönsgen: Function-Point Analyse. Ein Praxishandbuch. 2nd edn. Dpunkt.verlag, Heidelberg (2012)

[5] N. Balaji, N. Shivakumar & V. Vignaraj Ananth: Software Cost Estimation using Function Point with Non Algorithmic Approach, Global Journal of Computer Science and Technology Software & Data Engineering Volume 13 Issue 8 Version 1.0 Year 2013

[6] IFPUG: Function Point Counting Practices Manual, Release 4.0. Couting Practices Committee, The international Function Points Users Group (2000)

[7] D. Longstreet: Fundamentals of Function Point Analysis, Total Metrics

[8] S. Palmquist, M.A. Lapham, S. Miller, T. Chick, I. Ozkaya: Parallel Worlds: Agile and Waterfall Differences and Similarities. Software Engineering Institute (2013)

[9] P. Vickers: An introduction to function point analysis, Northumbria University, https://www.researchgate.net/publication/250894239_An_Introduction_to_Function_Point_Analysis, last accessed 2018/03/24

[10] C. Dekkers: Counting Function Points for Agile / Iterative Software Development; http://www.ifpug.org/Articles/Dekkers-CountingAgileProjects.pdf; last accessed 2018/05/06

[11] B. Molnár: Funkciópont elemzés a gyakorlatban; http://www.mtaita.hu/hu/Publikaciok/FunkcioPont_1.pdf, last accessed 2018/05/08

[12] Gregor v. Bochmann: Non-Functional Requirements

[13] http://www.site.uottawa.ca/~bochmann/SEG3101/Notes/SEG3101-ch3-4%20-%20Non-Functional%20Requirements%20-%20Qualities.pdf, last accessed 2018/10/20

[14] https://www.scaledagileframework.com/nonfunctional-requirements/, last accessed 2018/10/20

[15] http://www.ifpug.org/Conference%20Proceedings/ISMA3-2008/ISMA2008-16-Maschino-can-SAP-be-function-point-counted.pdf , last accessed 2018/10/20

[16] Ömüral, Demirörs: Effort estimation methods for ERP projects based on function points; http://openaccess.iyte.edu.tr/bitstream/handle/11147/6734/6733.pdf?sequence=1&isAllowed=y, last accessed 2018/10/20,