Super Fast Size and Effort Estimation

Pekka Forselius and Erkki Savioja

Finnish Software Measurement Association

pekka.forselius@fisma.fi and erkki.savioja@fisma.fi

Abstract:

Functional size of software is one of the most important cost drivers of software development projects. Every clever and knowledgeable person agrees with this statement. It is just common sense that says that developing bigger piece of software takes more effort and money than developing a small one. However, function points are very rarely used in the industry, although they are the only internationally standardized way to measure size of software. The main reason is that neither customers nor top managers of the developer companies require use of systematic functional size measurement (FSM). In most of the cases the decision makers don't even know that any ISO standardized methods for FSM exist, and that there would be trained measurement experts available, often even closer than they could imagine. Sometimes the decision makers have heard about function points, but their knowledge of them is old, or they have been misguided to believe that the methods are difficult to use, extremely time consuming, and that the counting results may vary significantly between two measurement experts. Their belief is all wrong.

Finnish Software Measurement Association (FiSMA) arranged a speed evaluation for software size estimation and project effort estimation in summer 2013. The results are impressive and very promising. During the three summer months all together 22 persons (all trained for FSM) estimated five projects in two steps: first the functional size of the target software and secondly the related development effort for the project. Size measurement speed varied from 45 seconds to 8 minutes, and the complete effort estimation, including the size calculation, from less than 3 to 18 minutes. An additional great thing is that not only the estimation results but also all the estimation parameters used by each of the 22 evaluators have been documented carefully. If any remarkable differences occur between the evaluators, the reasons can be checked and explained. In addition to the pure estimating speed we also wanted to compare the estimation results, i.e. the sizes and the effort estimates. The two main organizers of the evaluation generated separately the "teacher's solutions" for the estimates, and as soon as they matched perfectly, these were used as benchmarks for the other evaluators. 99 out of 110 size estimates were within +/- 2 % of the benchmark and in the effort estimation evaluation 93 estimates differed less than 20 % from the benchmark. Isn't this something to research in more details, or even to try by yourself? It is easy to arrange new speed evaluation events when ever and where ever there are ECQA accredited northernSCOPE™ CSM training partners available to help and coordinate the arrangements (ECQA = European Certification and Qualification

Association[60], and CSM = Certified Scope Manager, one of many ECQA certified job roles).

Key words

Software project estimation, Functional Size Measurement, function points, scope management, software metrics

1 Introduction

Finnish Software Measurement Association is a registered association: „FiSMA, for better management“. The full members of FiSMA are industrial and academic, software intensive organizations, including both software customer and supplier companies, and public organizations. Common to all members is their interest in use of metrics and measurement, to enable better management. Representatives of FiSMA members collaborate in several theme-based working groups that have been established around different topics, e.g. ISO/IEC JTC1/SC7 standards, process improvement, ICT research. Scope Manager Forum is one of the FiSMA working groups. Its members are all Certified Scope Managers, kind of ‘software engineering economics experts’, whose special interest areas are ICT acquisition management and ICT program management [40,50]. They are all familiar to Functional Size Measurement (FSM) methods and a program management concept called northernSCOPE™, which is one of the trademarks of FiSMA. The northernSCOPE™ concept creates the theoretical framework for the ECQA CSM training and certification. About 50 % of the 120 Finnish Certified Scope Managers are members of the FiSMA Scope Manager Forum, where they can share their measurement related worries and joys with colleagues from other organizations. This Forum was the starting background and its members were the first target attendees for the FiSMA size and effort estimation speed evaluation in summer 2013.

Although common interest in software size estimation and measurement has always existed among the Scope Manager Forum members, the innovation to organize speed evaluation event(s) needed still some external impulse. For it we have to thank Capers Jones, who is one of the best known function point experts in the whole world, and who has been musing about the success factors of institutionalization of use of function points for years. He sent us an interesting research paper [10], where he claims that FSM should be the leading metrics in software industry. In his paper Capers Jones presents a well thought list of pros and cons of functional size measurement, and points out that the size estimation speed is often considered as the main barrier for final success of function points’ systematic use. He published very interesting and impressive results from his own speed experiments: he had counted function points for 40 applications in just 75 minutes, reaching the average counting speed less than 2 minutes per application.

For the first eye we at FiSMA thought that it sounds unbelievable, but typical for Capers Jones, who loves to show exact measurement results from all kind of software issues. After more careful thinking, some research and lot of discussion, we understood that his ideas made perfect sense. Excellent estimation speed and very low estimation cost are definitely the key drivers that the industry needs to get convinced of the necessity of functional size estimation and measurement. We decided to take the challenge, and try how fast could our own methods and tools show up in use.

Caper Jones used his own pattern matching, patent-pending SRM method and the Software Risk Master™ tool in his experiment. The outcome size from the method and tool is expressed in IFPUG function points in his case. FiSMA Scope Managers prefer FiSMA 1.1 FSM method, which is the ISO/IEC 29881:2010 standard for functional size measurement [20]. The main reason for this preference is that functional user requirements matching to the Base Functional Component (BFC) types of FiSMA 1.1 are easier to recognize than those matching to BFC types of other ISO standardized FSM methods, especially in case of very early specifications of user requirements. Other reasons to prefer FiSMA 1.1 method are the scalability of the method [30] and the better customer controllability over the functions from early size estimates to final functional size measurement of the delivered software. So, it was natural to choose FiSMA FSM method in our case. However, size measurement without any other information connected to the result is not very interesting or useful for anyone. That's why we decided to complete the size estimation result with other measures needed to provide a project effort estimate. That was easy to arrange, because the Experience® tools included in the CSM training support both size estimation and effort estimation. Figure 1 illustrates the effort estimation process of the northernSCOPE™ concept as an Input-Process-Output chart.

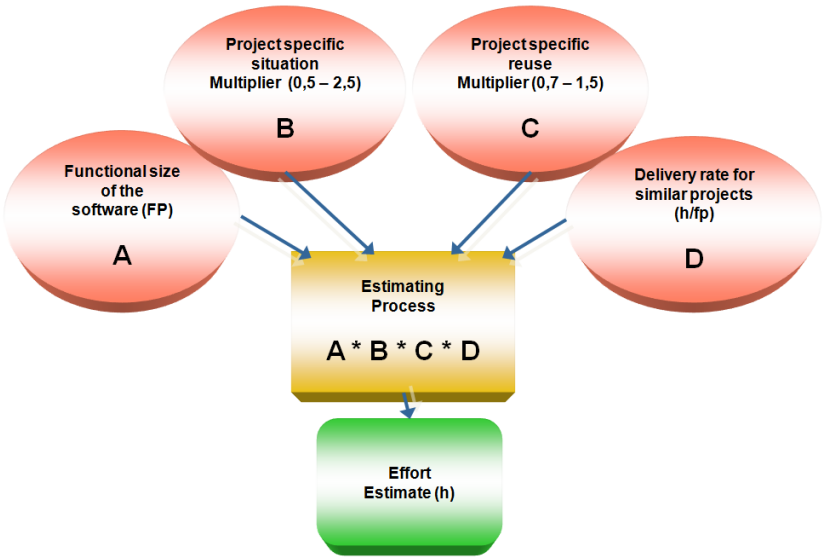

Figure 1 Effort estimation process of the northernSCOPE™ concept

As seen in Figure 1, the other three input parameters for effort estimation, in addition to functional size, are project specific situation and reuse multipliers, and an applicable delivery rate (h/FP), which can be determined by using a history database. We decided to use the new, web based Experience® Service tool in the speed evaluation, although it was not yet familiar to most of the target attendees. However, it is easy to learn to anyone, who knows the estimation methods and understands the process in overall, and there is a safe, separate training environment, so called Experience® Academy, where the evaluations could be easily organized without causing any harm for the production service.

2 Preparation of the speed evaluation

Hannu Lappalainen, chairman of the FiSMA Scope Manager Forum, and Pekka Forselius, the first author and a Senior Advisor of FiSMA, were selected to be the main organizers of the size and effort estimation speed evaluation. They chose five software projects that represented the real world quite well. The sizes of the target software varied from a very small (about 60 function points) to reasonably large (more than 4000 function points), including new development and maintenance projects, and also different system structures to be considered during the size and effort estimation. For each five projects the organizers provided two separate information sheets: one for size estimation and one for additional measures needed to estimate the project effort. The two other, more administrative documents were the list of user-ID's and passwords, and the data collection form for collecting the results. The information asked on the data collection form was:

- Name of the evaluator
- For each project there was one line with the project name, asking for:
    - Software size estimated, in FP
    - Minutes used to estimate the size
    - Effort estimated, in hrs, and
    - Minutes used to complete the effort estimate.

The main organizers created user accounts for 10 evaluators in the Experience® Service Academy, and initiated all five projects for each evaluator. This was made as a service for the evaluators, so that they could concentrate fully on size measurement and effort estimation, without any unnecessary, administrative burden. Initiation of one single project, including its classification and initial

method selection took approximately one minute from the main organizers, who both were already familiar with the tool. The tool allows deleting only the information entered by the evaluators, leaving the initial project information untouched, so that the same user accounts with initiated projects can be reused from event to another, with only minor additional effort from the main organizers.

Invitations were sent to members of Scope Manager Forum in May 23, as part of the normal meeting invitation letter. In the invitation e-mail we explained briefly the background, motivation and our goals for the speed evaluation. With this invitation we attracted eight attendees to sign in the first evaluation event. Later, for the next events, we extended the invitation to other Finnish Scope Managers and selected experts in the size measurement and estimation area.

3 Speed evaluation events

All the evaluations were arranged during the summer holiday season in Finland. It was clear in advance that it is impossible to get all interested to make evaluation at the same time. It is also good to remember that all the attendees were on voluntary basis, pushed only by their own, professional and personal interest. The attendees got scattered to four separate sessions between June 13 and August 8, 2013. All together 22 persons from almost as many organizations made the evaluation. Table 1 presents the complete list of evaluators in alphabetical order, their background organizations, and their principle viewpoint (A = Acquirer, S = Supplier, C = Consultant). All the events were identical, following the same manuscript, instructed by one of the main organizers.

Every evaluation session was started with a short introduction, reminding the attendees of the components of northernSCOPE™ effort estimation process. Then they were guided to find an available network connection and sign in the Experience® Service, so that they could find the list of initiated projects.

Name		Company	Role
Autio	Vesa	JaVePro	C
Forselius	Pekka	4SUM Partners	A
Hinkkala	Pekka	Anglo-Nordic	A
Lappalainen	Hannu	Albitech	C
Lehikoinen	Kirsi	Expericon	A
Manneri	Esa	Tieto	S
Mäki	Hannu	CGI	S
Männistö	Paula	Qentinel	C
Nevalainen	Risto	FiSMA	C
Nummenpalo	Kari	Fortum	A
Oksa	Seppo	Oracle	S
Paalasmaa	Mikko	BDO Consulting	C
Piipari	Jukka	SAS Institute	S
Ranta	Tapani	Edapco	C
Savioja	Erkki	FiSMA	C
Soikkeli	Raimo	Ilmarinen	A
Suonkoski	Tuomo	HK Ruokatalo	A
Tuomi-Sarja	Raija	KELA	A
Uutela	Tiina	Artii Consulting	C
Valsta	Anne	Haaga-Helia	C
Viljakainen	Aija	Tieto	S
Wuolab	Arvi	RAY	A

Table 1 List of evaluators with their organizations and principle viewpoints

In each event the evaluations were performed in same order, starting from size estimation of project 1, followed by the other size estimations in sequential order. Example of the typical contents of size estimation information given to the evaluators is presented in Figure 2. It is important to realize that gathering this kind of information from software documentation or requirements specification is not always equally easy. When a Product Requirements Document (PRD) is well organized, including the most important specifications, the necessary information is rather fast to collect and present at same level as in Figure 2, which is the level needed for size estimation. If there are no, or only very rough documents available, gathering the information needed for size estimation may be inaccurate and time consuming. However, the quality of PRD impacts not only to the estimation accuracy and speed, but even more to the development effort and speed. In this evaluation we wanted to concentrate only on the size and effort estimation speed, which was the reason to present the information as we did.

FiSMA Size and effort evaluation 2013-06-13		
Project ID and name	**PX05 Example project**	
Software ID and nan	**AX05 Sample application**	
	SIZE ESTIMATION	
Project classification:		
	Annual maintenance	
	Banking	
	Network	
	Java	
		number
FUNCTIONS:	Online messages from other systems	70
	Online messages to other systems	60
	Batch records from other systems	5
	Reports	15
	Calculation routines	8
	Other algorithmic routines	4
INSTRUCTIONS:	Estimate the software size based on functionality above, fill the data collection form by recording the size and estimation time in minutes.	

Figure 2 Exemplar software size estimation input information sheet

The second phase, completing the software size estimate to a project effort estimate, was performed using another input information sheet. Example of the additional information given to the evaluators is presented in Figure 3.

FiSMA Size and effort evaluation 2013-06-13	
Project ID and name	**PX05 Example project**
Software ID and name	**AX05 Sample application**
	EFFORT ESTIMATION
IMPACT OF REUSE	External method used, 20% of functions can be provided by using reusable components.
DELIVERY RATE	Use ISBSG dataset, analogy: customer specific new development, HTML
PRODUCTIVITY FACTORS:	
1.1	Customer and user representatives participation active
1.2	Development environment less equipped than normally
1.3	Availability of the key IT experts is good
1.5	Some pressure on schedule
2.3	Tools are better than in average
2.5	Requirements management is under the standard level
3.1	Functionality requirements are remarkably higher than usually
3.4	Efficiency requirements are very hard
3.5	Software must be extremely easy to maintain
4.1	Analysis skills of the development team are better than in average
INSTRUCTIONS:	Estimate the project effort based on functionality above, fill the data collection form by recording the effort and estimation time in minutes.

Figure 3 Exemplar project effort estimation input information sheet

Documentation of the results was made not only using the specific data collection form, but also with the standard estimation reports of the tool. All the evaluators reported their results immediately after each individual estimate provision, and in the end of the session they took the requested estimation reports out of the tool and sent them to the main organizers by email, for archiving the details, and for any further analysis.

4 Results

Although the main focus in our evaluation was on estimation speed, we also wanted to measure accuracy of the provided estimates, because speed without any accuracy is not what the industry needs. Fast estimation with good enough accuracy is. In this kind of evaluation it is not necessary, or even possible to compare estimates with actual software size and project effort, because the estimated pieces of software were not really developed. However, some kind of estimation accuracy can be measured by comparing the provided estimates with each other, or with a benchmark. In this case the organizers created the benchmarks for both size and effort estimates. This was made by estimating everything first separately, and then comparing results with each other, and then fixing the differences, if occurred. Either interpretation or typing may source such differences. In our case, only three of ten estimates had minor differences, although also the organizers wanted to estimate fast and measure their own estimation speed.

Tables 2 and 3 present the evaluation results. The “teacher solutions”, i.e. the benchmarks were given value 100,0 and all the others were compared against them, getting the percentage value in relation to the benchmark. The two last lines, Evaluator-ID’s Stu021 and Stu022, in both tables represent the organizers. All evaluators are in the same order in both tables, having the same Evaluator-ID. Colors in the tables indicate the accuracy of the estimate. Green is extremely good, yellow is good, and red is questionable, mapping to deviation ranges < 2%, 2-5%, and >5% for the size estimates. For the effort estimates we were more generous using deviation ranges <10%, 10- 20%, and >20%, because there are more components and opportunities for different interpretation and/or typing errors.

All speed numbers are on white background, and the measurement unit is 1 minute. The fastest size estimates were provided in less than 1 minute, and the longest time was 8 minutes. Average size estimation speed by application varied from 1,4 to 5,1 minutes, and the overall average estimation speed was 2,37 min/application, which appeared to be surprisingly close to the speed 1,87 min/application that Capers Jones reported in his blog [10].

	PX01		PX02		PX03		PX04		PX05	
	RelFP	Time/FSM	RelFP	Time/FSM	RelFP	Time/FSM	RelFP	Time/FSM	RelFP	Time/FSM
Stu001	100,0	4	100,0	2	56,3	8	81,1	3	100,0	2
Stu002	100,2	1,75	100,0	0,75	99,9	6	74,6	0,9	100,0	0,9
Stu003	100,2	3	100,0	2,2	97,9	3,6	93,1	2,6	100,0	1
Stu004	100,2	3	100,0	2	100,0	6	100,0	2	100,0	1,5
Stu005	100,2	3	100,0	2	100,0	6	100,0	2	100,0	1,5
Stu006	100,0	3	100,0	2	32,3	5	100,0	3	100,0	2
Stu007	100,0	3	100,0	3	100,0	5	100,0	1	100,0	3
Stu008	100,0	3	100,0	3	100,0	5	100,0	1	100,0	3
Stu009	100,2	1	100,0	1,5	100,0	7	100,0	1,25	100,0	1,1
Stu010	100,2	1	100,0	1,5	100,0	7	100,0	1,25	100,0	1,1
Stu011	100,0	2	100,0	1,5	100,0	5	90,9	1	100,0	1
Stu012	100,0	2	100,0	1,5	100,0	5	90,9	1	100,0	1
Stu013	100,2	1,5	100,0	1	100,0	7	100,0	1,5	100,0	1
Stu014	100,0	1	100,0	1	100,0	4	100,0	2	100,0	1
Stu015	100,2	2	100,0	2	100,0	5	100,0	3	100,0	1
Stu016	100,2	2	100,0	1	100,0	3	100,0	2	100,0	2
Stu017	100,2	1	100,0	1	100,0	2	100,0	1	100,0	1
Stu018	100,2	1	100,0	1	102,7	8	100,0	1	100,0	2
Stu019	100,0	2	100,0	2	102,5	5	110,8	2	100,0	1
Stu020	100,0	1,2	100,0	1,3	100,0	3,5	100,0	2	100,0	1,2
Stu021	100,0	1	100,0	1,3	100,0	3,5	100,0	1,5	100,0	1
Stu022	100,0	1	100,0	1	100,0	3	100,0	1,5	100,0	1
		2,0		1,6		5,1		1,7		1,4

Table 2 Results of the FiSMA size estimation speed evaluation in summer 2013

Software development projects are still mainly effort driven, i.e. development effort is the most important cost driver. Sometimes it is very difficult to understand how the supplier candidates have generated their fixed-price proposals. Knowing that acquiring organizations sometimes have reported software acquisition cases where the ratio between the lowest and the highest proposal is 1 to 25, and ratios like 1to 4 or 1 to 5 are rather common, the results of the FiSMA effort estimation speed evaluation presented in Table 3 should be quite encouraging, if somebody wants to try this type of estimating. In our estimation speed evaluation the ratio between the lowest and highest effort estimate was 1 to 3, and it was the only one of 110 estimates, where the ratio was worse than 1 to 2.

If two software project effort estimates differ less than 10 % from each other, they are considered to be close enough to be taken seriously. In our case, 66 of 110 were within +/- 10% range, and another 27 not far from that. However, as the Table 3 shows, there were 17 questionable cases, 14 of which were underestimated and three overestimated. Only two of these questionable cases can be explained directly by underestimation or overestimation in size. More common reasons for divergent effort estimates where related either to misinterpretation of information related to productivity factors or use of wrong history data. In this

paper we will not discuss reasons for deviation in individual estimates, but a more detailed analysis may follow in the future research. For example the results of first and last project in Table 3 (PX01 and PX05) make us thinking that there may be possible prove of learning by doing. For most of the evaluators the first project PX01 was their first ever project estimated using Experience® Service tool, and the PX05 was already the fifth. Looks like both estimation accuracy and speed would have improved, because the input information for these two projects was very similar.

	PX01		PX02		PX03		PX04		PX05	
	RelEffort	Total time	RelEffort	Total time	RelEffort	Total time	RelEffort	Total time	RelEffort	Total time
Stu001	64,8	16	86,5	6	177,2	14	97,2	10	95,5	7
Stu002	95,5	9,25	104,9	3,75	118,7	9	70,1	3,9	104,8	3,9
Stu003	100,0	9	100,0	6,2	130,2	8,6	94,2	7,6	100,0	4,7
Stu004	79,8	10	100,0	5	116,5	9	116,5	7	104,8	3,5
Stu005	79,8	10	100,0	5	116,5	9	116,5	7	104,8	3,5
Stu006	90,6	11	100,0	8	92,1	10	100,0	8	100,0	6
Stu007	100,0	10	100,0	5	116,9	9	100,0	6	104,8	7
Stu008	100,0	10	100,0	5	116,9	9	100,0	6	104,8	7
Stu009	91,1	9,5	100,0	5,5	111,1	10,5	89,7	8,75	104,8	3,6
Stu010	91,1	9,5	100,0	5,5	111,1	10,5	89,7	8,75	104,8	3,6
Stu011	88,9	18	99,3	5,5	111,1	11	86,1	9	100,0	5
Stu012	88,9	18	99,3	5,5	111,1	11	86,1	9	100,0	5
Stu013	84,2	6,5	104,9	4	111,1	10	89,7	6,5	100,0	4
Stu014	79,8	10	104,9	4	90,9	8	90,0	7	81,9	6
Stu015	81,8	12	100,0	5	148,5	8	70,5	9	81,9	5
Stu016	90,5	7	104,9	5	116,9	6	100,0	7	95,5	7
Stu017	94,8	5	104,9	3	105,3	4	100,0	4	100,0	4
Stu018	35,8	12	74,1	4	72,3	13	58,3	6	70,7	6
Stu019	88,3	10	104,9	4	86,6	9	109,2	6	104,8	4
Stu020	76,8	3,7	94,9	2,9	103,6	5,7	71,8	5,3	66,5	3,2
Stu021	100,0	5	100,0	3,9	100,0	6,5	100,0	5	100,0	3,5
Stu022	100,0	5	100,0	3	100,0	5	100,0	4	100,0	3
		9,8		4,8		8,9		6,9		4,8

Table 3 Results oft he FiSMA effort estimation speed evaluation in summer 2013

5 Conclusions and further research

Typical development effort for a 1000 FP government project varies from 7.700 to 19.800 hours based on historical project data [30, Table B-1]. Variance for a same size Java-Multiplatform project is from 5.700 to 8.100 hours [30, Table B-20]. Total cost of development project can be easily derived from the effort, if we know the hour rate. However, even in the lowest-cost countries, the development cost for 1000 FP software is hundreds of thousands Euro or US dollars. Based on our evaluation, the effort needed for a size-based effort estimate is only some minutes, and the estimation cost would be less than 50 Euro or US dollars even in the most expensive countries. In any case time, effort, and cost needed for

software size estimation with the best FSM methods and tools are so small compared to the total development figures that using them as excuses for ignoring size estimation make no sense. Based on our evaluation, although there were only 22 first evaluators in this research, we recommend functional software size estimation and size-based effort estimation to all software intensive projects. With applicable tool assistance they are very fast and cost-effective management components as well for the customer and supplier representatives, or independent acquisition consultants. However, to avoid unnecessary disappointments, all the estimates are worth of peer review. As the red cells in Tables 2 and 3 point out, interpretation and typing errors may occur, but they are easy to check and fix.

Requirements management is an important knowledge area within software industry, and very closely related to both software size and project effort estimation. Figure 4 illustrates the required levels of estimation accuracy during the development life-cycle. Similar models for estimation accuracy have been published also by e.g. Hill [40] and Boehm [70]. If the numbers of different types of functionality don't change, a "super fast size estimate" does not differ from the "final size measurement" remarkably, hardly ever more than +/- 10%. If the volatility of requirements is high, the accuracy of early (super fast or fast) estimates is naturally lower. However, studying this type of estimation accuracy was out of scope of this evaluation, and needs further research. The northernSCOPE™ concept includes several tools (e.g. pricing models, size measurement methods, history data) that can help keeping requirements volatility low, or at least managed.

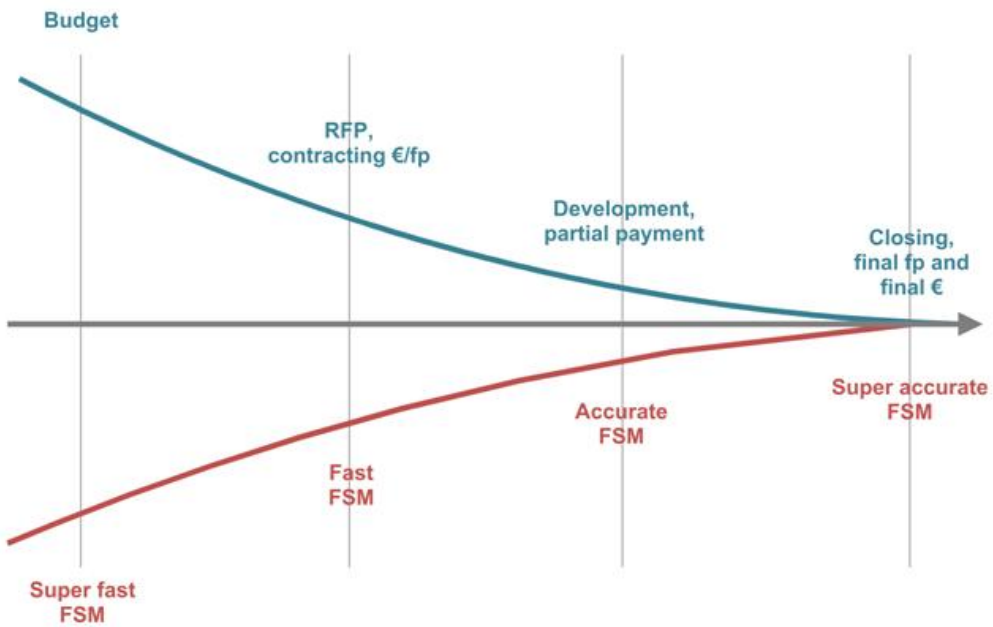

Figure 4 Required accuracy levels of FSM over software development life-cycle

Acknowledgements

Thank you to Capers Jones for inspiring us to try something new. Doing things differently is the only way to learn and to improve. Thank you also to FiSMA Scope Managers, those who attended and those who showed their interest, though they couldn't make it yet. Special thanks to Hannu Lappalainen, the Chairman of the FISMA Scope Manager Forum, who made great job in preparing and organizing evaluation events. There will be new testing opportunities and real world requests as well, as soon as the decision makers in their companies start to understand, how easy and cost-effective, but important and useful a size based effort estimate is for every software intensive activity. Then, but only then, functional size will become the leading metrics in software industry.

References

10 Jones, Capers: "Function Points as a Universal Software Metric", Blog 15.7.2013, <www.namcookanalytics.com>

20 ISO/IEC 29881:2010, Information technology - Software and systems engineering - FiSMA 1.1 functional size measurement method.

30 Hill, Peter, Practical Software Project Estimation, McGraw-Hill, 2011

40 Finnish Software Measurement Association, <http://www.fisma.fi/in-english/scope-management/>

50 Scope Management, <http://www.4sumpartners.com/ ?page_id=5>

60 Certified SCOPE Manager Job Role, European Certification and Qualification Association, http://www.ecqa.org/index.php?id=38

70 Boehm, Barry W., et al., Software Cost Estimation with COCOMO II, Prentice Hall, 2000