Early Project Estimation with Early Function Point Prognosis

MANFRED BUNDSCHUH

President of DASMA e.V.

Abstract

The presented Function Point Prognosis was performed three times: with 20, 39 and 78 application system counts, as the number of Function Point counts increased during the years 1998 - 2002. It was possible to show from error calculations that the regression formulae delivered Function Points with 13 % error by only counting the EI's and EO's. Since AXA fortunately has an almost uniform development environment with all application systems being counted (both being prerequisites for this result) and thus knowing all the interfaces, this Function Point Prognosis can be done before or at project start with marginal effort.

This Function Points are used for effort estimation with an estimation tool (Capers Jones' Checkpoint for Windows, CKWIN). Since this tool allowed only to estimate and view single projects or their comparison with other projects, templates or portfolios of projects within this tool database, there was a lack of overview. Thus the here presented enhancement was developed, showing the overview of the estimated projects for certain project categories like e.g. new development, enhancement, maintenance, project post mortem, and templates, each for different platforms like e.g. host, PC, C/S, etc. This is also a prerequisite for the development of own templates and portfolios.

The Challenge of Early Estimation

Imagine, your project leaders deliver you a profound and well documented tool- and measurement based estimation before the start of their projects. How high would you suggest the economic benefit for your organization?

This vision became reality some years ago in the IT department of the AXA Service AG, the non-insurance part of the AXA insurance in Cologne, Germany. Twice in the past years project leaders came to me and asked the similar question: "I have to estimate a project for portfolio decision. The project starts next year and I don't know much about it. Can you help me ?" Luckily I could, since I had developed our so called Function Point Prognosis and since we fortunately have an almost uniform development environment with all application systems being counted – both prerequisites to cut this gordic knot. As well as in alike other early estimations we could do the estimation within half a day, documented in our estimation tool and in a so called estimation log – a plain word document showing the suggestions we made and about the way how we proceeded during the estimation. This Function Point Prognosis is well accepted and often used since.

Project estimations are required as early as possible – not only from the contractors but also from every project leader as well as from management. Because of the importance of early estimation methods Meli and Santillo [Meli99] published a comparative overview of Function Point estimation methods that shows a valuable collection of worldwide efforts in this direction. Since Function Point counts are based on the requirements documentation, so-called Function Point prognoses or approximations proved to be helpful in practice to aid early estimations [Bund98b], [Bund99].

Our experience shows, that the necessary information for such approximations can be gained very early, at the beginning of a project (or, in stable environments, even before that) in discussions with the project leader. In a few cases we used this Function Point Prognosis about a year before project start. Of course, we added large percentages for each error, uncertainty, early estimation and risk [Eber04]. This additions can easily sum up to 100 % at this early stage of a project. This can be verified by the experience that failed projects often costed more than twice the proposed price. Newspapers often headline such examples.

Getting started with measurements

This chapter describes the experiences made during the introduction and progress of an estimation and measurement program inside our IT organization. About june 1995 the IT department of the AXA Service AG in Cologne, Germany started a metrics program with Function Point (FP) metrics, the introduction of a handbook for measurement and estimation, and the evaluation of tools. Beginning of the following year the Function Point Workbench and the estimation tool Checkpoint for Windows (CKWIN) were introduced and the base counts of all application systems was started.

6 years later, in 2001 the IT department completed the total counts of all 78 application systems (excluding Enterprise Resource Planning (ERP) applications, e.g. SAP, Peoplesoft) with a total of about 100,000 unadjusted Function Points (FPs). In 2002 productivity measures were developed, and a baseline for annual productivity measurement was completed. By 2003 the counted portfolio had increased to 98 application systems with about 150,000 unadjusted FPs. Also in 2003 a first version of a self developed metrics database was implemented and since then effort, defects and FP's were recorded every month for each development, enhancement and maintenance task [Eber04].

Since 1995 more than 100 staff members were trained in FP counting and project estimation. In each group of developers there is at least one FP and estimation “expert“. The competence center that fostered the measurement program consisted of two persons. Since the end of 2004 regular reports about quality, effort and size are available – FP counting is a routine job now.

Development Environment

The IT department of AXA Service AG includes about 500 IT professionals with approximately 50 project leaders. In addition, there is an outsourced computing center with about 250 staff members. The insurance branches deliver about 160 IT coordinators supporting the IT projects.

IT development is mostly host-based with COBOL programming. There exist about 2,700 databases (1,600 CICS, 1,100 IMS), and 2,500 DB2 tables (1,600 production, 900 disposition), and about 7,800,000 transactions per day (5,600,000 CICS and 2,200,000 IMS). PC projects use Optima++, a C++ shell, for programming and Internet programming is done with Java [Eber04].

Projects develop mostly interactive database administration systems for e.g. car, health or life insurance or claims management, within a very complex environment with centralized data bases for e.g. insurance partners, cash systems, document management etc.

IT Project Estimation Basics

The basic problems of estimation are shown in Fig. 1:

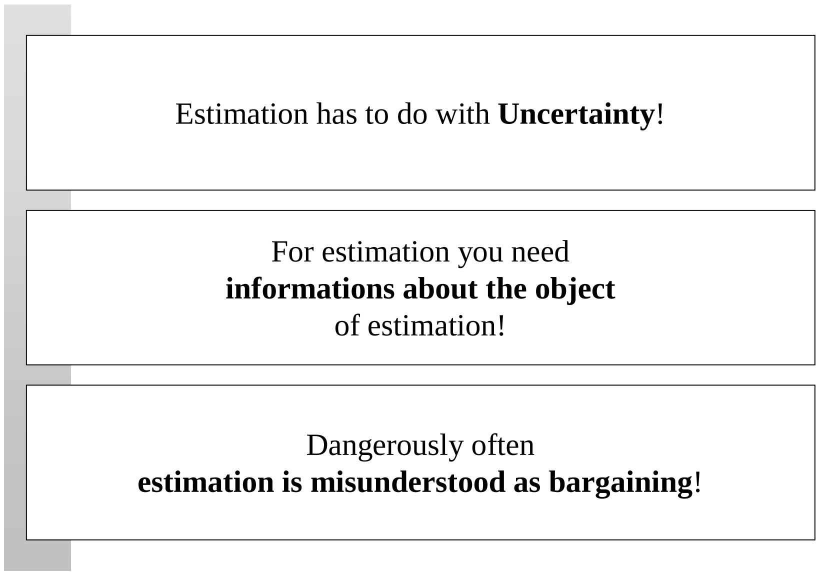


Fig. 1: The Basic Problems of Estimation [translated from Bund04a]

These problems are accompanied by the fact that empirical data are missing and that estimation is done too early and seldom. Three important measures should always be considered. First, the effort for measurement and estimation must be planned from the beginning on as a project management task. Second, a method without tool support has only little chances for survival. Third: size has to be measured before estimation. But estimate what?

- effort
- duration
- FTE's
- Costs
- ….

But on which basis?

Measurement is the heart of IT planning and the sound basis for professional estimations. It is always important to do estimations in different ways: expert estimation and using estimation conferences as well as other estimation methods [translated from Bund04a].

Expert Estimations

Expert estimations are usually performed in the following steps:

- definition of project structure and deliverables as well as estimators (at least two independent estimators)
- definition of the conditions and restrictions of the estimation

- independent estimation of the effort for each task by the different estimators
- comparison of their estimations and elaboration af a consensus
- addition of add on's for uncertainty and risks as well as for requirements creep. Before project start this may add up for doubling the before elaborated consensus (each of the three about 33% if a project is to run for a year, e.g.), a reason why projects often cost more than double as much as expected.
- Calculation of the total effort

Expert estimations often deliver lower estimations than experienced based estimations. Reasons for this underestimations are that estimators overestimate productivity and that high efforts are not appreciated by managers, customers and sales departments. Hence estimations often are political, which is a great obstacle for the development of a fair estimation culture [translated from Bund04a].

Political Estimations

A political estimation is any estimate provided with the prime objective to prove a guess or judgment, while obscuring reality. Especially in large organizations there may be many reasons for this, e.g., lust for power, human vanity and others. The following list presents some associated problems [translated from Bund04a].

- Estimation is often mistaken for bargaining. Missing historical data often result in the dictation of unrealistic deadlines.
- The size of the project is often obviously/consciously wrongly estimated (trimmed estimations).
- When cutbacks of the IT project occur (e.g. budgets or deadlines) the estimation is often erroneously trimmed according to the cutbacks instead of by reducing the other primary goals (quality, functionality, costs, time).
- Voluntary unpaid overtime will be planned but not considered in the estimation.
- Goal conflicts often arise from another irrational factor: human vanity. The desire for success and acknowledgement often leads to "turf wars" in the IT project environment to the end that project leaders must consider power politics in the environment of their IT project.
- There is a widespread prejudice that application systems, software and hardware cost more in the host environment than in client/server (C/S) applications for software or hardware. This leads to more bargaining in the C/S environment instead of estimations and thus necessitates more effort to convince management that C/S environment costs much the same as the host environment [Eber04].

A main cause for this underestimation is the fact that often a political estimation is done instead of a realistic one. In reality, the effort of an IT project is often underestimated in order to gain approval for the initiation and performance of this IT project. What a crazy world: here the decision makers are not guided by the estimations, but the effort estimators are guided by the criteria for decision making. The practical result is that the effort is not estimated but is determined by bargaining.

Political estimations and project decisions not based on facts definitely ruin trust in a company. Management and staff should thus avoid these obstacles in order to foster a good estimation culture [translated from Bund04a].

Estimation Conferences

Estimations can be done by different individuals and the average of their estimations can be used. But there exists an approved alternative: an estimation conference. Several persons from the project team (e.g., leaders of parts of the project) discuss together, how to estimate the estimation object in view of the total IT project. This leads to an estimation that is accepted by all involved persons, which is more objective than the above-mentioned average and hence can be better defended against other opinions. The results may not differ very much, as we found in some cases [Bund00a].

Another benefit of the estimation conference is that the involved estimators gain awareness of the uncertainties and possible risks of the IT project. Furthermore they all get the same information. An estimation conference is a team-building experience. An estimation conference also promotes the estimation culture in an organization, since it helps to solve acceptance problems by finding a consensus through discussions in a team. These benefits can often be gained in only one estimation conference lasting not more than two hours [translated from Bund04a]!

Controlling and Estimation

Estimation is the major tool of project controlling. When time elapses everything tends to get off course. A controlled plan is necessary to prevent this deviation and to stay on due course. Estimation is also the basis of the economic benefit of project, measured by cost benefit analysis, Function Point Method and diverse estimation methods.

The classic business functions of controlling are:

- planning and definition of key figures
- collection of informations – in time, cumulated and problem adequate
- control
- controlling

In order to reach this goals controlling has to perform the following tasks:

- definition of planned effort for tasks of the project based on sound estimation and planning
- continuous measurement of actual effort (collection of information)
- comparison of actual and planned effort (control)
- search for reasons of deviations and proposal of further project progress (controlling)

If this project management tasks are not performed professionally the project manger behaves gross negligent.

The same holds for the project management task documentation. Especially the measurement of project size as basis for estimation is a review of the functional requirements. And it is a prerequisite for the management of requirements creep as well. The documentation is important for quantification in order to understand the system to be developed. Basic experiences can be gained from good documentation and can be filled from there into project management manuals.

If it is not possible to measure functional size because of missing (existing, actual, understandable) documentation or missing know how in the IT project then this is a sign that the requirements analysis is not yet finished. Or it is an indicator showing that the IT project has lost it’s overview (or is out of control) already at the start of the project. In this case the project leader should better consider not to start the project [translated from Bund04a].

The most important Themes of Estimation

The determining factors of estimations are shown in Fig. 2. It demonstrates obviously that the constraints of estimation are the same variables which we are free to define and they all contribute to the result of the estimation, besides the size.

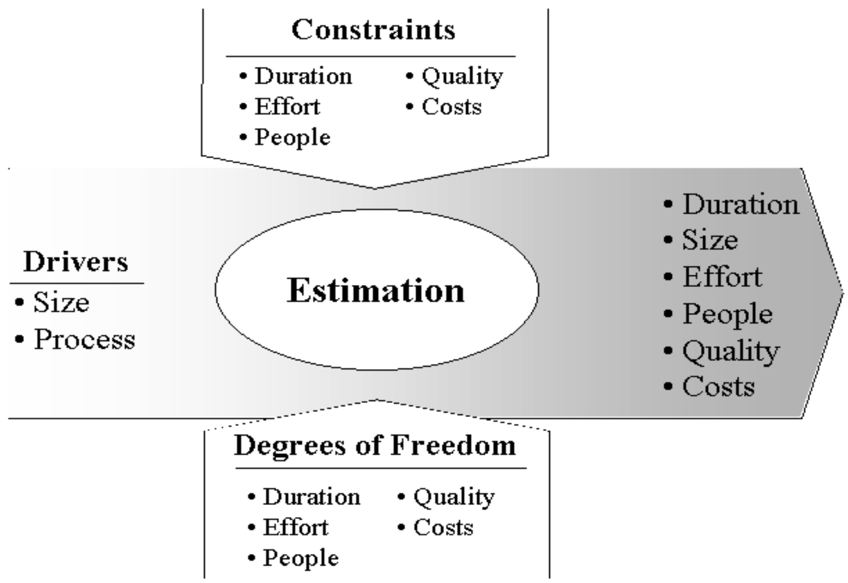


Fig. 2: Determining Factors of Estimations [translated from Bund04a]

Fig. 3 summarizes the most important themes of estimation. Some of them which are relevant for this paper have been mentioned above.

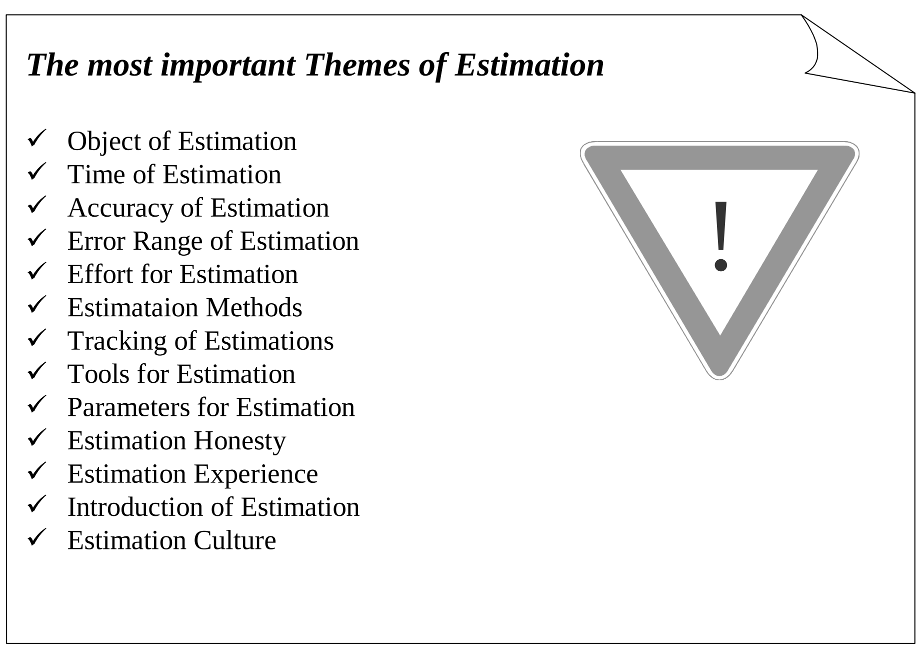


Fig. 3: The most important Themes of Estimation [translated from Bund04a]

Function Point Prognosis

FP counts are obligatory in the AXA Service AG, at least at the end of the requirements analysis and at the project post mortem. Function Point Prognosis, – as described here – instead of only FP count is mandatory during the feasibility study and at the start of a project. The counts and their details are documented in a centrally administered Tool, the Function Point Workbench. Function Point prognoses are only documented in a so called counting log, a plain word document showing the suggestions we made about the number of EIs and EOs and about the way how we proceeded during the estimation.

Throughout this chapter Function Points mean IFPUG 4.0 unadjusted FPs. AXA didn't adopt the IFPUG Rel. 4.1 and 4.2 since the old counts would have been not comparable then. A decision for this progress makes more sense in the following years, if there are major changes in the IFPUG standard and because FP's are counted now in routine. Hence actualizations of application counts and new developments can be counted according a newer IFPUG release and the FP portfolio thus be renewed.

Function Point Database

To get access to historical data, we first used a project register database (Excel), which showed detailed information (extracted from the Function Point Workbench) for each FP count. Later this dtabase was transformed to a MS Access database and actually the metrics database is designed new in order to develop it on an Oracle SQL platform. It contains the quantity as well as the Function Points of EIs (External Inputs), EOs (External Outputs), EQs (External Inquiries), ILFs (Internal Logical Files) and EIFs (External Interface Files) for each complexity (low, average high) of these components and some additional information (platform, VAF (Value Adjustment Factor), adjusted Function Points). When a project was counted repeatedly only the most recent count is shown and older counts are kept in a history table. There are sums of the quantity of EIs and EOs, of ILFs and EIFs, which were needed for our research.

IO means throughout this chapter the sum of the number (quantities) of EI and EO of a Functgion Point count. The data of the counts were always exported from the Function Point Workbench in Excel format and then imported into the follow up application (first Excel, later MS Access).

We started in 1997 with 20 application counts and a total of 26,575 FP's [Bund98a], which increased to 39 counts and a total of 77,865 FP's a year later [Bund99]. A balanced scorecard was introduced for the department leaders, combining their success in counting all of their application systems with 20% of their financial bonus. This was essential management support for the success of our metrics program. In 2001 we had counted 78 application systems with more than 100,000 FP's and 2003 all 98 application systems (except technical applications and except SAP and other COTS application systems (COTS = Commercial off-the-shelf)) with a total of more than 150,000 FP's.

We collect the informations about interfaces and parts of the project together with the Competence Center staff and document it together with a diagram of the application boundary which is also used for our architecture atlas and together with a so called counting log which is a simple Word document giving important informations about special decisions concerning the the FP counts [Bund02a].

Indicators

In addition to the investigation of the formulae for prognosis we also investigated the FP proportions and average function complexity of our FP counts and compared it, e.g., with the International Software Benchmarking Standards Group (ISBSG) data. The ISBSG (http://www.isbsg.org) publishes every year a collection of metrics. The actual Release 9 is based on more than 3,000 projects worldwide. Since we did this research several times over the past years, we now have at least three historical annual metrics of our own data for comparison, as can be seen in tables 1 to 3, which will be discussed in the following.

Value Adjustment Factor

One of the first results of our data collection was the perception that the value adjustment factor (VAF) of our counts is typically in the range of 0.73–1.22, with an average of 0.95 in the 2001 data (0.93 in 1998), and an average of 0.94 for host and 0.96 for PC environment. The average for migrations is 0.73. We have used this metrics as a rule of thumb for quality assurance of our Function Point counts since 1998 [Eber04].

Fig. 4 shows as an example the comparison (Vergleich) of the 14 General System Characteristics (Systemmerkmale) of the host applications (Bewertung = Value). The grey bars show the range of the values for the differnet counts and the horizontal line inside the bar shows the median.

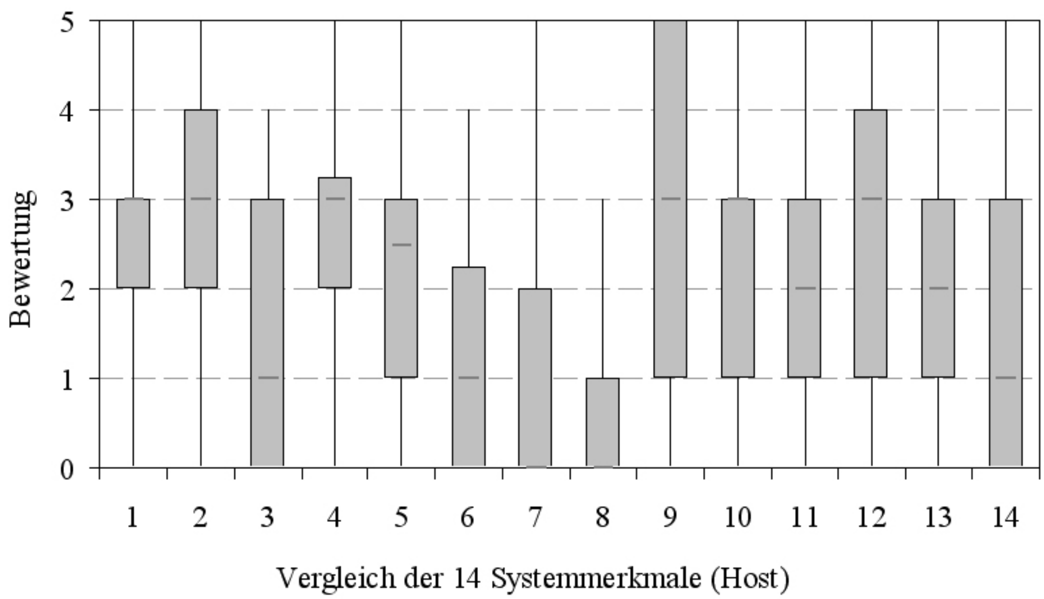


Fig. 4: Comparison of the 14 General System Characteristics of Host Application Systems [Bund04a]

Function Component Proportions

Table 1 shows the historic development of the function component proportions in AXA Service AG. Of course, the first two years are not very representative. The figures from 1998 and 2001 are similar, which we take as an indicator for a stable environment. The division into Host and PC development shows differences that should be carefully observed in future. The domination of the EIs and EOs (61% together) seems to be the reason for the strong correlation between IOs and the unadjusted FPs – the main result of this research.

It can be seen that EOs dominate in AXA Service AG (39%) compared with the results obtained by Morris and Desharnais [Morr96] (22%–24%), and the quick estimation mode of Checkpoint for Windows (CKWIN), the estimation tool from SPR (Software Productivity Research) in Burlington, MA, (20%), whereas ILFs are of minor importance (16% versus 24% and 43%, respectively). Because of this peculiarity one conclusion was not to use CKWIN (EIF + ILF = 46% versus 23% in AXA Service AG) in Quick Estimate mode for the estimation of FPs. The reason for the major importance of EOs may be that AXA Service AG has many centralized management information systems. On the other hand side the reason for the minor importance of the ILF's may be that AXA Service AG has some large centralized data bases which are widely referenced by almost all applications.

In 2000 we accomplished an error calculation with the 1998 data by using the percentage of each component to calculate 100% from it and compared the result with the actual Function Point count. Errors range from 37% (EOs) to 48% (EQs) [Bund00b]. Hence we do use the percentages of the components only as a rule of thumb for the quality assurance of our Function Point counts [Eber04]. Some figures from literature are added for comparison (ISBSG Rel. 6 and an article from M. Morris and J.M. Desharnais in the metricviews (IFPUG) in summer 1996).

2001		**Percent of Function Points**				
Platform	**Number of Application Systems**	**EI**	**EO**	**EQ**	**ILF**	**EIF**
Total	78	22	39	8	16	14
Host	69	21	40	8	16	15
PC	9	28	31	12	19	10
ISBSG Rel. 6	238 New development projects	33.5	23.5	16	22	5
Metricviews [Morr96]		26–39	22–24	12–14	24	4–12
Checkpoint for Windows		20	24	10	43	3
History:						
1998 Total	39	25	39	14	17	6
1996/97 Total	20	27	39	11	18	5
1997 Total	12	18	43	12	18	9
1996 Total	8	34	35	11	18	2

Table 1: Function Component Proportions

Average Function Complexity

We used the Excel problem solver to calculate from the project register database the average function complexity of the five components, i.e., how many FPs a "typical" EI, EO, EQ, ILF or EIF has in our environment. It is widely agreed that this measure is stable and can be used as a rule of thumb for quick estimation of counts, since the components need not be classified as low, average or high. SPR Function Points, e.g., use the average IFPUG classification for Function Point estimation. We advise our project leaders to use our AXA typical average

Function Points instead of the IFPUG's since a comparison with the average IFPUG Function Points (=SPR Function Points) show a lower error.

Table 2 shows that the average Function Point size increased over time, which may be caused by growing complexity in the application development environment. In 1998 we tested the applicability of the typical FPs for estimation purposes by multiplying it with the quantities of the EIs, EOs, EQs, ILFs and EIs, respectively, and compared the results with the unadjusted Function Points of the counts [Eber04]. The error was less than 26% [Bund02b]. For reason of comparison the ISBSG Rel. 5 figures are also shown.

2001		**Average Function Points**				
Platform	**Number of application systems**	**EI**	**EO**	**EQ**	**ILF**	**EIF**
Total	78	4.7	5.9	4.4	8.6	6.5
Host	69	4.7	5.9	4.6	8.7	6.5
PC	9	4.3	5.7	3.8	7.6	6.5
IFPUG		4	5	4	10	7
ISBSG	Release 5	4.3	5.4	3.8	7.4	5.5
	Release 5 Europe	4.2	4.9	3.8	7.2	5.3
1998	**Number of application systems**	**EI**	**EO**	**EQ**	**ILF**	**EIF**
Total	39	4.6	5.7	4.3	8.2	6.1
Host	28	4.8	5.7	4.5	8.5	6.2
PC	11	4.0	5.7	3.9	7.3	5.4
1997	**Number of application systems**	**EI**	**EO**	**EQ**	**ILF**	**EIF**
Total	20	4.6	5.5	4.3	8.1	5.7

Table 2: Average Function Complexity

Function Point Ratios.

One would expect three inputs (add, change, delete) at least, one output and one EQ for the maintenance of a file. The following results in table 3 show the averages in AXA Service AG compared to ISBSG Rel. 5.

Ratios of Components.

There are remarkable differences between the before-mentioned expectations and also some differences between the ratios in our application systems and the ISBSG findings [ISBS00, ISBS02], as can be seen from table 3. Hence we can advise other organizations to better develop their own metrics for better estimations. This know how can easily be gained by an analysis of the Function Point counts, as shown here.

Application systems	AXA Service AG			ISBSG Rel. 5	
	2001	1998	1997	Europe	Total
Quantity	78	39	20	32	238
EI per ILF	2.6	2.7	2.7	3.8	2.9
EO per ILF	3.6	3.3	3.7	2.6	1.5
EQ per ILF	0.9	1.4	1.2	1.9	1.1
EIF per ILF	0.6	0.5	0.4	–	–
Ratios per input and ratios per output					
78 Application Systems	2001		78 Application systems	2001	
EO per EI	1.3		EI per EO	0.7	
EQ per EI	0.3		EQ per EO	0.3	
ILF per EI	0.4		ILF per EO	0.3	
EIF per EI	0.2		EIF per EO	0.2	

Table 3: Ratios of Components

Ratios of Function Points per Component.

The ratios of Function Points per ILF, input and output were also calculated and are shown in table 4.

78 AS	**2001**		**78 AS**	**2001**		**78 AS**	**2001**
EI FPs per ILF	12.2		**EO FPs per EI**	8.0		**EI FPs per EO**	3.4
EO FPs per ILF	21.0		**EQ FPs per EI**	1.5		**EQ FPs per EO**	1.1
EQ FPs per ILF	4.0		**ILF FPs per EI**	3.3		**ILF FPs per EO**	2.4
EIF FPs per ILF	4.2		**EIF FPs per EI**	1.6		**EIF FPs per EO**	1.2

Table 4: Ratios of Functions Points per Component (AS = application systems)

Regression Analysis

Regression analysis on our project register database was used to find correlations between the number of components and the unadjusted Functions Points of the counts. We found later on a verbal information that comparable regression analyses had been performed by Gaffney, J. E. Jr.: [Gaff94].

The result of our research was that **the sum of the quantities of EIs and EOs (IOs in our terminology)** is correlated with about $R^2 \geq 0.95$ ($R \geq 0.97$) to the total amount of FPs of a count and can thus be used as a Rule of thumb for the prognosis of FPs when the FPs of EQs, ILFs and EIFs are not known. Correlations were analysed with the lower and upper half of the data (< 1,200; >1,800 FP's ; resp.) and all counts (exept migrations, because they are built for obsolesence and are batch - systems). The correlation coefficients of different regression analyses are shown in table 5 [Bund98b].

An interesting result was that the correlation was not as reliable (R^2 mostly less than 0.9) for other components as for data subsets of small, medium and large counts, and it was not better with polynomial regression. Of course, the use of FPs instead of the IOs for the prognosis gives a stronger correlation, but the higher effort for classification (low, average, high) of the components instead of only counting the number of EIs and EOs is not adequate for the higher precision. One should always keep in mind that estimation has to do with uncertainty per se.

	R^2									R								
	Quantitiy			FP´s unadjusted			SPR FP´s			Quantitiy			FP´s unadjusted			SPR FP´s		
	<1200	>1800	all	<1200	>1800	all	<1200	>1800	all	<1200	>1800	all	<1200	>1800	all	<1200	>1800	all
EI´s	0,28	0,11	0,77	0,30	0,51	0,86				0,53	0,33	0,88	0,55	0,71	0,93			
EO´s			0,84	0,90	0,44	0,85						0,92	0,95	0,66	0,92			
EQ´s			0,33	0,01	0,31	0,41						0,57	0,10	0,56	0,64			
EI´s+EO´s	0,95	0,71	0,95	0,95	0,90	0,97	0,91	0,70	0,93	0,97	0,84	0,97	0,97	0,95	0,98	0,95	0,84	0,96
EI´s+EO´s+EQ´s			0,94			0,96						0,97			0,98			
ILF´s	0,86	0,19	0,75	0,14	0,75	0,83				0,93	0,44	0,87	0,37	0,87	0,91			
EIF´s			0,05	0,34	0,84	0,05						0,22	0,58	0,92	0,22			
ILF´s+EIF´s			0,47	0,92	0,47	0,62						0,69	0,96	0,69	0,79			

R^2: Determination coefficient

$R^2 \geq 0,90$ bold surrounded

R: probability for correlation

Table 5: Correlation Coefficients

Prognosis Formulae

The 1998 data were analyzed independently by Noel [Noel98] in a joint research with the Software Engineering Management Research Laboratory, Department Informatique, Universite du Quebec a Montreal (UQAM), Canada, who obtained the same results. He applied the same method to seven projects with COSMIC Full Function Points (FFPs), in order to find a similar correlation for FFPs, but the sample seemed to be too small for reliable results. He reported in his thesis an error margin of 20%. Table 6 gives a historical overview of the prognosis formulae of AXA Service AG.

2001	**Number of counts**	**R^2**	**Error in%**	**Formula for Prognosis**
Total	78	0.9483	13	FP = 7.8 * IO + 43
Host	69	0.9498	12	FP = 7.9 * IO + 40
PC	9	0.9503	21	FP = 6.4 * IO + 172
1998	39	0.9589	20	FP = 7.6 * IO + 50
Host	28	0.9580		FP = 7.9 * IO + 11
PC	11	0.9760		FP = 6.5 * IO + 134
1997	20	0.9525	13 (Median 11)	FP = 7.3 * IO + 56

Table 6: Prognosis Formulae

Fig. 5 visualizes the regression analysis result for all counts.

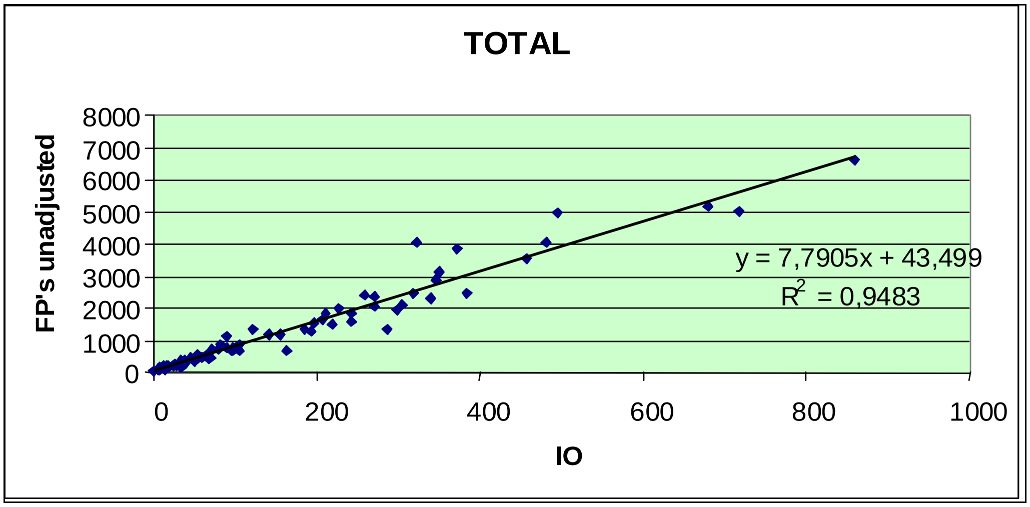


Fig. 5: Regression Analysis Example

The error margins were analysed by using only the quantity of EI's an EO's (summarized, =IO's in our diction) for each count, computing it with the regression formula and determing the difference between this computation and the counted FP's.

Our findings compared with those from other organizations show that such data collections can be used to find heuristic solutions for FP - Prognosis, either using Function Point proportions („typical FP's"), function ratios or regression formulae. But there is evidence that different environments demand for according solutions. Hence each organization should develop its own know-body of heuristic solutions and should distingnish between different development platforms etc. when doing so.

Tool based Estimation

A science is as mature as its tools for measurement. (Louis Pasteur)

No method will be commonly used if it is not supported by tools. Hence the usage of measurement and estimation tools is in any case a valuable investment, albeit the benefit might not be quantifiable. Nevertheless a warning ahead:

"A fool with a tool remains a fool."

This bonmot reminds us that information about and training in estimation is a prerequisite before a tool can be used with benefit. The first discovery in trainings is mostly that the tool doesn't perform the intellectual work. This is much the same with estimation as with planning tools since project leaders have to evaluate estimation parameters as well as as they must plan the work breakdown structure of their projects before they can give it as input in a tool.

Tools used without appropriate knowledge deliver non-usable results. Especially estimation tools afford careful customization and critical reflection of the estimation parameters since estimation has to do with uncertainty per se. Estimation results with decimal points produce illusions of a not existing accuracy and let the user feel in a treacherous safety.

The results of the tool can only be as good as the informations available and used for the estimation with the tool. For this reason estimations with tools should also be performed in different ways for plausibilization. When we discover differences in this estimations the benefit will usually be important learning effects about the object of estimation, prerequisites of the estimation, the environment or process of estimation as well as about the evaluation of the estimation parameters. This may give valuable hints for project risk assessment, too.

The Benefits of Estimation Tools

The beneficial aspects of estimation tools are manifold. They can help the project leader to do his work efficient and they have strategic benefits for the organization as well:

- earlier (toolbased and hense more actual) documentation
- transparency and evidence
- standardization
- mastering complexity

This fosters the common use of estimation methods. Common use of a method is a major key success factor for its acceptance.

Estimation tools provide operative benefits for the project leader by supporting his plans regarding the

- project size
- development tasks
- project complexity (work b reakdown structure)
- resource management
- time schedules and milestones
- simulation of alternatives

Hence some estimation tools offer interfaces to project management tools (as e.g. KnowledgePLAN to MS Project). Thus estimation tools support quicker and more efficient planning. And they assist in gaining overview when using many parameters for estimations.

Figure 6 shows the priciple of tool based estimations.

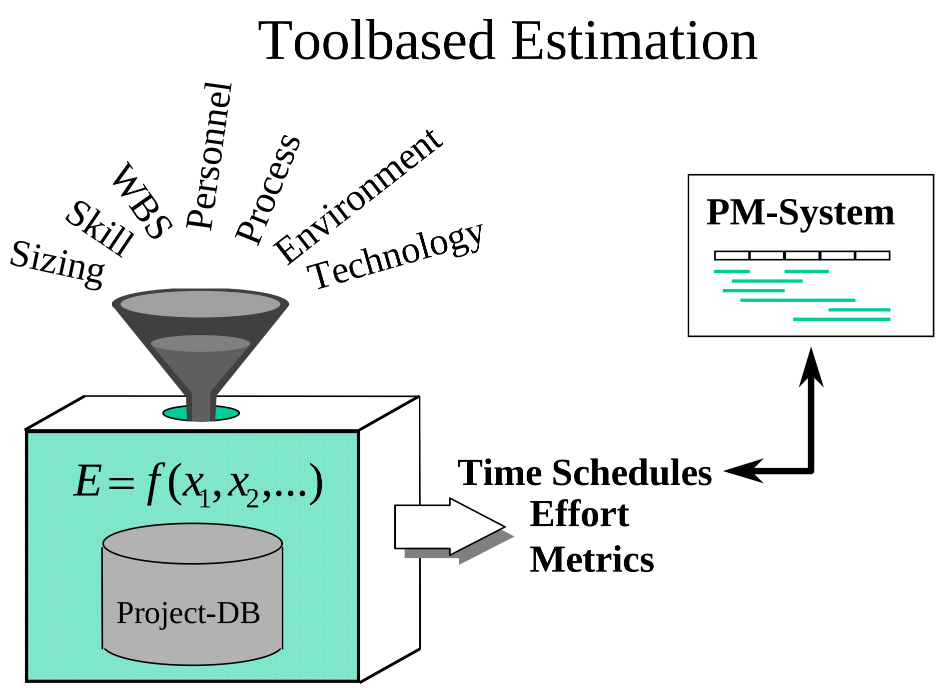


Fig. 6: The Principle of Toolbased Estimations

The major benefits (time saving, quality and efficiency improvement) are thus harvested project by project over time and summon up accordingly in an organization. Besides this there emerge beneficial side effects, as e.g. improved acceptance or improved transparency, standardization and a culture for estimation.

The practical benefits of estimation tools are:

- reduced time for planning (time to market)
- improved planning results (no extra costs because of delays)
- efficient project management, improved user acceptance and no project failures
- improved software development, less maintenance effort
- transparency in management of many informations

To harvest the full benefit one has to follow strictly the method which underlies the estimation tool and one has to collect enough historical data of project sizes.

Estimation Tools

A method without tool support has only a little chance for survival. Hence the experiences of the author are based on the personal (good) experiences with the dedicated tools used in the own organization. Nevertheless this experiences can be transferred to any other estimation tool.

Estimation tools are used in following six steps [Jone02a]:

1. measurement of project size
2. selection of project tasks (work breakdown strucgture, WBS)
3. plannning of resources
4. estimation of effort
5. estimation of costs
6. time planning

They commonly use two strategies:

1. Micro estimation (bottom up)
2. Macro estimation (top down)

Micro estimation aggregates the partial estimations for the project tasks. Hence this kind of estimation is more complex as macro estimation. The benefit is that errors impact only single tasks. Since the estimations can comprise any detailed project task this kind of estimation is the more precise one of the strategies.

Macro estimation is used for the estimation of large projects. The estimated effort or duration is distributed to the project phases. Hence macro estimation is easier done but errors impact all phases of the project. The benficial use of macro estimation is thus best for quick and early estimations [ISBS99].

Estimation tools use 3 fundamental relationships:

1. Assignment scope: This is the amount of work (size) a person is responsible for.
2. Production rate: This is the amount of work (size) a person can accomplish in a given time.
3. Duration: This is the effort for a task divided by the number of persons assigned to it.

This relationships are normally used in estimation tools in the following sequence in estimation equations:

1. Size divided by assignment scope delivers the number of the required persons.
2. Size divided by production rate delivers the effort.
3. Effort divided by the number of persons delivers the duration.

This equations are simple constructions but complicated in handling since assignment scope and production rate are often not known precise enough for the delivery of usable results. The value of an estimation tool thus depends on it's flexibility to consider different estimation parameters. [Jone02a]

A prerequisite for tool acceptance is its calibration to company standards (customizing). Most important is the establishment of the time accounting, e.g. how many working hours a person day, person month and person year comprises (considering holidays and average absence days etc.). A common rule is that a person is available for effective project work on only 75% of its presence due to holidays, sickness, general meetings, trainings etc.

Best for comparisons is a time accounting based on person hours, since this avoids the incomparability of person months or person years and thus allows unlimited benchmarking.

On international benchmarks within an organization it was found e.g., that the Japanese branch reported figures with 120 hours per person month without counting the work overtime. Such differences are torpedoing every learning effect of comparisons with estimations from other organizations.

Other standards to be customized may include deliverables and tasks as well as documentation and quality requirements. Some tools support this standards delivering the opportunity to develop company own estimation portfolios and templates (see according chapter following pages).

Simulations for Estimation

Simulations with estimation tools provide very useful data for the analysis of influences of single estimation parameters (estimation metrics, quality metrics, duration, effort) and thus help to identify the critical success factors for different goals. The estimation Tool Checkpoint for Windows (CKWIN)/KnowledgePLAN from SPR e.g., contains a sensitivity analysis delivering a hit list of criteria that can be used for improvement measures. Questions thus as how project duration can be reduced by reduction of requirements creep and project complexity can be investigated.

Hence the concrete goal for the simulations is to find out the most effective parameters affecting project duration using the estimation of a typical IT project. The following steps were chosen in the simulation project.

We once performed simulations of a finished project in order to find out the parameters which fit best for shorter project duration. This task was performed for gaining knowledge for further projects. We started the project simulation with the sensitivity analysis in order to see the parameters that have the greatest influence on project duration. With the sensitivity analysis Checkpoint showed the 16 strongest parameters for the four project goals: duration, effort, productivity and quality. Hence we got a hit list of parameters mostly influencing the matching goal, independent of the actual parameter value. For our investigations only the goal project duration was of relevance.

Next this parameters were improved successively by about one unit at a time, documented in tables and reset afterwards. When modifying the parameters, according to the hit list of the sensitivity analysis we found that 3 of the 16 parameters could not be used for shorter durations since they had the best value from the start on.

The hit list of parameters is sorted in decreasing order. The first parameter has the most effect for shorter duration. The last parameter astonishingly delivered a three-day longer duration for some reason. The next step was the summation of the parameters, followed by step-by-step improvement to the best value. All simulations were documented in analog tables.

The simulations proceeded with an alternative for the improvement of an IT project with the report on weaknesses. In this case the weaknesses are the parameters with values worse then the average. Again, these parameters were modified step-by-step and in sum.

The simulations clearly demonstrated that there are a large number of ways to finish projects earlier. But not all parameters can be influenced by senior managers, project mangers or the project team, as e.g., the involvement of the users. Based on the modification of only one parameter, the project duration could have been improved by 33% with equal quality and more staff (138 people instead of 86). Obviously we would not recommend to do this in full extreme, but the result can be used to proceed a bit in this direction.

The lesson learned is that tools should be used more frequently for simulations. This rule is also valid for project planning tools. We found in daily project life that this rule is almost neglected by project leaders, leaving them without an essential aid for project survival[Eber04].

Precision of Tool Based Estimations

Interesting is a comparison of the precision of such tools which was performed by C.F: Kemerer at the University of Calgary [Keme93]. In this comparison ESTIMACS (FP-based) showed an average error of 85%, COCOMO (LOC-based) 601% and SLIM (LOC-based) 701% for estimations in early software life cycle. An insurance company in Philadelphia compared esgtimations with CKWIN(KnowledePLAN and S.M.A.R.T. Predictor of DDB Software Inc. and found a variance of less than 7% for the estimated costs and effort. Both results were also very close to the actual values at project post mortem.

The organizations Bachmann and Texas Instruments developed tools for automatic counts of FPs. Such tools are regarded critically, since it cannot be guaranteed that the counts are delivered from “user view”. Hence IFPUG found no software for automatic FP counts according to the CPM (Counting Practises Manual). FP counting tools using code analysis strictly follow the IT-technical view and not the user view. When code is already available the counting of LOC should be a more reasonable choice. But this is a technical measure and not a functional one.

Estimation Portfolios and Templates

One of the most important tasks for preservation of the organizational estimation know how is the customizing of the estimation tool by the development of own estimated project portfolios and templates in addition to the delivered basis of projects inherent in the tool. The tool vendors normally are proud to publish the number of the inherent projects used for the deliverance of the results, but never deliver the details, as e.g. the ISBSG does, selling the database together with analysis tools.

Hence it is of vivid importance for an organization to group “equal” projects into a portfolio. This is done based on the idea, that projects of similar kind will also show similar behaviour and thus can be estimated in the same manner (with similar estimation parameters). Checkpoint/KnowledgePLAN e.g. delivers the functionality to group projects into a portfolio in order to compare other projects with this portfolio (the average of the projects in that portfolio). A next step is to extract an estimation template from such a portfolio in order to use this template for the estimation of further projects. Such a template is like an own database of the estimation tool.

Now, what are similar projects or asked in another manner: which projects should be put together in a portfolio? Looking into the literature you’ll find a lot of project characteristics which can be candidates for categorizations, e.g.:

- kind of development
    - new program development
    - enhancement
    - migrations

  - project post mortems

- platform
  - batch systems
  - PC systems
  - Client/Server systems
  - Data warehouse applications
  - OO systems
  - web applications

We used the kind of development and the platform as categorizations and grouped the projects according to size (small, medium, large). A prerequisite was to find out all the projects of a common category. Our tool, Checkpoint for Windows, was not much help in finding out which projects fit together in one portfolio, since it just showed each project. Hence we developed a tool which extracted the estimated projects from Checkpoint and imported it into an MS Access database (CKWIN Reader/Writer). Here we could see the projects according to the categories and thus evaluate which projects should be taken into a portfolio. This was an important enhancement to our estimation tool, since from then on we had our own templates for estimation.

Figures 8 to 12 give an overview from our tool CKWIN Reader/Writer of some of the categories, showing estimation parameters of e.g., kind of development and platform with productivity, quality, staff. All figures show in the first 4 columns the project scope (Projekt-Art: new program development, enhancement, maintenance), the platform (host, PC, C/S, DW), the Project short name and the start date of the project.

Just for better understanding the Checkpoint for Windows estimation results "View Totals" screen is shown in Fig. 7

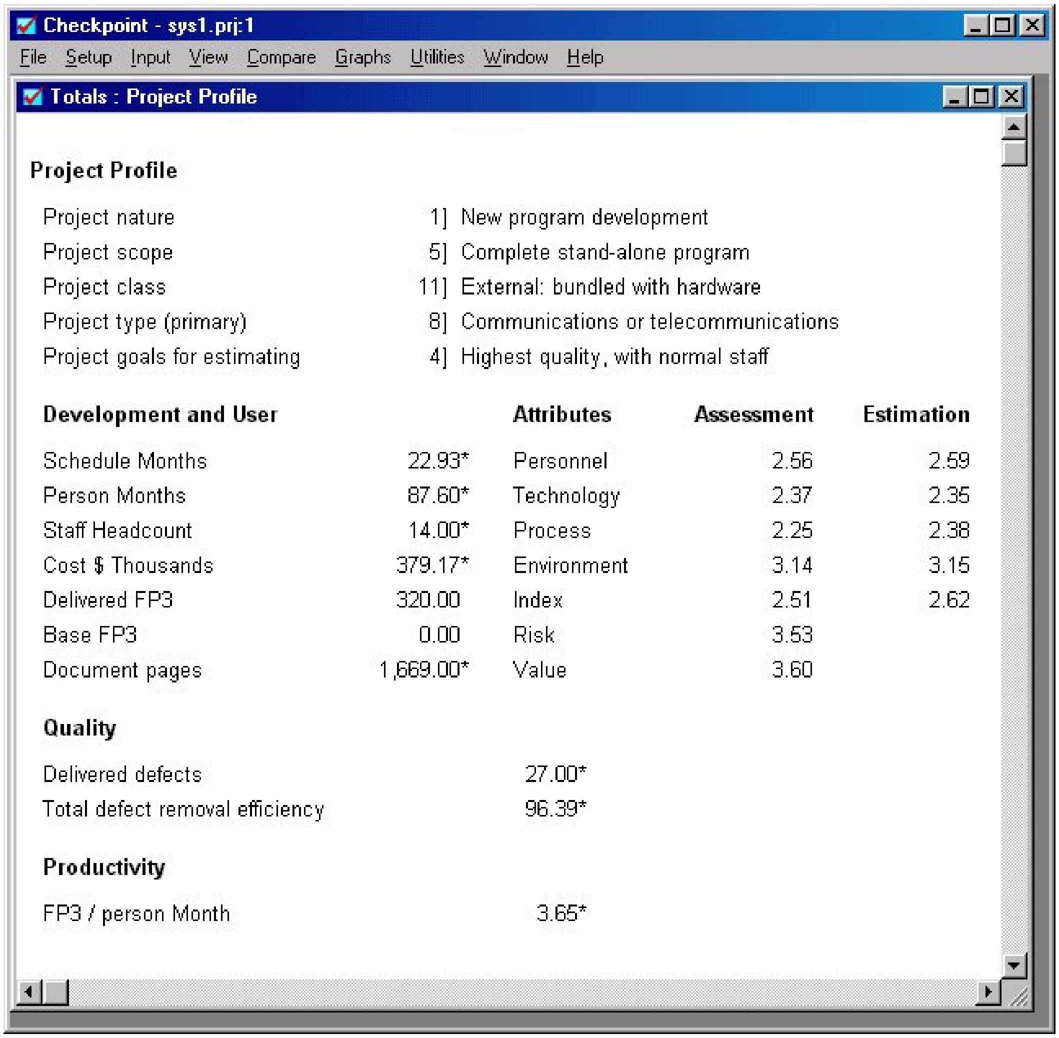


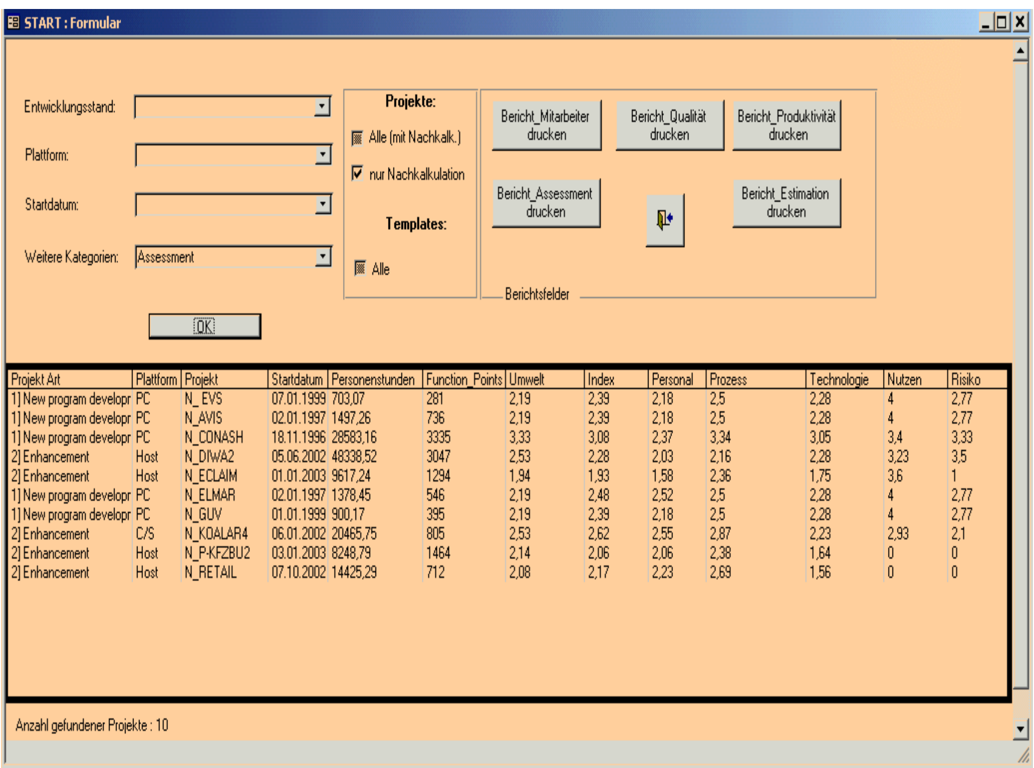
Fig. 7: Checkpoint for Windows "View Totals"

Fig. 8 shows all 10 project post mortems in our project database in CKWIN with the estimation figures of the CKWIN assessment estimation:

- person hours,
- Function Points, environment (Umwelt),
- assessment index (Index: refers to all attribute questions in personnel, process, technology, environment sections),
- personnel,
- process,
- technology,
- benefit (Nutzen),
- risk.

This CKWIN figures range from 1 to 5 with the database average of 3,0. Thus figures above 3,0 are better than the average and larger ones show worse results.

There are 5 PC New Development
and 4 Host Enhancement project post mortems,
the PC projects having sizes of 281, 736, 3,335 and 546, 395 Function Points.
The Host projects have 3,047 and 1,249 and 1,464 and 712 Function Points respectively.

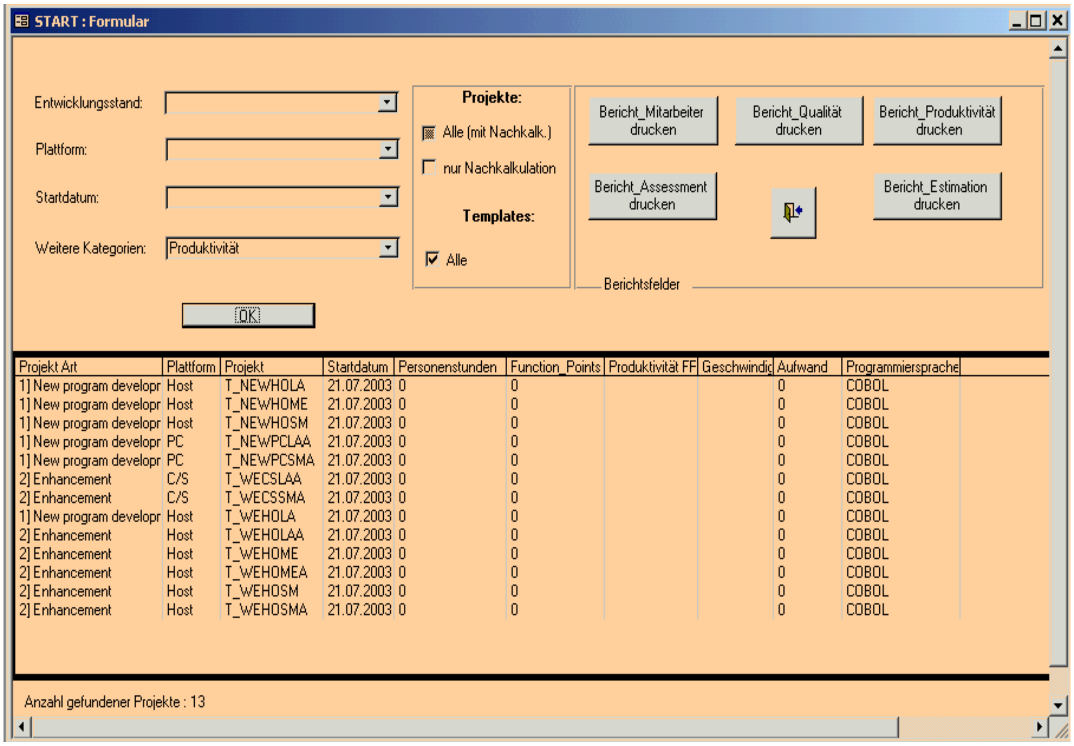


Projekt Art	Plattform	Projekt	Startdatum	Personenstunden	Function_Points	Umwelt	Index	Personal	Prozess	Technologie	Nutzen	Risiko
1] New program developr	PC	N_ EVS	07.01.1999	703,07	281	2,19	2,39	2,18	2,5	2,28	4	2,77
1] New program developr	PC	N_AVIS	02.01.1997	1497,26	736	2,19	2,39	2,18	2,5	2,28	4	2,77
1] New program developr	PC	N_CONASH	18.11.1996	28583,16	3335	3,33	3,08	2,37	3,34	3,05	3,4	3,33
2] Enhancement	Host	N_DIWA2	05.06.2002	48338,52	3047	2,53	2,28	2,03	2,16	2,28	3,23	3,5
2] Enhancement	Host	N_ECLAIM	01.01.2003	9617,24	1294	1,94	1,93	1,58	2,36	1,75	3,6	1
1] New program developr	PC	N_ELMAR	02.01.1997	1378,45	546	2,19	2,48	2,52	2,5	2,28	4	2,77
1] New program developr	PC	N_GUV	01.01.1999	900,17	395	2,19	2,39	2,18	2,5	2,28	4	2,77
2] Enhancement	C/S	N_KOALAR4	06.01.2002	20465,75	805	2,53	2,62	2,55	2,87	2,23	2,93	2,1
2] Enhancement	Host	N_P-KFZBU2	03.01.2003	8248,79	1464	2,14	2,06	2,06	2,38	1,64	0	0
2] Enhancement	Host	N_RETAIL	07.10.2002	14425,29	712	2,08	2,17	2,23	2,69	1,56	0	0

Fig. 8: Project Post Mortems

Fig. 9 shows all templates (the estimation figures of CKWIN are 0 since this are templates to be used for start of an estimation).

There are 6 New Program Development
and 5 Enhancement Templates,
9 for Host (4 New Program Developments)
and each 2 for PC and C/S environment.

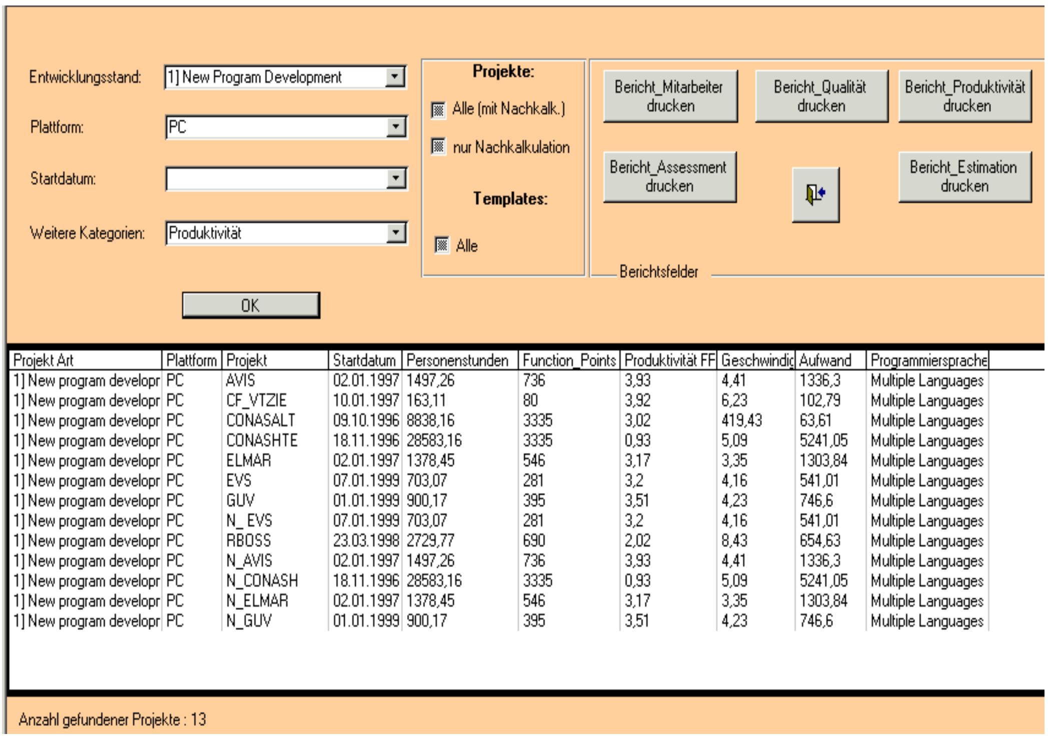


Projekt Art	Plattform	Projekt	Startdatum	Personenstunden	Function_Points	Produktivität FF	Geschwindig	Aufwand	Programmiersprache
1] New program developr	Host	T_NEWHOLA	21.07.2003	0	0			0	COBOL
1] New program developr	Host	T_NEWHOME	21.07.2003	0	0			0	COBOL
1] New program developr	Host	T_NEWHOSM	21.07.2003	0	0			0	COBOL
1] New program developr	PC	T_NEWPCLAA	21.07.2003	0	0			0	COBOL
1] New program developr	PC	T_NEWPCSMA	21.07.2003	0	0			0	COBOL
2] Enhancement	C/S	T_WECSLAA	21.07.2003	0	0			0	COBOL
2] Enhancement	C/S	T_WECSSMA	21.07.2003	0	0			0	COBOL
1] New program developr	Host	T_WEHOLA	21.07.2003	0	0			0	COBOL
2] Enhancement	Host	T_WEHOLAA	21.07.2003	0	0			0	COBOL
2] Enhancement	Host	T_WEHOME	21.07.2003	0	0			0	COBOL
2] Enhancement	Host	T_WEHOMEA	21.07.2003	0	0			0	COBOL
2] Enhancement	Host	T_WEHOSM	21.07.2003	0	0			0	COBOL
2] Enhancement	Host	T_WEHOSMA	21.07.2003	0	0			0	COBOL

Fig. 9: Templates

Fig. 10 shows all 13 new program development projects on PC platform with the estimation figures of the CKWIN productivity estimation parameters:

- person hours,
- Function Points,
- productivity (Function Points per person day),
- speed of delivery (Geschwindigkeit: Function Points per scheduled day),
- scheduled hours (Aufwand) and
- programming language.

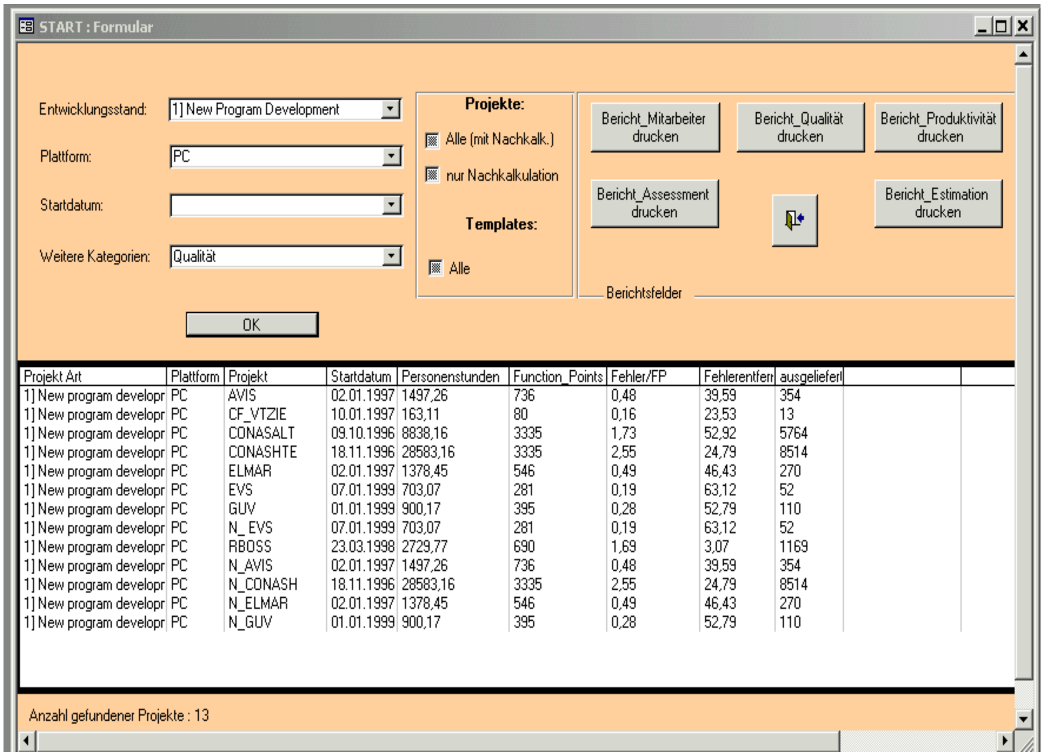


Projekt Art	Plattform	Projekt	Startdatum	Personenstunden	Function_Points	Produktivität FF	Geschwindig	Aufwand	Programmiersprache
1] New program developr	PC	AVIS	02.01.1997	1497,26	736	3,93	4,41	1336,3	Multiple Languages
1] New program developr	PC	CF_VTZIE	10.01.1997	163,11	80	3,92	6,23	102,79	Multiple Languages
1] New program developr	PC	CONASALT	09.10.1996	8838,16	3335	3,02	419,43	63,61	Multiple Languages
1] New program developr	PC	CONASHTE	18.11.1996	28583,16	3335	0,93	5,09	5241,05	Multiple Languages
1] New program developr	PC	ELMAR	02.01.1997	1378,45	546	3,17	3,35	1303,84	Multiple Languages
1] New program developr	PC	EVS	07.01.1999	703,07	281	3,2	4,16	541,01	Multiple Languages
1] New program developr	PC	GUV	01.01.1999	900,17	395	3,51	4,23	746,6	Multiple Languages
1] New program developr	PC	N_EVS	07.01.1999	703,07	281	3,2	4,16	541,01	Multiple Languages
1] New program developr	PC	RBOSS	23.03.1998	2729,77	690	2,02	8,43	654,63	Multiple Languages
1] New program developr	PC	N_AVIS	02.01.1997	1497,26	736	3,93	4,41	1336,3	Multiple Languages
1] New program developr	PC	N_CONASH	18.11.1996	28583,16	3335	0,93	5,09	5241,05	Multiple Languages
1] New program developr	PC	N_ELMAR	02.01.1997	1378,45	546	3,17	3,35	1303,84	Multiple Languages
1] New program developr	PC	N_GUV	01.01.1999	900,17	395	3,51	4,23	746,6	Multiple Languages

Fig. 10: New Program Development PC – Productivity Parameters

Fig. 11 shows the same project like Fig. 10 but this time with the estmation figures of the CKWIN quality estimation parameters:

- person hours,
- Function Points, defects per Function Point (Fehler/FP),
- total defect removal efficiency in % (Fehlerentfernungsrate),
- delivered defects (ausgelieferte Fehler).

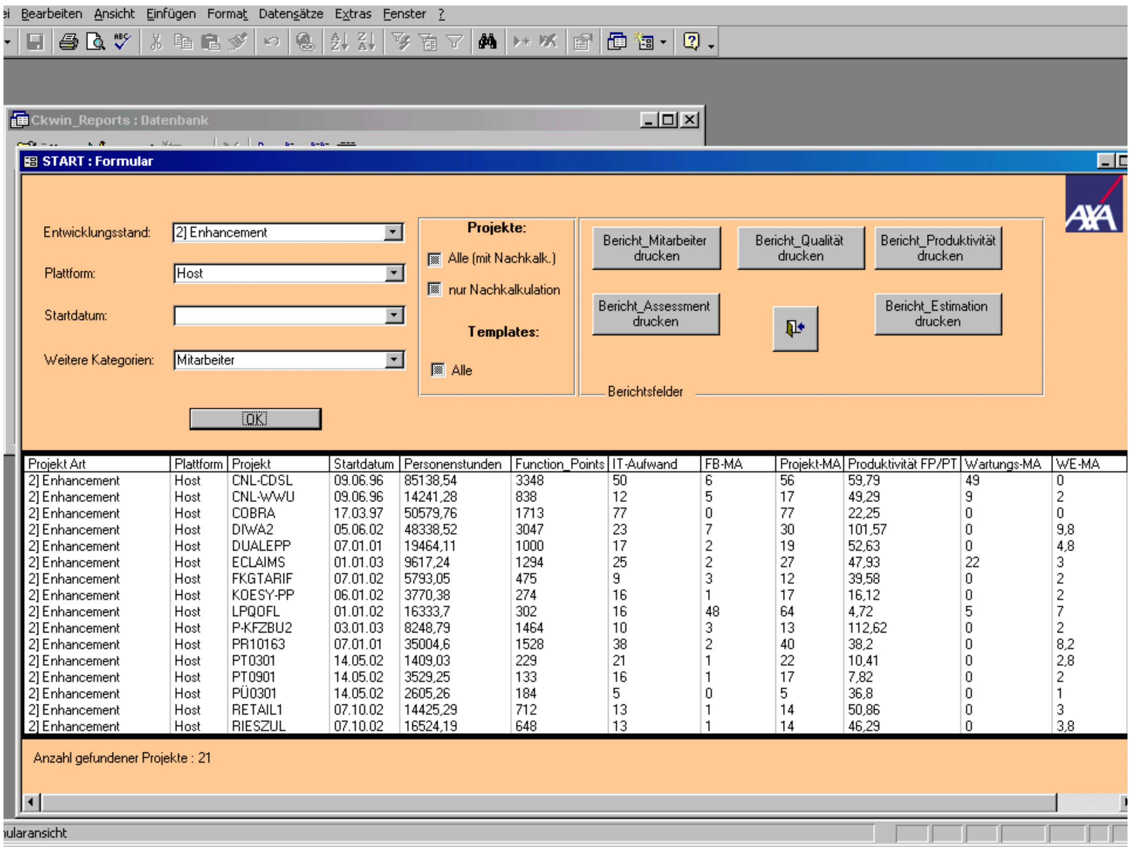


Projekt Art	Plattform	Projekt	Startdatum	Personenstunden	Function_Points	Fehler/FP	Fehlerentfern	ausgeliefert
1] New program developr	PC	AVIS	02.01.1997	1497,26	736	0,48	39,59	354
1] New program developr	PC	CF_VTZIE	10.01.1997	163,11	80	0,16	23,53	13
1] New program developr	PC	CONASALT	09.10.1996	8838,16	3335	1,73	52,92	5764
1] New program developr	PC	CONASHTE	18.11.1996	28583,16	3335	2,55	24,79	8514
1] New program developr	PC	ELMAR	02.01.1997	1378,45	546	0,49	46,43	270
1] New program developr	PC	EVS	07.01.1999	703,07	281	0,19	63,12	52
1] New program developr	PC	GUV	01.01.1999	900,17	395	0,28	52,79	110
1] New program developr	PC	N_EVS	07.01.1999	703,07	281	0,19	63,12	52
1] New program developr	PC	RBOSS	23.03.1998	2729,77	690	1,69	3,07	1169
1] New program developr	PC	N_AVIS	02.01.1997	1497,26	736	0,48	39,59	354
1] New program developr	PC	N_CONASH	18.11.1996	28583,16	3335	2,55	24,79	8514
1] New program developr	PC	N_ELMAR	02.01.1997	1378,45	546	0,49	46,43	270
1] New program developr	PC	N_GUV	01.01.1999	900,17	395	0,28	52,79	110

Fig. 11: New Program Development PC – Quality Parameters

Fig. 12 shows the first part of the 21 enhancement projects on host platform with the estmation figures of the CKWIN staff estimation parameters:

- person hours,
- Function Points,
- IT-effort (IT-Aufwand),
- number of end users,
- development and user involvement (Projekt-MA: number of developers and users),
- productivity (Function Points per person day),
- number of maintenance staff (Wartungs-MA),
- number of enhancement staff (WE-MA).



Projekt Art	Plattform	Projekt	Startdatum	Personenstunden	Function_Points	IT-Aufwand	FB-MA	Projekt-MA	Produktivität FP/PT	Wartungs-MA	WE-MA
2] Enhancement	Host	CNL-CDSL	09.06.96	85138,54	3348	50	6	56	59,79	49	0
2] Enhancement	Host	CNL-WWU	09.06.96	14241,28	838	12	5	17	49,29	9	2
2] Enhancement	Host	COBRA	17.03.97	50579,76	1713	77	0	77	22,25	0	0
2] Enhancement	Host	DIWA2	05.06.02	48338,52	3047	23	7	30	101,57	0	9,8
2] Enhancement	Host	DUALEPP	07.01.01	19464,11	1000	17	2	19	52,63	0	4,8
2] Enhancement	Host	ECLAIMS	01.01.03	9617,24	1294	25	2	27	47,93	22	3
2] Enhancement	Host	FKGTARIF	07.01.02	5793,05	475	9	3	12	39,58	0	2
2] Enhancement	Host	KOESY-PP	06.01.02	3770,38	274	16	1	17	16,12	0	2
2] Enhancement	Host	LPQOFL	01.01.02	16333,7	302	16	48	64	4,72	5	7
2] Enhancement	Host	P-KFZBU2	03.01.03	8248,79	1464	10	3	13	112,62	0	2
2] Enhancement	Host	PR10163	07.01.01	35004,6	1528	38	2	40	38,2	0	8,2
2] Enhancement	Host	PT0301	14.05.02	1409,03	229	21	1	22	10,41	0	2,8
2] Enhancement	Host	PT0901	14.05.02	3529,25	133	16	1	17	7,82	0	2
2] Enhancement	Host	PÜ0301	14.05.02	2605,26	184	5	0	5	36,8	0	1
2] Enhancement	Host	RETAIL1	07.10.02	14425,29	712	13	1	14	50,86	0	3
2] Enhancement	Host	RIESZUL	07.10.02	16524,19	648	13	1	14	46,29	0	3,8

Fig. 12: Enhancement Host – Staff Parameters

Since the time during which we use this tool we can easily discuss with the project leader which project or template we should use for the estimation of his projects. This is a valuable aid since they often come with the notion of “my projects is one half (or one third) of project x”. The overview also supports the acceptance of the tool based estimations since the project leaders see, that there is some know how present and that the others used the help of tool based estimation, too.

Benefits of Early Function Point Prognosis for Early Project Estimation

In the first part of this chapter the benfits of the Function Point method will be shown in order to deliver counter arguments against the widespread killer phrases which one encounters in many organizations. In the then following three parts about the benefits of early Function

Points and early project estimations as well as lessons learned the according informations from the previous chapters are summarized.

Benefits of the FPM

The usage of the FPM can easily be understood, learned and practiced. These are valuable factors considering the fact that counts will be audited and accepted in quality assurance measures and views. On the other hand FP's define the application systems objectively from user view and do this consistently with the perception of the users for their functionality and in their own language. The whole project team is thus forced to model the application system from user view. The functional elementary processes map the business requirements (bussiness processes) accordingly. This leads definitely to better designs and to better chances for controlling during project progress.

When the FPM is performed together with the endusers they will be motivated for teamwork and engagement. Hence their acceptance of the whole project can be improved. This is a very important project success criterion.

Nevertheless there exist a lot of prejudices against Function Points. They are mostly based on the assumptions of the project leaders who fear to become measurable and comparable regarding their productivity as well as transparent regarding their estimations. In this cases the management must deal with their fears and commit to measure projects and not persons. On the other hand there exist concrete concerns because of widespread killer phrases, which can be discussed by counter arguments, as shown in Fig. 13.

Function Points are disliked, since ...

PREJUDICE	COUNTER ARGUMENT
... they are developed by teoreticians and not practically usable.	Originally developed by A. Albrecht in a project for the development of system software.
... They produce administrative overhead.	The effort is compared to the benefit and the whole project effort neglieably low.
... They are not usable for object oriented application development.	FP's are a meta-model, they allow a mapping of the requirements, no matter in which description.

Fig. 13: Function Points are disliked

The principal benefit of Function Points is that they are independent from the development environment as well as from the developers. By consequent usage of the IFPUG Function Point Method and careful documentation of the counts the organization gets project portfolio wide consistent data about the size of already developed as well as to be developed applications. This informations are not only a solid basis for professional estimations of the required effort but also a necessary prerequisite for inter-organizational benchmarking based on this international standard.

The knowledge of project size can help software developers to decide which functionality may be left out in order to reach milestones. Function Points thus can be used to calculate different scenarios and plan releases. They bring both sides, users and developers, transparency and a commen understandig.

Fig. 14 shows the different possible areas of application of measured FP's in the software engineering process as well as in project management.

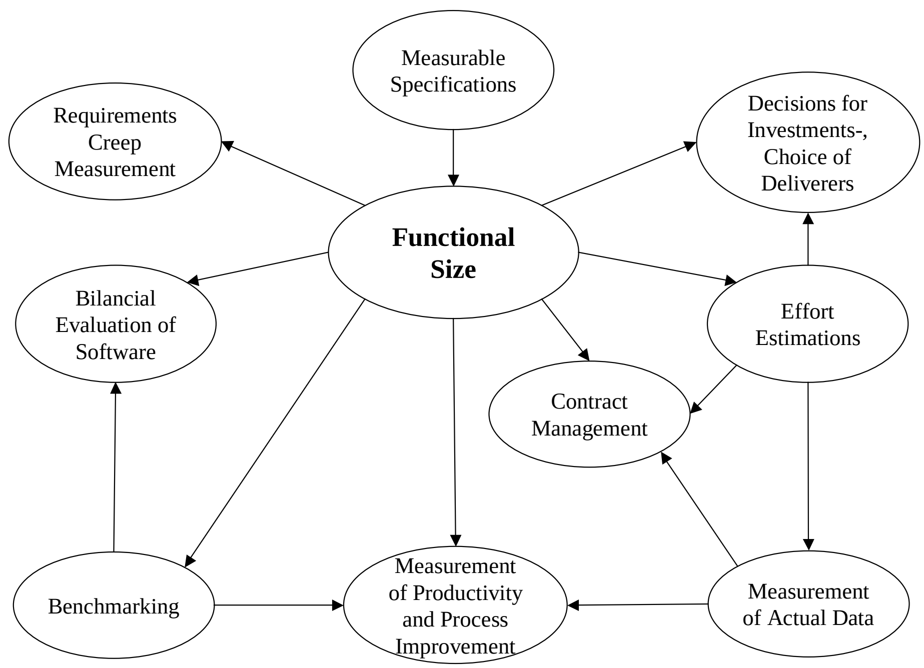


Fig. 14: Usability of Function Points

Function Points support easier and improved estimation as well as ease of work and improvement of quality, since business cases for enhancement tasks and projects can be declared without misunderstandings, test cases are easily deductible and productivity improvements measurable**.** Besides that proper documentation of counts in a tool allows easy measurement of small enhancements with one mouse click.

There are many benefits of Function Points:

- FP's as input for estimation and project management
- FP's as basis for project planning and architectural considerations
- FP's as basis for project structuring and release planning
- FP's as basic metrics for quality planning and management reviews
- FPM for implicit control of requirements (completeness and misunderstanding) and hence for quality improvement

- FPM for design of test cases and estimation of test effort and improved planning of the test phase
- FPM delivers metrics for stability and reliability
- FP's as basis for benchmarking and risk studies
- FPM for user oriented documentation of the applications
- FP's for measurement of the requirements creep
- FP's for measurement of productivity and quality

FP's support reuse in application development through early and unique representation of business cases for estimation at project start, for proposals, for project effort estimations, for test case definition, for enhancement projects, for project changes and for documentation.

Fig. 15 describes the benefits of the Function Point method in an overview.

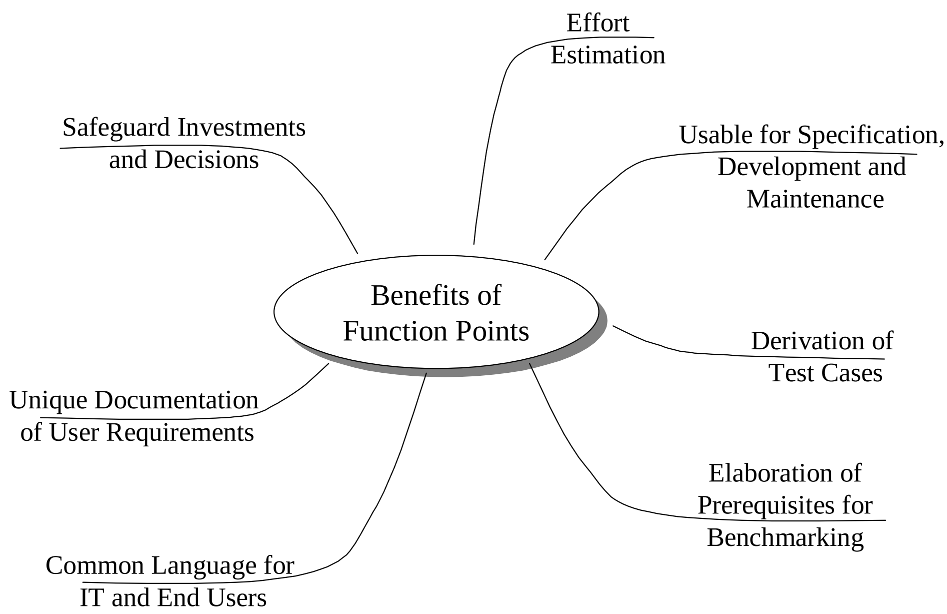


Fig. 15: Benefits of the Function Point method

The knowledge of project size is a decision aid for functionality that can be left out in order to reach the milestones of the project. With FP's thus several scenarios can be calculated and release planning can be performed. FP's thus bring both partners of development contracts transparency and common understanding.

Thus Function Points can also be used for contract metrics:

- costs per delivered FP for new development or enhancement
- fixed price for a fixed number of FP's
- fixed price for maintenance of a portfolio of a FP measured size
- different prices if proposed sizes come not up to the estimate or exceed it
- for process improvements (targets for bonusses e.g.)
    - a certain number of FP's for one hour of new development or enhancement
    - cost per FP for one hour of new development or enhancement

- a certain number of FP's per FTE for the maintenance of an application system
- a certain number of errors per FP

Generally contracts should comprise also following topics (besides all usual themes):

- number of Function Points (functional size)
- different measures for less or more Function Points
- estimated costs, effort, staff and its qualification and dates of availability
- tracking of project progress and measures in case of delay

Another interesting usage of Function Points is the comparison of prices when buying COTS software (COTS = Commercial off-the-shelf) as well as for contract development. Due to Capers Jones [Jone02b] this is increasingly done in the USA. Fig. 16 shows the principles of the usage.

With the price per Function Point one is able to discuss the contracts with the partners. Capers Jones [Jone02b] reports that usual standard software like spreadsheets can be bought for $ 0.25 per Function Point. More specialized niche packages can cost in the range of $ 10 to $ 300 per Function Point. Software development costs can run from less than $ 200 for small end user applications to more than $ 5,000 per Function Point for large military and defense applications. This prices can be compared with software development costs of $ 1,500 in Western Europe as well as $ 350 per Function Point in Eastern Europe.

Function Points are also used to solve disputes between contractors. Capers Jones [Jone02b] reports that Function Point metrics have probably been used in at least 15 breach-of-contract cases and 10 tax cases in the United States and that both civilian and military cases have used it.

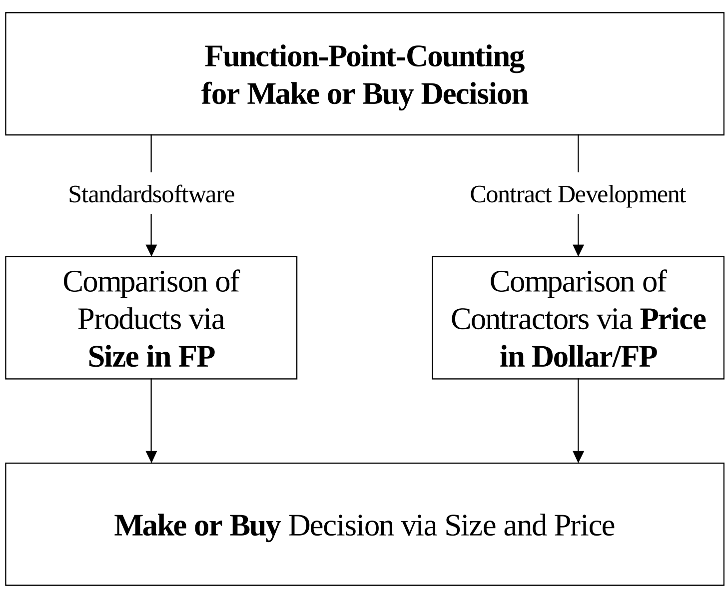


Fig. 16: Function Points for Contract Metrics

Last but not least there is must be considered and calculated that the effort for planning and performing as well as for the documentation of Function Point counts is always negligeable compared to the enormous benefits.

Benefits of early Function Points

In this chapter some of the before mentioned results of the exploration of measured data are summarized with the focus on the early usage of estimated Function Points.

Early collection of the informations about interfaces and parts of the project and documention of it together with a diagram of the application boundary which is also used for our architecture atlas is a prerequisite for early Function Point prognosis. Together with a so called counting log which is a simple Word document giving important informations about special decisions concerning the the FP counts a valuable overview about metrics for software measurement can thus be achieved. In general this might be reference numbers (key figures) for quality assurance (percentages,…).

Another well known side effect of careful measurement and documentation of the Function Point counts is the higher quality of the requirements documents afterwards, since they are used, checked and have to be revised. Often they are not available or not actual at the start of Function Point counting and thus have to be revised. This leads also to a higher quality and efficiency in software development, a beneficial side effect of Function Point counting.

With the available details one can calculate with the Excel problem solver e.g., the average function complexity of the five components, i.e., how many FPs a “typical“ EI, EO, EQ, ILF or EIF has in the actual environment. It is widely agreed that this measure is stable and can be used as a rule of thumb for quick estimation of counts, since the components need not be classified as low, average or high. SPR Function Points, e.g., use the average IFPUG classification for Function Point estimation. But it is always better to use the own typical average Function Points instead of the IFPUG’s since mostly there will be differences.

In 1998 we tested the applicability of the typical FPs for estimation purposes by multiplying it with the quantities of the EIs, EOs, EQs, ILFs and EIs, respectively, and compared the results with the unadjusted Function Points of the counts[Eber04]. The error was less than 26% [Bund02b].

One of the first results of our data collection was the perception that the value adjustment factor (VAF) of our counts is typically in the range of 0.73–1.22, with an average of 0.95 in the 2001 data (0.93 in 1998), and an average of 0.94 for host and 0.96 for PC environment. The average for migrations is 0.73. We have used this metrics as a rule of thumb for quality assurance of our Function Point counts since 1998 [Eber04].

Another experience learned was that the average Function Point size increased over time, which may be caused by growing complexity in the application development environment. Such results can only be gained by continual recoding and examination of the measured data during a time range.

Our findings compared with those from other organizations show that such data collections can be used to find heuristic solutions for FP - Prognosis, either using Function Point proportions („typical FP’s“), function ratios or regression formulae. But there is evidence that different environments demand for according solutions. Hence each organization should develop its own know-body of heuristic solutions and should distingnish between different development platforms etc. when doing so.

Benefits of early Project Estimation

In this chapter some of the before mentioned results of the usage of our Function Point Prognosis are summarized with the focus on the early estimation of projects.

There is a demanding business need for early estimations. One of the most influential parameters of estimation is the size, e.g., Function Points. Thus FP prognoses or FP approximations practically proved to be helpful aids for early estimations. Centrally collected informations about FP counts allow research that can deliver useful metrics for quality assurance, estimation and benchmarking. Hence the measurement of software size is the most important parameter of estimation. We heard from other organizations that they did similar research. There is strong evidence that different environments will lead to other results, hence each organization should develop its own heuristic solutions. Standards for the distribution of the estimated effort to the project phases can be used to develop an organization specific percentage method for estimation.

As we have shown before, our project leaders are able to deliver a profound and well documented tool- and measurement based estimation before the start of their projects. In some cases we used this for project portfolio decisions during portfolio planning for the next year. As well as in alike other early estimations we could do the estimation within half a day, documented in our estimation tool and in a so called estimation log – a plain word document reporting about the suggestions we made and about the way how we proceeded during the estimation.

Project estimations are required as early as possible – not only from the contractors but also from every project leader. Because of the importance of early estimation methods Meli and Santillo [Meli99] published a comparative overview of Function Point estimation methods that shows a valuable collection of worldwide efforts in this direction. Since Function Point counts are based on the requirements documentation, so-called Function Point prognoses or approximations proved in practice to be helpful to aid early estimations [Bund98b], [Bund99].

A method without tool support has only a little chance for survival. Hence the usage of measurement and estimation tools is in any case a valuable investment, albeit the benefit might not be quantifiable. The experiences of the author are based on the personal (good) experiences with the dedicated tools used in the own organization. Nevertheless this experiences can be transferred to any other estimation tool.

Tools used without appropriate knowledge deliver non-usable results. Especially estimation tools afford careful customization and critical reflection of the estimation parameters since estimation has to do with uncertainty per se. Estimation results with decimal points produce illusions of a not existing accuracy and let the user feel in a treacherous safety. The results of the tool can only be as good as the informations available and used for the estimation with the tool. For this reason estimations with tools should also be performed in different ways for plausibilization. When we discover differences in this estimations the benefit will usually be important learning effects about the object of estimation, prerequisites of the estimation, the environment or process of estimation as well as about the evaluation of the estimation parameters. This may give valuable hints for project risk assessment, too.

Some estimation tools offer interfaces to project management tools (as e.g. KnowledgePLAN to MS Project). Thus estimation tools support quicker and more efficient planning. And they assist in gaining overview when using many parameters for estimations.

The major benefits (time saving, quality and efficiency imporovement) are thus harvested project by project over time and summon up accordingly in an organization. Besides this there emerge beneficial side effects, as e.g. improved acceptance or improved transparency, standardization and a culture for estimation.

Lessons Learned

In this chapter some of the before mentioned experiences of the usage of our Function Point Prognosis and early estimation of projects are summarized.

Our experience shows, that the necessary information for such approximations can be gained very early, at the beginning of a project (or, in stable environments, even before that) in discussions with the project leader. In a few cases we used this Function Point Prognosis about a year before project start. Of course, we added large percentages for each error, uncertainty, early estimation and risk [Eber04]. This additions can easily sum up to 100 % at this early stage of a project. This can be verified by the experience that failed projects often costed more than twice the proposed price. Newspapers often headline such examples.

One obstacle hindering early estimations is the fact that empirical data are missing and that estimation is done too early and seldom. Three important measures should always be considered. First, the effort for measurement and estimation must be planned from the beginning on as a project management task. Second, a method without tool support has only little chances for survival. Third: size has to be measured before estimation.

One important experience is that the people in the organization need time to develop a culture for measuring and estimation. We needed almost 8 years to complete the total counts of all 98 application systems (excluding Enterprise Resource Planning (ERP) applications, e.g. SAP, Peoplesoft) with a total of about 150,000 unadjusted Function Points (FPs). By this time productivity measures were developed, and a baseline for annual productivity measurement was completed. One year later a self developed metrics database was implemented and since then effort, defects and FP's were recorded every month for each development, enhancement and maintenance tasks.

Expert estimations often deliver lower estimations than experienced based estimations. Reasons for this underestimations are that estimators overestimate productivity and that high efforts are not appreciated by managers, customers and sales departments. Hence estimations often are political, which is a great obstacle for the development of a fair estimation culture.

Estimation conferences lead to an estimation that is accepted by all involved persons, which is more objective than the above-mentioned average and hence can be better defended against other opinions. The results may not differ very much, as we found in some cases.

Another benefit of the estimation conference is that the involved estimators gain awareness of the uncertainties and possible risks of the IT project. Furthermore they all get the same information. An estimation conference is a team-building experience. An estimation conference also promotes the estimation culture in an organization, since it helps to solve acceptance problems by finding a consensus through discussions in a team. These benefits can often be gained in only one estimation conference lasting not more than two hours!

Estimation is the basis of the economic benefit of projects, measured by cost benefit analysis, Function Point Method and diverse estimation methods. If this project management tasks are not performed professionally the project manger behaves gross negligent.

The same holds for the project management task documentation. Especially the measurement of project size as basis for estimation is a review of the functional requirements. And it is a prerequisite for the management of requirements creep as well. The documentation is important for quantification in order to understand the system to be developed. Basic experiences can be gained from good documentation and can be filled fron there into project management manuals.

If it is not possible to measure functional size because of missing (existing, actual, understandable) documentation or missing know how in the IT project then this is a sign that the requirements analysis is not yet finished. Or it is an indicator showing that the IT project has lost it's overview (or is out of control) already at the start of the project. In this case the project leader should better consider not to start the project [translated from Bund04a].

Literature

[Bund00a] Bundschuh, M., Fabry, A.: Aufwandschätzung von IT-Projekten. MITP Bonn (2000) p. 331

[Bund00b] Bundschuh, M.: Function Point Approximation with the five Logical Components, FESMA 2000, Madrid, Spain, October 18-20, (2000)

[Bund02a] Bundschuh, M.: Function Point Prognosis Approved, in: Metrics News, Vol. 7, No. 1, July 2002

[Bund02b] Bundschuh, M., Estimation of Maintenance Tasks, in: Dumke, R. et al. (eds.) Software Measurement and Estimation – Proceedings of the 12th International Workshop on Software Measurement, 2002, Shaker Aachen (2002) ISBN 3-8322-0765-1, pp. 125-136

[Bund04] Bundschuh, M., Fabry, A.: Aufwandschätzung von IT-Projekten, MITP Bonn (2004) ISBN 3-8266-0864-X (2nd edition)

[Bund98a] Bundschuh, M.: Function Point Prognosis, FESMA 98 "Business Improvement through Software Measurement", Antwerp, Belgium, May 6-8, (1998), pp. 463–472

[Bund98b] Bundschuh, M.: Function Point Prognosis. In: Metrics News, Vol. 3, No. 2, December 1998

[Bund99] Bundschuh, M.: Function Point Prognosis Revisited, FESMA 99, Amsterdam, Netherlands, October 4-7, 1999, pp. 287-297

[Eber04] Ebert, C., Dumke, R., Bundschuh, M., Schmietendorf, A. (Ed.), Best Practices in Software Measurement, How to use metrics to improve projct and process improvement, Springer Berlin (2004) ISBN 3-540-20867-4

[Gaff94] Gaffney, J. E. Jr.: A Simplified Function Point Measure. In the Proceedings of the IFPUG 1994 Fall Conference, Oct. 19-21, 1994, Salt Lake City, Utah

[ISBS00] ISBSG, The Benchmark, Release 6, ISBSG, Warrandyte, Victoria (2000) ISBN 0 9577201 6 5

[ISBS02] ISBSG: The Software Metrics Compendium (The Benchmark, Release 7), ISBSG, Warrandyte, Victoria (2002) ISBN 0 9577201 2 2

[ISBS99] ISBSG, Software Project Estimation – A Workbook for Macro-Estimation of Software Development Effort and Duration, ISBSG, Melbourne, Victoria, 1999, ISBN 0-9577-2010-6

[Jone02a] Jones, C.: How Software Estimation Tools work, Technical Report, Software Productivity Research Inc. Burlington, MA, April 6, 2002

[Jone02b] Jones, C.: The Expanding Roles of Function Point Metrics in: IFPUG, IT Measurement – Practical Advice from the Experts, Addison-Wesley Pearson Education Inc., Indianapolis, 2002, ISBN 9-780201-741582, pp. 3–30

[Keme93] Kemerer, C.F.: Reliability of Function Point Measurement: A Field Experiment, in: Communications of te ACM (CACM), 36 (2), 1993, p. 85-87

[Meli99] Meli, R., Santillo, L.: Function Point Estimation Methods: A Comparative Overview. In: Proceedings of the FESMA Conference 1999, Amsterdam, (1999) pp. 271–286

[Morr96] Morris, M., Desharnais, J. M.: Validation of the Function Point Counts. In: Metricviews (IFPUG), summer 1996, (1996), p. 30

[Noel98] Noel, Damien: Analyse statistique pour un design plus simple de la methode de measure de taille fonctionelle du logiciel dans des contextes homogenes. Doctoral dissertation, UQUAM, Montreal, Cananda, 2.8.1998 (1998)

Biography

Manfred Bundschuh works since 22 years in the IT Department of the AXA Insurance, Cologne, Germany, being responsible for project management tools and the introduction of the Function Point metrics program as well as IT project estimation.

Before this job he worked 7 years as Controller, IT Trainer and Consultant in 2 IT Consultancies in Hamburg.

Besides he is appointed professor at the University of Applied Sciences in Cologne since 22 years and lectures about requirements analysis, IT design and IT project management as well as about basic skills for personal management and teamwork in IT projects.

Since 7 years he is also president of the DASMA e.V., the metrics organization of the german speaking countries (http://www.dasma.org).

He has given more than 100 public lectures (some of them in english), has published more than 50 papers (most of them at international conferences) and is co-author of more than 10 books (some of them in english).

His homepage is http://www.gm.fh-koeln.de/~bundschu.
His email address is manfred.bundschuh@freenet.de or bundschuh@dasma.org.