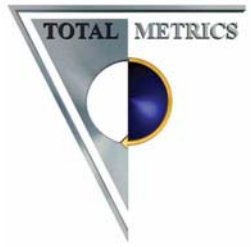

Methods for Software Sizing

How to Decide which Method to Use

1 Why Measure Software Size?

Software is the output product from the software development and/or enhancement activity that is delivered and/or supported by IT.

Traditionally the 'amount of product' produced from a software development project was perceived as being the amount of programming source code written. I.e. Source lines of Code (SLOC or KLOC). In early homogeneous software development environments the amount of SLOC and the perceived relative size of software had a fairly direct relationship. However, as technology progressed, and software was built using hybrid languages, re-usable modules, COTs components, utilities, code generators and high level languages, the relationship between the SLOC and the relative size of the software became less predictable.

Developers need to be able to accurately estimate the effort and cost to deliver a software product and to compare different solutions, technology and tools for their efficiency and cost effectiveness. But in order to do this they needed to be able to quantify 'what' they are building i.e. how 'much' software, since the resources required are related to 'how much' software is built or maintained.

In the late 1970s Allan Albrecht from IBM developed a method to size the functional requirements, called Function Point Analysis. This concept of sizing the functional user requirements to establish relative size of software has advanced over the past three decades so that now the concept has been transformed into an ISO standard called "Functional Size Measurement".

There are five ISO Methods for Functional Sizing, that fall into two main groups, those derived from Albrecht's original methodology (IFPUG Function Point Analysis, and NESMA Function Point Analysis) and those that derive from extensions of his method (MK II and COSMIC Functional Sizing Methods).

2 Why use Functional Size Measurement (FSM)?

Functional Size Measurement (FSM) is the only internationally recognised and [1]ISO standardised technique to measure the size of the Users Functional Requirements. Since Functional Requirements are independent of any constraints of *how* the software is built (i.e. Independent of the non-functional technical and quality requirements), then it enables software

[1] ISO/IEC Functional Size Measurement standards 14143 Parts 1 to 6

to be measured accurately and repeatably over time, as developers utilise different tools, techniques and technologies to build software.

FSM also enables comparisons of IT efficiency and cost effectiveness using different development environments and strategies.

The recognition of the value of Functional Size Measurement is such that it is the ***only*** measurement unit for software development and support processes that has been formalised to be the level of an ISO standard. All other common measures such as effort, duration, defects or speed of delivery do not have an internationally agreed method of collection, validation, and comparison.

3 Why isn't everyone using FSM?

Many critics of the technique claim that FSM:

1. Is no longer relevant to current development technology environments
2. Is no longer relevant to the *type* of software built today
3. Does not adequately 'measure' highly algorithmic complex functionality
4. Is too time intensive
5. Does not provide any value if you do not know how much effort it takes to build a unit of software
6. Requires too much detailed information, much of which is often not available
7. Is outdated and other newer methods such as 'Use Case' Points are better
8. Does not take into account the constraints on the users requirements such as stringent quality requirements and highly complex architectural solutions.
9. Has five ISO approved methods and it is too hard to choose which one to use
10. Needs highly skilled trained measurement experts to measure accurately
11. The resulting software project and application sizes are difficult to maintain over time

Some of these comments are valid criticisms and others are excuses for avoiding the hard work of measurement. We have addressed each of these points in turn below.

4 The Value of FSM as a Size Measurement

FSM has become the method of choice for organisations worldwide to measure their software development product. It has been unequivocally proven to be the most reliable and effective method to estimate software and to compare productivity. Whilst its exponents admit it is still not perfect, there continues to be [2]teams of experts around the world, working daily on improving and certifying FSM related methods, tools, training, and standards.

4.1 Is FSM relevant to current development environments?

Since the Functional User Requirements are independent of the development environment, then the measurement of these requirements is not influenced by 'how' the software is built. However the effort and cost to build and support is influenced by the environment; therefore for estimation and comparison purposes, different productivity factors are applied for the different environments.

[2] IFPUG Counting Practices Committee, COSMIC Core Measurement Team, NESMA Counting Practices Committee, ISO/SC7 Working Group 12

4.2 Is FSM relevant to the *Type* of Software being built today?

Since FSM measures the Functional User Requirements, then as long as there are requirements, they can then be measured. However different software can have different *types* of requirements and this is why we have different Functional Size Measurement Methods. Different methods are attuned to measuring different types of requirements. Identify the functional domain of the software you are building and select the appropriate FSM method.

4.3 How does FSM deal with Algorithmic complexity of Software?

All the current ISO Standard FSM Methods, measure the relative size of software functions based on the amount of different data types processed (enter, leave, read and written to storage) by the function. The intermediate algorithmic transformations, translations and conversions of those data types are not included in the measurement. Only the actual individual data movements from one form to another are considered. The reason for not considering algorithms is because there is no internationally accepted way of defining or quantifying their complexity. In reality for most applications, complex algorithms only exist in a very small proportion of the software. One commonly accepted way to address the impact of the algorithms on effort and cost is to isolate the functional area and apply a different cost factor to those requirements that have algorithmic complexity.

4.4 How much time will it take to functionally size my software?

The act of measuring requires the dissection of the Functional Requirements into their elementary processes, which are in turn catalogued and assessed for size. The effort to do this will be directly proportional to the magnitude of the Functional Requirements, but will be also be strongly influenced by the quality (completeness, ambiguity, consistency) of the specification of those requirements. Actually doing the measurement, highlights any gaps in the requirements and is often the only fully documented list of functions for an implemented system. Time consumed in the measurement is in most cases compensated by the time saved by identifying requirements defects early in the life cycle of a project.

Typically an experienced functional size measurement expert can measure, document and report between 200 and 300 function points per day. This is equivalent to a project that would consume 12 to 18 person months of effort. I.e. Measurement is < 0.5% of project effort.

However in our experience, the cost of measurement is far outweighed by the [3]benefits the measurement provides the project manager in optimising the success of the project and minimising risk of project failure[4].

[3] Uses and Benefits of Function Points – Pam Morris www.totalmetrics.com
[4] Metrics Based Project Governance – using measurement to minimize project risk - Pam Morris www.totalmetrics.com

4.5 How can I use the Functional Size if I do not have any other measures?

This is a 'chicken and egg' problem, where a project manager needs to estimate a high risk project but even if they measure the functional size, it will be of little benefit if they have no idea of how many hours or dollars it take to build a functional software unit.

We recommend to start measuring now and with time you will build your own metrics repository. This will enable you to know, for each development environment, your organisations number of effort hours to build or change a functional unit and the relative cost per unit. However, until you can collect enough data, there is always industry data available to assist in project estimations and productivity comparisons. For example the International [5]Software Benchmarking Standards Group (ISBSG) has freely available productivity data on over 4,000 projects and all the latest development environments.

4.6 How can I use FSM if I do not have detailed functional specifications?

Whilst accurate Functional Sizing does require detailed functional specifications to identify each data group used by a functional process, there [6]are ways to do less accurate sizing with less detailed specifications. Since the accuracy of the measured size is directly related to the accuracy and completeness of the specification, then these less accurate [7]'approximation' methods will result in a size that is anywhere between 15% to 30% different from true value. However, at early cost benefit analysis of a project, a ball-park size provides valuable information when assessing the viability of a project budget or schedule.

4.7 Has FSM been superseded by alternative methods for sizing software?

The use of FSM world wide is increasing daily, from India, China, Korea and Brazil to the United Kingdom and the USA. In terms of methods to size software it is estimated to have >99% of market share. All Functional Sizing methods work on the same principle i.e. to measure the functionality delivered by the software. Whether they do this by measuring Use Cases or by identifying elementary processes, the principle is the same. The advantage of using the ISO standard Functional Size Methods is that the functional process measured by these methods is clearly defined such that requirements can be accurately and repeatably decomposed to the elementary process level, no matter how they are specified. This is also true for implemented software, it can be analysed and decomposed to this same elementary process level. The methodology to measure Use Cases and assign points, has the limitation that there is no internationally accepted standardised level of granularity for defining Use Cases and as such there tends to be no external consistency in the measured size, making it difficult to compare externally or utilise industry data.

4.8 How can I estimate the impact of stringent Quality and Technical Requirements?

FSM only measures the functional requirements; it does not take into account variations in quality and technical requirements. Software size however, is based only on functionality

[5] ISBSG –- ISBSG Estimating, Benchmarking and Research Suite R10 – www.isbsg.org
[6] Levels of Counting – Total Metrics – www.totalmetrics.com
[7] SCOPE Outline™ a software tool that calculates the approximate functional size of your software from a series of questions about the software project or application.

delivered. In software development the productivity rate is adjusted relative to the impact of the quality and technical constraints. The [8]degree of adjustment can be determined from similar projects available in Industry Data.

4.9 Which FSM Method should I choose?

There are currently five FSM Methods approved by the International Organisation for Standardisation (ISO):

- **COSMIC-FFP** Functional size measurement method v 3.0.1 [10]
  - ISO/IEC 19761:2009 - Information technology -- Software and systems engineering ---- COSMIC-FFP -- A functional size measurement method
- **IFPUG CPM 4.3** [11]
  - ISO/IEC 20926 : 2009 - Information technology -- Software and systems engineering --— Software measurement — IFPUG functional size measurement method 2009
- **Mk II Function Point Analysis** 1.3.1 Unadjusted [12]
  - ISO/IEC 20968:2002- Software engineering – Mk II Function Point Analysis - Counting Practices Manual
- **NESMA FPA Method** 2.1 Unadjusted [13]
  - ISO/IEC 24570:2005 - Information technology -- Software and systems engineering --- NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis
- **FiSMA FSM** 1.1 [3]
  - ISO/IEC 29881:2008 - Information technology -- Software and systems engineering -- FiSMA 1.1 functional size measurement method

ISO has recently published a [9]guide to choosing the method most appropriate for your needs. Key points needed to be considered are the following:

- **Applicability to the domain of software that you need to measure**. As a guide most organisations that develop or support 'data rich' software (i.e. Management Information Systems, Banking, financial, CRM, Asset Management, Work flow systems etc) use the IFPUG method. Organisations that develop or support 'process rich' or real time software tend to use the COSMIC method.
- **Availability of equivalent Industry data.** If you need to use Industry data for comparison of productivity or as input into estimates then this may be a deciding factor. The ISBSG repository has >90% of projects measured with the IFPUG method, but the proportion of COSMIC projects is increasing rapidly as the COSMIC method becomes more widely used.
- **Availability of training and FSM tools**. Most of the Industry leading tools e.g. SCOPE Project Sizing Software™ have been written to measure using the IFPUG method, but will also offer a version in the future that uses the COSMIC method, as its use increases.

[8] ISBSG Practical Project Estimation 2nd edition – www.isbsg.org
[9] ISO/IEC 14143-6: SOFTWARE ENGINEERING — SOFTWARE MEASUREMENT — FUNCTIONAL SIZE MEASUREMENT — PART 6: GUIDE FOR USE OF ISO/IEC 14143 SERIES AND RELATED INTERNATIONAL STANDARDS

- **Availability of trained experienced certified Metrics experts**. Currently most people are trained and certified in the IFPUG method which is offered in most countries, but COSMIC training and certification is now being offered in Europe, North America and Australia.

4.10 How hard is it to measure Functional Size?

Functional Size measurement requires specialised training of 2 to 3 days duration. After training, it typically takes several months of using the technique, measuring in a variety of situations to become proficient. International accreditation usually requires the measurer to have a skill level of at least 2 years experience in using the technique. Most organisations train a select group of software developers for the measurement role. It is similar to other specialist activities e.g. Database Design, it takes a person skilled in business analysis and attention to detail to make a good Functional Size Measurement Specialist.

4.11 How do we effectively manage the Functional Size Results?

Whilst Functional sizing can be performed by recording the results in a spreadsheet, experience shows that this approach has proved to be very high risk which are mitigated by using a specialist FSM tool such as SCOPE Project Sizing Software™. Problems we have experienced with clients using spreadsheets are as follows:

- **Results are hard to check for correctness or completeness**: There is no capability to 'model' the functional requirements, and long lists of functional processes (often thousands) are very hard to track. It is very difficult for the applications expert to determine if the measurer has missed any of the requirements from the measured project size. Omissions will result in under estimating the effort and cost of the project.
- **Size Calculation formulas are prone to Corruption**: Formulas totalling the components of the measurement are prone to have their ranges corrupted by inexperienced users and maynot calculating as originally designed.
- **Significant time is wasted every measurement as there is little opportunity for re-use of previous measurements**: Each time a new enhancement project is started a new spreadsheet is started. There is rarely a library or cross-reverencing capability to determine if a previous project measurement could be leveraged and re-used. Inconsistent and duplicate measurement can result as there is no cross-referencing to decisions made in the original baseline to ensure functions are counted consistently.
- **Baseline Sizes are quickly out of date**: If a baseline measurement was completed for an application or project, change requests are rarely retrospectively applied to keep the baseline size current; since on a spreadsheet this involves two different acts of measurement recording (Baseline updates are performed automatically using FSM tools such as SCOPE).
- **Configuration control of multiple concurrent projects is very difficult.** If multiple Projects are applied to the same application and several people develop concurrent measures of the impact of their projects, then it is a significant cross-referencing activity to determine if the same function is impacted by several projects and to determine the final impact on the baseline measurement

5 Summary

The past three decades of use of functional size measurement has shown that it is currently the only proven method of sizing software that gives consistent and reliable results for project

estimation and productivity comparisons. The FSM method for sizing is supported and continually enhanced by the international community and is the method of choice for major software estimation tools and benchmarking organisations.

For further information contact:

Contact:

+61 3 9882 7611
Admin@totalmetrics.com

WWW.Totalmetrics.com