Counting "Non-Functional" Requirements when they are implemented as Software

Version 2.2 October 2006

Pam Morris – Total Metrics (Australia)

Email : Pam.Morris@Totalmetrics.com **or visit** www.totalmetrics.com

Introduction

[1]ISO/IEC 14143-1 (2006) states that Functional Size Measurement sizes the Users Functional Requirements. (Note this is a later version of the standard than currently quoted by the IFPUG CPM)

*Ie "**Functional User Requirements***
a sub-set of the User Requirements. Requirements that describe what the software shall do, in terms of tasks and services.
NOTE: Functional User Requirements include but are not limited to:

- *data transfer (for example Input customer data, Send control signal);*
- *data transformation (for example Calculate bank interest, Derive average temperature);*
- *data storage (for example Store customer order, Record ambient temperature over time);*
- *data retrieval (for example List current employees, Retrieve aircraft position)."*

*Examples of User Requirements that are **not** Functional User Requirements include but are not limited to:*

- *quality constraints (for example usability, reliability, efficiency and portability);*
- *organizational constraints (for example locations for operation, target hardware and compliance to standards);*
- *environmental constraints (for example interoperability, security, privacy and safety);*
- *implementation constraints (for example development language, delivery schedule).*

However the restriction of what *can* be sized and what *cannot* be sized is limited to the piece of software that is within the scope of this measurement exercise. It excludes within the *same piece of software* the measurement of non-functional requirements. However it does not preclude the measurement of the software used to implement the non-functional requirements; it just precludes it from being within the same measurement of the software from the External User's View. If the non-functional requirements are implemented in software, then the way that the data within this software is seen is always from another

[1] ISO/IEC 14143-1 : 2006 Software and Systems Engineering — Software measurement — Functional size measurement Part 1: Definition of concepts

perspective (usually from the User view of the application not the external User) and therefore needs to be measured from that perspective.

We can consider these 'utility' applications used to deliver the non-functional requirements as being in a different software Layer. What is 'non-functional' to one layer of software is "functional' to another layer and can be counted within the 'viewpoint' of that layer.

The concept of Layers is generic to all functional size measurement methods and has been defined within the ISO/IEC 19761 COSMIC FSM Method (see [2]COSMIC functional size measurement methodology) as follows:

DEFINITION – Layer
A layer is a partition resulting from the functional partitioning of a software architecture which together with hardware forms a whole computer system where: • *layers are organized in a hierarchy that has only one layer at each level in the hierarchy;* • *there is a 'superior/subordinate' hierarchical dependency between the functional services provided by software in any two layers in the software architecture that exchange data directly;* • *the software in any two layers in the software architecture that exchange data interpret only part of that data identically.*

What we measure with Functional Size Measurement depends upon the purpose of the count. What is included within scope of the measurement activity will depend on how the resultant measurements will be used. Ie. If the project is required to build the non-functional requirements as software utilities and then the support teams is required to support them then they should be counted BUT not within the same count as the users application is counted from an External Business User view. These utilities are usually developed with different technologies and typically have different productivity rates for their development. Whilst the effort for their development can be aggregated with the effort to build the applications software their sizes ***cannot be aggregated*** and should always be recorded and reported separately. Only size within the same application layer (ie. Peer software) can have their sizes aggregated.

Examples of Software that is *Non-Functional* From the External Business Users viewpoint but *Functional* from the software applications viewpoint. Are Codes Data, Help, Security, Audit etc? The following examples give guidance to how these utilities can be counted.

1.1 Codes Data

If the Codes Data *application* is to be counted then we need to apply the same counting rules as would normally be applied to data maintained by business applications.

I.e. Data groups are identified based on the user view. In this case it will be an 'administrative user' who is required to maintain codes and descriptions. If the elementary process to perform this maintenance is identical for different code types i.e. the user can add, change and delete a code type and description, then irrespective of the number of different types of codes there is only one logical file, one set of maintenance functions (add, change and delete) and one Enquiry.

[2] Measurement Manual v2.3 – draft 2006

If some of these Codes/Codes tables store different attributes (e.g. Include a Value field and commencement and expiry dates), or if their maintenance invokes different business rules then they would be considered to be a different Internal Logical File, their maintenance and enquiry functions would also be considered to be unique.

Note: The Codes Data tables are NOT counted as FTRs for the elementary processes within the target application, which invoke enquiries on the tables since only logical files within 'peer applications' within the same functional layer may be counted as file types referenced.

Note that in the target application all kinds of drop down Codes MUST NOT be counted as EQs within the target application.

Note that the *Codes Data Logical File(s)* are considered to be within the *system* and the development project, which builds the *Codes Data Application* can include its functionality in the project count. If the user is delivered the functionality of the maintenance of the *Codes Data Logical File(s)* then the functionality delivered by this application can be included within the total functional size of the set of applications (ie. *system*).

1.2 Help Facility

The Help Facility is counted in a similar way to the Codes Data Utility. I.e.:

- it is considered to be in a different functional layer which provides services to the target application
- it is counted as a separate application
- a logical file is counted for each entity that has different data attributes and/or a unique set of maintenance and/ or enquiry functions.
- One set of maintenance and one enquiry function is counted for each logical file. Enquiry on different help text is not considered to be different enquiries.
- Help data files are not counted as file types referenced for elementary processes within an application in another layer.
- The Help Facility may service one or more applications in a superior layer.
- Functions within the Help Facility may only be counted as part of the Project scope if they are impacted by project changes. Note: Adding additional text records to the help message files is not considered to be an enhancement, only changes in data definitions are considered a change to the data file.

1.3 Error Message Facility

The Error Message Facility is counted identically to the Help Facility.

1.4 Security

The Security Facility would include Logon facilities and would be counted identically to the Help Facility.

1.5 Transaction Logging

Users often request that all access to their elementary processes be logged as well as a record of access to their system and files. This information can then be analyzed and reported. The tracking and logging

functionality of the Transaction Logging Facility is counted from the perspective of the administrative users i.e. auditors rather than view of the business users. Transactions are usually categorized and access is recorded generically depending on category. Count different types of logging as different elementary processes if they process different data or use different business rules.