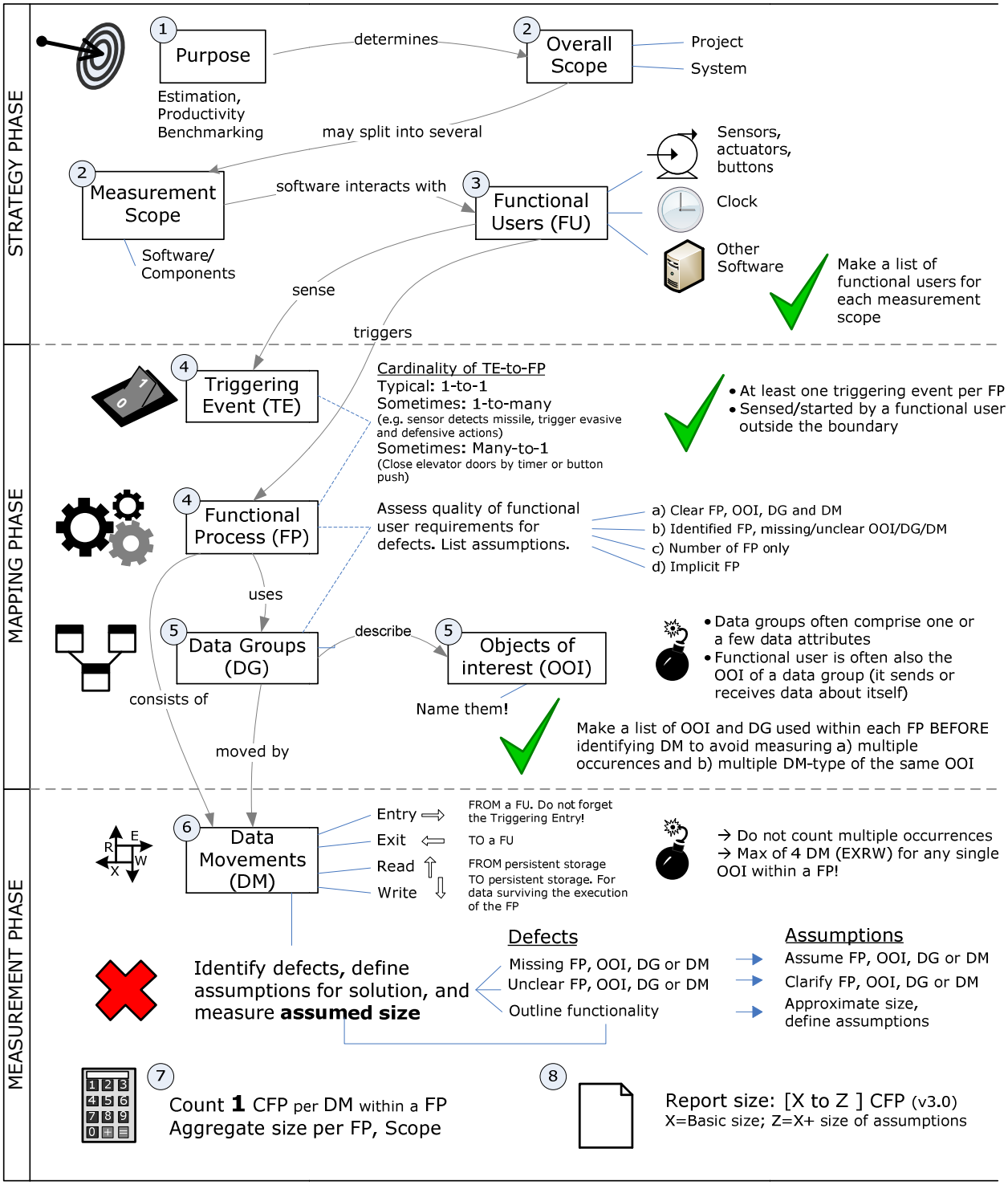
Quick Reference Guide to the COSMIC method for sizing Real-Time Application Software



Introduction

This Quick Reference Guide provides a simple 'recipe book' for the **main** steps when measuring real-time application software. It is based on the Measurement Manual, version 3.0.1.

References such as 'Ref. MM 2.1' refer to the Measurement Manual v3.0.1, section 2.1. The numbers in the diagram correspond with the numbers of the steps (in italics) below.

The Measurement Strategy phase

Aim: to define the parameters of the measurement so that the resulting size can always be understood and used correctly in the future. The parameters are the *purpose* and *scope* of the measurement, the *level of decomposition* of the software and the *level of granularity* of the Functional User Requirements (FUR).

- Finalizing all parameters of the Measurement Strategy may require iteration to ensure self-consistency
- Agree the parameters of the Measurement Strategy with the measurement sponsor, and record them
- Record the artifacts that will be used as the source of the FUR

1 Define the purpose of the measurement (Ref. MM 2.1)

Example: 'Measure the size from a statement of requirements for project estimating purposes'

2 Define the scope of the measurement (Ref. MM 2.2)

Identify the scopes of the individual pieces of software to be measured separately within the overall scope to be measured. Checks:

- the requirements are stated at the software level rather than at the system level (i.e. the requirements have explicitly been allocated to hardware or software)
- each separate piece of software to be measured exists wholly in one layer
- the levels of granularity of the FUR of the individual pieces of software are at the level of functional processes; if not, consider using an approximation variant of the COSMIC method
- the level of decomposition is the same for all pieces whose sizes must be compared

3 Identify the functional users (Ref. MM 2.3)

Examples of functional users are sensors, actuators, lamps, communications terminals, push buttons, a clock and other interfacing applications. Checks:

- all sources of triggering events have been identified
- all devices that must be polled and that must receive data have been identified
- if the software being measured must obtain data from or pass data to another piece of software, then the latter has been identified as a functional user

The Mapping phase

Aim: to define the functional processes, the data groups moved and the objects of interest described by the data groups, for each piece of software to be measured.

4 Identify triggering events and their functional processes (Ref. MM 3.2)

This is the most critical step in the whole measurement process. Be sure to:

- identify *types* of events, functional processes, etc, required in the FUR, *not occurrences*
- record all assumptions when the FUR to be measured are uncertain or appear to be incomplete

A triggering event:

- occurs outside the software; it is indivisible; it has either happened or not happened
- is detected by a sensor functional user (e.g. it senses reaching a pre-set temperature) or is generated by a functional user (e.g. a clock reaching a pre-set point in time, or another system reporting an event has occurred)
- results in the functional user sending a triggering Entry to start one functional process (possibly > 1 functional process, e.g. if an event triggers multiple functional processes to execute in parallel, as in an emergency)

A functional process:

- is triggered by receipt of a unique triggering Entry. Some functional processes can be triggered by > 1 event type (e.g. close elevator doors by timer or button push)
- is complete when the software has done all that is required to respond to the event

5 Identify objects of interest and data groups (Ref. MM 3.3)

A data group comprises all data attributes that describe a single object of interest. An object of interest is any 'thing' (physical or conceptual) in the real world *about which* the software is required to store or process data.

- In real-time software a data group often comprises one or a few data attributes, sent from an input device (e.g. a sensor) to software or from software to an output device (e.g. an actuator).
- In real-time software often a functional user is also the object of interest of a data group (it is sending or receiving data about itself). For instance, an indicator lamp (functional user) receives a 'lamp on' message (a data group consisting of one attribute).

The Measurement phase

Aim: to identify the data movement types (Entry, Exit, Read and Write) of each functional process and to add them up to calculate the functional size.

6 Identify the data movements (Ref. MM 4.1)

For each functional process, identify the movements of data groups from/to functional users (Entries/Exits) and from/to persistent storage (Reads/Writes). Checks:

- each functional process must have a triggering Entry, which may carry data about the event; there may be other Entries for other entered data groups
- if any entered data attributes required to be validated against persistent data, this implies a Read
- the FUR may state that, exceptionally, more than one data group of a given type must be moved in the same functional process (Ref. MM 4.1.7)
- one Write for a requirement to delete a data group must be identified

7 Measure the size (Ref. MM 4.3)

- The size of a functional process is the count of all its data movements of all types
- The size of a piece of software is the sum of the sizes of all its functional processes
- The size of a required change to a piece of software is the sum of all its added, modified and deleted data movements. A data movement is 'modified' if the data group it moves and/or the associated data manipulation is modified in any way. (Ref. MM 4.4)
- Determine the range of uncertainty in the size due to the assumptions made. See section 1.2 'Quality of the software artifacts' in the 'Guideline for Assuring the Accuracy of Measurements'

8 Report measurement results, archive documents (Ref. MM 5)

Check that the measurement data are correctly registered and that all relevant documents are present and archived. See section 1.3 'Quality of the measurement process' in the 'Guideline for Assuring the Accuracy of Measurements'.

References

Besides the Measurement Manual, relevant guidelines for sizing real-time application software are the

- Guideline for Sizing Real-time Application Software
- Guideline for Assuring the Accuracy of Measurements
- Guideline for Sizing Service Oriented Architecture software

All COSMIC documentation, including translations into other languages, is available for free download at the portal of www.cosmicon.com. Among the documentation available on this website are

- COSMIC method standards: Measurement Manual, Method Overview, Documentation Overview and Glossary, including translations into many languages
- Method Update Bulletins (MUB's, describe updates on the method that will be worked into the next version of relevant documents)
- Case studies
- Research and conference papers, conference announcements
- Measurement recording and reporting tools
- A list of known users of the COSMIC method
- Annual reports and Newsletters
- Notes on applications of functional sizing, such as effort estimation and benchmarking. For the latter a data collection form is available

Acknowledgements

The diagram on page 1 is derived from a diagram originally published by Sylvie Trudel and Alain Abran, included in Dr. Trudel's PhD thesis 'Using the COSMIC Functional Size Measurement method (ISO 19761) as a software requirements improvement mechanism', Montreal, Jan 9th 2012.