Software Development Velocity

with

COSMIC Function Points.

By: Alain Abran.

(April 19, 2019)

COSMIC Function Points provide an objective and standardized method of measuring a functional size of software. Using COSMIC Function Points, software developers can calculate objectively their own development velocity and use it for planning and work-in-progress monitoring.

Velocity can be calculated as the Effort over COSMIC Function Points and can be used to estimate the software functionality that can be delivered within a time period.

Example with:

a) a 1 week timeframe with 2 full time staff, each available 30 hours a week, and

b) an average velocity of 4 hours per COSMIC Function Points (4 hours per CFP).

This team can then deliver, on average within this timeframe, 15 CFP of software functions.

(Available hours ⁄ Average velocity = 60 hrs ⁄4 hours per CFP = 15 CFP).

This document presents a 'light' version of the COSMIC Function Points. For situations beyond basic measurement**,** that raise questions or that require conformity to ISO 19761, refer to the official COSMIC Measurement Manual with its full set of definitions and rules.

The Common Software Measurement International Consortium (COSMIC).

COSMIC is a voluntary, not-for-profit organization of software measurement experts from around the world, founded in 1998. All its publications are 'open' and available for free distribution subject to copyright and acknowledgement restrictions. For more on COSMIC and its organization, see www.cosmic-sizing.org .

Table of Contents

1. COSMIC Overview. 3
2. COSMIC Method. 3
2.1 COSMIC Principles. 3
2.2 Identifying functional users. 4
2.3 Identifying functional processes. 4
2.4 Identifying Data Groups. 4
2.5 Identifying data movements. 5
2.6 COSMIC unit of measurement. 7
3. Modifying functionality. 7
4. Other rules. 8
4.1 The standard level of measurement is the 'functional process'. 8
4.2 Data or groups of data not candidate for data movements. 8
4.3 Data movement uniqueness. 9
4.4 Data manipulation associated with data movements. 9
4.5 Documenting a COSMIC size measurement. 9

APPENDIX A: EXAMPLES OF COSMIC MEASUREMENT RESULTS 10
A) Example from the Rice Cooker case study. 10
B) Example from the Course Registration case study. 11
Examples of Data movements: Types versus occurrences. *..... 11*
Examples of: Modifying functionality. *..... 12*
Example of: Data or groups of data not candidates for data movements. *..... 12*
Examples of: Data movements uniqueness. *..... 12*

APPENDIX B: COSMIC DOCUMENTATION & GUIDANCE. 13

Copyright 2019. All Rights Reserved. Alain Abran. Permission to copy all or part of this material is granted provided that the copies are not made or distributed for commercial advantage and that the title of the publication, its version number, and its date are cited and notice is given that copying is by permission of Alain Abran. To copy otherwise requires specific permission.

Acknowledgement: This document reproduces, with permission of COSMIC copyrights, significant texts 'as is' or modified from: "The COSMIC Functional Size Measurement Method, Version 4.0.2 Measurement Manual, the COSMIC implementation guide for ISO/IEC 19761:2017", December 2017.

1. COSMIC OVERVIEW.

A software consists of a number of functional processes - see Figure 1.

- A functional process:
  - has one or more functional users, which are the senders to and/or receivers of data from the functional process.
  - consists of a number of 'data movements' of four kinds: 'Entry', 'Exit', 'Read' and 'Write' ('E', 'X', 'R' and 'W').
  - can have many E, X, R and W data movements.
  - each data movement is measured as one 'COSMIC Function Point' (1 CFP).
- When a functional user:
  - sends a data group to the functional process, one 'Entry' is identified.
  - receives a data group from the functional process, one 'Exit' is identified.
- When the functional process:
  - stores a data group in persistent storage, one 'Write' is identified.
  - retrieves a data group from persistent storage, one 'Read' is identified.

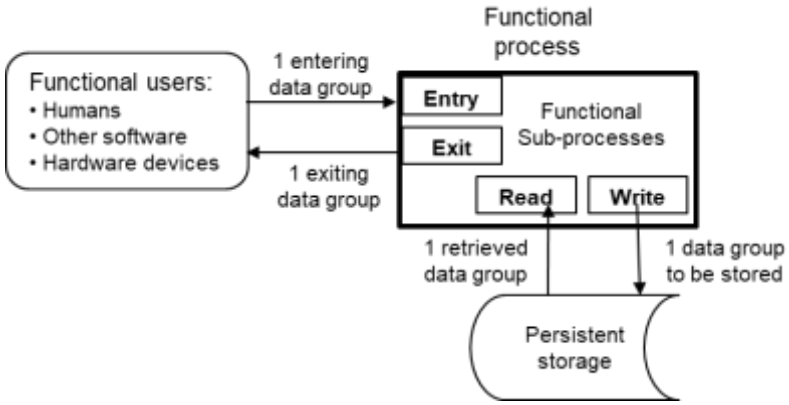

Figure 1: COSMIC Overview: four kinds of data movement, each moving 1 data group.

2. COSMIC METHOD.

2.1 COSMIC Principles.

a) A piece of software interacts with its functional users and with persistent storage.
b) Functional user requirements of a piece of software to be measured can be mapped to a set of unique functional processes.
c) Each functional process consists of sub-processes.
d) A sub-process may be either a data movement or a data manipulation.
e) A data movement moves a single data group.
f) The are four kinds of data movement: Entry, Exit, Read and Write.
g) COSMIC distinguishes between types and occurrences. (For ease of reading, we normally omit 'type'.)
h) The size of a functional process is equal to the addition of its data movements.
i) A functional process shall include at least the triggering Entry data movement and either a Write or an Exit data movement, i.e. a minimum size of two (2) data movements.
j) There is no upper limit to the number of data movements in a functional process and hence no upper limit to its size.
k) Data manipulation is not separately measured; the functionality of any data manipulation is assumed to be accounted for by the data movement with which it is associated. (See section 'Data manipulations associated to data movements'.)

Note. When a functional process is not concerned with where the data is stored to or retrieved from, the term ‘persistent data’ is used.

2.2 Identifying functional users.

DEFINITION – FUNCTIONAL USER.
A functional user of the piece of software is a sender and/or a recipient of data processed by that software.

The identification of functional users usually depends on the software domain:

- For business application software, the functional users are usually humans, plus other software with which the business application interfaces.
- For real-time software, the functional users would usually be hardware devices or other software that interact directly with the real-time software.

2.3 Identifying functional processes.

DEFINITION – FUNCTIONAL PROCESS.
a) A set of data movements that is unique within the software being measured and that can be defined independently of any other functional process. b) A functional process shall have only one triggering Entry. Each functional process starts processing on receipt of a data group moved by its triggering Entry. c) The set of all data movements of a functional process is the set that is needed for all the possible responses to its triggering Entry. Note: It is an *occurrence* of a functional process that starts *executing* on receipt of an *occurrence* of a data group moved by an *occurrence* of a triggering Entry.

Identifying functional processes starts with identifying the following:

‘a functional user – (moves a) data group – to initiate a functional process’.

2.4 Identifying Data Groups.

DEFINITION – DATA GROUP.
Any set of data attributes describing one ‘thing’ in the world of a functional user. It may be any physical thing, as well as any conceptual object or part of a conceptual object. Note: a piece of software may have formed new data groups from any combinations of data attributes, as required by different functional processes.

DEFINITION – DATA ATTRIBUTE.
The smallest parcel of information, within an identified data group, carrying a meaning from the perspective of the functional user of the software.

In practice, a data group can take many forms:

Copyright © 2018

a) a physical record structure on a hardware storage device (file, database table, ROM memory, etc.).

b) a physical structure within the computer's volatile memory (data structure allocated dynamically or through a pre-allocated block of memory space).

c) the clustered presentation of functionally-related data attributes on an input/output device (display screen, printed report, control panel display, etc.).

d) a message or signal between a device and a computer, or over a network, etc.

RULE - IDENTIFYING DIFFERENT DATA GROUPS MOVED IN THE SAME FUNCTIONAL PROCESS.
a) For all the data attributes appearing in the input of a functional process, sets of data attributes that describe the same 'thing' and have the same frequency of occurrence belong to the same data group type and are moved by the same one Entry. b) Sets of data attributes that have different frequencies of occurrence define different things and therefore belong to different data group types which are moved by different Entries. Note: This rule applies for all the data attributes appearing in all data movements kinds (Entries, Exits, Reads and Writes).

2.5 Identifying data movements.

This step consists in identifying the data movement kinds (Entry, Exit, Read and Write) of each functional process.

DEFINITION – DATA MOVEMENT.
A base functional component which moves a single data group. Four kinds of data movement: • Entry (E). A data movement that moves a data group from a functional user into the functional process where it is required. • Exit (X). A data movement that moves a data group from a functional process to the functional user that requires it. • Read (R). A data movement that moves a data group from persistent storage into the functional process which requires it. • Write (W). A data movement that moves a data group from inside a functional process to persistent storage.

2.5.1 Identifying Entries (E).

An Entry data movement must comply with the following rules:

RULES – ENTRY (E).
a) If the input to a functional process comprises more than one data group, identify one Entry for each unique data group in the input. b) The data group of a triggering Entry may consist of only one data attribute, or may include many data attributes. c) A clock-tick that initiates a functional process shall always be associated with an Entry.

Copyright © 2018

Note 1: for example, a clock-tick occurring every 3 seconds shall be associated with an Entry moving a data group of one data attribute.

Note 2: It makes no difference whether the clock-tick is generated periodically by hardware or by another piece of software.

d) Unless a specific functional process is necessary, obtaining the date and/or time from the system's clock shall not be considered to cause an Entry or any other data movement.

2.5.2 Identifying Exits (X).

An Exit data movement must comply with the following rules:

RULE – EXIT (X).
If the output of a functional process comprises more than one data group, identify one Exit for each unique data group in the output.

DEFINITION – ERROR/CONFIRMATION MESSAGE.
An Exit issued by a functional process to a human functional user that either confirms only that entered data has been accepted, or only that there is an error in the entered data.

RULES – ERROR/CONFIRMATION MESSAGES & OTHER INDICATIONS OF ERROR CONDITIONS.
a) One Exit shall be identified to account for all types of error/confirmation messages issued by any one functional process of the software being measured from all possible causes. E.g. successes or failures of validation of entered data or for a call to retrieve data or to make data persistent, or for the response from a service requested of another piece of software. b) If a message to a human functional user provides data in addition to confirming that entered data has been accepted, or that entered data is in error, then this additional data should be identified as a data group moved by an Exit.

2.5.3 Identifying Reads (R).

A Read data movement must comply with the following rules:

RULES – Read (R).
a) Identify a Read when the software must retrieve a data group from persistent storage. b) If the functional process must retrieve more than one data group from persistent storage, identify one Read for each unique data group that is retrieved. c) A Read data movement always includes any 'request to Read' functionality (so a separate data movement shall never be counted for any 'request to Read' functionality).

2.5.4 Identifying Writes (W).

A Write data movement must comply with the following rules:

Copyright © 2018

RULES – WRITE (W).
a) Identify a Write when the software must move a data group to persistent storage. b) If the functional process must move more than one data group to persistent storage, identify one Write for each unique data group that is moved to persistent storage. c) A requirement to delete a data group from persistent storage shall be measured as a single Write data movement. d) Do not identify a Write when: • Creation or update of variables or intermediate results that are internal to the functional process. • Storage of data by a functional process resulting only from the implementation, rather than from the functional requirements. (Example: the storage of data temporarily during a large sort process in a batch-processed job.)

2.6 COSMIC unit of measurement.

DEFINITION – COSMIC UNIT OF MEASUREMENT.
1 CFP (Cosmic Function Point), which is the size of one data movement.

PRINCIPLE – THE COSMIC MEASUREMENT PRINCIPLE.
a) The size of a functional process is equal to the number of its data movements. b) The functional size of a piece of software is equal to the sum of the sizes of its functional processes.

3. MODIFYING FUNCTIONALITY.

Any data movement (E, X, R and W) involves two kinds of functionality: it moves a single data group and it has some associated data manipulation. For a *change* to data manipulation, determine which type of data manipulation is associated with which type of data movement and consider as a modified Data Movement.

DEFINITION – MODIFICATION (OF THE FUNCTIONALITY OF A DATA MOVEMENT).
a) A data movement is considered to be functionally modified if at least one of the following applies: • the data group moved is modified, • the associated data manipulation is modified. b) A data group is modified if at least one of the following applies: • one or more new attributes are added to the data group, • one or more existing attributes are removed from the data group, • one or more existing attributes are modified, e.g. in meaning or format (but not in their values). c) A data manipulation is modified if it is functionally changed in any way.

Copyright © 2018

RULES – MODIFYING A DATA MOVEMENT.
a) If a data movement must be modified due to a change of the data manipulation associated with the data movement and/or due to a change in the number or type of the attributes in the data group moved, one changed CFP shall be measured, regardless of the actual number of modifications in the one data movement. b) If a data group must be modified, data movements moving this group whose functionality is not affected by this modification shall not be identified as changed movements. Note: The minimum size of a change to a piece of software is 1 CFP.

RULES – AGGREGATING SIZES OF CHANGES.
a) For any functional process: Size (Change(functional process)) = Σ size (added data movements) + Σ size (modified data movements) + Σ size (deleted data movements). b) The sizes of pieces of software shall be added together only if it makes sense to do so for the purpose of the measurement.

4. OTHER RULES.

4.1 The standard level of measurement is the 'functional process'.

DEFINITION - FUNCTIONAL PROCESS LEVEL OF MEASUREMENT.
The level of the description of a piece of software at which: • its functional user (-types) are individual humans or engineered devices or pieces of software (and not any groups of these) AND • single functional processes can be identified (and not any groups of these).

4.2 Data or groups of data not candidate for data movements.

Any data appearing on input or output screens or reports that are not related to a functional user should not be identified as indicating a data group, so should not be measured.

When identifying Entries in a screen that enables human functional users to input data into functional processes, analyze only screens that are filled with data:

Ignore any screen that is formatted but otherwise 'blank' except for possible default values, and ignore all fields and other headings that enable human users to understand the input data required.

Copyright © 2018

4.3 Data movement uniqueness.

RULES – DATA MOVEMENT UNIQUENESS AND POSSIBLE EXCEPTIONS.
a) All data describing the same data group type that is required to be entered into one functional process shall be identified as one data group moved by one Entry. Note 1: A functional process may, of course, have multiple Entries, each moving a different data group type. Note 2: The same equivalent rule applies to any Exit, Read ir Write data movement in any one functional process.
b) Repeated occurrences of any data movement type when it is being executed shall not be counted. Note: This applies even if multiple occurrences of the data movement type differ in their execution because different values of the data attributes of the data group moved result in different processing paths being followed through the functional process type.

4.4 Data manipulation associated with data movements.

RULES – DATA MANIPULATION ASSOCIATED WITH DATA MOVEMENTS.
a) An Entry accounts for all data manipulation to enable a data group to be entered by a functional user (e.g. formatting and presentation manipulations) and to be validated.
b) An Exit accounts for all data manipulation to create the data group attributes to be output and/or to enable the data group to be output (e.g. formatting and presentation manipulations) and to be routed to the intended functional user.
c) A Read accounts for all computation and/or logical processing needed in order to retrieve a data group from persistent storage.
d) A Write accounts for all computation and/or logical processing to create or to update a data group to be written, or to delete a data group.
e) The data manipulation associated with any of these data movements does not include any data manipulation that is needed after the data movement has been successfully completed, nor does it include any data manipulation associated with any other data movement.

4.5 Documenting a COSMIC size measurement.

A table can be used to record the results of a measurement for each identified functional process – see Appendix A. The Knowledge base of www.cosmic-sizing.org has a number of spreadsheet tools for measurement recording. – see https://cosmic-sizing.org/cosmic-publications/overview/

Copyright © 2018

Appendix A: Examples of COSMIC Measurement results.

A) Example from the Rice Cooker case study.

Functional Requirement 1:

On receipt of the Start signal, the software: Rice Cooker Case

a) sends a 'Turn ON' signal to the Cooking Lamp.

b) sends a 'Turn ON' signal to the heater.

The functional size of this requirement is measured as follows with COSMIC:

Functional user	Data Movement	Data Group moved	Data Movement Kind	CFP
Start button	Receive Start signal	Start signal	E (Entry)	1
Heater	Send a Turn ON command to the Heater	Heater command	X (eXit)	1
Cooking Lamp	Send a Turn ON command to the Cooking lamp	Cooking lamp command	X (eXit)	1
			Size of Functional Requirement 1 = 3 CFP	

Functional Requirement 2:

On receipt of a 30-second signal:

Starting at t = 0, and at each following 30-second interval, the software:

a) receives the 30-second signal;

b) receives the elapsed time signal;

c) gets the cooking mode from persistent storage;

d) selects a new target temperature, for the cooking mode by reading it in a table in persistent storage.

e) puts this new target temperature into persistent storage, which becomes the current target temperature until the next 30-second signal.

The functional size of this requirement is measured as follows with COSMIC:

Functional user	Data Movement	Data Group moved	Data Movement Kind	CFP
Timer	Receive 30-second signal	30-second signal	E (Entry)	1
Timer	Receive Current elapsed time signal	Current elapsed time signal	E (Entry)	1
	Get cooking mode	Cooking mode	R (Read)	1
	Get New target temperature from table	New target temperature	R (Read)	1
	Store the (new) Current target temperature	Current target temperature	W (Write)	1
			Size of Functional Requirement 2 = 5 CFP	

Copyright © 2018

B) Example from the Course Registration case study.

Functional Requirement:

Enquire on a Professor's details. - Course Registration System Case

a) When the Registrar enquires on the details of a Professor, he must first select the sub-option 'Enquire on a Professor' and he must enter a Professor ID.

b) The software searches for a Professor with the specified ID and displays the Professor's name and address and other details.

c) Alternatively, if a Professor with the specified ID is not found, the software displays an error message, "Professor Not Found". The Registrar can then type in a different ID or cancel the operation.

The functional size of this requirement is measured as follows with COSMIC:

Functional Requirement	Functional user	Data Movement	Data Group moved	Data Movement Kind	CFP
Enquire on a Professor's details	Registrar	The Registrar enters Professor ID	Professor ID	E (Entry)	1
		The software retrieves the Professor details	Professor details	R (Read)	1
	Registrar	The software displays the Professor details	Professor details	X (eXit)	1
	Registrar	Display error message	Error Messages	X (eXit)	1
Size of Functional Requirement = 4 CFP.					

Examples of Data movements: Types versus occurrences.

EXAMPLE 1:

A tire-pressure system monitors the pressure of each of the four tires of a car. Each wheel has a sensor which obtains the pressure of its tire. Once per second each sensor sends its status (e.g., the sensor id for each wheel, and tire pressure) to the software.

The four sensor functional users are of the same type *(e.g., 'Tire pressure sensor') and represent four* occurrences (e.g., 1 sensor type and 4 occurrences, one for each of the 4 tires).

EXAMPLE 2: Navigation, Control commands & Application-general data.

The following do not involve any data group, hence shall not be considered as data movement or manipulation:

- *Commands to 'page up/down' or between physical screens.*
- *Hitting a Tab or Enter key, or pressing a button to continue.*
- *Clicking on an 'OK' button to confirm or cancel a previous action, or to acknowledge an error message or to confirm some entered data, etc.*
- *Functions that enable a user to control the display (or not) of a header or of sub-totals that have been calculated.*
- *Menu commands that enable a user to navigate to one or more specific functional processes but which do not themselves initiate any one functional process.*
- *Commands to display a blank screen for data entry.*

Copyright © 2018

- Movement or manipulation of constants or variables which are internal to the functional process and/or that can be changed only by a programmer.
- Computation of intermediate results in a calculation.
- Data stored by a functional process resulting only from the implementation, rather than from the functional requirements.
- *Movement or manipulation of screen titles, screen colour, etc.*
  - When the screen colour for all screens is changed, this change should not be measured.
  - Change the calculation, the specific formatting, presentation, and/or validation of the data. 'Presentation' can mean the font, background colour, field length, field heading, number of decimal places, etc.

Examples of: Modifying functionality.

EXAMPLE 3: A change in a piece of software consists of:

A. add one new functional process of size 6 CFP, and

B. in another functional process add one data movement (1 CFP), make modifications to three other data movements (3CFP) and delete two data movements (2 CFP).

Total size of the change in these 2 functional processes = 6 + 1 + 3 + 2 = 12 CFP.

Note: Changes to control commands and application-general data should not be measured.

EXAMPLE 4: The change is: addition or modification of the attributes of a data group D_1, such that after modification it becomes D_2.

In the functional process where this modification is done, all data movements affected by the modification should be identified and measured as modified. So, if the changed data group D_2 is in both a Write and an Exit data movement in this functional process:

- identify one changed Write, and/or
- one changed Exit data movement respectively.

Size of the change = 2 CFP.

Example of: Data or groups of data not candidates for data movements.

EXAMPLE 5: 'Application-general data' such as headers and footers (company name, application name, system date, etc.) appearing on all screens.

Examples of: Data movements uniqueness.

EXAMPLE 6:

In a 'Create Employee' functional process, a name, address and city must be entered in a first screen and, in a subsequent screen, the date of birth and the email address.

Identify one Entry (1 CFP).

Copyright © 2018

APPENDIX B: COSMIC DOCUMENTATION & GUIDANCE.

Most of the COSMIC documents listed below, including translations, are included in COSMIC Knowledge Base at https://cosmic-sizing.org/cosmic-publications/overview/

- References 2 to 15: Guidelines for various software domains and related topics
- References 16 to 23: Case Studies

[1] Measurement Manual
[2] Introduction to the COSMIC Method of measuring software
[3] What is a COSMIC Function Point
[4] Guideline for Sizing Real-time Software
[5] Guideline for 'Measurement Strategy Patterns
[6] Guideline for early or rapid COSMIC functional size measurement
[7] Guideline for Sizing Business Application Software
[8] Guideline for sizing Data Warehouse Application Software
[9] Guideline for Sizing Service-Oriented Architecture Software
[10] Guideline: introducing COSMIC size measurement to help manage Agile software activities
[11] Guideline on Non-Functional & Project Requirements
[12] Quick Reference Guide to the COSMIC method for sizing Business Application Software
[13] Quick Reference Guide to the COSMIC method for sizing Real-Time Application Software
[14] Guideline on the accuracy of COSMIC Function Points
[15] Guideline on how to convert 'First Generation' Function Point sizes to COSMIC sizes
[16] ACME Car Hire Case
[17] Automatic Line Switching Case
[18] Course Registration System Case
[19] Industrial Automation (Robot) Case
[20] Web and Mobile Applications for Restaurant System Case (not on COSMIC Sizing)
[21] Rice Cooker Case
[22] Valve Control Case
[23] Web Advice Module Case
[24] ISO/IEC 14143/1:2011 Information technology – software measurement – functional size measurement. Part 1 Definition of concepts, International Organization for Standardization – ISO, Geneva.

Copyright © 2018