



COSMIC In a Nutshell

Part 1 - Fundamentals

1 COSMIC in a Nutshell - Part 1 - Fundamentals

Introduction

The COSMIC Method is defined in the COSMIC Measurement Manual 4.2 (MM). The MM contains the Principles, Definitions, Rules, Guidance, Notes, Examples and the Measurement Process and is the ultimate reference for the methods.

While that document is necessary as a reference it is difficult for the lay-reader easily to extract specific information.

The COSMIC in a Nutshell (CIAN) documentation set is designed to distil the information into discreet areas of interest so enabling the reader to home in on the information that is important for the purpose at hand.

Scope

The scope of this part is explain the elements of the COSMC model of software and to describe how those elements are used to derive a functional Size

.

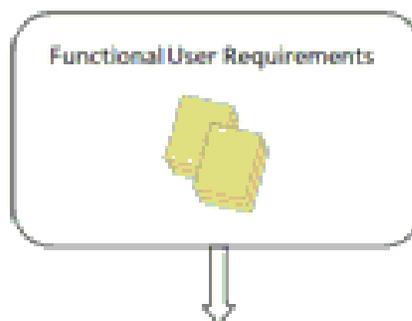
Audience

This document addresses the fundamentals of the COSMIC method and is suitable for those needing a quick reference to the essentials of the method

2 The COSMIC Model

Introduction

The elements of the COSMIC Model and their relationships are illustrated in Fig 1 below



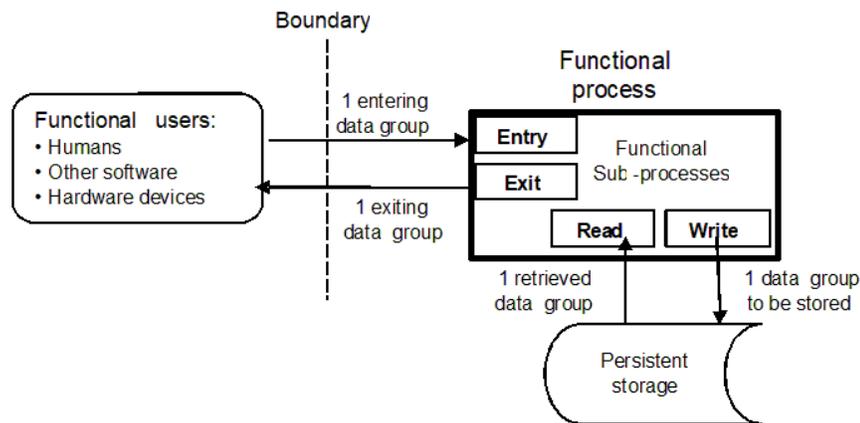


Fig 1 - The COSMIC Model

Explanation of the Elements of the COSMIC Model

- A Piece of Software (the software) is the Software being measured and is defined by its Functional User Requirements (FUR).
- A Functional User (FU) is anyone or anything that interacts with the software by providing information to it or receiving information from it across the Boundary.
- The Boundary is conceptual and separates the software from the users interacting with it.
- The software comprises one or more Functional Processes (FP) each defined by its own FUR.
- A Functional Process (FP) is a set of unique Data Movements.
- A Data Movement (DM) moves a Data Group (DG) to/from a FU or to/from Persistent Storage (see below).
- A Data Group is a unique set of attributes describing something the software processes. The word “something” is assigned the term Object of Interest (OOI). Although not needed for measurement knowledge of the attributes determine whether a DG is unique for the same OOI.
- The uniqueness of a DG is determined by the uniqueness of its collection attributes in the same OOI.
- There are four types of DM
 - Entry DM moves a DG provided to the Software **from** a FU across the Boundary
 - Exit DM moves moves a DG provided by the Software **to** a FU across the Boundary
 - Read DM moves a DG retrieved by the FP **from** Persistent Storage

- Write DM moves a DG to Persistent Storage by the FP

Persistent Storage

Persistent Storage (PS) is a set of Data Groups saved by a FP by a Write DM during execution and which persists when that FP terminates.

Functional Process Characteristics

- The FP is the smallest elementary part of the FUR that can be the subject of a measurement. The outcome of measuring a FP is its Functional Size.
- A FP It is triggered when a Functional User (FU) detects an Event to which the software must respond by providing an Entry DM (called the **Triggering Entry (TE)**). The FP is complete when it has satisfied its FUR.

Note that in certain domains, the Event may not be explicit and for all practical purpose can be assumed to have occurred, in other domains it is essential to identify the Event. E.g.

In Data Rich a Customer may require a new account, the operator responds by entering the details triggering a FP. Implicitly the Customer request is the Event, but for all practical purpose this can be ignored.

In Real-Time – Event driven (as the name implies) the Event is explicit and essential for understanding the operation of the software so must be identified.

- Once the FP has been triggered by its TE it may perform any number of additional DM Types in any sequence
- The FP must have a TE plus at least 1 Read DM or Exit DM. the minimum size of a FP is therefore 2. There is no maximum limit
- Each subsequent DM contributes to the size therefore there is no upper limit to the size.

Functional Size

- The Functional Size of a FP is the sum of its DMs.
- The Functional Size of a Piece of Software is the sum of the sizes of all the FPs.