2014

Simple Function Point Functional Size Measurement Method

Reference Manual

SiFP-01.00-RM-EN-01.01

TABLE OF CONTENTS

Rights of use

Versions

SiFP-01.00-RM-EN-01.01, 2014 March

0 Introduction

0.1 Information on the SiFP Method

0.1.1 Goals of the Method

The method called Simple Function Point (SiFP) is a Functional Size Measurement Method for software sizing (FSMM - according to ISO/IEC standards) that assigns a numeric value to a software application by identifying, classifying and weighting the User Functional Requirements that characterize it. However, the method does not consider the Non-Functional software Requirements and specifically the Quality Requirements and Technical Requirements.

0.1.2 Characteristics of the Method

Organizations interested in functional measurement of software applications require fast, agile, lightweight, easy to use measuring methods with low impact on production processes, which do not require very specialized skills, that are easy to learn, provide reliable results, are not dependent on the opinions of those who perform the measurements, on the technologies used and technical design principles, and which are adequately correlated to effort, cost, duration and staffing of a project. The Simple Function Points method helps to achieve all these goals.

0.2 Information about the SiFP Method Reference Manual

0.2.1 Goals of the Manual

This document aims to describe the Simple Function Point Measurement method for software called SiFP (Simple Function Point). The document has been drawn up in accordance with the ISO/IEC 14143-1:2007 standard.

The main objectives of the reference manual are:

- to provide a clear and detailed description of the Simple Function Point measurement method;
- to promote consistent and homogeneous application of the rules of the SiFP method by different practitioners using it.

0.2.2 Manual Features

The structure and contents of this document are based on the following criteria:

- *ISO/IEC 14143 compliance*: consistency with the provisions of the standard ISO/IEC 14143-1:2007 standard "Information technology – Software measurement – Functional size measurement Part 1: Definition of concepts";
- *Self consistency*: the presentation of the method is thorough and self consistent. This means that it is not necessary to read other document sources to understand its terms, definitions, concepts and measurement process;
- *Simplicity*: the document is easy to read, with a direct style and particular attention to the concision and agility.

0.2.3 STRUCTURE OF THE MANUAL

The manual is divided into three parts:

- Part 0 - Introduction
- Part 1 - Description of the method
- Part 2 - Glossary of terms and License Agreement.

Part 0 - Introduction

This section contains general information about the method and the reference manual.

Part 1 – Description of the method

This section contains a description of the SiFP method, the principles on which it is based, its scope, the model of the software on which the measurement is based, the description of basic types of Base Functional Components (BFC), the operating procedure for measurement, the assignment function and aggregation of the elementary values, the measurement process more in general, the documentation standards for the measurement and the convertibility of the method.

Part 2 – Glossary of Terms and License Agreement

This section contains the reference glossary for the software measurement activity and the user license for the SiFP method.

Due to higher update rate, a separate document shows Examples and Case Studies. Specific interpretation guidelines for the method for different contexts of software production will be issued along with the Reference Manual of the method.

0.3 VERSION MANAGEMENT RULES

The Method and the Manual have independent version numbers, since, with the same method version it may be necessary to make different versions of the Reference Manual for purely editorial reasons (correction of grammatical errors, style variations, usability improvements, reorganization of content, etc.) or for representation improvements (adding examples, changes to descriptions, adding additional diagrams, etc.) which however do not alter rules, procedures and application results of the method itself. For example, the SiFP method version 01.02 method could be described in the manuals with version numbers 01.00, 01.01 01.02, 01.03, 02.01. Therefore the identifier of the document that describes a particular version of the method has the form: SiFP-XX.XX-RM-YY.YY. For example, the code SiFP-01.01-RM-02.01 refers to the Reference Manual version 02.01 that describes version 01.01 of the method. The first pair of numbers in the code XX.XX or YY.YY indicates a version with significant changes while the second pair indicates minor amendments. Whenever the version of the method rises, the version number of the manual that describes it, will be resetted and will start again from 01.00.

1 DESCRIPTION OF THE METHOD

1.1 GENERAL CONCEPTS

In accordance with the orientation of the ISO/IEC 14143-1:2007 international standard, user requirements, related to a software application, can be divided into three main classes: Functional Requirements, Technical Requirements and Quality Requirements. The second and third are also known as Non-Functional Requirements. Functional measurements of software (FSM – Functional Size Measurements), to which Simple Function Point belong, are related exclusively to the first of the three categories.

The purpose of Simple Function Point is to provide an objective measurement of the quantity of functions that a software application offers to its users (humans and/or other software systems) by quantifying "what"it makes possible, in terms of available data and operations on it.

A fundamental component of a functional software measurement method is the concept of Base Functional Component (BFC). The ISO/IEC 14143-1:2007 standard defines the BFC as "elementary unit of Functional User Requirements...". The term "elementary" is, in this context, a synonym of "atomic" — in the original sense of the philosophical term - meaning that "it can not be further decomposed ". Actually, just as an atom of matter is made up of subatomic particles, a BFC may still be splitted into components that are however used only for the purpose of identifying internal complexity. In the Simple Function Point method, sub components (such as the data elements that make up a logical store) are not identified specifically nor used in the measuring function. The BFC is the elementary entity to which numeric values are attributed, based on the measurement function.

By analyzing the different types of functional requirements it becomes clear that there are three basic categories: the requirements that represent data flows or movements, those that represent data processing rules and those relating to permanent data storage. The first two categories are generally interlinked as there may be features that move data without changing it while the opposite never occurs, i.e. processing that produces data that are not in any way moved (input or output or into / from permanent storage). The SiFP method identifies and gives a value to only the first and the third category of user requirements: logical transactions that move data (logical flows) and the logical stores that keep it for an undefined period, but higher than the duration of a transaction that creates or uses the assigned data. The category of logical processes (algorithms - processes - transformations etc.) is not expressly measured but is considered for the assessment of identity of two BFC candidates.

The SiFP method adopts the assumption that the functional value of a software object[1] is proportional only to the number of types of logical transaction and types of logical stores required and not to their internal structure in terms of component data processed or different type (e.g. depending on the primary intent of processing). In essence, the principle for which computational BFC may have greater or lesser functional value by virtue of some attribute is recognized as valid (e.g. the complexity of used algorithms) but it is not granted that it can depend significantly on the number of elementary data types processed or on their internal aggregations that in SiFP are not therefore explicitly identified and measured.

[1] In this document, the term software "object" is used generically to mean a computer program accompanied by documentation that describes it. There is no specific reference to Object Oriented Programming.

1.2 SCOPE OF APPLICATION

The SiFP method applies to a wide range of application domains as defined in the ISO/IEC TR 14143-5:2004 Technical Report. In particular, it applies to all those described, as examples, in table A.3 of that document, shown here to facilitate retrieval of that information.

Table A.3 — Functional Domains

Functional Domain	Control- and Communication- Rich	Data-Rich	Manipulation- and Algorithm-Rich
Pure Data Handling System	negligible	dominant	negligible
Information System	negligible	dominant	present
Data Processing System	negligible	present	present
Controlling Information System	present	dominant	negligible
Controlling Data System	present	present	negligible
Complex Controlling Information System	present	dominant	present
Non-Specific (Complex) System	present	present	present
Simple Control System	dominant	negligible	negligible
Control System	present	negligible	present
Complex Control System	dominant	negligible	present
Data Driven Control System	dominant	present	negligible
Complex Data Driven Control System	dominant	present	present
Pure Calculation System	negligible	negligible	dominant
Controlling Calculation System	present	negligible	dominant
Scientific Information System	negligible	present	dominant
Scientific Controlling Data Processing System	present	present	dominant
NOTE This table provides examples of functional domains and is not intended to be an all-inclusive list.			

1.3 NORMATIVE AND AUXILIARY REFERENCES

To develop this version of the Reference Manual the following documents have been used:

- ISO/IEC 14143-1:2007 "Information technology — Software measurement — Functional size measurement Part 1: Definition of concepts"
- ISO/IEC TR 14143-5:2004 "Information technology — Software measurement — Functional size measurement Part 5: Determination of functional domains for use with functional size measurement"
- DPO Srl, "Research and Development Report: Simple Function Point - A new Functional Size Measurement Method fully compatible with IFPUG FP©" , 2011
- International Software Benchmarking Standard Group, Wordwide Software Development: The Benchmark, Release 11, 2009

1.4 The Software Model on which the Measurement is Based

To adequately describe the measuring function, we must firstly identify the software object on which it operates. A software object is, generally, the result of planning, design, implementation, testing and maintenance. In each of these phases the software object is at a progressively more complete state of progress. Frequiently the process starts out with a very brief and coarse description of the requirements, sometimes as a result of a feasibility study, then it progresses through functional and technical design to implementation, testing and commissioning of the software and then its maintenance begins. The main activities are therefore classified as: New development and Maintenance.

1.4.1 (Custom) Software Development

By "Software Development" we intend:

- the development of whole new application systems, or autonomous parts of the same (in the case that the development process is divided in lots), which address specific needs in terms of functionalities not yet computationally acquitted;
- complete overhaul of the application systems, the functions of which are not satisfied with the required features after an evaluation that it is not convenient to implement a Functional Enhancement Maintenance on existing software.

1.4.2 (Custom) Software Maintenance

By"Software Maintenance" we intend the set of activities needed to preserve the correspondance between the functional, performance and quality requirements desired by the user, in a general sense, for some existing information system and the various elements or logical and physical components - programs, database and documentation - that make the information system usable.

Maintenance can be classified into four main classes:

- Functional Enhancement Maintenance (FEM)
- Non-functional Enhancement Maintenance (NFEM)
- Corrective Maintenance
- Adaptive Maintenance

- **Functional Enhancement Maintenance** is defined as a set of activities aimed at creating new features or to modify or delete existing ones, in relation to pre- existing software application.
- **Non-functional Enhancement Maintenance** is defined as a set of activities aimed at preserving or improving the efficiency of existing programs and procedures to changing operational conditions of use and workloads or , more generally , to improve non-functional performance (usability , maintainability etc. .).
- **Corrective Maintenance** is defined as a set of activities aimed at removing the causes and effects of malfunctions of computer procedures and software programs.
- **Adaptive Maintenance** is defined as a set of activities aimed at ensuring the continued compliance of procedures and software programs to evolving technological environment of the integrated information system .

Simple Function Point apply directly to the software developed from scratch or to the software "changed" by a Functional Enhancement Maintenance or to the software asset[2]. For other types of maintenance, that are not involved with a change in the functional requirements, it is not possible to "calculate" a functional measure of the product impacted by the maintenance, but in many cases the functional measure of the "software asset" can be of help to the government of the production processes.

Simple Function Point is a product measurement method, not a process measurement method, therefore:

- the "measurement of the development project" is meant as a "measurement of the software features provided by development project"
- the "measurement of functional enhancement maintenance project" is meant as "measurement of the software features provided by the functional enhancement maintenance project (FEM)"
- the "measurement of application" is meant as the "measurement of the software features that are available after the initial development and the subsequent FEM activities". It is an "asset" type value.

The following sections introduce some important concepts needed for the application of the SiFP method. Section 1.4.3 presents a generic "transparent box" model ot the software seen from the point of view of the processing and storage of data as required by any FSMM; section 1.4.4, instead, presents the concept of Measurable Software Application (MSA) which is the actual measured object of each functional measurement exercise; section 1.4.5 introduces the concept of layer while sections 1.4.6, 1.4.7 and 1.4.8 allow to understand how this concept influences the way of measuring functional size; finally section 1.4.9 summarizes these points.

[2] "Software asset" refers to the set of software applications developed and available to an organization for use in an enterprise or institutional information system.

1.4.3 A GENERIC SOFTWARE MODEL

The following illustration shows a simple representation of what has been described above, highlighting the components related to the functional requirements of "moving" data, "processing" data and data "storage".

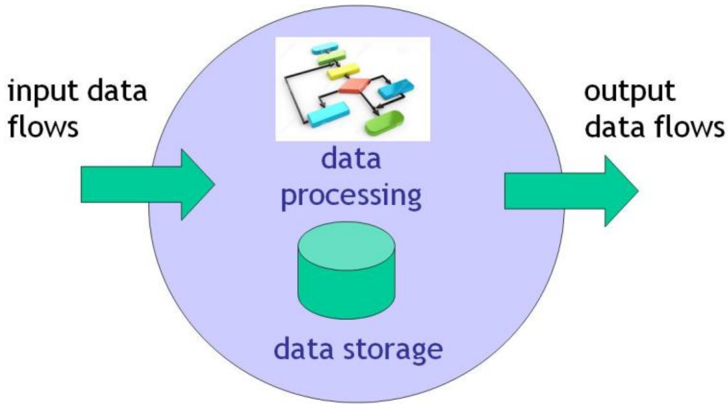


Figure 1

The circle represents a software application. A software application is characterized by a conceptual demarcation line that we may define "boundary" of the application and that separates the inside and outside of the application. There are many different ways to combine elementary software "objects" into software applications. Many of these derive from technical viewpoints, for example you can combine components that share allocation on a technology platform or programming language or type of service provided. A functional method of measurement, should instead use "non technological" criteria, related to a "logical" or user point of view of the services provided by a software application. For this reason the Simple Function Point method introduces the concept of Measurable Software Application (MSA) which is needed to distinguish an aggregate of software components that has the properties needed to be measured from the functional point of view. An essential task in governing software measurement is to maintain a mapping between the catalog of MSAs and the various operational and technical catalogs that eventually already exist.

1.4.4 MEASURABLE SOFTWARE APPLICATION (MSA)

In common IT language, the concept of Software Application can be represented by a collection of documentation, modules and procedures grouped not necessarily from the user perspective, therefore it is necessary to introduce the notion of Measurable Software Application (MSA).

MSA is defined as "an aggregate of logical features based on the business and identified from the point of view of its users, of their goals and informational and operational needs ".

"User" means what the ISO 14143 family of standards aims, namely: "Any person or object that communicates or interacts with the software at any time."

The definition of the MSAs is a preliminary mandatory step for any measurement in order to identify identical BFCs that will be measured only once per application even if encountered again and again in the functional menu structure for the use of a software system. The same BFCs eventually present in several MSAs must be measured for each application that contains them, instead.

Defining the MSA catalog is under the responsibility of the "owner" of the Software Applications. This catalog documents the boundaries between the applications of the assets that will form, therefore, the basis for all the measurements that will be done on those assets. The redefinition of the boundaries among MSAs usually changes the values of the functional measurements of the total assets for the same features provided to users, because of duplication of elements between the various MSAs. Since this activity, therefore, makes it impossible to comparable the values of assets calculated before and after the change of boundaries, it may be carried out only in exceptional cases. As a general rule, therefore, the boundaries among MSAs must remain stable to provide continuity and congruence to the measurements of the assets.

1.4.5 Layers

Current software architectures are characterized by the distribution of data and processing components on separate, cooperating technological platforms. Frequently, a process is performed dynamically on the most suitable architectural element at any given time. This organization makes it possible to reuse generalized components (often referred to as services) by standardizing and specializing both the features that contribute to achieving the application objectives and their interfaces. The models that describe these architectures use the concept of layers which is a way to aggregate these components on the basis of criteria of homogeneity of representation and usage methods. A layer is characterized, therefore, by a certain level of abstraction in the representation of data and functions that is linked, in turn, to the typical user's perspective associated with that particular layer. For example, the top application layer is associated to the requirements and usage behavior by a so-called business or end user. A DBMS layer is related to the processing and data storage requirements regardless of their semantic content for the end user; it is, in other words, a layer that considers the information more from a structural than from a business point of view.

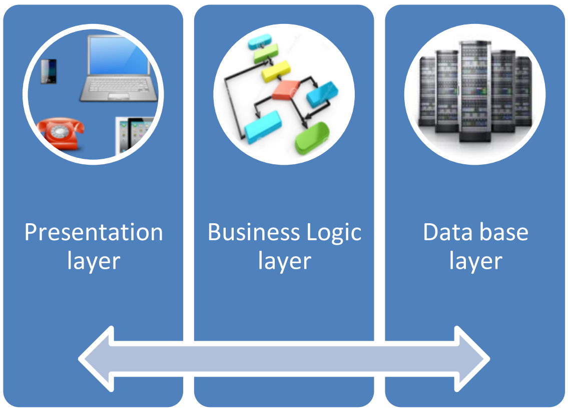


Figure 2 - Technical Layers

The layers most frequently used to aggregate service level components are:

Presentation Layer: it contains the user interface, for example an Internet browser. This invokes the services in the Business Layer.

Business Layer: it contains the services that perform required processing functions. They can be invoked by one or more services in the Presentation Layer or also by services in the same layer.

Data Access Layer: it contains the services that enable management of the DB data. They can be invoked by the Business Layer services.

This model, however, is seen from a technological perspective oriented to the design and "smart" implementation of software code and not to that of a business user. Therefore a similar approach, which emphasizes aspects related to distribution and relationship of client and server components residing on specific physical nodes of a data processing network, does not facilitate determining the software objects to measure from the application point of view, as required by international standards of functional measurement. An elementary "user functional process" typically starts with the activation, operated by a business actor, of functions offered by the user interface manager to collect information for searching or for writing data, it then proceeds through functions that analyze the requests and formulate, on the basis of application rules, the procedural steps necessary to provide a response to the user's request and ends up by returning information to the requester or to another actor using again the interface manager functions. This set of steps, considered significant and inseparable from the application end user's point of view, crosses several times the presentation, business and data layers previously identified. This means that a decomposition of the software by following those criteria does not allow identification of the proper software objects to be measured from the functional perspective.

However, it will be possible to use the indicated criteria, to perform a functional measurement of a software component at the middleware level.

1.4.6 Reference Model for Layered Architectures

A more usable approach to map applications on the generic model used as a foundation for a functional measurement is shown in the following illustration.

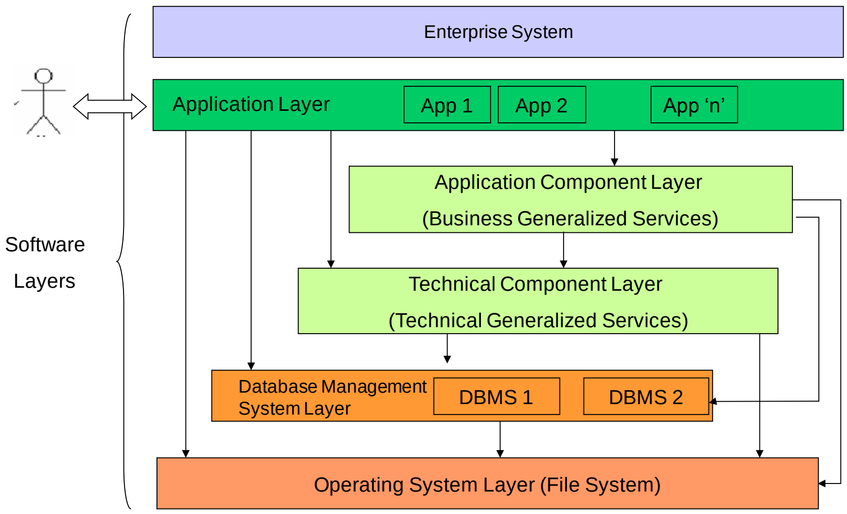


Figure 3 - Software architecture from a functional measurement perspective

The diagram shows that an enterprise system can be considered as an interface to activate a set of applications that appear to users through a variety of channels and that, in turn, are based on a set of underlying software layers, each of which can provide "services" to the layer above it either directly or indirectly. The arrows indicate the "call" direction of the components on the underlying layers. Two intermediate layers have been added between the "classical" application layer and the "environment" software: that of generalized business components and that of generalized technical components. The former are business functions that are recognizable at the application layer by the users of the system but are not autonomous enough to be considered part of the upper level i.e. independent MSAs; they are more like "recognizable" pieces of software that need to be "composed" and "aggregated" in order to respond to user requirements (e.g. a component to verify a tax code to be included in various elementary processes at the application level). The latter are generalized technical functions that support the management of applications (such as print drivers or generic form managers but can also be physical security managers, input/output control systems, network services, access management and support services, client-server services).

In summary, in this model, a middleware layer contains a set of functions defined by the user/designer that work to support specific requirements of modularization and independence from the hardware or from operating systems or from the DBMS environments. Middleware capabilities, being generalized, can then be used by different applications, even initially not considered in defining the layers of the technical architecture.

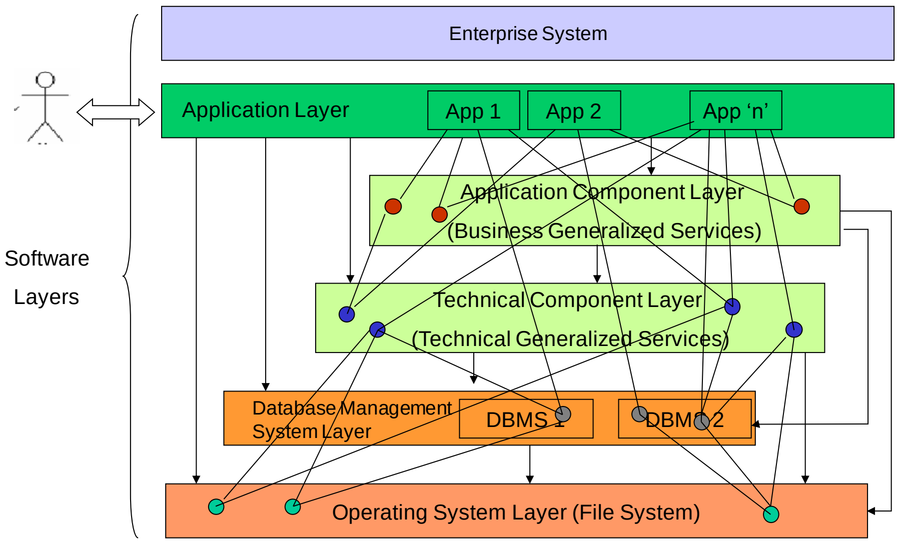


Figure 4 - Connection among components

The measurement model that the Simple Function Point Analysis is based on involves allocating all the features needed for the operation of a complex software system, upon layers each of which contains only "complete" and "significant" features from the perspective of the user of that level.

For example, an user at the business layer doesn't perceive the existence of the feature provided by the middleware (such as an optical reader driver), but enjoys the benefits arising from its presence in the system. In contrast, a logon transaction to authenticate users enabled to access the systems can be considered as a BFC from the business user's point of view while from the designer's point of view, there could be many other elementary functions and/or intermediate transactions, performed by the middleware and necessary for the completion of the authentication service.

Functional requirements can therefore be represented in the system specifications at uneven levels of aggregation and abstraction. Mapping of the functional requirements must allocate the functional user requirements (FUR) on the various software layers identifying components to be measured independently from each other.

1.4.7 Layers and Assets Measurement

The "assets" measurement of a software that belongs to a certain layer is expressed only as a function of the components that are perceived and measured on that layer, not of higher ones by which it is used or lower ones, which it uses. In other words, the asset size in FP of a software application (for example, valid for the calculation of service levels) must not be the result of the sum of the measurements in different layers.

1.4.8 Layers and Contractual Measurements

Measurements on different layers can, however, be added to each other, using the concept of scope, for contractual or management purposes besides asset assessment.

For example, a measurement on the Technical Generalized Services layer can be taken to account for the development of middleware components that a Supplier needs to build from scratch to handle particular technologies or non-functional user requirements that the production systems on the market or those that the Customer must use do not treat in a standardized way. This is the case of a particular georeferenced user interface or of a driver of technological equipment built for the purpose.

1.4.9 Conclusions

Any Measurable Software Application (MSA) may belong to one and only one layer but may therefore use services that are deployed on multiple layers, each of which contains generalized software components (technical or business) designed to provide a specific and reusable treatment of particular non-functional or functional requirements of the application layer. For example, presentation management components serve to free the GUI from dependence on physical devices or implement georeferencing requirements (GIS functions). Identification of generalized components belonging to layers below the business layer is also crucial in enhancing the amount or re-use attributable to each project measurement for the determination of contractual fees. All this, however, is not part of the SiFP measurement method but rather of the way the measurements are used to manage and drive production processes.

1.5 BFC Types

The SiFP method uses only two BFCs:

- UGEP: Unspecified Generic Elementary Process
- UGDG: Unspecified Generic Data Group

The term "Unspecified", in the case of the UGEP, highlights that it is not necessary to distinguish whether a process is mainly for input, or output, or what is its data processing primary intent. Similarly, in the case of the UGDG, it means that it is not necessary to distinguish between internal and external logical storage with respect to the boundary of the MSA. The term "Generic", on the other hand, indicates that for any BFC there is no need to identify subcomponents in order to determine BFC complexity: all the BFCs weight equally within the same type of BFC. Future developments of the methodology may lead to define different functional weights for each specific BFC depending on elements related to the processing component of transactional BFCs that, at present, is not quantitatively taken into account.

1.5.1 UGEP: Unspecified Generic Elementary Process

An Unspecified Generic Elementary Process is defined as:

"An atomic set of functional user requirements conceived for processing purposes. It refers to an informational or operational goal considered significant and unitary by the user and includes all automated, mandatory and optional activities needed to meet the goal. After an UGEP is concluded, the MSA to which it belongs must be in a logically consistent state."

1.5.2 UGDG: Unspecified Generic Data Group

An Unspecified Generic Data Group is defined as:

"An atomic set of user requirements having a storage purpose. It refers to a single logical data set of interest to the user, for which information must be kept persistently."

When identifying the UGDGs it must be clear that two different types of logical sets of data can be found in the user requirements, which we can call:

- Fundamental data group
- Auxiliary or non functional data group

The former keeps the information considered "of merit" in relation to the user's application related requirements, its processes and its perspective of interest. Fundamental data sets are used to satisfy the functional user requirements. For example, the following can be identified as fundamental data sets: Clerk, Sale, Supply contract, Car, Blast furnace, Missile, Telephone.

The latter are sets of data aimed at implementing non-functional requirements such as usability (data for drop-down lists, numeric range boundaries, stylesheets etc.) or performance (data access indexes) or maintainability (configuration parameters), and so on.

Only the first category (the fundamental data group) meets user functional requirements and can therefore be identified as a UGDG.

1.6 The SiFP Measurement Operating Procedure

The following illustration is a diagram of the measurement procedure that is explained in detail later.

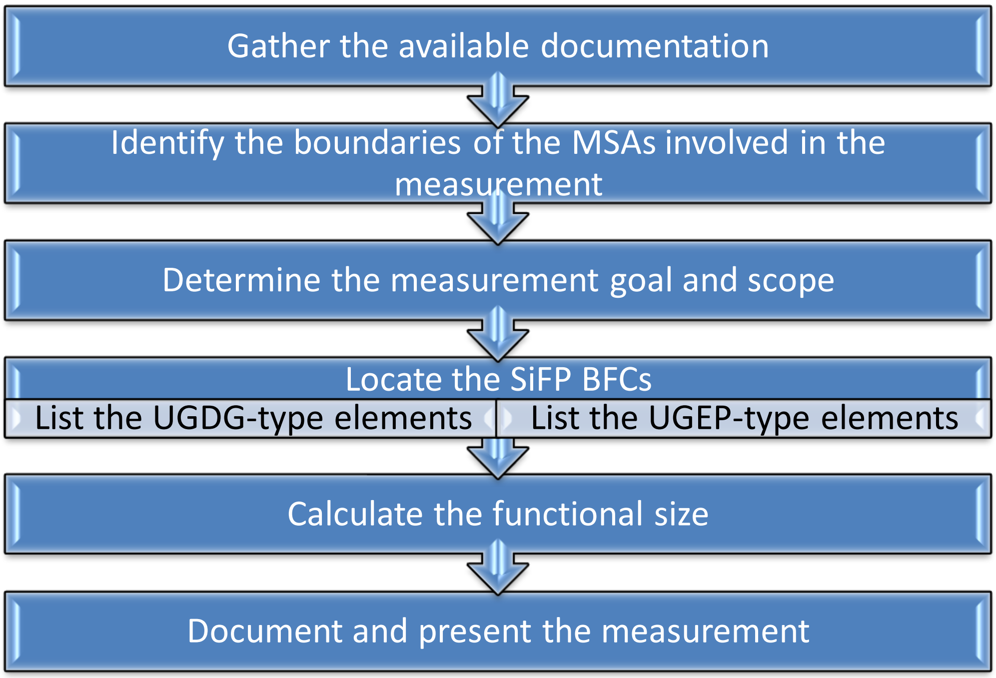


Figure 5 - Measurement procedure

1.6.1 Gather the Available Documentation

This step involves gathering all the information needed for a reliable functional measurement. The SiFP method is not dependent on any technology or analysis, design and user requirements' representation method. The measurer, before enacting the actual measurement, must put in place a scouting process aimed at locating all design or operational documents and the people who can be useful in the subsequent steps of the method. For as much as SiFP measurement is independent of the way user requirements are represented, it is still true that having the "right" documents and people for the measurement needs can facilitate or hinder the productivity and measurement quality. After all, the information requirements for the measurement are few and relate to the completeness and granularity of the functional requirements that must reach the level of refinement that makes it possible to identify with certainty the individual BFCs set out in the method and decide on their logical uniqueness. Similarly, information should be available to clearly establish the boundaries of the MSAs.

1.6.2 Identify the Boundaries of the MSAs Involved in the Measurement

As stated before, the identification of the MSA and of its boundaries is driven by logical principles, not technical, and focused on the user's standpoint (at any level of abstraction or layer it is). The focus is on what the user can understand and describe.

These general principles can be combined with the following operational recommendations.

To define the boundary of an MSA, aggregate functionalities and data based on the presence of organizational, functional and semantic similarities of the information that is shown/handled by those functionalities.

Identification of the boundaries of the MSAs should respect the principles of structured software design known as: minimization of coupling and maximization of cohesion. In other words, functional and operational interdependencies between separate MSAs should be minimal while inside an MSA there should not be any parts that are totally autonomous and independent, operationally and semantically; "container" type MSAs should be kept to a minimum. In these containers the different functionalities are kept together just because they can not be somewhere else or because of the technological ways they are used or because of other factors outside the "logic" of the user's point of view.

1.6.3 Determine the measurement goal and scope

A specific measurement exercise is characterized by a "scope" which may relate to one or more MSAs. The choice of measurement scope does not redefine the boundaries among the MSAs.

For example, a certain project may relate, at the same time, to the development of a new MSA and functional enhancement maintenance of pre-existing MSAs that must interface with it. The scope of the measurement includes both the software made from scratch and the pre-existent software that has been changed. The measurement must be made for each MSA separately and then the SiFP values will be summed up.

The scope is closely related to the measurement goal in the sense that it is determined by it. The goal of the measurement is, generally, a knowledge goal aimed at a management action and does not affect the measurement rules but only how the measurements are split, joined and connected to each other. For example, if a knowledge goal were to compare the level of on line interactive features to the batch processes for each application of the catalog, the scope would exclude the measurement resulting from the data part and would separate the online BFC measurements from the batch BFC measurements.

1.6.4 Locate the SiFP BFCs

There is no mandatory sequence, for this step. It is possible to start from the storage BFCs, working on to the processing BFCs, or vice versa, or even mixing the two approaches. One element of attention should be the identification of identical BFC candidates, that must be measured only once regardless of how many times they occur in the basic documentation for the measurement.

1.6.5 List the UGEP-type Elements

Using the gathered documentation, identify the UGEPs compiling their list.

1.6.5.1 Uniqueness Rules

- Each UGEP must appear once and only once in the organized list of BFC for MSAs. Within one MSA two UGEPs are identical when they process the same data in the same way and could be used interchangeably. The functional design normally identifies identical UGEP candidates.
- An UGEP which has the same computational behavior on the same data as another, and differs only by the technology used, or by the platform used (mobile, web, satellite, intelligent terminal, audio etc.) or the representation format (paper, electronic, etc.) is considered identical and must be listed only once.
- An UGEP must appear in each MSA in which it is used.

1.6.6 List the UGDG-type Elements

Using the gathered documentation, identify the UGDGs, compiling their list. Logical sets of data that are used in any way by the UGEPs of the MSA should be treated as UGDGs. There is no difference between the UGDGs that are only read or read and written by the UGEPs.

1.6.6.1 Uniqueness Rules

Each UGDG must appear once and only once in the MSAs organized list. Within an MSA two UGDGs are identical when they refer to the same object of interest to the user.

1.6.7 Calculate the Functional Size

Once the UGEP and UGDG lists are complete, the scores are assigned to the individual BFCs and added together as shown below. The scores to assign to each individual BFC are:

UGDG = 7,0 SiFP

UGEP = 4,6 SiFP

1.6.8 Document and Present the Measurement

The measurement must be documented with all the assumptions and measurement decisions taken, the standards used, the guidelines adopted, the links to the design documentation as set forth in the special section of the Reference Manual.

1.7 Calculation of the Functional Measurement

In SiFP method there are different formulas to calculate the functional measurement, depending on the type of measurement required.

1.7.1 New Development Activity

When creating a new MSA there will be two components to consider for the software involved by the activity: new (ADD) and auxiliary (AUX) features supporting start-up of usage of the MSA, such as logical files population, feature configuration, format translation, and initialization features. The latter will be part of the measurement of the features released by the the development activity but not of the assets measurement after development.

DEV SiFP = ADD + AUX

1.7.2 MSA Assets Measurement after New Development

At the end of development activity, the measurement of the assets released will be that of the activity that has generated it minus the auxiliary (AUX) component.

MSA SiFP = ADD

1.7.3 Functional Enhancement Maintenance Activity

For Functional Enhancement Maintenance activity (FEM) of an existing MSA there are 4 components to consider for measurement: features added (ADD), those changed (CHG), those deleted (DEL) and the auxiliary functions (AUX).

FEM SiFP = ADD+CHG+DEL+AUX

1.7.4 MSA Asset Measurement after FEM

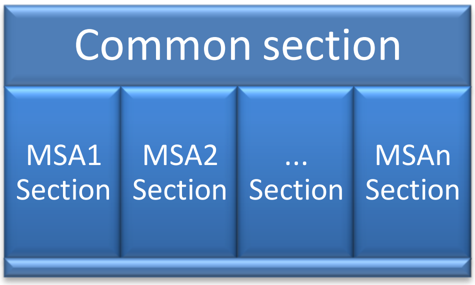
After Functional Enhancement Maintenance activity, the measurement of the assets released will be that of the assets measurement before the project activity that created it plus the new features (ADD) and minus the features removed (DEL).

MSA SiFP after = MSA SiFP before + ADD - DEL

1.8 DOCUMENTATION OF THE FUNCTIONAL MEASUREMENT

The last step of the SiFP functional measurement procedure is to: document and present the measurements. Below is the minimum set of information needed for this task.

Since a measurement is tied to a goal and this determines a scope that could include even more than one MSA, the measurement document must have a modular structure that includes a common part and a part that is repeated for each MSA involved in the scope.



1.8.1 COMMON SECTION

- Executive summary
- Objectives of the overall measurement
- Customer of the overall measurement
- Scope of the overall measurement
- MSAs involved by overall measurement
- Release date of the report
- Report authors
- Personnel involved in measurement activity
- Standards used (versions of the method)
- General documentary references

1.8.2 FOR EACH MSA

- Executive summary
- MSA Identification
- Type of specific measurement
- Any identifier of the measurement task (Development or FEM)
- Release date of the measurement
- Date of approval of the measurement
- Authors of the measurement
- Personnel involved in measurement and their role and position
- Standards used (versions of the method)
- Specific document references on which the measurement was based
- List of BFCs with corresponding functional weights

    - UGDG section (optionally with link to documentation of the related functional requirements)
    - UGEP section (optionally with link to documentation of the related functional requirements)
- Final outcome (calculation formula) with indication of the number of release of the method
- List of assumptions, critical aspects
    - overall
    - for each BFC

1.9 CONVERTIBILITY WITH OTHER FSMMS

Convertibility has been studied for version 4.x of the IFPUG method (meaning from version 4.0 to version 4.3.1). Convertibility for the COSMIC method is being studied.

Analysis of the convertibility for the IFPUG method, in accordance with the ISO/IEC 14143-1:2007 standard, identified a "statistical convertibility" with a high degree of reliability. In other words, there is an algorithm that applied to the elements of an IFPUG measurement leads to determine a SiFP measurement that is impressively close on a statistical basis. We must make it clear that the convertibility is not two-way or symmetrical because although you can pass from an IFPUG measurement to a SiFP measurement maintaining full correspondence between IFPUG BFCs and SiFP BFCs, it does not work the other way round, i.e. it is not possible to generate a list of IFPUG BFCs from a list of SiFP BFCs to assign the IFPUG measuring function to. SiFP --> IFPUG convertibility is possible, therefore just for overall final measurement values, precisely because of the very good overall correlation between the two measurements. The convertibility study findings are detailed below.

1.9.1 METHODOLOGICAL CORRESPONDENCE

Analysis of the theoretical correspondence between elements of the IFPUG method and of the SiFP method found the following evidences.

1.9.1.1 CORRESPONDENCE OF OBJECTS AND TYPES OF MEASUREMENT

The IFPUG software application and MSA concepts are closely related. The MSA definition is more complex but both definitions lead to identify the same measurement objects. The scope, border and goal concepts of the measurement are extremely similar. The types of measurement are identical (development, enhancement functional maintenance, assets).

1.9.1.2 BFC CORRESPONDENCE

The IFPUG transactional BFCs (EI, EO, EQ) match the UGEP BFC of the SiFP method. The definition of elementary process for IFPUG identifies an element that would be identified as UGEP in the SiFP method. There is, therefore, a 3:1 match between IFPUG BFCs and SiFP BFCs.

The data type BFCs of IFPUG (ILF and EIF) match the UGDG BFCs of the SiFP method. The definition of logical data set of IFPUG identifies a corresponding logical data set in the SiFP method. There is, therefore, a 2:1 match between IFPUG BFCs and SiFP BFCs. The IFPUG business and reference data types match the fundamental data type of the SiFP method, the decoding data type of the IFPUG method matches that of the auxiliary, nonfunctional data of the SiFP method. These rules lead to include the same elements in the list of processing and storage BFCs for both the methods.

The elimination rules for "identical logical functions" are very similar and lead to the same eliminations.

From these considerations it appears that the list of BFCs that would be produced by the identification phase in the IFPUG method has the same quantity and type correspondence (transactional and data) of that of SiFP.

1.9.1.3 CORRESPONDENCE OF CALCULATION FORMULAS

The calculation formulas for development activity, for assets after development, and for functional enhancement maintenance are similar. There is a difference only in the formula used to update assets after functional enhancement maintenance, that does not consider a complexity diversity of the functions changed since SiFP doesn't use a triad of FP values for each BFC but rather only one value.

1.9.1.4 Methodological Analysis Conclusions

From the above it appears that the two methods are very similar and conversion from IFPUG to SiFP is algorithmic with a margin of error due to the diversity of functional weights assigned to the BFCs.

1.9.2 Empirical Correspondence

To empirically assess the degree of convertibility of the IFPUG FP measurements to SiFP a sample of 766 ISBSG (International Software Benchmarking Standards Group) rel 11 points was used, for which IFPUG detailed measurements in terms of BFCs and therefore of SiFP BFCs were available. The distribution of IFPUG UFP values (Unadjusted Function Point) and SiFP was not normal as the ISBSG database is unbalanced on projects of small and medium-sized (more numerous) than large ones (less numerous). For this reason, the Spearman 's test and Kendall 's were used to verify the correlation between UFP and SiFP . The Spearman 's rank correlation test gave a value of rho = 0.988 (p - value < 10^{-15}) . The Kendall 's rank correlation test gave a value of tau = 0.907 (p - value < 10^{-15}) . It can be concluded , therefore, that UFP and SiFP are very strongly correlated.

In order to determine the numerical relation between UFP and SIFP, OLS linear regression forcing the passage of the intercept through the axis origin was used. After eliminating 321 outliers - according to the Cook's distance - we attived to the following ratio.

$$SiFP = 0.998\ UFP$$

The model has an adjusted $R^2 = 0.994$.

To verify the statistical significance of the hypothesis SiFP = UFP, interval for the coefficient was calculated at 95% confidence. It turned out that in the model SiFP = K × UFP, K belongs to the interval [0.9907 , 1.0052] with 95% confidence .

Thus, we can safely assume that SiFP = UFP , with a 95% confidence that the error is less than 1% . To give an idea of the extent of this error, by adopting the model SiFP = UFP , the difference to the pattern found by OLS regression is at best 1 SiFP - that is, practically negligible for sizes up to 732 UFP .

The asset difference (i.e. the difference between the sum of all signed measurements using the IFPUG method and the sum of all the signed measurements using the SiFP method) is -1123 FP out of 284,005 FP, which is -**0.4%.** This means that the positive and negative errors are compensated by combining the measurements together as if they were a large portfolio of applications.

2 Glossary of Terms

2.1 Glossary of Terms

2.1.1 Scope of Measurement

The scope of measurement defines the user features that will be included in a particular Simple Function Points measurement. The scope:

- defines a (sub)set of the software to be measured;
- is determined by the goal set for the measurement;
- identifies which user functions should be included in the measurement so as to provide relevant answers to the measurement goal;
- it could include more than one measurable software application (MSA).

2.1.2 Measurable Software Application (MSA)

A functional group suitable for the measurement in SiFP.

An MSA is defined as "an aggregate of logical functions based on business, on organizational management criteria of application domains and is conceived from a user point of view".

2.1.3 Base Functional Component (BFC)

Elementary Unit of Functional User Requirements (FUR) defined and used by an FSM method for measurement purposes.

2.1.4 Layer

A layer is an aggregate of software objects that share a strong functional focus and specialization. Each layer carries out specific and homogeneous tasks, can communicate with other layers and defer to them any actions it is not responsible for. There is a certain hierarchy among the different layers, in the sense that, generally speaking, the relationship between the layers is not peer to peer, but is governed by a set of dependencies that identify an order of relationship. This means that each level "relies on" one or more layers in order to perform its tasks, it depends on them and communicates with them.

2.1.5 Functional Enhancement Maintenance

Functional modification activity of an existing MSA

2.1.6 Measurement method

Logical sequence of operations performed to produce measurements.

2.1.7 Value

The numeric value assigned to an attribute of an entity as a result of a measurement

2.1.8 Measurement

The task of measuring and its result is assigning a value to an attribute in accordance with a scale of reference.

2.1.9 FUNCTIONAL USER REQUIREMENTS (FUR)

A subset of User Requirements. FURs are user practices and procedures that the software must perform to meet the user requirements. They do not include quality requirements and any technical requirement.

2.1.10 STORAGE PURPOSE

It is the goal that characterizes the set of functional requirements that identifies a UGDG: to preserve information about objects of interest to the user.

2.1.11 COMPUTATIONAL PURPOSE

It is the goal that characterizes the set of functional requirements that identifies a UGEP: to move and process information about objects of interest to and from the user.

2.1.12 ESTIMATE

The estimate can be regarded as an approximate measurement of a certain variable performed under non-standard rules but considered compatible and consistent with them. An estimate of a variable is less accurate than its standard measurement.

2.1.13 DEVELOPMENT

Activity to create a MSA.

2.1.14 UNIT OF MEASUREMENT

Conventional quantity with which similar quantities can be compared, to express the extent of their value. Units of measurement are assigned a name, a symbol (eg. person *day – pd*) and any multiples and submultiples (e.g. person month – pm, person hour - ph).

2.1.15 USER

Any person or object that communicates or interacts with the software at any time.

3 LICENSE AGREEMENT

creative commons

Attribution-NoDerivatives 4.0 International

Creative Commons Attribution-NoDerivatives 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NoDerivatives 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

c. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

d. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

e. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

f. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

g. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

h. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

i. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

j. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. **License grant**.

   1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

      A. reproduce and Share the Licensed Material, in whole or in part; and

      B. produce and reproduce, but not Share, Adapted Material.

   2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

   3. Term. The term of this Public License is specified in Section 6(a).

   4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

   5. Downstream recipients.

      A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

      B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. **Other rights**.
1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
2. Patent and trademark rights are not licensed under this Public License.
3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. **Attribution**.
1. If You Share the Licensed Material, You must:
   A. retain the following if it is supplied by the Licensor with the Licensed Material:
      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
      ii. a copyright notice;
      iii. a notice that refers to this Public License;
      iv. a notice that refers to the disclaimer of warranties;
      v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
   B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
   C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

   For the avoidance of doubt, You do not have permission under this Public License to Share Adapted Material.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database, provided You do not Share Adapted Material;
b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
   1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
   2. upon express reinstatement by the Licensor.

   For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.