



## **Manuel de mesurage COSMIC**

**ISO/IEC 19761**

### **Partie 2 : Directives**

**Version 5.0 - Mai 2020**

**Modifications mineures Février 2021**

## Avant-propos.

Le manuel de mesurage COSMIC pour ISO/IEC 19761: 2011 comprend trois parties :

Partie 1 : Principes, définitions et règles.

Partie 2 : Directives d'application

Partie 3 : Exemples de mesures COSMIC.

Cette partie 2 du manuel de mesurage COSMIC présente l'ensemble des directives élaborées par le groupe COSMIC pour faciliter la précision, la répétabilité et la reproductibilité des résultats de mesure COSMIC.

Mai 2020 : Modifications mineures

- Sous-section 3.7, le titre « Directive sur les règles 21 à 24 » a été remplacé par « Directives sur la règle 24 »
- Trois points ajoutés à la section sur l'identification des mouvements de données (texte provenant de MM4.0.2).

### Éditeurs

Alain Abran École de technologie supérieure – U. Québec, Canada	Peter Fagg Pentad Royaume-Uni	Arlan Lesterhuis Pays-bas
--	-------------------------------------	------------------------------

### Membres du comité des pratiques de mesure COSMIC

Diana Baklizky TI Metrics Brésil	Jean-Marc Desharnais École de Technologie Supérieure – U. Québec, Canada	Cigdem Gencel Université libre Bozen- Bolzano, Italie
Dylan Ren Measures Technology LLC Chine	Bruce Reynolds Tecalote Research États-Unis	Hassan Soubra, Université allemande du Caire, Égypte
Sylvie Trudel UQAM, Canada	Francisco Valdés Souto Spingere Mexique	Frank Vogelesang Metri Pays-Bas

<b>Membres du Comité de traduction en français du Guide de mesurage COSMIC</b>		
Alain Abran Ecole de technologie supérieure – U. Québec, Canada	Anna Chapelle (*) Estimancy, France	Jean-Marc Desharnais École de Technologie Supérieure – U. Québec, Canada
Patrick Hamon (*) Estimancy, France	Safae Laqrichi (*) Estimancy, France	Bernard Londeix, Telmaco, UK
Sylvie Trudel UQAM, Canada		

(\*) Un merci spécial au personnel d'Estimancy pour leur aide dans la traduction de ce document.

#### Contrôle de version

La Table suivante récapitule les modifications apportées à cette « ligne directrice ».

DATE	Réviser(s)	Modifications / Additions
	COSMIC MPC	Création "Part 2 – Rules v1"
Avril 2020	Estimancy - France	Révision – "Fr - Part 2 – Rules v1.1"
Avril 2020	École de Technologie Supérieure – U. Québec, Canada	Révision – "Fr - Part 2 – Rules v1.1JMD"
Mai 2020	Estimancy - France	Révision – "Fr - Part 2 – Rules v1.1"

Cette partie 2 a été approuvée par le MPC en Février 2021.

Copyright 2021. Tous droits réservés. Consortium international de mesure commune de logiciels (COSMIC). La permission de copier tout ou partie de ce matériel est accordée à condition que les copies ne soient pas faites ou distribuées à des fins commerciales et que le titre de la publication, son numéro de version et sa date soient cités et qu'un avis soit donné que la copie est autorisée par le Consortium international pour la mesure commune des logiciels (COSMIC). La copie nécessite une autorisation spécifique.

Une version du domaine public du Manuel de mesure COSMIC et d'autres rapports techniques, y compris des traductions dans d'autres langues, peuvent être trouvés dans la base de connaissances de [www.cosmic-sizing.org](http://www.cosmic-sizing.org).

## Table des matières

Avant-propos. ....	2
Contrôle de version.....	3
<b>1. INTRODUCTION. ....</b>	<b>5</b>
1.1 But. ....	5
1.2. Le processus du mesurage COSMIC. ....	5
1.3 Types et occurrences. ....	5
<b>2. LA PHASE DE STRATÉGIE DE MESURAGE. ....</b>	<b>5</b>
2.1 Aperçu de la phase de stratégie de mesurage. ....	5
2.2 Détermination de la raison d'être et du périmètre de la MTF. ....	6
2.3 Identification des EFU à partir des artefacts logiciels. ....	6
2.4 Exigences Non Fonctionnelles (ENF). ....	7
2.5 Identification des couches. ....	8
2.6 Identification des utilisateurs fonctionnels. ....	10
2.7 Niveaux de décomposition.....	10
2.8 Diagrammes contextuels. ....	10
2.9 Identification du niveau de granularité. ....	11
<b>3. LA PHASE DE MISE EN CORRESPONDANCE. ....</b>	<b>11</b>
3.1 Mise en correspondance des EFU avec le modèle générique COSMIC. ....	11
3.2 Identification des processus fonctionnels. ....	12
3.3 Identification des groupes de données. ....	13
3.3.1 <i>Identification des groupes de données. ....</i>	<i>13</i>
3.3.2 <i>À propos de l'identification des objets d'intérêt et des groupes de données. ....</i>	<i>14</i>
3.3.3 <i>Données ou groupes de données qui ne sont pas candidats à des mouvements de données.....</i>	<i>15</i>
3.3.4 <i>Identification des attributs de données (facultatif). ....</i>	<i>16</i>
3.4 Identification des mouvements de données. ....	16
3.6 Mesurer les composants d'un système logiciel distribué. ....	18
3.7 Mesure de la réutilisation du logiciel. ....	19
3.8 Mesure de la taille des modifications apportées au logiciel. ....	19
<b>4. AUTRES SUJETS. ....</b>	<b>20</b>
4.1 Extension de la méthode de mesure COSMIC - Extension locale. ....	20
4.2 COSMIC avec Agile. ....	20
<b>5. PHASE DE MESURAGE.....</b>	<b>21</b>
<b>6. RAPPORTS DE MESURE. ....</b>	<b>21</b>

## 1. INTRODUCTION.

### 1.1 But.

Le but de ce document est de fournir des conseils aux praticiens pour l'application de la norme COSMIC ISO 19761 Function Points présentée dans la partie 1. Cette partie 2 est complémentaire de la partie 1 et est complétée par la partie 3 qui présente un grand nombre d'exemples.

### 1.2. Le processus du mesurage COSMIC.

Le processus de mesurage COSMIC présenté dans la partie 1 est reproduit à la figure 1.1

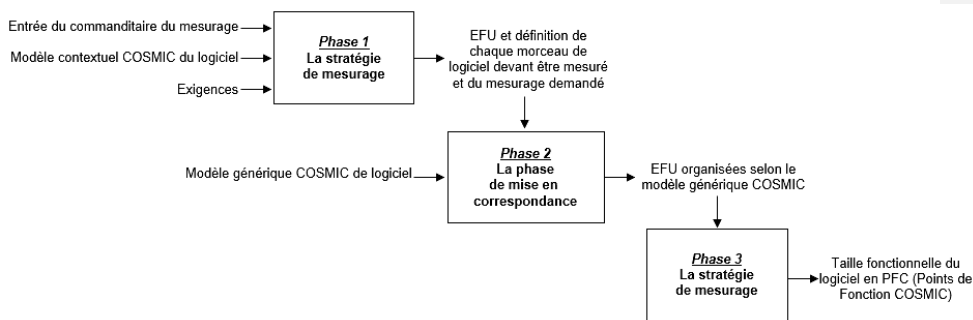


Figure 1.1 - Le processus de mesurage de la méthode COSMIC.

### 1.3 Types et occurrences

En général, le « type » d'une chose est une classe abstraite de toutes les choses qui partagent une caractéristique commune, de sorte que les choses sont soumises à la même EFU. (Synonymes de « type » : « catégorie », « genre »). Une « occurrence » d'une chose se produit lorsqu'elle devient réelle en attribuant des valeurs à ses attributs. Connue dans le monde orienté objet sous le nom d'instanciation – « la création d'une instance ».

## 2. LA PHASE DE STRATÉGIE DE MESURAGE.

### 2.1 Aperçu de la phase de stratégie de mesurage.

La figure 2.1 présente graphiquement la phase de la stratégie de mesurage.

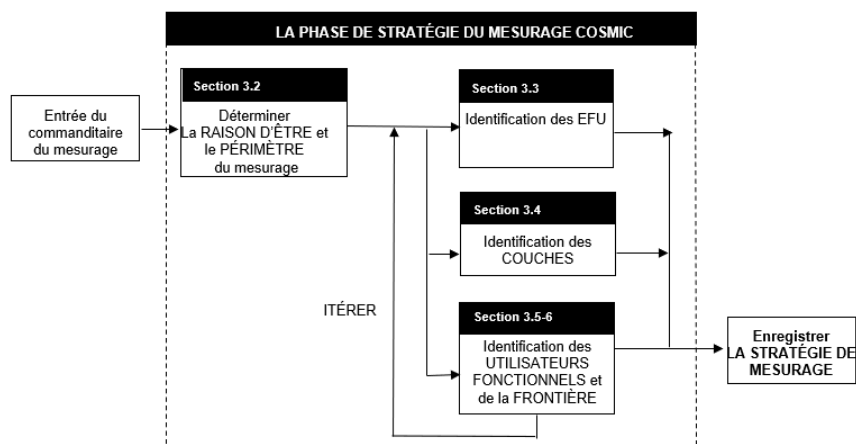


Figure 2.1 - Le processus de détermination d'une stratégie de mesure.

## 2.2 Détermination de la raison d'être et du périmètre de la MTF.

La raison d'être aide le mesureur à déterminer :

- le périmètre de la mesure et donc les artefacts qui seront nécessaires pour la mesure,
- les utilisateurs fonctionnels,
- les changements fonctionnels,
- le moment dans le cycle de vie du projet où la mesure aura lieu,
- la précision requise de la mesure, et donc s'il faut utiliser la méthode COSMIC standard, ou si une version approximative (par exemple au début du cycle de vie d'un projet, avant que les EFU soient entièrement élaborées).

Afin de faciliter l'élaboration de la stratégie de mesure, il est recommandé d'utiliser des « modèles de stratégie de mesure ». Il s'agit de groupes de paramètres standards spécifiques à chaque type de logiciel et permettant de mesurer une taille logicielle. L'utilisation cohérente des mêmes modèles permet aux mesureurs de s'assurer que :

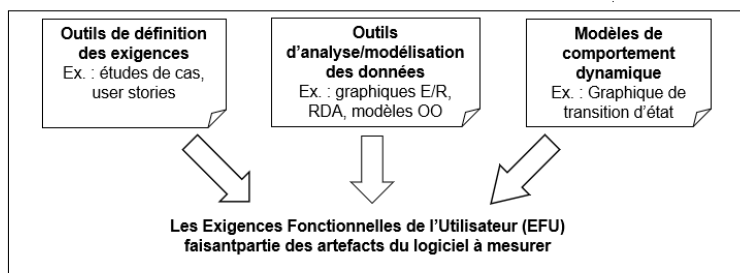
- les mesurages effectués dans le même but sont conduits de manière cohérente,
- les mesures peuvent être comparées en toute sécurité avec d'autres mesures découlant du même modèle
- les mesures seront correctement interprétées en vue de futures utilisations.

Un avantage secondaire de l'utilisation d'un modèle standard est que l'effort pour déterminer les paramètres de la stratégie de mesure est beaucoup moins important.

Le groupe COSMIC recommande aux mesureurs d'étudier et de maîtriser la méthode COSMIC, en particulier les paramètres de la stratégie de mesure, avant d'utiliser les modèles standard.

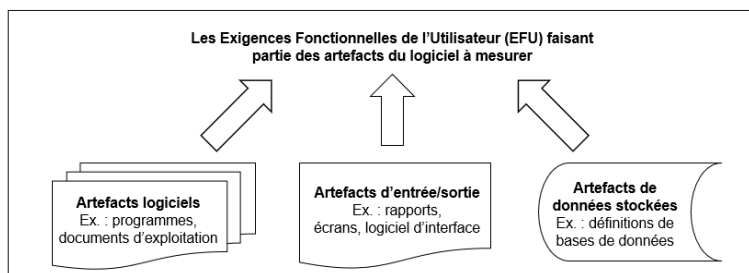
## 2.3 Identification des EFU à partir des artefacts logiciels.

Comme l'illustre la figure 2.2, les EFU peuvent être dérivées d'artefacts d'ingénierie logicielle qui ont été produits avant que le logiciel existe. Ainsi, la taille fonctionnelle d'un logiciel peut être mesurée avant sa mise en œuvre dans un système informatique.



**Figure 2.2 - Sources de pré-implémentation des besoins fonctionnels des utilisateurs.**

REMARQUE : Certains logiciels existants peuvent devoir être mesurés sans aucun, ou avec seulement quelques artefacts d'architecture ou de conception disponibles, et les Exigences Fonctionnelles de l'Utilisateur peuvent ne pas être documentées (par exemple pour les logiciels hérités). Dans de telles circonstances, il est toujours possible de dériver les EFU des artefacts du système informatique, même après sa mise en œuvre, comme illustré dans la figure 2.3.



**Figure 2.3 – Sources post-implémentation des Exigences Fonctionnelles de l'Utilisateur (EFU).**

Le processus à utiliser, et donc l'effort, étant requis pour extraire les EFU de différents types d'artefacts d'ingénierie logicielle ou pour les dériver du logiciel installé varieront. Ces processus ne peuvent pas être traités dans le manuel de mesure. La méthode COSMIC suppose que les EFU du logiciel à mesurer existent ou qu'elles peuvent être extraites ou dérivées de ses artefacts, à la lumière de l'objectif de la mesure.

Si le mesureur comprend ces deux modèles, il sera toujours possible de dériver les EFU d'un logiciel à mesurer à partir des artefacts disponibles, bien qu'il puisse avoir à formuler des hypothèses en raison d'informations manquantes ou peu claires.

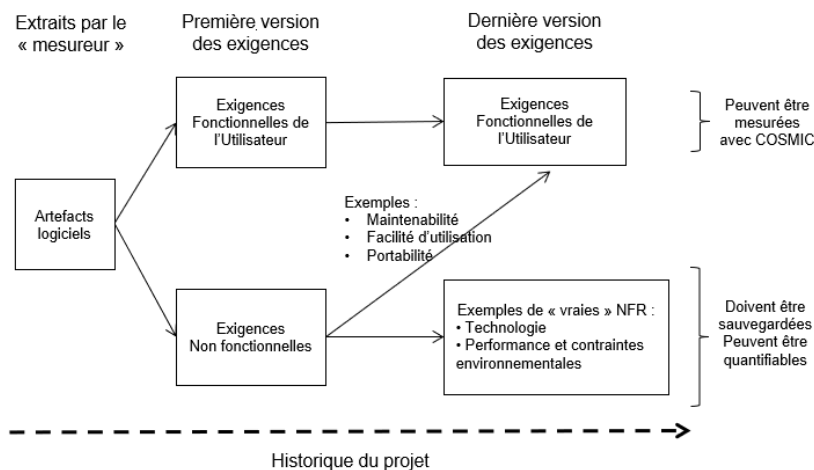
## 2.4 Exigences Non Fonctionnelles (ENF).

Les ENF peuvent être très nombreuses pour un projet logiciel. Dans les cas extrêmes, un énoncé des exigences pour un système à forte intensité logicielle peut nécessiter autant de documentation pour les ENF que pour les Exigences Fonctionnelles de l'Utilisateur. La méthode COSMIC peut être utilisée pour mesurer certaines exigences qui peuvent être non

fonctionnelles de prime abord. Plusieurs études ont montré que certaines ENF peuvent évoluer au fur et à mesure du projet, vers un mélange d'exigences dont certaines pourront être implémentées dans des fonctions logicielles, alors que d'autres sont bel et bien « non fonctionnelles ». Voir figure 2.4.

Cela est vrai pour de nombreuses contraintes de qualité du système, telles que le temps de réponse, la facilité d'utilisation, la maintenabilité, etc. Une fois identifiées, ces fonctions logicielles qui, au début du projet étaient « cachées » dans les ENF, peuvent être dimensionnées en utilisant la méthode COSMIC, comme toute autre fonction logicielle. C'est en partie à cause de cette taille logicielle « invisible » que la taille d'un logiciel peut sembler augmenter à mesure qu'un projet progresse.

Des discussions plus approfondies sur les Exigences Non Fonctionnelles des systèmes et des logiciels (ENF) sont présentées dans les « Directives sur les Exigences Non Fonctionnelles et projet » (<https://cosmic-sizing.org/publications/guideline-on-non-functional-project-requirements/>)



**Figure 2.4 - De nombreuses exigences initialement semblables à des ENF évoluent en EFU au fur et à mesure de l'avancement du projet.**

## 2.5 Identification des couches.

Les architectures logicielles peuvent être très complexes, mais COSMIC utilise une vue très simplifiée qui est suffisante aux fins du mesurage. C'est le mesureur qui doit effectuer cette simplification. Cette section donne des conseils au mesureur.

Étant donné que le périmètre d'un logiciel à mesurer doit être limité à une seule couche logicielle, le processus de définition de la ou des étendues de la MTF peut exiger que le mesureur commence par déterminer quelles sont les couches de l'architecture du logiciel à mesurer. Cette sous-section traite des « couches » de logiciel. Ces termes sont utilisés dans la méthode COSMIC parce que :

- le mesureur peut être confronté au mesurage de certains logiciels dans un environnement « hérité » de logiciels et ayant évolué pendant de nombreuses années sans jamais avoir été conçu selon une architecture sous-jacente.



- les expressions « couche » et « architecture en couches » ne sont pas utilisées de manière cohérente dans l'industrie du logiciel. Lorsque le mesureur doit mesurer un logiciel décrit comme étant dans une « architecture en couches », il est conseillé de vérifier que la définition des « couches » de cette architecture est compatible avec la méthode COSMIC. Pour ce faire, le mesureur doit établir l'équivalence entre des objets architecturaux spécifiques dans le paradigme de l'architecture en couches et le concept de couches tel que défini dans ce manuel.

Dans une architecture logicielle définie, chaque couche peut avoir les caractéristiques suivantes :

a) Le logiciel dans une couche fournit un ensemble de services qui est cohérent selon un critère défini, et que les logiciels dans d'autres couches peuvent utiliser sans pour autant savoir comment ces services sont mis en œuvre.

b) La relation entre les logiciels dans deux couches quelconques est définie par une « règle de correspondance » qui peut être :

- « hiérarchique », c'est-à-dire que les logiciels de la couche A sont autorisés à utiliser les services fournis par les logiciels de la couche B, mais pas l'inverse. Relation communément appelée « client-serveur » ;
- « bidirectionnelle », c'est-à-dire que les logiciels de la couche A sont autorisés à utiliser les logiciels de la couche B et vice versa. Relation communément appelée « poste à poste » (P2P).

c) Un logiciel d'une couche échange des groupes de données avec des logiciels d'une autre couche via leurs processus fonctionnels respectifs.

d) Les logiciels d'une couche n'utilisent pas nécessairement tous les services fonctionnels fournis par les logiciels d'une autre couche.

e) Les logiciels d'une couche d'une architecture logicielle définie peuvent être partitionnés en d'autres couches selon une autre architecture logicielle définie.

#### **DIRECTIVES sur la règle 4 : Identification des couches**

Si la portée globale de la MTF s'étend sur plusieurs couches, le mesureur doit procéder comme suit.

- Si le logiciel à mesurer existe dans une architecture en couches établie et pouvant correspondre aux principes de stratification COSMIC définis ci-dessus, alors cette architecture doit être utilisée pour identifier les couches à des fins de mesure.
- Si toutefois le but exige que soient mesurés des logiciels qui ne sont pas structurés selon les principes de stratification COSMIC, le mesureur doit essayer de partitionner le logiciel en couches en appliquant les principes définis ci-dessus.
- Classiquement, les logiciels d'infrastructure tels que les systèmes de gestion de base de données, les systèmes d'exploitation ou les pilotes de périphériques, qui fournissent des services pouvant être utilisés par d'autres logiciels dans d'autres couches, sont chacun situés dans des couches distinctes.

Normalement, dans les architectures logicielles, la couche « supérieure », c'est-à-dire, la couche qui n'est subordonnée à aucune autre couche dans une hiérarchie de couches, est appelée couche « d'application ». Le logiciel de cette couche d'application s'appuie sur les services des logiciels de toutes les autres couches pour fonctionner correctement. Les logiciels de cette couche « supérieure » peuvent eux-mêmes être superposés. Par exemple, une « architecture à trois couches » composée des couches : interface utilisateur, règles métier et services de données.

Une fois identifiée, chaque couche peut être notifiée dans le rapport de mesure, avec l'étiquette correspondante.

## 2.6 Identification des utilisateurs fonctionnels.

### DIRECTIVE sur la règle 7 : Identification des utilisateurs fonctionnels.

L'identification de l'utilisateur fonctionnel (ou des utilisateurs fonctionnels) est déterminée par les Exigences Fonctionnelles de l'Utilisateur devant être mesurées et la raison d'être de la mesure.

NOTE : Il n'y a rien d'absolu à propos d'un utilisateur fonctionnel, c'est-à-dire qu'il faut identifier les utilisateurs fonctionnels par processus fonctionnel. Dans le même logiciel mesuré, un émetteur / récepteur de données peut être un utilisateur fonctionnel pour certains processus fonctionnels, mais pas pour d'autres.

## 2.7 Niveaux de décomposition.


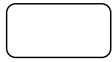
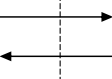
Les mesures de la taille des composants d'un logiciel ne sont directement comparables que pour des composants d'un même niveau de décomposition. Ceci est important car les tailles de logiciels de différents niveaux de décomposition ne peuvent pas être additionnées sans tenir compte des règles d'agrégation de la section 5.1. De plus, la performance (par exemple, la productivité = taille/effort) des projets de développement de différents logiciels ne peut être comparée que si tous les logiciels sont au même niveau de décomposition.

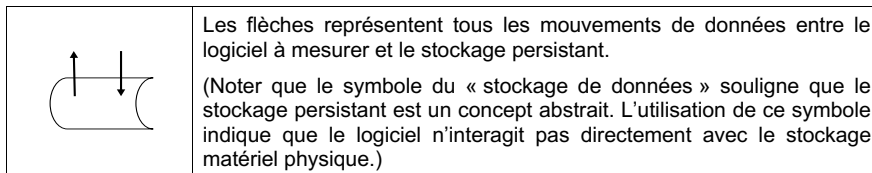
Différents niveaux de décomposition d'un logiciel peuvent correspondre à différentes « vues » des couches du logiciel (exemple : la figure 2.4 de la partie 3). Cependant, les logiciels peuvent être décomposés en « niveaux », qu'ils soient conçus ou non à l'aide d'un modèle d'architecture en couches.

## 2.8 Diagrammes contextuels.

Lors de la définition du périmètre de la MTF et des utilisateurs fonctionnels, il peut être utile de dessiner un « diagramme contextuel » pour le logiciel à mesurer. Ces diagrammes définissent et clarifient les frontières du système logiciel. Ils identifient les flux d'informations entre le logiciel et ses utilisateurs fonctionnels (humains, périphériques matériels ou autres logiciels) et les mouvements de données entre eux. (Les diagrammes contextuels montrent généralement un stockage persistant, le cas échéant.)

Un diagramme contextuel est une instance d'un modèle de mesure appliqué au logiciel mesuré. Les symboles utilisés dans les diagrammes contextuels sont présentés dans la figure 2.5.

Symbole	Interprétation
	Le logiciel à mesurer (contour épais), autrement dit, le périmètre de la mesure
	Tout utilisateur fonctionnel du logiciel à mesurer
	Les flèches représentent tous les mouvements de données traversant la frontière (ligne en pointillé) entre l'utilisateur fonctionnel et le logiciel à mesurer



**Figure 2.5 - Symboles des diagrammes contextuels.**

### 2.9 Identification du niveau de granularité.

COSMIC exige que les EFU soient exprimées à un niveau de détail suffisant pour créer les modèles de mesure COSMIC : c'est ce qu'on appelle le niveau de granularité.

Pour dériver une taille fonctionnelle en utilisant la mesure de la taille fonctionnelle (MTF) COSMIC en utilisant les RÈGLES de l'ISO 19761, le niveau de granularité nécessaire est celui auquel les processus fonctionnels individuels et leurs mouvements de données peuvent être identifiés et définis. Lorsque des détails fonctionnels manquent à d'autres niveaux de granularité des exigences, les mesures doivent être effectuées en utilisant l'une des techniques d'approximation de la taille décrites dans le guide du praticien pour une mesure rapide des Points de Fonction COSMIC en utilisant des techniques d'approximation.

REMARQUE 1 : Au cours des étapes initiales d'un projet de développement logiciel, les exigences réelles sont spécifiées « à un niveau élevé », c'est-à-dire, de manière succincte ou peu détaillée. Au fur et à mesure de l'avancement du projet, les exigences réelles sont affinées (par exemple via les versions 1, 2, 3, etc.), révélant de plus en plus de détails « aux niveaux inférieurs ». Ces différents degrés de détail des besoins réels sont les « niveaux de granularité ».

REMARQUE 2 : Les mesureurs doivent savoir que lorsque les exigences évoluent au début de la vie d'un projet logiciel, il est fréquent que certaines EFU soient documentées à différents niveaux de granularité.

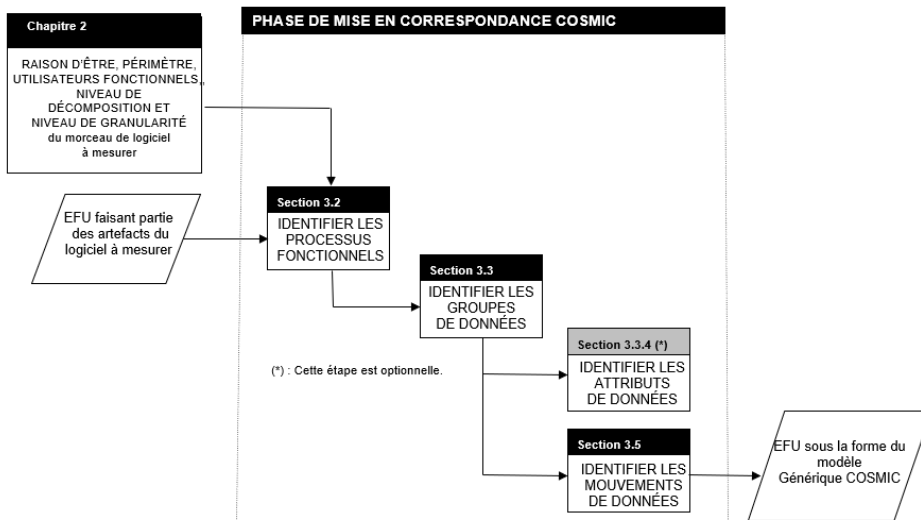
Pour un exemple de mesurage à différents niveaux de granularité et de décomposition, voir celui du système de télécommunication dans le document « Lignes directrices pour la mesure précoce ou rapide de la taille fonctionnelle COSMIC à l'aide d'approches d'approximation »

## 3. LA PHASE DE MISE EN CORRESPONDANCE.

Les processus fonctionnels sont composés de sous-processus qui déplacent des données (« mouvements de données ») et peuvent éventuellement manipuler des données (« manipulation de données »).

### 3.1 Mise en correspondance des EFU avec le modèle générique COSMIC.

La figure 3.1 montre les étapes de mise en correspondance des EFU dans les artefacts logiciels disponibles au format requis par le modèle générique COSMIC du logiciel.



**Figure 3.1 - Le processus de la phase de mise en correspondance COSMIC.**

Plusieurs directives décrivent comment mapper à partir de diverses méthodes d'analyse de données et de détermination des exigences, utilisées dans différents domaines, avec les concepts de la méthode COSMIC :

- [Guideline for Sizing Business Application Software](#),
- [Guideline for Sizing Data Warehouse Application Software](#),
- [Guideline for Sizing Service-Oriented Architecture Software](#), et
- [Guideline for Sizing Real-time Software](#).

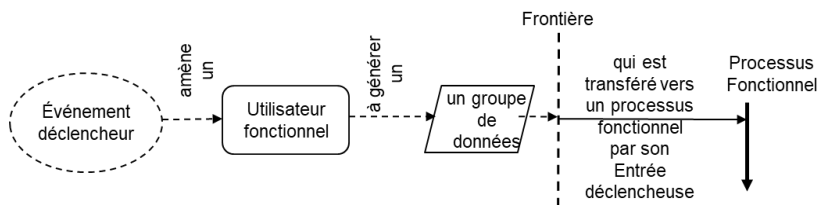
Pour les domaines métier ([business](#)) et temps réel ([real-time](#)), il existe également des guides de référence rapide qui donnent un aperçu du processus en quelques pages.

### 3.2 Identification des processus fonctionnels.

Cette section donne des conseils supplémentaires pour aider le mesureur.

La première étape de la phase de mesurage consiste à identifier l'ensemble des processus fonctionnels du logiciel à mesurer, à partir des EFU.

La relation entre un événement déclencheur, l'utilisateur fonctionnel et le mouvement de données d'Entrée qui déclenche un processus fonctionnel mesuré est présentée dans la figure 3.2 où : un événement déclencheur amène un utilisateur fonctionnel à générer un groupe de données qui est déplacé par l'entrée déclencheuse pour démarrer le processus fonctionnel.



**Figure 3.2 - Relations entre un événement déclencheur, un utilisateur fonctionnel et un processus fonctionnel.**

REMARQUE 1 : Pour faciliter la lecture, la référence au groupe de données est omise lorsqu'un utilisateur fonctionnel génère une entrée de déclenchement d'un processus fonctionnel, ou, plus simplement, lorsqu'un utilisateur fonctionnel déclenche un processus fonctionnel.

#### DIRECTIVES sur la règle 10 : Identification des processus fonctionnels

Après identification des utilisateurs fonctionnels et compte tenu des EFU du logiciel mesuré, le processus d'identification des processus fonctionnels, suit la chaîne de la figure 3.2, à savoir :

1. Identifier les événements distincts dans le monde des utilisateurs fonctionnels auxquels le logiciel mesuré doit répondre - les « événements déclencheurs ».

REMARQUE : Les événements déclencheurs peuvent être identifiés dans les diagrammes d'état et dans les diagrammes de cycle de vie d'entité, car certaines transitions d'état et de cycle de vie d'entité correspondent à des événements déclencheurs auxquels le logiciel doit réagir.

2. Identifier le ou les utilisateurs fonctionnels du logiciel qui peuvent répondre à chaque « événement déclencheur » ;

3. Identifier la ou les entrées de déclenchement que chaque utilisateur fonctionnel peut lancer en réponse à l'événement ;

4. Identifier le processus fonctionnel démarré par chaque entrée de déclenchement.

Il convient d'utiliser les vérifications suivantes pour s'assurer que les processus fonctionnels candidats (PF) ont été correctement identifiés pour le logiciel mesuré :

1. Tous les PF identifiés résident-ils dans la même couche ?
2. Tous les PF identifiés comprennent-ils une entrée et au moins un mouvement de données d'écriture ou de Sortie ?

### 3.3 Identification des groupes de données.

#### 3.3.1 Identification des groupes de données.

Après avoir identifié les processus fonctionnels, l'étape suivante consiste à identifier leurs mouvements de données. Les conseils suivants aident à identifier les groupes de données et donc les objets d'intérêt, en particulier au niveau de la sortie des processus fonctionnels.

#### DIRECTIVES sur la règle 11 : Identifier différents groupes de données déplacés dans le même processus fonctionnel.

Pour tous les attributs de données apparaissant dans une Entrée-Sortie-Lecture-écriture d'un processus fonctionnel :

- a) des ensembles d'attributs de données qui ont différentes fréquences d'occurrence décrivent différents objets d'intérêt ;
- b) des ensembles d'attributs de données qui ont la même fréquence d'occurrence mais des attributs clés d'identification différents décrivent différents objets d'intérêt ;
- c) tous les attributs de données d'un ensemble résultant de l'application des parties a) et b) de ce guide appartiennent au même type de groupes de données, sauf si les EFU spécifient qu'il peut y avoir plus d'un type de groupes de données décrivant le même objet d'intérêt dans l'entrée du processus fonctionnel. (Cf directives des règles 13 et 14 – Unicité des mouvements de données, cas b) et c)).

REMARQUE 1 : Un utilisateur fonctionnel du logiciel mesuré peut être l'objet d'intérêt d'un groupe de données envoyé ou reçu par l'utilisateur fonctionnel.

REMARQUE 2 : En théorie, un groupe de données peut contenir un seul attribut de données si c'est tout ce qui est requis par les EFU pour décrire l'objet d'intérêt. En pratique, de tels cas se produisent couramment dans les logiciels d'application temps réel (par exemple, le groupe de données entré pour transmettre le tic d'une horloge en temps réel ou l'entrée de l'état d'un capteur) ; ils sont moins courants dans les logiciels d'application métier.

REMARQUE 3 : Il n'y a rien d'absolu à propos d'un objet d'intérêt, c'est-à-dire d'identifier les objets d'intérêt par processus fonctionnel. Une 'chose' peut être un objet 'd'intérêt' pour un utilisateur fonctionnel via un ou plusieurs processus fonctionnels, mais pas être un objet 'd'intérêt' pour un autre utilisateur fonctionnel via d'autres processus fonctionnels, même dans le même logiciel mesuré.

En pratique, l'origine d'un groupe de données peut prendre plusieurs formes, par exemple :

- a) Une structure d'enregistrement physique sur un périphérique de stockage matériel (fichier, table de base de données, mémoire ROM, etc.).
- b) Une structure physique dans la mémoire volatile de l'ordinateur (structure de données allouée dynamiquement ou via un bloc pré-alloué d'espace mémoire).
- c) Une présentation groupée d'attributs de données fonctionnellement liés sur un périphérique d'entrée/sortie (écran d'affichage, rapport imprimé, affichage du panneau de commande, etc.).
- d) Un message en transmission entre un appareil et un ordinateur, ou via un réseau, etc.

### **3.3.2 À propos de l'identification des objets d'intérêt et des groupes de données.**

La définition et les principes des objets d'intérêt et des groupes de données sont intentionnellement larges afin d'être applicables à la gamme de logiciels la plus large possible. Il en résulte qu'il est parfois difficile d'appliquer la définition et les principes lors de la mesure d'un logiciel spécifique. Voir la partie 3 pour des exemples pour aider à l'application des principes à des cas spécifiques.

Face à un besoin d'analyser un groupe d'attributs de données qui est déplacé dans ou hors d'un processus fonctionnel ou qui est déplacé par un processus fonctionnel vers ou depuis un stockage persistant, il est extrêmement important de décider si les attributs transmettent tous des données sur un seul « objet d'intérêt », car ce sont ces derniers qui déterminent le nombre de « groupes de données » distincts définis par la méthode COSMIC qui seront déplacés par les mouvements de données.

Par exemple, si les attributs de données à saisir dans un processus fonctionnel sont des attributs de trois objets d'intérêt distincts, alors trois mouvements de données d'Entrée

distincts doivent être identifiés. Décider du nombre de groupes de données peut être difficile lors de l'analyse de la sortie d'un processus fonctionnel d'une application métier pouvant inclure :

- plusieurs groupes de données, chacun décrivant un objet d'intérêt différent, par exemple, un rapport montrant les totaux à différents niveaux d'agrégation ;
- les résultats de requêtes où la sortie variera en fonction de l'entrée ;
- des groupes de données qui peuvent même être indépendants les uns des autres, par exemple une facture qui comprend une publicité pour un service indépendant.

Quand il faut analyser des sorties complexes, par exemple, des rapports avec des données de plusieurs objets d'intérêts, considérer chaque groupe de données séparément, comme s'il s'agissait de sorties de processus fonctionnels distincts. Tous les groupes de données identifiés ainsi doivent aussi être comptés pour mesurer le rapport.

### ***3.3.3 Données ou groupes de données qui ne sont pas candidats à des mouvements de données.***

Les données apparaissant sur les écrans d'entrée ou de sortie ou les rapports qui ne sont pas liés à un objet d'intérêt pour un utilisateur fonctionnel ne doivent pas être identifiées comme étant un groupe de données et ne doivent donc pas être mesurées.

Le modèle générique COSMIC suppose que toute manipulation de données au sein d'un processus fonctionnel est associée aux quatre types de mouvements de données. Par conséquent, aucun groupe de données ne peut être identifié comme résultant de la manipulation de données au sein d'un processus fonctionnel en plus des groupes de données déplacés par les Entrées, Sorties, Lectures et écritures du processus.

### 3.3.4 Identification des attributs de données (facultatif).

Dans la méthode COSMIC, il n'est pas obligatoire d'identifier les attributs de données. Cependant, la compréhension du concept d'« attribut de données » est nécessaire pour appréhender le « mesurage des changements ». Une demande pour modifier un attribut de données peut résulter en un mesurage du mouvement de données auquel appartient l'attribut à « modifier ».

En outre, il peut être utile d'analyser et d'identifier les attributs de données dans le processus de distinction des groupes de données et des objets d'intérêt.

### 3.4 Identification des mouvements de données.

Cette étape consiste à identifier les mouvements de données (Entrée, Sortie, Lecture et Écriture) de chaque processus fonctionnel.

#### DIRECTIVES concernant la règle 12 : les mouvements de données

Les conseils suivants permettent de confirmer le statut d'un mouvement de données d'entrée candidat :

#### DIRECTIVES concernant la règle 16 : Entrée (E).

- Le groupe de données d'une entrée déclencheuse peut être constitué d'un seul attribut de données qui informe simplement le logiciel qu'un événement Y s'est produit.
- Pour les tics d'horloge qui déclenchent des événements, identifier une entrée provenant d'un utilisateur fonctionnel, dans ce cas, l'horloge.
- À moins qu'un processus fonctionnel spécifique ne soit nécessaire, l'obtention de la date et/ou de l'heure à partir de l'horloge du système n'est pas considérée comme une entrée ou tout autre mouvement de données COSMIC.

REMARQUE : Très souvent, en particulier dans les logiciels d'application métier, le groupe de données de l'entrée déclencheuse a plusieurs attributs de données qui informent le logiciel qu'un événement Y s'est produit et voici les données sur cet événement particulier.

#### DIRECTIVES concernant la règle 17 : Sortie (S).

- Pour une requête qui produit du texte fixe (où « fixe » signifie que le message ne contient pas de valeurs de données variables), identifier une Sortie pour la sortie de texte fixe.
- Lors de l'identification des sorties, ignorer tous les champs et autres en-têtes qui permettent aux utilisateurs humains de comprendre les données de sortie.

#### DIRECTIVE sur la règle 18 : Lecture (L).

Ne pas identifier de Lecture lorsque les EFU du logiciel mesuré spécifient un utilisateur fonctionnel logiciel ou matériel comme source d'un groupe de données ou comme moyen de déplacer le groupe de données vers un stockage persistant. L'interaction avec d'autres utilisateurs fonctionnels se fait par définition à travers une frontière (modèle logiciel générique, principe 1), qui est gérée par un mouvement de données d'Entrée. La lecture réelle a lieu dans les limites du logiciel mesuré.

#### DIRECTIVE sur la règle 19 : Écriture (C).

Ne pas identifier d'Écriture lorsque les EFU du logiciel mesuré spécifient un utilisateur fonctionnel logiciel ou matériel comme destination du groupe de données ou comme moyen de récupérer un groupe de données stocké de manière permanente. L'interaction avec d'autres utilisateurs fonctionnels se fait par définition à travers une frontière (modèle logiciel

**Commenté [SL1]:** Si le contenu est supprimé, le titre doit être aussi supprimé



générique, principe 1), qui est gérée par un mouvement de données de Sortie. L'écriture réelle a lieu dans les limites du logiciel mesuré.

REMARQUE : Lorsque les EFU exigent que les données soient stockées ou extraites du stockage, le mesureur doit rechercher si les données peuvent être stockées ou récupérées à l'intérieur de ses propres limites, c'est-à-dire, vers/depuis le « stockage persistant », ou si les données doivent être stockées/récupérées à l'aide d'un utilisateur fonctionnel du logiciel mesuré (c'est-à-dire via un autre logiciel, ou directement vers ou depuis un périphérique matériel).

#### **DIRECTIVES sur les règles 16 à 19 - Manipulation des données associée aux mouvements de données.**

La manipulation de données associée à l'un de ces mouvements de données n'inclut aucune manipulation de données nécessaire une fois le mouvement de données terminé avec succès, ni aucune manipulation de données associée à tout autre mouvement de données.

Les directives suivantes couvrent la situation la plus courante (règle a)) et d'autres cas valides possibles (directives b) et c)).

- En a), les occurrences de groupes de données sont relatives à une même EFU : ainsi un groupe de données et un mouvement de données sont identifiés.
- En b) et c), la même chose s'applique à chaque groupe de données séparément : ainsi un groupe de données et un mouvement de données sont identifiés pour chaque groupe de données.

#### **DIRECTIVES sur les règles 13 à 14 - Unicité des mouvements de données**

a) Sauf si les Exigences Fonctionnelles de l'Utilisateur sont telles que données dans les directives b) ou c), toutes les données décrivant tout objet d'intérêt devant être entré dans un processus fonctionnel sont identifiées comme un groupe de données déplacé par une entrée.

REMARQUE 1 : Un processus fonctionnel peut, bien sûr, avoir plusieurs entrées, chacune des données mobiles décrivant un objet d'intérêt différent.

REMARQUE 2 : Les mêmes instructions équivalentes s'appliquent à tout mouvement de Lecture, d'écriture ou de Sortie de données dans un processus fonctionnel.

b) Si les Exigences Fonctionnelles des Utilisateurs spécifient que différents groupes de données doivent être entrés dans un processus fonctionnel, chacun provenant d'un utilisateur fonctionnel différent, où chaque groupe de données décrit le même objet d'intérêt, une Entrée est identifiée pour chacun de ces différents groupes de données.

REMARQUE 1 : Les directives équivalentes s'appliquent aux Sorties de données vers différents utilisateurs fonctionnels d'un processus fonctionnel.

REMARQUE 2 : Tout processus fonctionnel a une seule entrée déclencheuse.

c) Si les besoins fonctionnels des utilisateurs spécifient que différents groupes de données doivent être déplacés du stockage persistant vers un processus fonctionnel, chacun décrivant le même objet d'intérêt, alors une lecture est identifiée pour chacun de ces différents groupes de données.

REMARQUE 1 : Les directives équivalentes s'appliquent aux écritures dans tout processus fonctionnel donné.

REMARQUE 2 : Ce guide est analogue à la règle b). Dans le cas des EFU, pour lire

différents groupes de données décrivant le même objet d'intérêt, ils proviendront probablement de différents utilisateurs fonctionnels. Dans le cas où les EFU décrivent différents groupes de données, ils seront probablement mis à disposition pour être lus par différents utilisateurs fonctionnels.

**DIRECTIVES sur les règles 16 et 17 : Processus fonctionnel nécessitant des données d'un utilisateur fonctionnel.**

- a) Lorsque le processus fonctionnel n'a pas besoin de dire à l'utilisateur fonctionnel quelles données envoyer, une seule Entrée suffit (par objet d'intérêt).
- b) Lorsque le processus fonctionnel doit indiquer à l'utilisateur fonctionnel quelles données envoyer, une sortie suivie d'une entrée sont nécessaires.

**DIRECTIVES sur les règles 16 à 19 : Commandes de contrôle dans les applications avec une interface humaine.**

Dans une application avec une interface humaine, les « commandes de contrôle » sont ignorées car elles n'impliquent aucun mouvement de données sur un objet d'intérêt.

Les messages d'erreur et de confirmation sont des formes spécifiques de participation et les règles régissant les identifications s'appliquent.

**DIRECTIVES sur les règles 16 à 19 : Messages d'erreur/de confirmation et autres indications de conditions d'erreur**

- a) Une sortie est identifiée pour tenir compte de tous les types de messages d'erreur/de confirmation émis par un processus fonctionnel du logiciel mesuré à partir de toutes les causes possibles en fonction de ses EFU.
- b) Si un message à un utilisateur fonctionnel humain fournit des données en plus de confirmer que les données saisies ont été acceptées, ou que les données saisies sont erronées, ces données supplémentaires sont identifiées comme un groupe de données distinct déplacé par une Sortie.
- c) Toutes les autres données, émises ou reçues par le logiciel mesuré, vers/depuis ses utilisateurs matériels ou logiciels fonctionnels doivent être analysées conformément aux EFU respectivement en tant que Sorties ou Entrées, conformément aux règles COSMIC, indépendamment du fait que les valeurs de données indiquent une condition d'erreur.
- d) Les Lectures et écritures sont considérées comme tenant compte de tout rapport associé aux conditions d'erreur. Par conséquent, aucune entrée dans le processus fonctionnel mesuré n'est identifiée pour une indication d'erreur reçue à la suite d'une Lecture ou d'une écriture de données persistantes.
- e) Aucune Entrée ou Sortie n'est identifiée pour tout message indiquant une condition d'erreur qui pourrait être émis lors de l'utilisation du logiciel mesuré mais qui ne doit en aucun cas être traité par les EFU de ce logiciel, par exemple, un message d'erreur émis par le système d'exploitation.

**3.6 Mesurer les composants d'un système logiciel distribué.**

Lorsque le but d'un mesurage est de mesurer séparément la taille de chaque composant d'un système logiciel distribué, un périmètre distinct de la MTF doit être défini pour chaque composant. Dans un tel cas, le dimensionnement des processus fonctionnels de chaque composant suit toutes les règles déjà décrites.

Du processus pour chaque mesurage (définition du périmètre, puis des utilisateurs fonctionnels et de la frontière, etc.), il s'ensuit que si un logiciel se compose de deux ou

plusieurs composants, il ne peut y avoir de chevauchement entre les périmètres des MTF de chaque composant. Le périmètre de la MTF de chaque composant doit définir un ensemble de processus fonctionnels complets. Par exemple, il ne peut pas y avoir de processus fonctionnel avec une partie dans un périmètre et une partie dans un autre. De même, les processus fonctionnels dans le périmètre de mesure d'un composant ne disposent d'aucune information sur les processus fonctionnels dans le périmètre d'un autre composant, même si les deux composants échangent des messages.

Le ou les utilisateurs fonctionnels de chaque composant sont déterminés en examinant où se produisent les événements déclencheurs des processus fonctionnels dans le composant examiné. (Les événements déclencheurs ne peuvent se produire que dans le monde d'un utilisateur fonctionnel.)

### **3.7 Mesure de la réutilisation du logiciel.**

Deux ou plusieurs processus fonctionnels dans le même logiciel mesuré peuvent avoir des fonctionnalités identiques ou très similaires dans chaque processus et décrites séparément ailleurs dans les exigences. Ce phénomène est appelé « partage » fonctionnel ou « similitude » fonctionnelle.

Cependant, chaque processus fonctionnel est défini, modélisé et mesuré indépendamment, c'est-à-dire, sans référence à toute autre EFU dans le même logiciel mesuré.

Par conséquent, si une EFU pour un processus fonctionnel donné fait référence à une EFU ailleurs dans les exigences, la taille de cette fonctionnalité référencée doit être incluse dans la taille du processus fonctionnel mesuré.

### **3.8 Mesure de la taille des modifications apportées au logiciel.**

Un « changement fonctionnel » d'un logiciel existant est interprété dans la méthode COSMIC comme « toute combinaison d'ajouts de nouveaux mouvements de données ou de modifications ou de suppressions de mouvements de données existants, y compris la manipulation de données associée ». Les termes « amélioration » et « maintenance » sont souvent utilisés pour ce que nous appelons ici un « changement fonctionnel ».

La nécessité d'un changement de logiciel peut provenir :

- d'une nouvelle EFU (c'est-à-dire uniquement des ajouts à la fonctionnalité existante), ou
- d'une modification des EFU (impliquant éventuellement des ajouts, des modifications et des suppressions) ou
- d'un besoin de « maintenance » pour corriger un défaut.

Les règles de dimensionnement de ces modifications sont les mêmes, mais le mesureur doit distinguer les différents contextes lors de la mesure et de l'estimation des performances.

Un mouvement de données est considéré comme fonctionnellement modifié comme suit.

#### **DIRECTIVES sur la règle 24 : Modification d'un mouvement de données.**

- a) Un mouvement de données est considéré comme fonctionnellement modifié si au moins une des conditions suivantes s'applique :
- Le groupe de données est modifié,
  - La manipulation de données associée est modifiée.
- b) Un groupe de données est modifié si au moins une des conditions suivantes s'applique :
- Au moins un nouvel attribut a été ajouté au groupe de données,

- Au moins un attribut a été retiré du groupe de données,
- Au moins un attribut a été modifié, par exemple sur le fond ou la forme (mais pas sa valeur).

- c) Une manipulation est modifiée si elle a été fonctionnellement modifiée.
- d) Si un mouvement de données doit être modifié à la suite d'une modification de la manipulation de données associée et/ou suite à la modification du nombre ou du type des attributs du groupe de données déplacé, on mesure 1 PFC modifié, peu importe le nombre de modifications apportées au mouvement de données.
- e) Si un groupe de données doit être modifié, les mouvements de données déplaçant le groupe de données modifié dont la fonctionnalité n'est pas affectée par la modification du groupe de données ne sont pas identifiés comme des mouvements de données modifiés.

REMARQUE : Une modification des données apparaissant sur les écrans d'entrée ou de sortie et qui ne sont pas liées à un objet d'intérêt associé à un utilisateur fonctionnel n'est pas un PFC modifié.

- f) Une convention de mesure est que la taille fonctionnelle d'un logiciel ne change pas lorsqu'il faut changer le logiciel pour corriger une anomalie qui le rend conforme aux EFU. La taille fonctionnelle change lorsqu'on corrige une anomalie d'une EFU.

Les mouvements de données modifiés n'ont aucune influence sur la taille du logiciel car ils existent à la fois avant et après les modifications.

Lorsqu'un logiciel est complètement remplacé, par exemple en le réécrivant, avec ou sans extension et/ou omission de fonctionnalité, la taille fonctionnelle de ce changement est la taille du logiciel de remplacement, mesurée selon les règles normales de dimensionnement du nouveau logiciel.

REMARQUE : La différence entre la taille de la modification fonctionnelle (discutée ici) et la modification de la taille fonctionnelle du logiciel. Habituellement, elles sont différentes.

## 4. AUTRES SUJETS.

### 4.1 Extension de la méthode de mesure COSMIC - Extension locale.

La méthode COSMIC de mesure d'une taille fonctionnelle ne suppose pas de mesurer tous les aspects possibles de la « taille » d'un logiciel. Ainsi, la méthode n'est pas conçue pour mesurer séparément et explicitement la taille des EFU des sous-processus de manipulation de données. L'influence sur la taille des sous-processus de manipulation des données est prise en compte via une hypothèse simplificatrice valable pour un large éventail de domaines logiciels.

Néanmoins, la mesure de taille COSMIC est considérée comme une bonne approximation de l'objectif déclaré et des domaines d'applicabilité de la méthode. Pourtant, il se peut, qu'il soit souhaitable de tenir compte de cette fonctionnalité d'une manière significative en tant que norme locale au sein de l'environnement local d'une organisation utilisant la méthode de mesure COSMIC. Lorsque de telles extensions locales sont utilisées, les résultats de mesure doivent être communiqués conformément à la convention spéciale présentée à la section 6.

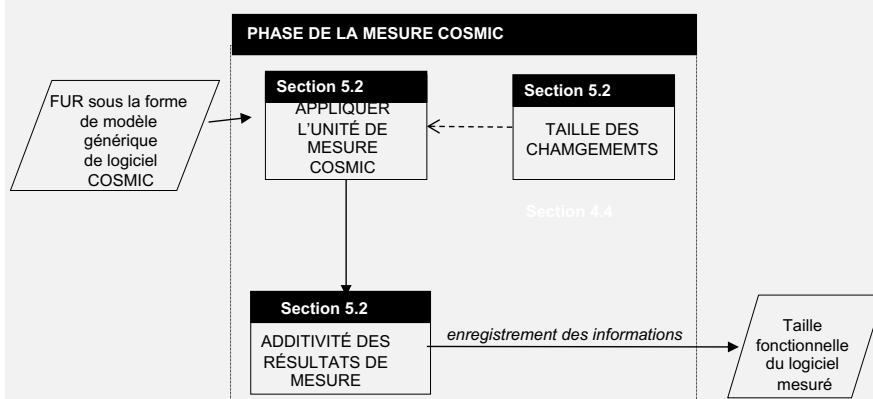
### 4.2 COSMIC avec Agile.

La méthode COSMIC est utilisée avec succès pour mesurer la taille des User Stories dans les développements logiciels Agile, qui peuvent avoir très peu de mouvements de données. Cette pratique est bien établie, avec de nombreux rapports sur les tailles PFC des itérations Agiles (ou « sprints »), agrégées à partir des tailles des User Stories individuelles, qui correspondent

très bien à l'effort pour développer l'itération (et bien mieux que la corrélation avec l'effort de tailles mesurées à l'aide de Story Points).

## 5. PHASE DE MESURAGE.

Le processus général de mesure d'un logiciel lorsque ses EFU ont été exprimées en termes de modèle de logiciel générique COSMIC est résumé dans la figure 5.1 ci-dessous.



5.1 – Le processus de la phase de mesurage COSMIC

### DIRECTIVE sur la règle 23 : Agrégation des tailles fonctionnelles

- Les tailles de logiciels ou de modifications de logiciels ne peuvent être additionnées que si elles sont mesurées au même niveau de granularité de processus fonctionnel de leurs EFU.
- Les tailles de logiciels et/ou les modifications des tailles de logiciels au sein d'une même couche ou de différentes couches ne sont additionnées seulement si cela est logique, aux fins de la mesure.
- La taille d'un logiciel est obtenue en additionnant les tailles de ses composants (quelle que soit la façon dont le morceau est décomposé) et en éliminant les contributions de taille des mouvements de données inter-composants.

Au sein de chaque couche identifiée, la fonction d'agrégation est entièrement évolutive. Un sous-total peut être généré pour des processus fonctionnels individuels ou pour tous les processus fonctionnels du logiciel, selon le but et la portée de l'exercice de mesure.

## 6. RAPPORTS DE MESURE.

Le résultat doit être communiqué et les données sur la mesure enregistrées afin de garantir que le résultat est toujours interprétable sans ambiguïté. Les résultats des mesures COSMIC doivent être rapportés et archivés conformément aux conventions suivantes.

### DIRECTIVE concernant la règle 25 : Étiquetage des mesures COSMIC.

Un résultat de mesure COSMIC est noté « x PFC (v) », où :

- « x » représente la valeur numérique de la taille fonctionnelle,
- « v » représente l'identification de la version de la méthode COSMIC standard utilisée pour obtenir la valeur numérique de la taille fonctionnelle « x ».

REMARQUE : Si une méthode d'approximation locale a été utilisée pour obtenir la mesure, mais que la mesure a été effectuée en utilisant les conventions d'une version COSMIC standard, la convention d'étiquetage ci-dessus est utilisée, mais l'utilisation de la méthode d'approximation doit être spécifiée quelque part.

**DIRECTIVE sur la règle 25 : Étiquetage des extensions locales COSMIC.**

Un résultat de mesure COSMIC utilisant des extensions locales est noté comme suit :

« X PFC (v.) + Z PF local », où :

- « x » représente la valeur numérique obtenue en agrégeant tous les résultats de mesure individuels selon la méthode COSMIC standard, version v,
- « v » représente l'identification de la version de la méthode COSMIC standard utilisée pour obtenir la valeur numérique de la taille fonctionnelle « x ».
- « z » représente la valeur numérique obtenue en agrégeant tous les résultats de mesure individuels obtenus à partir des extensions locales de la méthode COSMIC.