

ΤΙΤΛΟΣ ΕΡΓΟΥ: Υπηρεσίες Εμπειρογνωμοσύνης Για Ανάπτυξη Έργων Λογισμικού Σε Περιβάλλον SOA

ΑΝΑΘΕΤΟΥΣΑ ΑΡΧΗ: Γενική Γραμματεία Πληροφοριακών Συστημάτων

ΑΝΑΔΟΧΟΣ: Ινστιτούτο Πληροφοριακών Συστημάτων - Ι.Π.ΣΥ / Ε.Κ. «Αθηνά»

ΠΑΡΑΔΟΤΕΑ: Π1: Μεθοδολογία προμέτρησης και επιμέτρησης έργων SOA

Π2: Εγχειρίδιο εφαρμογής μεθοδολογίας προμέτρησης και επιμέτρησης έργων SOA

Έκδοση 2.0, 13/07/2012

ΤΟ ΕΡΓΟ ΣΥΓΧΡΗΜΑΤΟΔΟΤΕΙΤΑΙ ΑΠΟ ΤΟ ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ
(Ε.Τ.Π.Α.) ΚΑΙ ΑΠΟ ΕΘΝΙΚΟΥΣ ΠΟΡΟΥΣ



ΕΛΛΗΝΙΚΗ
ΔΗΜΟΚΡΑΤΙΑ



ΕΥΡΩΠΑΪΚΗ
ΕΝΩΣΗ



Ε.Π.
ΨΗΦΙΑΚΗ ΣΥΓΚΛΙΣΗ



ΕΘΝΙΚΟ ΣΤΡΑΤΗΓΙΚΟ
ΠΛΑΙΣΙΟ ΑΝΑΦΟΡΑΣ

ΤΑΥΤΟΤΗΤΑ ΤΟΥ ΕΓΓΡΑΦΟΥ

ΠΑΡΑΔΟΤΕΟ	Π1 - Μεθοδολογία προμέτρησης και επιμέτρησης έργων SOA Π2 - Εγχειρίδιο εφαρμογής μεθοδολογίας προμέτρησης και επιμέτρησης έργων SOA
ΕΚΔΟΣΗ	2.0
ΗΜΕΡΟΜΗΝΙΑ	13/07/2012
ΗΛΕΚΤΡΟΝΙΚΑ ΑΡΧΕΙΑ	FSM ΓΓΠΣ.v2.doc FSM Template ΓΓΠΣ.v2.xlt

ΙΣΤΟΡΙΚΟ ΑΛΛΑΓΩΝ

ΕΚΔΟΣΗ	ΗΜΕΡΟΜΗΝΙΑ	ΣΥΓΓΡΑΦΕΙΣ	ΑΙΤΙΑ ΑΛΛΑΓΗΣ
1.0	20/03/2012	ΟΜΑΔΑ ΕΡΓΟΥ	1 ^η Έκδοση
2.0	13/07/2012	ΟΜΑΔΑ ΕΡΓΟΥ	2 ^η Έκδοση μετά από παρατηρήσεις της ΕΠΠΕ

0	Εισαγωγή.....	6
0.1	Αντικείμενο και στόχοι του έργου.....	6
0.2	Σκοπός του παρόντος εγγράφου.....	6
P1.	Μεθοδολογία προμέτρησης και επιμέτρησης έργων SOA.....	8
1	Γενική περιγραφή της Υπηρεσιοστραφούς Αρχιτεκτονικής (Service Oriented Architecture - SOA).....	8
1.1	Ένας ορισμός για την Υπηρεσιοστραφή Αρχιτεκτονική.....	8
1.2	Χαρακτηριστικά και πλεονεκτήματα της SOA.....	9
1.3	Δομή επιπέδων της SOA.....	10
1.4	Οι Διαδικτυακές Υπηρεσίες (Web Services).....	11
2	Αλγοριθμικές μεθοδολογίες εκτίμησης Μεγέθους Λογισμικού.....	12
2.1	Το μοντέλο COCOMO II.....	12
2.2	Εφαρμοσιμότητα του μοντέλου COCOMO II σε έργα SOA.....	13
3	Νεότερες μεθοδολογίες που προτάθηκαν για την εκτίμηση κόστους SOA.....	15
3.1	Τεχνική SMART (Software Engineering Institute's Service Migration and Reuse Technique).....	15
3.1.1	Μεθοδολογία «Divide and Conquer».....	15
3.1.2	Μεθοδολογία SMAT-AUS.....	16
3.2	Εμπειρικές μεθοδολογίες και άλλες τεχνικές.....	17
3.3	Μεθοδολογίες Function Size Measurement κατά ISO/IEC 14143.....	17
3.3.1	IFPUG FPA 4.3.....	17
3.3.2	COSMIC-FFP Functional size measurement method v 3.0.1.....	18
3.3.3	Mk II Function Point Analysis 1.3.1 Unadjusted.....	19
3.3.4	NESMA FPA Method 2.1 Unadjusted.....	19
3.3.5	FiSMA FSM 1.1.....	20
3.4	Συμπέρασμα – Επιλογή μεθοδολογίας.....	21
4	Εκτίμηση μεγέθους λογισμικού χρησιμοποιώντας τη μεθοδολογία IFPUG FPA 4.3.....	22
4.1	Βασικοί Ορισμοί.....	22
4.1.1	Λειτουργικό Μέγεθος.....	22
4.1.2	Λειτουργικό Σημείο.....	22
4.1.3	Στοιχειώδης Διαδικασία.....	22
4.1.4	Λειτουργικές Διαδικασίες Δεδομένων.....	22
4.1.5	Λειτουργικές Διαδικασίες Συναλλαγών.....	22
4.1.6	Στοιχείο Δεδομένων.....	23
4.1.7	Στοιχείο Εγγραφής.....	23
4.1.8	Αναφερόμενο Αρχείο.....	23
4.1.9	Λοιποί Ορισμοί.....	23
4.2	Μέτρηση Λειτουργικού Μεγέθους Λογισμικού.....	24

4.2.1	Συγκέντρωση της τεκμηρίωσης για το Μετρούμενο Λογισμικό	24
4.2.2	Προσδιορισμός του Τύπου της Μέτρησης	25
4.2.3	Προσδιορισμός των Ορίων του Μετρούμενου Λογισμικού	25
4.2.4	Μέτρηση επιμέρους Λειτουργικών Μεγεθών	26
4.2.5	Τελικός υπολογισμός Λειτουργικού Μεγέθους	30
5	Χρησιμοποιώντας τη μεθοδολογία IFPUG FPA 4.3 για έργα SOA	32
5.1	Πεδίο εφαρμογής της IFPUG FPA 4.3 σε έργα SOA.....	32
5.1.1	Δράσεις που δεν συνεισφέρουν Λειτουργικά Σημεία	32
5.2	Μετρώντας επαναχρησιμοποιούμενες μονάδες λογισμικού με την IFPUG FPA 4.3.....	33
5.3	Παραδείγματα μέτρησης επαναχρησιμοποιούμενων μονάδων λογισμικού με την IFPUG FPA 4.3 36	
5.3.1	Σενάριο 1 – Κλήση Web Service για την πρόσβαση σε δεδομένα.....	36
5.3.2	Σενάριο 2 – Κλήση Web Service για την αποστολή δεδομένων	36
5.4	Διαφοροποίηση στη μέτρηση Λειτουργικού Μεγέθους μεταξύ RESTful και SOAP Web Services .	37
6	Μεθοδολογίες για την προεκτίμηση αξιοποιώντας δεδομένα IFPUG	40
6.1	Ερευνητικές προσεγγίσεις.....	40
6.2	Χρήση των μεθόδων του ISBSG.....	40
6.3	Ορόσημα και μετρούμενα μεγέθη.....	41
6.3.1	Μέτρηση της παρέκκλισης του εύρους του έργου	42
6.3.2	Μέτρηση των σφαλμάτων υλοποίησης	42
6.4	Πρόταση εφαρμογής μεθόδου προμέτρησης και επιμέτρησης έργων στη Γ.Γ.Π.Σ.....	42
6.5	Εναλλακτική προσέγγιση.....	44
7	Εφαρμόζοντας τη μεθοδολογία IFPUG FPA 4.3 στη Γ.Γ.Π.Σ.....	45
7.1	Εσωτερική δομή στη Γ.Γ.Π.Σ. για την εφαρμογή της μεθοδολογίας.....	45
7.1.1	Αρχές λειτουργίας	45
7.1.2	Απαιτούμενοι πόροι.....	46
7.1.3	Προγραμματισμός και οργάνωση μετρήσεων	48
7.2	Οφέλη από τη λειτουργία της Ο.Ε.Μ.Ε.Λ.	49
8	Βιβλιογραφία & Αναφορές	50
P2:	Εγχειρίδιο εφαρμογής μεθοδολογίας προμέτρησης και επιμέτρησης έργων SOA	53
1	Εργαλείο Υπολογισμού Λειτουργικού Μεγέθους Λογισμικού	53
2	Οδηγίες Χρήσης Εργαλείου Υπολογισμού	53
2.1	Εισαγωγή	53
2.2	Καταχώρηση Ταυτότητας Έργου	53
2.3	Εκτέλεση Μέτρησης Λειτουργικών Σημείων.....	54
2.3.1	Μέτρηση ILFs.....	54
2.3.2	Μέτρηση EIFs.....	55

2.3.3	Μέτρηση Εξωτερικών Εισόδων	55
2.3.4	Μέτρηση Εξωτερικών Εξόδων	56
2.3.5	Μέτρηση Εξωτερικών Αναζητήσεων	56
2.4	Υπολογισμός Λειτουργικού Μεγέθους	57
3	Εικόνες (Screenshots) Χρήσης Εργαλείου Υπολογισμού	58

0 Εισαγωγή

Η παρούσα μελέτη «Υπηρεσίες εμπειρογνωμοσύνης για ανάπτυξη έργων λογισμικού σε περιβάλλον SOA» στο πλαίσιο της απόφασης ανάθεσης με αρ. πρωτ. Δ51011493 ΕΞ 2012 / 20-1-2012 (ΑΔΑ: ΒΟΝ8Η-9ΧΤ) περιλαμβάνει τα δύο παραδοτέα του έργου και συγκεκριμένα:

- Π1. «Μεθοδολογία προμέτρησης και επιμέτρησης έργων SOA» όπου τεκμηριώνεται η προτεινόμενη προς υιοθέτηση μεθοδολογία, με αναφορές σε συμμόρφωση με πρότυπα και ακολουθούμενες καλές πρακτικές (best practices).
- Π2. «Εγχειρίδιο εφαρμογής μεθοδολογίας προμέτρησης και επιμέτρησης έργων SOA» όπου παρέχονται υποδείγματα προς συμπλήρωση (templates) και οδηγίες εφαρμογής της μεθόδου για την διενέργεια της προεκτίμησης ή της αποτίμησης του μεγέθους του λογισμικού.

Επιπροσθέτως, για την εφαρμογή της μεθόδου αναπτύχθηκαν φύλλα υπολογισμού χρησιμοποιώντας Microsoft Excel ως υποδείγματα μέτρησης (templates) εφοδιασμένα με συναρτήσεις υπολογισμού σύμφωνα με τις οδηγίες της προτεινόμενης μεθοδολογίας εκτίμησης Λειτουργικού Μεγέθους. Οι εκτελούμενες συναρτήσεις (μετά από στοιχειώδεις παραδοχές) καταλήγουν σε εκτίμηση ανθρωπομηνών για το υπό εξέταση έργο, ενώ τα αποτελέσματα καταγράφονται σε αυτοτελές φύλλο εργασίας.

Το ψηφιακό αρχείο υπολογισμού υλοποιημένο με πρότυπο (template) του Microsoft Excel επισυνάπτεται ως παράρτημα των παραδοτέων του έργου.

0.1 Αντικείμενο και στόχοι του έργου

Αντικείμενο του έργου είναι ο προσδιορισμός της προσφορότερης μεθοδολογίας για την προεκτίμηση και την αποτίμηση του μεγέθους λογισμικού που θα αναπτύσσεται σε περιβάλλον Service Oriented Architecture (SOA) τεχνολογιών.

Μέχρι πρόσφατα για αντίστοιχους σκοπούς χρησιμοποιούνταν ευρύτατα η μέθοδος COCOMO II, η οποία όμως εμφανίζει σημαντικά προβλήματα κατά την εφαρμογή της σε έργα ΤΠΕ με Υπηρεσιοστραφή Αρχιτεκτονική (SOA)..

Η εκτίμηση του μεγέθους (άρα κατ' επέκταση και του κόστους) υλοποίησης, ενός έργου λογισμικού SOA, αποδεικνύεται πολυπλοκότερη διαδικασία σε σύγκριση με τις κλασσικές εφαρμογές (web, desktop, rich internet). Για το λόγο αυτό είναι απαραίτητη (αλλά και επίκαιρη) η υιοθέτηση μιας πιο σύγχρονης προσέγγισης για τις ανάγκες προεκτίμησης και αποτίμησης του μεγέθους ανάπτυξης λογισμικού σε περιβάλλοντα SOA.

0.2 Σκοπός του παρόντος εγγράφου

Στο Π1 «Μεθοδολογία προμέτρησης και επιμέτρησης έργων SOA» παρουσιάζεται η Υπηρεσιοστραφής Αρχιτεκτονική (Service-Oriented Architecture – SOA) και περιγράφονται οι μέθοδοι εκτίμησης κόστους για έργα λογισμικού που βασίζονται σε κλασσικούς τρόπους ανάπτυξης και της κυριότερης από αυτές, την COCOMO II.

Εν συνεχεία αναλύονται οι πιο σύγχρονες μέθοδοι εκτίμησης μεγέθους λογισμικού για περιβάλλοντα SOA, όπως αυτές έχουν παρουσιασθεί σε επιστημονικά περιοδικά (IEEE), στην καθημερινή πρακτική και από διεθνώς αναγνωρισμένα πρότυπα (ISO).

Τέλος προσεγγίζεται τόσο θεωρητικά όσο και με παραδείγματα η προτεινόμενη μεθοδολογία για την εκτίμηση μεγέθους λογισμικού από τη Γ.Γ.Π.Σ., λαμβάνοντας υπόψη τα ιδιαίτερα χαρακτηριστικά της Υψηλοισοσταφούς Αρχιτεκτονικής αλλά και το περιβάλλον του Φορέα.

Στο Π2 «Εγχειρίδιο εφαρμογής μεθοδολογίας προμέτρησης και επιμέτρησης έργων SOA» παρέχονται οδηγίες για την χρήση του πρότυπου (template) εκτίμησης μεγέθους λογισμικού, προς χρήση κατά την διενέργεια της προεκτίμησης ή της αποτίμησης.

Π1. Μεθοδολογία προμέτρησης και επιμέτρησης έργων SOA

1 Γενική περιγραφή της Υπηρεσιοστραφούς Αρχιτεκτονικής (Service Oriented Architecture - SOA)

Στη σημερινή επιχειρηματική πραγματικότητα ένας συνεχώς αυξανόμενος αριθμός επιχειρήσεων, φορέων και οργανισμών αναζητούν τρόπους ώστε να ενοποιήσουν και να επεκτείνουν τις πληροφοριακές υποδομές και αρχιτεκτονικές τους. Τις περισσότερες φορές η προσπάθεια αυτή καταλήγει στην προσθήκη ενός επιπλέον επιπέδου τεχνολογίας στις ήδη υπάρχουσες αυξάνοντας την πολυπλοκότητα του πληροφοριακού συστήματος του οργανισμού.

Η εμφάνιση της Υπηρεσιοστραφούς Αρχιτεκτονικής (Service-Oriented Architecture – SOA), αποτέλεσε τον καταλύτη για την απλοποίηση των πληροφοριακών συστημάτων, δίνοντας έμφαση στην ευελιξία, εστιάζοντας στις υπηρεσίες / επιχειρησιακές δραστηριότητες.

Η SOA ενσωμάτωσε ένα πλήθος πρότυπα (standards), καλύπτοντας τις απαιτήσεις ολοκλήρωσης και ενοποίησης που απαιτούνται για τη σχεδίαση, υλοποίηση και θέση σε λειτουργία σύνθετων εφαρμογών Αυτοματοποίησης Επιχειρησιακών Διαδικασιών.

1.1 Ένας ορισμός για την Υπηρεσιοστραφή Αρχιτεκτονική

Παρότι δεν υπάρχει ένας απολύτως αποδεκτός ορισμός για την Υπηρεσιοστραφή Αρχιτεκτονική, δεν μπορούμε να παραβλέψουμε ότι ο ίδιος ο τίτλος της εμπεριέχει τη θεμελιώδη αρχή για έμφαση στις «υπηρεσίες».

Ο Οργανισμός OASIS (Organization for the Advancement of Structured Information Standards) ορίζει [1] την Υπηρεσιοστραφή Αρχιτεκτονική ως

Ένα πρότυπο για την οργάνωση και χρήση καταναμημένων δυνατοτήτων, που πιθανόν να βρίσκονται υπό τον έλεγχο διαφορετικών τομέων ευθύνης, προσφέροντας ένα ενιαίο τρόπο για την παροχή, εξεύρεση, διάδραση και αξιοποίηση δυνατοτήτων ώστε να επιτυγχάνονται τα επιθυμητά αποτελέσματα με τρόπο συνεπή προς μετρήσιμες προϋποθέσεις και προσδοκίες¹.

Πρακτικά η SOA είναι ένα τεχνολογικό πλαίσιο που επιτρέπει σε περισσότερα του ενός πληροφοριακά συστήματα (εντός ή/και εκτός ενός οργανισμού), να κοινοποιούν και να έχουν πρόσβαση, σε υπηρεσίες σαφώς καθορισμένες, καθώς και δεδομένα και πληροφορίες συνδεδεμένες με αυτές τις υπηρεσίες.

Η SOA στηρίζεται στο σχεδιασμό των εφαρμογών με επίκεντρο τις υπηρεσίες, οι οποίες αποτελούν υλοποιήσεις επιχειρησιακών διαδικασιών, που καθορίζονται σε σχέση με το τι κάνουν, προσδιορίζονται στο πλαίσιο μηνυμάτων που ανταλλάσσονται και είναι προσπελάσιμες μέσω δικτύου.

¹ «A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations»

1.2 Χαρακτηριστικά και πλεονεκτήματα της SOA

Μέχρι την δημιουργία της Υπηρεσιοστραφούς Αρχιτεκτονικής, και ακολουθώντας τις συμβατικές αρχιτεκτονικές πληροφορικής, οι επιχειρησιακές διαδικασίες, οι εφαρμογές και τα δεδομένα ήταν συχνά «κλειδωμένα» σε ανεξάρτητες και ασύμβατες μεταξύ τους περιοχές. Αυτές οι λύσεις ήταν δαπανηρές στη συντήρησή τους και υποχρέωναν τους χρήστες να περιηγούνται σε ξεχωριστά δίκτυα, εφαρμογές και βάσεις δεδομένων προκειμένου να εκτελέσουν συγκεκριμένες εργασίες.

Πλέον υιοθετώντας τη SOA οι χρήστες δε χρειάζεται να συνδέονται σε πολλά συστήματα, να αναζητούν σχετικά δεδομένα και να ενοποιούν οι ίδιοι τα αποτελέσματα. Τα δεδομένα για τις δραστηριότητες επιχειρηματικών διαδικασιών μπορούν να προσφέρονται πλέον με τη μορφή μιας ενοποιημένης υπηρεσίας, μέσα από μία και μοναδική εφαρμογή.

Τα συστήματα που υλοποιούνται με βάση την Υπηρεσιοστραφή Αρχιτεκτονική, έχουν τα εξής εγγενή χαρακτηριστικά [2] [3] [4]:

- Επαναχρησιμοποίηση (Reusability): Η επιχειρησιακή λογική επιμερίζεται σε υπηρεσίες ώστε να προωθείται η επαναχρησιμοποίηση κώδικα και αυτόνομων δομικών μονάδων λογισμικού (components).
- Σύμβαση Υπηρεσιών (Service contract): Οι υπηρεσίες υπακούουν σε μια συμφωνία επικοινωνίας, όπως αυτή περιγράφεται συλλογικά σε ένα ή περισσότερα έγγραφα, καλύπτοντας στοιχεία για την αναγνώριση της υπηρεσίας, την ιδιοκτησία και την περιγραφή της υπηρεσίας τις λειτουργικές και τις μη λειτουργικές απαιτήσεις, της διαδικασίες και τους τρόπους κλήσης της υπηρεσίας.
- Χαλαρή Συνδεσιμότητα (Loose coupling): Οι υπηρεσίες διατηρούν μια σχέση που περιορίζει στο ελάχιστο τις εξαρτήσεις και απαιτεί μόνο να γνωρίζουν ή μία την «ύπαρξη» της άλλης. Έτσι είναι δυνατόν να γίνονται αλλαγές στον τρόπο υλοποίησης των υπηρεσιών χωρίς να επηρεάζονται άλλα μέρη της εφαρμογής.
- Αφαίρεση (Abstraction): Οι υπηρεσίες αλληλεπιδρούν με το «εξωτερικό περιβάλλον» αποκλειστικά μέσω των δημοσιευμένων διεπαφών, αποκρύπτοντας κάθε εσωτερική λογική λειτουργίας.
- Δυνατότητα Σύνθεσης (Composability): Ομάδες υπηρεσιών μπορούν να συντονίζονται και να ενορχηστρώνονται ώστε να δημιουργούν σύνθετες υπηρεσίες.
- Αυτονομία (Autonomy): Οι υπηρεσίες έχουν απόλυτο έλεγχο στην λογική που ενσωματώνουν.
- Απουσία Μνήμης (Statelessness): Οι υπηρεσίες για την εκτέλεσή τους απαιτούν την διακράτηση (αποθήκευση) των ελάχιστων δυνατών δεδομένων.
- Δυνατότητα Ανακάλυψης (Discoverability): Οι Υπηρεσίες σχεδιάζονται και περιγράφονται εξωστρεφώς, ώστε να είναι δυνατή η εύρεσή τους από ένα Μητρώο Υπηρεσιών (service Registry) ή παρόμοιο μηχανισμό. Με τον τρόπο αυτό ανακαλύπτονται και αρχικοποιούνται οι συνδέσεις μεταξύ του παρόχου της υπηρεσίας και του «καταναλωτή» της.

Τα βασικότερα πλεονεκτήματα της SOA περιλαμβάνουν μεταξύ άλλων:

- Μείωση της προσπάθειας για ανάπτυξη εφαρμογών λογισμικού από την αρχή και την ολοκλήρωσή τους στις υφιστάμενες υποδομές ΤΠΕ, αξιοποιώντας τα χαρακτηριστικά επαναχρησιμοποίησης υπηρεσιών.
- Ευελιξία στην αλλαγή των επιχειρησιακών διαδικασιών με ταχύτητα και κατά περίπτωση, λαμβάνοντας υπ' όψιν τις υφιστάμενες υπηρεσίες και ροές πληροφοριών.

- Δυνατότητα δυναμικής παρακολούθησης σε πραγματικό χρόνο των σημείων ανταλλαγής πληροφορίας ή των υπηρεσιών. Με τον τρόπο αυτό εξασφαλίζεται η σωστή λειτουργία των επιχειρησιακών διαδικασιών, ενώ σε δεύτερο επίπεδο δίνεται η δυνατότητα τροποποίησης ή ρύθμισης των διαδικασιών για την βελτιστοποίηση της λειτουργίας του οργανισμού.
- Απομακρυσμένη πρόσβαση των υπηρεσιών έτσι ώστε να δίνεται η δυνατότητα σε εξωτερικούς του οργανισμού φορείς να διαλειτουργούν και να μοιράζονται τις υπηρεσίες που παρέχει ο οργανισμός αυτός.
- Διαφάνεια φυσικής τοποθεσίας υλοποίησης υπηρεσίας, υπό την έννοια ότι ο χρήστης μιας υπηρεσίας δεν γνωρίζει πού βρίσκεται η υλοποίηση της υπηρεσίας.

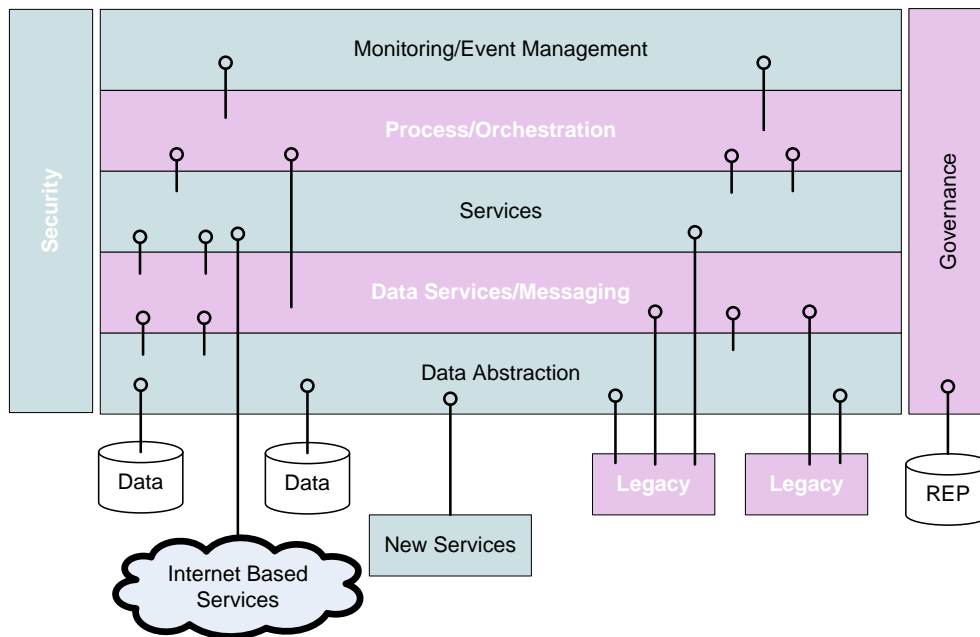
1.3 Δομή επιπέδων της SOA

Η προσπάθεια για το διαμοιρασμό κοινών διαδικασιών δεν είναι καινούρια. Στο παρελθόν υπήρξαν αντίστοιχες προσπάθειες με χαρακτηριστικότερο παράδειγμα την αρχιτεκτονική πελάτη εξυπηρετητή πολλαπλών επιπέδων (multi-tier client/server), η οποία παρέχει ένα σύνολο διαμοιραζόμενων υπηρεσιών σε ένα κοινό εξυπηρετητή ο οποίος αποτελεί την υποδομή που αναλαμβάνει το διαμοιρασμό και την επαναχρησιμοποίηση τους.

Η επαναχρησιμοποίηση είναι η λέξη κλειδί, καθώς στην περίπτωση της SOA αφορά την τόσο τις υπηρεσίες όσο και την πληροφορία με την οποία είναι συνδεδεμένες. Όπως είναι λογικό μια ομάδα υπηρεσιών οι οποίες διαμοιράζονται ανάμεσα στις εφαρμογές ενός φορέα, επιτυγχάνουν την αυξημένη επαναχρησιμοποίηση και μειώνουν την ανάγκη για πλεονάζουσες και επικαλυπτόμενες εφαρμογές.

Η μοναδικότητα της SOA έγκειται στο γεγονός ότι αποτελεί, όχι μόνο στρατηγική αλλά ταυτόχρονα και ένα σύνολο τεχνολογιών που μπορούν να χρησιμοποιηθούν για τη δημιουργία και τη διαχείριση διαδικτυακών υπηρεσιών.

Όπως προκύπτει από το επόμενο διάγραμμα, η SOA επιτρέπει χρησιμοποιώντας ένα ενδιάμεσο δυναμικό επίπεδο (Process/Orchestration), να προκύψουν σημαντικές αλλαγές στις επιχειρησιακές διαδικασίες (Business Processes) χωρίς να δρομολογηθούν τροποποιήσεις σε όλα τα συνδεδεμένα συστήματα.



Σχήμα 1: Μεταμοντέλο της αρχιτεκτονικής SOA (SOA metamodel)

Οι διαφορετικές εφαρμογές που συμμετέχουν σε μια τέτοια αρχιτεκτονική ανταλλάσσουν δεδομένα συμμετέχοντας στις επιχειρησιακές διαδικασίες ανεξάρτητα από την πλατφόρμα υλικού (hardware), το λειτουργικό σύστημα και περιβάλλον ή ακόμα και την γλώσσα προγραμματισμού που υλοποιούν και υποστηρίζουν τις εφαρμογές αυτές.

1.4 Οι Διαδικτυακές Υπηρεσίες (Web Services)

Θεμελιώδες συστατικό της SOA αρχιτεκτονικής είναι η διαδικτυακή υπηρεσία. Μια διαδικτυακή υπηρεσία (Web Service) είναι ένα στοιχείο λογισμικού που αναγνωρίζεται από ένα Ενιαίο Αναγνωριστικό Πόρου (Uniform Resource Identifier – URI), ενώ οι δημόσιες διεπαφές και συνάψεις της ορίζονται και περιγράφονται με χρήση XML. Ο ορισμός της διαδικτυακής υπηρεσίας γίνεται με τρόπο ώστε να μπορεί να «ανακαλυφθεί» από άλλα συστήματα. Στην περίπτωση αυτή τα συστήματα μπορούν να αλληλεπιδράσουν με τις διαδικτυακές υπηρεσίες, χρησιμοποιώντας τον τρόπο που περιγράφεται στον ορισμό της και χρησιμοποιώντας XML μηνύματα τα οποία διαβιβάζονται μέσω διαδικτύου [5].

Η χρήση Υπηρεσιών Διαδικτύου είναι ιδιαίτερα αξιόπιστη, λόγω της χρήσης συγκεκριμένων διαδεδομένων πρωτοκόλλων (λ.χ. SOAP - Simple Object Access Protocol, WSDL – Web Services Description Language,, UDDI – Universal Description, Discovery, and Integration, XML, HTTP, κ.α.). Αξίζει να σημειώσουμε ότι παρότι η Υπηρεσιοστραφής Αρχιτεκτονική συνήθως στηρίζεται στη χρήση Υπηρεσιών Διαδικτύου, αυτές μπορούν να αξιοποιηθούν και να χρησιμοποιηθούν και από άλλες μεθοδολογίες και αρχιτεκτονικές σχεδίασης λογισμικού.

2 Αλγοριθμικές μεθοδολογίες εκτίμησης Μεγέθους Λογισμικού

Η εκτίμηση του κόστους υλοποίησης ενός έργου λογισμικού κατά κανόνα εκτελείται στο στάδιο του σχεδιασμού και αφορά την κατά το δυνατό λεπτομερέστερη πρόβλεψη και αποτύπωση του κόστους των πόρων που θα απαιτηθούν για την ολοκλήρωση του έργου.

Συχνά όμως καταγράφονται αστοχίες στην εκτίμηση κόστους για έργα λογισμικού, οδηγώντας τελικά σε υλοποιήσεις με μεγάλη χρονική υστέρηση, αποκλίσεις προϋπολογισμού και προβληματική ποιότητα των τελικών παραδοτέων.

Μεγάλη διάδοση σημείωσαν μετά τη δεκαετία του 1980 οι αλγοριθμικές μεθοδολογίες για την εκτίμηση του μεγέθους (και του κόστους) ενός λογισμικού. Επρόκειτο για το πρώτο σημαντικό άλμα στον τρόπο υπολογισμού κόστους έργων λογισμικού. Καθεμιά από τις μεθοδολογίες αυτές είχε ως κεντρικό πυρήνα έναν αλγόριθμο υπολογίζει την εκτίμηση για το μέγεθος/κόστος σαν συνάρτηση ιστορικών δεδομένων καθώς και του τύπου του λογισμικού.

Από το σύνολο των αλγοριθμικών μοντέλων υπολογισμού, τη μεγαλύτερη διάδοση σημείωσε το COConstructive COSt MOdel (COCOMO) που αναπτύχθηκε από τον Barry Boehm τη δεκαετία του 1970 και δημοσιεύτηκε το 1981, αξιοποιώντας δεδομένα από 63 έργα λογισμικού της εταιρείας TRW Aerospace.

Το COCOMO έχει τρεις εκδόσεις, Βασική, Ενδιάμεση και Προηγμένη (η διαφοροποίηση σχετίζεται με το βαθμό ασάφειας και την ταχύτητα εκτέλεσης της εκτίμησης). Βασίζεται σε εξισώσεις με κύρια παράμετρο τις εκτιμώμενες γραμμές πηγαίου κώδικα (Source lines of code – SLOC). Το μέγεθος προσαρμόζεται σύμφωνα με 15 παράγοντες που επηρεάζουν το κόστος (cost drivers).

2.1 Το μοντέλο COCOMO II

Το τελευταίο διάστημα Οργανισμοί και Εταιρίες, υιοθετούν για την υλοποίηση των έργων λογισμικού νέες διαδικασίες και πλατφόρμες. Αυτές περιλαμβάνουν: συνεργατικές διαδικασίες υλοποίησης λογισμικού (collaborative software processes), γλώσσες προγραμματισμού 4ης γενιάς, γεννήτριες κώδικα, χρήση λογισμικού commercial off the shelf (COTS). Οι νέες αυτές πρακτικές οδηγούν στην παραγωγή καλύτερης ποιότητας λογισμικού εισάγοντας νέα μοντέλα κύκλου ζωής (life cycle models) [6].

Παράλληλα με την εξέλιξη των τεχνικών υλοποίησης, προέκυψε και η ωρίμανση των αλγοριθμικών μεθοδολογιών εκτίμησης μεγέθους λογισμικού. Άλλωστε το COCOMO 81 αναπτύχθηκε με την παραδοχή ότι χρησιμοποιείται το μοντέλο κύκλου ζωής του «καταρράκτη» και ότι όλο το λογισμικό κατασκευάζεται εκ του μηδενός. Έτσι το 1995 (και ενώ είχε προηγηθεί η ανακοίνωση του Ada COCOMO το 1989) ξεκίνησε η αναθεώρηση του μοντέλου COCOMO και τελικά το 2000 ανακοινώθηκε μια νεότερη παραλλαγή του μοντέλου.

Το αλγοριθμικό μοντέλο COCOMO II είναι προσαρμοσμένο για τον υπολογισμό του κόστους στις νέες μορφές υλοποίησης λογισμικού, ενσωματώνοντας νέες δυνατότητες.

Οι κύριοι στόχοι πίσω από τη δημιουργία του μοντέλου COCOMO II ήταν:

- Να καθιερώσει ένα μοντέλο εκτίμησης κόστους και χρονοπρογραμματισμού, εναρμονισμένο με τις σύγχρονες τεχνολογικές εξελίξεις.
- Να αναπτύξει μια βάση δεδομένων με στοιχεία κόστους ανάπτυξης λογισμικού καθώς και σχετικών εργαλείων, για την διαρκή υποστήριξη και βελτίωση του μοντέλου.
- Να παρέχει ένα πλαίσιο ποσοτικής ανάλυσης καθώς και ένα σύνολο εργαλείων και τεχνικών για την εκτίμηση της επίδρασης των τεχνολογικών αλλαγών τόσο στα κόστη κύκλου ζωής του λογισμικού όσο και στα χρονοδιαγράμματα.

Το αλγοριθμικό μοντέλο COCOMO II χρησιμοποιεί τρία «μεγέθη» (sizing quantities) για την εκτίμηση του μεγέθους/κόστους ενός έργου λογισμικού: Σημεία Αντικειμένων (Object Points), Μη Σταθμισμένα Λειτουργικά Σημεία (Unadjusted Function Points) και Γραμμές Πηγαίου Κώδικα (Source Lines of Code).

Το COCOMO II ουσιαστικά αποτελείται από τρία επιμέρους υπο-μοντέλα [7], καθένα εκ των οποίων χρησιμοποιείται, ανάλογα με το βαθμό ωριμότητας της υλοποίησης:

1. Το υπο-μοντέλο Σύνθεσης Εφαρμογής (Application Composition model) αφορά την προκαταρκτική εκτίμηση για ένα έργο (λ.χ. κατά τη φάση δημιουργίας «πρωτότυπου» - prototyping) με βασικό σκοπό τον εντοπισμό και την πρόληψη ζητημάτων υψηλού ρίσκου (επιλογή περιβάλλοντος ανάπτυξης, διεπαφών χρήστη κ.ο.κ).
2. Το υπο-μοντέλο Πρώιμης Σχεδίασης (Early Design model) αφορά την εκτίμηση σε μια φάση αρχικού σχεδιασμού του έργου, όταν υπάρχει μια σαφέστερη εικόνα για τις απαιτήσεις αλλά λίγα είναι οριστικοποιημένα για τους συντελεστές κόστους του έργου. Στο υπο-μοντέλο αυτό αξιοποιούνται Λειτουργικά Σημεία (function points) σε συνδυασμό με συμπληρωματικές παραμέτρους κόστους.
3. Το υπο-μοντέλο Μετά-Αρχιτεκτονικής (Post-Architecture model) αφορά την εκτίμηση κατά τη φάση υλοποίησης και συντήρησης ενός έργου λογισμικού, όταν η τελική αρχιτεκτονική του συστήματος έχει οριστικοποιηθεί και είναι διαθέσιμες περισσότερες πληροφορίες για το έργο.

2.2 Εφαρμοσιμότητα του μοντέλου COCOMO II σε έργα SOA

Παρά τη μεγάλη διάδοσή του, το αλγοριθμικό μοντέλο COCOMO II δέχεται πλέον ισχυρή κριτική ως προς την αποτελεσματικότητά του σε ένα τεχνολογικά περιβάλλον που αλλάζει ραγδαία. Βασικότερες αδυναμίες του μοντέλου θεωρούνται:

- Βασικές ιδέες του μοντέλου παραμένουν ως και σήμερα ασαφείς, υπηρετώντας μια λογική «μαύρου κουτιού» (black box) [8].
- Στηρίζεται σε μεγάλο βαθμό στην μέτρηση γραμμών πηγαίου κώδικα, μετρική που δεν είναι πλέον ασφαλής λόγω των διαφοροποιήσεων που εισάγει η επιλογή γλώσσας προγραμματισμού καθώς και άλλες τεχνολογικές επιλογές για την αυτόματη δημιουργία κώδικα (automatic generation of code, coding based on frameworks κλπ) [9].
- Η «καρδιά» του μοντέλου βασίζεται στην παραδοχή υιοθέτησης το μοντέλο κύκλου ζωής του «καταρράκτη» (waterfall) [10].
- Ο χρόνος που απαιτείται για την εκτίμηση μεγέθους μικρών έργων λογισμικού είναι δυσανάλογα μεγάλος [10].
- Κάποιες επεκτάσεις ή παραλλαγές του μοντέλου που έχουν προταθεί για τον εκσυγχρονισμό του, βρίσκονται ακόμη σε πειραματικό στάδιο και δεν θεωρούνται ασφαλείς [10].

Πλέον των ανωτέρω και ιδίως για έργα υλοποίησης λογισμικού που αξιοποιούν την Υπηρεσιοστραφή Αρχιτεκτονική εντοπίζονται και οι εξής αδυναμίες για το αλγοριθμικό μοντέλο COCOMO II:

- Για τα έργα λογισμικού Υπηρεσιοστραφούς Αρχιτεκτονικής ο όγκος παραγωγής πηγαίου κώδικα δεν αποτελεί ασφαλές κριτήριο του μεγέθους.
- Η χρήση των Object points ή Application points που προβλέπουν τα υπο-μοντέλα της COCOMO II δεν είναι δόκιμη σε έργα Υπηρεσιοστραφούς Αρχιτεκτονικής, καθώς οι μετρικές αυτές προκύπτουν από την καταμέτρηση λειτουργικών οθονών, αναφορών, ομάδων program modules.
- Τα μοντέλα COCOMO II αξιοποιεί και στηρίζεται σε «ιστορική πληροφορία» για το μέγεθος και το κόστος υλοποίησης έργων λογισμικού, παραγνωρίζοντας ότι τα μέσα και τα εργαλεία κατά την ανάπτυξη έργων SOA είναι μη συγκρίσιμα.
- Το μοντέλο COCOMO II δεν επαρκεί για να εκτιμήσει ορθά την προσπάθεια που χρειάζεται για την επαναχρησιμοποίηση υπηρεσιοστραφών πόρων [11].
- Τέλος η Υπηρεσιοστραφή Αρχιτεκτονική υιοθετεί μια διαδικασία πιο σύνθετη από την «απλή» επαναχρησιμοποίηση (reuse) των services. Παρέχει τη δυνατότητα στα services να αλλάζουν για να εξυπηρετήσουν μια καινούρια λειτουργία (adapt), να κατηγοριοποιούνται (categorize), να ανακαλύπτονται (discover) και τέλος να μπορούν να βελτιώνουν και να εξελίξουν (evolve) τις λειτουργίες τους.

3 Νεότερες μεθοδολογίες που προτάθηκαν για την εκτίμηση κόστους SOA

Για την ακριβέστερη εκτίμηση κόστους σε ένα έργο λογισμικού Υπηρεσιοστραφούς Αρχιτεκτονικής έχουν προταθεί τα τελευταία χρόνια αρκετές διαφορετικές μεθοδολογίες, χρησιμοποιώντας ένα συνδυασμό των δύο αυτών κατηγοριών. Αν η ανάλυση των προδιαγραφών είναι εκτενής τότε η αποτελεσματικότητά τους βελτιώνεται. Παρακάτω παρουσιάζονται μερικές από τις πιο γνωστές μεθοδολογίες.

3.1 Τεχνική SMART (Software Engineering Institute's Service Migration and Reuse Technique)

Η μεθοδολογία “Service-Oriented Migration and Reuse Technique” (SMART) δημιουργήθηκε για να βοηθήσει τους οργανισμούς να αναλύσουν τις δυνατότητες των υπάρχοντων εφαρμογών για να χρησιμοποιηθούν σαν υπηρεσίες (services) σε μια Υπηρεσιοστραφή Αρχιτεκτονική. Η μεθοδολογία SMART προήλθε από την μεθοδολογία Options Analysis for Reengineering (OAR) η οποία χρησιμοποιήθηκε για την ανάλυση της επαναχρησιμοποίησης υπάρχοντων οντοτήτων λογισμικού (legacy components) [12]. Η SMART συγκεντρώνει ένα μεγάλο εύρος πληροφοριών για τα υπάρχοντα στοιχεία (legacy components), την αρχιτεκτονική της πλατφόρμας SOA, και τις πιθανές υπηρεσίες που μπορούσαν να παραχθούν από την πολιτική μετάβασης (migration) σαν βασικό παραδοτέο της μεθοδολογίας. Η SMART παράγει και κάποια αποτελέσματα τα οποία είναι χρήσιμα σε έναν οργανισμό ώστε να αποφασίσει αν θα προχωρήσει στην μετάβαση στη νέα πλατφόρμα.

Οι διαδικασίες συλλογής των πληροφοριών περιγράφονται στον οδηγό Service Migration Interview Guide (SMIG). Ο οδηγός SMIG περιέχει ερωτήσεις οι οποίες περιγράφουν άμεσα και σκιαγραφούν το κενό μεταξύ της υπάρχουσας και της μελλοντικής αρχιτεκτονικής, το σχεδιασμό, την γλώσσα προγραμματισμού καθώς και την ανθρωποπροσπάθεια που χρειάζεται για τη μετάβαση. Η χρήση του SMIG διασφαλίζει την ευρύτερη κάλυψη όλων των παραμέτρων που επηρεάζουν τα κόστη μετάβασης των υπάρχοντων εφαρμογών σε υπηρεσίες (services).

3.1.1 Μεθοδολογία «Divide and Conquer»

Το βασικό σκεπτικό της μεθοδολογίας «Divide and Conquer» είναι η διαδοχική αποδόμηση του «προβλήματος» σε πολλά μικρότερα μέχρις ότου αυτά να γίνουν αρκετά απλά έτσι ώστε να μπορούν να επιλυθούν με εύκολες διαδικασίες [13]. Οι μερικές λύσεις των κατακερματισμένων προβλημάτων ανασυντίθενται για να δώσουν την συνολική λύση.

Το λογισμικό που είναι βασισμένο σε Υπηρεσιοστραφή Αρχιτεκτονική αποτελείται από τη φύση του από ένα σύνολο υπηρεσιών με χαλαρή συνδεσιμότητα (loosely coupled services). Αυτές οι δομικές μονάδες της SOA μπορούν να κατηγοριοποιηθούν βάσει των διαφορετικών ιδιοτήτων (features) τους και εστιάζοντας στην διαδικασία υλοποίησης μπορούμε να χωρίσουμε τις υπηρεσίες στις παρακάτω κατηγορίες:

- **Διαθέσιμη Υπηρεσία (Available Service)**, αποτελεί ένα βασικό τύπο υπηρεσίας που υπάρχει ήδη, είτε παρέχεται από ένα εξωτερικό φορέα (3rd party) ή έχει κληρονομήσει τη λειτουργικότητά της από υπάρχουσες εφαρμογές (legacy based system).
- **Υπηρεσία από Μετάπτωση (Migrated Service)**, είναι μια υπηρεσία η οποία δημιουργείται από την μετατροπή ή την ενθυλάκωση (wrapping) παραδοσιακών στοιχείων λογισμικού.
- **Νέα Υπηρεσία (New Service)**, είναι το είδος της υπηρεσίας που δημιουργείται εξ αρχής.

- **Συνδυασμένη Υπηρεσία (Combined Service)**, είναι το είδος της υπηρεσίας που αποτελεί ένα συνδυασμό των τριών παραπάνω ειδών υπηρεσιών.

Βασισμένο σε αυτό το είδος κατηγοριοποίησης των υπηρεσιών προκύπτουν τέσσερις (4) κατηγορίες κατακερματισμού της υλοποίησης για έργα SOA. Αυτές οι κατηγορίες αφορούν την Εύρεση της Υπηρεσίας (Service Discovery), την μετάπτωση της Υπηρεσίας (Service Migration), την ανάπτυξη / υλοποίηση της Υπηρεσίας (Service Development), καθώς και τη διαδικασία ανασύνθεσης των υπηρεσιών (Service Integration). Η εκτίμηση κόστους της συνολικής λύσης μπορεί να προκύψει από την εκτίμηση κόστους των κομματιών που την απαρτίζουν.

Αρχικά μέσω της ανάλυσης της Υπηρεσιοστραφούς Αρχιτεκτονικής το έργο χωρίζεται σε βασικές υπηρεσίες. Σε δεύτερο στάδιο ξεχωριστές μέθοδοι μέτρησης (metrics) υιοθετούνται για κάθε στάδιο της ανάπτυξης για το σωστό υπολογισμό του κόστους και της άνθρωπο – προσπάθειας.

Το συνολικό κόστος προκύπτει από την σύνθεση των επί μέρους υπολογισμών.

3.1.2 Μεθοδολογία SMAT-AUS

Η SMAT-AUS είναι μια μεθοδολογία που βρίσκεται σε εξέλιξη και βελτιώνεται συνεχώς. Δημιουργήθηκε για να χρησιμοποιηθεί στον καθορισμό του εύρους του αντικειμένου (score), της εκτίμησης του κόστους και της ανθρωποπροσπάθειας για έργα Υπηρεσιοστραφούς Αρχιτεκτονικής. Η μεθοδολογία αναδεικνύει όλες τις διαστάσεις υλοποίησης ενός λογισμικού σε περιβάλλον SOA που εκτός από την τεχνική περιλαμβάνει κοινωνικές και επιχειρησιακές παραμέτρους.

Η μεθοδολογία SMAT-AUS βασίζεται στις παρακάτω αρχές και παραδοχές [14]:

- Υπάρχουν διαφορετικές μορφές και τύποι έργων SOA και ο οργανισμός μπορεί να δεσμευτεί ότι θα υλοποιήσει ένα ή περισσότερα από αυτά. (SOA-based software development, Service Mining, Service Development, Service Integration, SOA Application Development)
- Για κάθε τύπο SOA που θα υλοποιηθεί ορίζονται:
 - Μέθοδοι (Methods) οι οποίες υποστηριζόμενες από τα κατάλληλα εργαλεία που προσδιορίζουν το αντικείμενο του έργου (project scope), καθώς και τα κέντρα κόστους (effort, cost).
 - Πρότυπα (Templates) για την συλλογή των δεδομένων σχετικά με το έργο (π.χ πληροφορίες σχετικά με τα services, τα υφιστάμενα συστήματα (legacy systems), τους εμπλεκόμενους στο έργο κλπ).
 - Ένα σύνολο από παραμέτρους κόστους (cost factors) συγκεκριμένους για κάθε τύπο έργου.
 - Ένα σύνολο από συναρτήσεις κόστους για τον προσδιορισμό του κόστους ανά τύπο έργου. Πολλές από τις παραμέτρους μπορούν να ομαδοποιηθούν και να χρησιμοποιηθούν οριζόντια για περισσότερους του ενός τύπους έργων.
- Ένα μοντέλο ωριμότητας SOA (SOA Maturity Model) χρησιμοποιείται οριζόντια για όλους τους τύπους των έργων έτσι ώστε να διαπιστωθεί η δυνατότητα του οργανισμού / εταιρείας να αναλάβει και να φέρει σε πέρας ένα έργο αρχιτεκτονικής SOA. Η ωριμότητα και η εμπειρία του φορέα μπορεί να έχει επιπτώσεις στο κόστος υλοποίησης ενός έργου.
- Για κάθε είδος έργου πρέπει να συνυπολογιστούν δύο διαστάσεις, η τεχνική που αφορά τη διαδικασία υλοποίησης και η κοινωνική / οργανωτική που αφορά το προφίλ των εμπλεκόμενων στο έργο.

3.2 Εμπειρικές μεθοδολογίες και άλλες τεχνικές

Το ενδιαφέρον για την διατύπωση μιας μεθοδολογίας εκτίμησης κόστους κατά την υλοποίηση εφαρμογών σε Υπηρεσιοστραφή Αρχιτεκτονική έχει προκαλέσει μια σειρά από δημοσιεύσεις προτάσεων για μεθολογίες με αρκετές ετεροαναφορές (citations) [12], [13], [14]. Η πλειοψηφία αυτών των προτάσεων βασίζονται σε γενικές κατευθύνσεις λαμβάνοντας υπόψη ως παραμέτρους τον αριθμό στοιχείων δεδομένων (data elements), την πολυπλοκότητα τεχνολογίας αποθήκευσης δεδομένων (data storage complexity), την πολυπλοκότητα του συστήματος (system complexity), την πολυπλοκότητα των Υπηρεσιών (service complexity) κ.α.

Το εύρος αποδοχής αυτών των εμπειρικών μεθοδολογιών είναι μικρό, ενώ τα αποτελέσματα εφαρμογής τους είναι μη αξιολογήσιμα αν λάβουμε υπόψη την έλλειψη σχετικών ιστορικών δεδομένων.

3.3 Μεθοδολογίες Function Size Measurement κατά ISO/IEC 14143

Οι μέθοδοι Function Size Measurement (Function Size Measurement Methods - FSMM) χρησιμοποιούνται για τη μέτρηση του Λειτουργικού Μεγέθους μιας μονάδας λογισμικού και βασίζονται στην ποσοτικοποίηση του «μεγέθους» της λειτουργικότητας που υλοποιεί και προσφέρει η υπό εξέταση μονάδα.

Το ISO/IEC 14143 [15] ορίζει τις θεμελιώδεις έννοιες για τις τεχνικές μέτρησης του μεγέθους και ουσιαστικά θέτει το πλαίσιο για την «επικύρωση» μιας τεχνικής FSM. Με το πρότυπο αυτό συμμορφώνονται μέχρι σήμερα πέντε μέθοδοι μέτρησης του λειτουργικού μεγέθους:

- **IFPUG FPA 4.3** [ISO/IEC 20926:2009]
- **COSMIC-FFP** Functional size measurement method v 3.0.1 [ISO/IEC 19761:2009]
- **Mk II Function Point Analysis 1.3.1 Unadjusted** [ISO/IEC 20968:2002]
- **NESMA FPA Method 2.1 Unadjusted** [ISO/IEC 24570:2005]
- **FISMA FSM 1.1** [ISO/IEC 29881:2008]

3.3.1 IFPUG FPA 4.3

Η «ιστορία» των λειτουργικών σημείων ξεκινάει από τα τέλη της δεκαετίας του 1970 όταν ο μηχανικός λογισμικού Allan J. Albrecht παρουσίασε τη μεθοδολογία για την κοστολόγηση των έργων λογισμικού της εταιρίας IBM. [16]

Από τότε έχουν μεσολαβήσει πολλά χρόνια εξέλιξης των αρχών και της μεθόδου που παρουσίασε ο Allan Albrecht. Το International Function Point Users Group (IFPUG) είναι μια μη κερδοσκοπική οργάνωση που ιδρύθηκε το 1986 και ανέλαβε να προωθήσει την χρήση της ανάλυσης λειτουργικών σημείων (Function Point Analysis) για τη μέτρηση της λειτουργικότητας του λογισμικού. Ήδη από το 1993 το JTC1 (διεθνής επιτροπή για τα πρότυπα της πληροφορικής) είχε ενεργοποιήσει μια προσπάθεια για την τυποποίηση της ανάλυσης λειτουργικών σημείων. Η προσπάθεια αυτή κατέληξε στην δημιουργία της μεθοδολογίας IFPUG και του σχετικού εγχειριδίου μέτρησης (Counting Practices Manual).

Η μέθοδος IFPUG FPA έχει τυποποιηθεί με το ISO/IEC 20926 (έκδοση 4.1 το 2003 και η ισχύουσα έκδοση 4.3 το 2009), ενώ υπεύθυνος φορέας για την συντήρησή της παραμένει το International Function Point Users Group (IFPUG) [17].

Στο πέρασμα του χρόνου η μεθοδολογία IFPUG FPA έχει σημειώσει ευρεία αποδοχή και χρησιμοποιείται για την εκτίμηση μεγέθους σε πλειάδα έργων λογισμικού, ενώ με βάση το ISO η μεθοδολογία εφαρμόζεται αποτελεσματικά σε όλους τους τύπους λογισμικού.

Η μεθοδολογία (βλ. και κεφ. 4 - Εκτίμηση μεγέθους λογισμικού χρησιμοποιώντας τη μεθοδολογία IFPUG FPA 4.3) λειτουργεί πάνω στην παραδοχή ότι το λογισμικό απαρτίζεται από βασικά λειτουργικά συστατικά (Basic Functional Components – BFC). Αυτά τα βασικά λειτουργικά συστατικά κατηγοριοποιούνται στις εξής πέντε κατηγορίες: Εξωτερικές εισόδους (External Inputs – EI), Εξωτερικές εξόδους (External Outputs – EO), Εξωτερικά ερωτήματα (External Inquiries – EQ), Εσωτερικά λογικά αρχεία (Internal Logical Files – ILF) και Εξωτερικά αρχεία (External Interface Files – EIF).

Στο σημείο αυτό θα πρέπει να επισημάνουμε ότι σύμφωνα με το ISO/IEC 20926 στην μέθοδο IFPUG λαμβάνονται πλέον υπόψη μόνο τα μη-σταθμισμένα Λειτουργικά Σημεία (unadjusted function points), ενώ προωθείται η χρήση του όρου «Λειτουργικό Μέγεθος». Η χρήση των Γενικών Χαρακτηριστικών Συστήματος (General System Characteristics – GSC) και η «προσαρμογή» της μέτρησης με χρήση του Συντελεστή Προσαρμογής (Value Adjustment Factor) κατέστη προαιρετική από το 2002 για λόγους συμμόρφωσης με το ISO/IEC 14143. Τα GSC ήταν ίσως το σημαντικότερο σημείο κριτικής για τις παλαιότερες εκδοχές της μεθόδου, καθώς επέτρεπε ευρύτητα ερμηνειών και κατ' επέκταση αποκλίσεις στις εκτιμήσεις για το μέγεθος του λογισμικού.

3.3.2 COSMIC-FFP Functional size measurement method v 3.0.1

Το Common Software Measurement International Consortium (COSMIC) δημιουργήθηκε το 1998 με σκοπό τη δημιουργία μιας νέας μεθόδου για τη λειτουργική μέτρηση μεγέθους λογισμικού. Η μέθοδος την οποία συνέθεσαν βασίστηκε στην ανάλυση των υπάρχουσών μεθοδολογιών εκείνη την περίοδο (IFPUG, Mark II, NESMA) και συγκεκριμένα στα καλύτερα χαρακτηριστικά κάθε μεθόδου.

Η νέα μέθοδος δημοσιοποιήθηκε το Νοέμβριο του 1999, ως η έκδοση 2.0 της Cosmic-FFP (COSMIC Full Function Points) διαθέτοντας παράλληλα ελεύθερα και το εγχειρίδιο μέτρησης (σε Αγγλικά, Γαλλικά, Ιαπωνικά και Ισπανικά). Η μέθοδος COSMIC-FFP έχει τυποποιηθεί με το ISO/IEC 19761 και υπεύθυνος φορέας για την συντήρησή της παραμένει το Common Software Measurement International Consortium [18].

Η COSMIC-FFP σχεδιάστηκε ώστε να εφαρμόζεται σε λογισμικά των εξής πεδίων:

- Λογισμικό επιχειρησιακής υποστήριξης (MIS), για την διοίκηση μιας επιχείρησης (π.χ. λογισμικό για την υποστήριξη τραπεζικών, ασφαλιστικών και λογιστικών αναγκών).
- Λογισμικό πραγματικού χρόνου (real time) τον έλεγχο και την υποστήριξη αντίστοιχων αναγκών (π.χ. εκτέλεση συναλλαγών και παραγωγικών διαδικασιών).

Η μεθοδολογία COSMIC-FFP δεν καλύπτει με τρόπο αποδοτικό την εκτίμηση μεγέθους για λογισμικό που βασίζεται στην εκτέλεση σύνθετων μαθηματικών αλγορίθμων καθώς και για λογισμικό που επεξεργάζεται συνεχείς μεταβλητές (όπως ήχο και κινούμενη εικόνα).

Η COSMIC-FFP λειτουργεί πάνω στην παραδοχή ότι το λογισμικό αποτελείται από λειτουργικές διαδικασίες, οι οποίες με τη σειρά τους απαρτίζονται από ροές δεδομένων. Οι ροές αυτές κατηγοριοποιούνται σε Εισόδους (data input type – E), Αναγνώσεις (data read type – R), Εξόδους (data

output type – X) και Καταγραφές (data write type – W). Στην COSMIC-FFP η μονάδα μέτρησης είναι μια «ροή δεδομένων» κάποιας εκ των παραπάνω τεσσάρων κατηγοριών (E, R, X, W).

Η έρευνα για την εξέλιξη της μεθοδολογίας COSMIC-FFP εστίασε στη δυνατότητα καλύτερης εφαρμογής της στην εκτίμηση μεγέθους λογισμικού πραγματικού χρόνου. Συγκεκριμένα το 2001 παρουσιάστηκε [24] μια εκδοχή της COSMIC-FFP πάνω στην Αντικειμενοστραφή Μοντελοποίηση σε Πραγματικό Χρόνο (Real-Time Object Oriented Modeling - ROOM). Η τυποποίηση αυτή περιορίζει σημαντικά το ενδεχόμενο αποκλίσεων σε διαφορετικές μετρήσεις για τις ίδιες προδιαγραφές λογισμικού (εξαιτίας τυχόν διαφορετικών ερμηνειών που μπορεί να δώσουν τα πρόσωπα που εκτελούν τη μέτρηση). Με τον τρόπο αυτό επετεύχθη η αυτοματοποίηση της εκτέλεσης μετρήσεων με την COSMIC-FFP για προσέγγιση έργων λογισμικού με ROOM. Επιπλέον η νέα τυπική εκδοχή της μεθοδολογίας παρέχει μια ξεκάθαρη προσέγγιση των εννοιών, διευκολύνοντας την εφαρμογή της COSMIC-FFP σε άλλες αντικειμενοστραφείς μοντελοποιήσεις όπως η UML.

3.3.3 Mk II Function Point Analysis 1.3.1 Unadjusted

Τη δεκαετία του 1980 η μέθοδος Mark II (ή MkII) σχεδιάστηκε από τον Charles Symons, έχοντας ως σημείο αφετηρίας την πρόταση και τις αρχές που έθεσε ο Allan J. Albrecht. Η μεθοδολογία αναπτύχθηκε εσωτερικά στην εταιρία KPMG μεταξύ 1985 και 1986 και αρχικά ήταν ιδιόκτητη όμως πλέον προσφέρεται δημόσια. Η μέθοδος MkII έχει τυποποιηθεί με το ISO/IEC 20968 και υπεύθυνος φορέας για την συντήρησή της είναι πλέον το UK Software Metrics Association (UKSMA) [19].

Η MkII εφαρμόζεται εκτεταμένα στο Ηνωμένο Βασίλειο και παρά το κοινό σημείο αφετηρίας με την μεθοδολογία IFPUG παρουσιάζει κάποιες διαφοροποιήσεις.

Λειτουργεί πάνω στην παραδοχή ότι το λογισμικό απαρτίζεται από λογικές συναλλαγές (logical transactions). Η μέθοδος εισάγει τρία επιμέρους μεγέθη τα οποία αποτιμά για να κάνει εκτίμηση μεγέθους του λογισμικού: Τύποι Δεδομένων Εισόδου (Input Data Element Types - Ni), Τύποι Αναφερόμενων Οντοτήτων (Entity Types Referenced - Ne) και Τύποι Δεδομένων Εξόδου (Output Data Element Types - No).

Με βάση το ISO η μεθοδολογία εφαρμόζεται αποτελεσματικά σε πληροφοριακά συστήματα όπου μπορούν να αναγνωριστούν λογικές συναλλαγές. Όμως η MkII δίνει μεγαλύτερα αποτελέσματα μέτρησης σε σύγκριση με την IFPUG, με την απόκλιση αυτή να μεγαλώνει όσο αυξάνει η πολυπλοκότητα των υπό μέτρηση συστημάτων ιδίως για μεγαλύτερα/πολυπλοκότερα συστήματα. [27]

3.3.4 NESMA FPA Method 2.1 Unadjusted

Η Netherlands Software Metrics Users Association ιδρύθηκε το 1989 και είναι η μεγαλύτερη ομάδα χρηστών για την ανάλυση λειτουργικών σημείων στην Ευρώπη. Σκοπός της NESMA είναι η συλλογή, συντήρηση, ανταλλαγή και ανάπτυξη της γνώσης γύρω από την ανάλυση λειτουργικών σημείων. Στο πλαίσιο αυτό ο οργανισμός παρουσίασε τη δική του μεθοδολογία μέτρησης λειτουργικών σημείων.

Η μέθοδος NESMA έχει τυποποιηθεί με το ISO/IEC 24570 και υπεύθυνος φορέας για την συντήρησή της παραμένει η Netherlands Software Metrics Users Association [20]. Με βάση το ISO η μεθοδολογία εφαρμόζεται αποτελεσματικά σε όλους τους τύπους λογισμικού.

Η μεθοδολογία NESMA προσφέρει τρεις εκδοχές της μεθόδου μέτρησης: (α) «αναλυτική» (detailed) μέτρηση λειτουργικών σημείων, (β) «εκτιμητική» (estimate) μέτρηση λειτουργικών σημείων και (γ)

«ενδεικτική» (indicative) μέτρηση λειτουργικών σημείων. Οι δύο τελευταίες εκδοχές σχεδιάστηκαν από τη NESMA για την απλοποίηση της εκτίμησης μεγέθους κατά τα πρώτα στάδια ανάπτυξης ενός λογισμικού.

Η Αναλυτική εκδοχή είναι εξαιρετικά συγγενής με την μεθοδολογία IFPUG FPA, και συγκεκριμένα σύμφωνα με το IFPUG είναι όμοιες κατά 95% [21].

Κατά την Εκτιμητική εκδοχή της μεθόδου, η «πολυπλοκότητα» (complexity) δεν προσδιορίζεται ανάλογα με την κάθε επιμέρους λειτουργία, αλλά σταθερές για όλο το σύστημα. Συγκεκριμένα μετά την αναγνώριση των λειτουργιών του συστήματος, οι πολυπλοκότητες για τα ILF και EIF ορίζονται ως χαμηλές (low) και οι πολυπλοκότητες για τα EI, EO και EQ ορίζονται ως μέσες (average).

Η Ενδεικτική εκδοχή της μεθόδου χρησιμοποιεί αποκλειστικά τον αριθμό των λειτουργιών ILF και EIF, χρησιμοποιώντας τον εξής μαθηματικό τύπο: Ενδεικτική Εκτίμηση = (35 x Πλήθος ILF) + (15 x Πλήθος EIF). Ο τύπος αυτό βασίζεται στην υπόθεση ότι για κάθε ILF θα προκύπτουν κατά μέσο όρο τρία EI (για να προσθέσουν, αλλάξουν και διαγράψουν πληροφορία στο ILF) δύο EO και ένα EQ. Αντιστοίχως η μέθοδος υποθέτει ότι για κάθε EIF θα προκύπτει κατά μέσο όρο ένα EO και ένα EQ. Τα βάρη (35/15) προκύπτουν εάν θεωρήσουμε τις πολυπλοκότητες για τις συναλλακτικές λειτουργίες ορίζονται ως μέσες (average) και οι πολυπλοκότητες για τις λειτουργίες δεδομένων ορίζονται ως χαμηλές (low). Επιπλέον η μέθοδος δίνει 2 Λειτουργικά Σημεία στα ILF και ένα λειτουργικό σημείο στα EIF, για την λειτουργικότητα «γενικής υποστήριξης».

3.3.5 FiSMA FSM 1.1

Η Finnish Software Measurement Association (FiSMA) ιδρύθηκε το 1992 (υπό το όνομα LATURI ως το 1998) και είναι μια ανεξάρτητη ένωση (Φινλανδικών ως επί το πλείστον) εταιριών που στοχεύει στην βελτίωση της ποιότητας και της μετρησιμότητας του λογισμικού και των συστημάτων. Αρχικά η FiSMA εξέδωσε μια μέθοδο μέτρησης λειτουργικών σημείων υπό το όνομα «Experience FPA», από την οποία προέκυψε η μεθοδολογία FiSMA που βασίζεται εξολοκλήρου στην προσέγγιση των λειτουργικών σημείων. Πρόκειται για το «νέοτερο μέλος» στην οικογένεια των μεθοδολογιών FSM που ικανοποιούν το ISO/IEC 14143.

Η μέθοδος FiSMA έχει τυποποιηθεί με το ISO/IEC 29881 και υπεύθυνος φορέας για την συντήρησή της παραμένει η Finnish Software Measurement Association [22]. Με βάση το ISO η μεθοδολογία εφαρμόζεται αποτελεσματικά σε όλους τους τύπους λογισμικού. Χρησιμοποιείται ευρέως στη Φινλανδία όμως δεν έχει γνωρίσει ιδιαίτερη διάδοση στον υπόλοιπο κόσμο.

Η FiSMA λειτουργεί πάνω στην παραδοχή ότι το λογισμικό απαρτίζεται από βασικά λειτουργικά συστατικά (Basic Functional Components – BFC). Αυτά τα βασικά λειτουργικά συστατικά κατηγοριοποιούνται στις εξής επτά κλάσεις:

- Διαδραστικές υπηρεσίες πλοήγησης και ερώτησης τελικού χρήστη (Interactive end-user navigation and query services «q»)
- Διαδραστικές υπηρεσίες παροχής εισόδων τελικού χρήστη (Interactive end-user input services «i»)
- Μη-διαδραστικές υπηρεσίες εξόδων προς τον τελικό χρήστη (Non-interactive end-user output services «o»)
- Υπηρεσίες διεπαφής προς άλλη εφαρμογή (Interface services to other application «t»)
- Υπηρεσίες διεπαφής από άλλη εφαρμογή (Interface services from other applications «f»)
- Υπηρεσίες αποθήκευσης δεδομένων (Data storage services «d»)
- Αλγοριθμικές και επεξεργαστικές υπηρεσίες (Algorithmic and manipulation services «a»)

Οι επτά βασικές κλάσεις χωρίζονται σε επιμέρους υπό-κλάσεις (συνολικά 28).

3.4 Συμπέρασμα – Επιλογή μεθοδολογίας

Μέσα από το πλήθος των μεθοδολογιών (και των παραλλαγών τους), όπως προαναφέρθηκαν, καταλήξαμε στην επιλογή της πλέον κατάλληλης για την προεκτίμηση και την αποτίμηση του μεγέθους και του κόστους ανάπτυξης λογισμικού υπηρεσιοστραφούς αρχιτεκτονικής (SOA) στη Γ.Γ.Π.Σ.

Από την εξέταση της βιβλιογραφίας προέκυψε ως πλέον κατάλληλη η «οικογένεια» των μεθόδων μέτρησης του μεγέθους λειτουργίας (Function Size Measurement - FSM), οι οποίες συμμορφώνονται με τις προβλέψεις του ISO/IEC 14143.

Από τις πέντε FSM μεθόδους προτείνουμε την υιοθέτηση της μεθόδου Function Point Analysis (FPA) του οργανισμού IFPUG (έκδοση 4.3) η οποία επικυρώθηκε με το πρότυπο ISO/IEC 20926:2009.

Η επιλογή της μεθοδολογίας FPA 4.3 έναντι των υπολοίπων μεθόδων που συμμορφώνονται με το ISO/IEC 14143 έγινε λαμβάνοντας υπόψη τους παρακάτω παράγοντες:

- Χαρακτηρίζεται από ιδιαίτερη ωριμότητα (η έκδοση 4.0 εκδόθηκε από τον IFPUG το 1994) αλλά και ευρεία αποδοχή στο χώρο της ανάπτυξης λογισμικού και της εκτίμησης μεγέθους λογισμικού.
- Είναι ιδιαίτερος δομημένη, οδηγώντας σε αξιόπιστα και σταθερά αποτελέσματα. Κατά κανόνα δύο ανεξάρτητες μετρήσεις από διαφορετικούς ανθρώπους πάνω στο ίδιο λογισμικό δεν αποκλίνουν περισσότερο του +/- 10%.
- Εφαρμόζεται για όλους τους τύπους συστημάτων (Online, Batch, Client/Server, Web, Mainframe, Data Warehouse, SOA), όλες τις γλώσσες λογισμικού και όλες τις μεθόδους ανάπτυξης (Agile, RAD, JAD, UML, Waterfall κλπ). Επίσης ισχύει και για περιπτώσεις παραμετροποίησης έτοιμου λογισμικού (customization).
- Μπορεί να εφαρμοστεί, ανεξάρτητα του σταδίου ανάπτυξης που βρίσκεται το λογισμικό. Ισχύει δηλαδή ακόμα και σε πρώιμες φάσεις ανάπτυξης (λ.χ. κατά την σύνταξη λειτουργικών προδιαγραφών ενός λογισμικού).
- Δεν προαπαιτεί ιδιαίτερες γνώσεις (λ.χ. ειδική εκπαίδευση ή χρονοβόρα κατάρτιση) ώστε να ξεκινήσει η εφαρμογή της σε ένα φορέα.
- Περιορίζει το χρόνο που απαιτείται για την εκτέλεση μιας εκτίμησης μεγέθους και δεν προκαλεί επιβάρυνση στην εκτέλεση ενός έργου ανάπτυξης. Αυτός ο χρόνος μειώνεται ακόμα περαιτέρω, όσο αυξάνεται η εμπειρία και η εκπαίδευση αρμόδιων στελεχών.
- Διαθέτει - σε σύγκριση με τις υπόλοιπες μεθόδους - ένα ιδιαίτερα μεγάλο πλήθος «δεδομένων αναφοράς» (είτε σε ειδικές «ιδιωτικές» βάσεις όπως το ISBSG είτε «ανοιχτών» στο διαδίκτυο), προς χρήση στην εσωτερική αξιολόγηση και την αυτοβελτίωση της διαδικασίας μέτρησης εντός ενός Φορέα.
- Συνδέεται με αρκετούς φορείς που παρέχουν πιστοποιημένη εκπαίδευση πάνω στη χρήση της μεθόδου. Επίσης είναι πλέον διαθέσιμα αρκετά πιστοποιημένα λογισμικά για την υποβοήθηση της εκτέλεσης της μεθόδου και την διαχείριση των πραγματοποιούμενων μετρήσεων.

4 Εκτίμηση μεγέθους λογισμικού χρησιμοποιώντας τη μεθοδολογία IFPUG FPA 4.3

Η μεθοδολογία Ανάλυσης Λειτουργικών Σημείων (Function Point Analysis - FPA) του οργανισμού IFPUG (έκδοση 4.3) [23] αφορά στην ποσοτικοποίηση του «μεγέθους» της λειτουργικότητας που υλοποιεί και προσφέρει το υπό εξέταση πληροφοριακό σύστημα/υποσύστημα/μονάδα/εφαρμογή λογισμικού.

Στις παραγράφους που ακολουθούν παρουσιάζουμε τα βασικά σημεία για τη μεθοδολογία αυτή σύμφωνα και με το ISO/IEC 20926:2009.

4.1 Βασικοί Ορισμοί

4.1.1 Λειτουργικό Μέγεθος

Το «μέγεθος» ενός λογισμικού όπως αυτό προκύπτει από την ποσοτικοποίηση των Λειτουργικών Απαιτήσεων Χρήστη (Functional User Requirements), ήτοι η «ποσότητα» επιχειρησιακής λειτουργικότητας που παρέχει ένα λογισμικό στον χρήστη του.

4.1.2 Λειτουργικό Σημείο

Λειτουργικό Σημείο (function point) ορίζεται η μονάδα μέτρησης του Λειτουργικού Μεγέθους.

4.1.3 Στοιχειώδης Διαδικασία

Στοιχειώδης Διαδικασία (elementary process) ορίζεται η μικρότερη μονάδα δραστηριότητας του συστήματος που μπορεί να αντιληφθεί ο χρήστης. Μια στοιχειώδης διαδικασία πρέπει να είναι ανεξάρτητη, αυτόνομη και να διατηρεί τη συνοχή του συστήματος μετά την εκτέλεσή της.

4.1.4 Λειτουργικές Διαδικασίες Δεδομένων

Οι **λειτουργικές διαδικασίες δεδομένων (Data Functions)** αντιπροσωπεύουν τη λειτουργικότητα που απαιτείται για να καλυφθούν οι εσωτερικές και εξωτερικές ανάγκες σε δεδομένα. Οι λειτουργίες δεδομένων διακρίνονται σε:

4.1.4.1 Εσωτερικά Λογικά Αρχεία

Εσωτερικά Λογικά Αρχεία (Internal Logical Files - ILF) ορίζονται τα σύνολα λογικά σχετιζόμενων πληροφοριών/δεδομένων που συντηρούνται από την εφαρμογή. Τα ILFs συντηρούνται (π.χ. εισάγονται, τροποποιούνται, διαγράφονται) μέσω μιας ή περισσότερων στοιχειωδών διαδικασιών⁴ του συστήματος.

4.1.4.2 Εξωτερικά Αρχεία Διεπαφών

Εξωτερικά Αρχεία Διεπαφών (External Interface Files – EIF) ορίζονται τα σύνολα λογικά σχετιζόμενων πληροφοριών/δεδομένων, τα οποία δεν συντηρούνται από το σύστημα, αλλά υπάρχει αναφορά σε αυτά από μία ή περισσότερες στοιχειώδεις διαδικασιών⁴ του συστήματος μέσω διεπαφών. Τα EIFs συντηρούνται από κάποιο άλλο σύστημα. Συνεπώς ένα EIF για ένα σύστημα αποτελεί ILF για κάποιο άλλο σύστημα.

4.1.5 Λειτουργικές Διαδικασίες Συναλλαγών

Οι **λειτουργικές διαδικασίες συναλλαγών** (Transactional Functions) απεικονίζουν τη λειτουργικότητα που παρέχεται στο χρήστη για την επεξεργασία των δεδομένων, και διακρίνονται σε:

4.1.5.1 Εξωτερικές Είσοδοι

Εξωτερικές Είσοδοι (External Inputs - EI) ορίζονται οι στοιχειώδεις διαδικασίες που επεξεργάζονται δεδομένα που εισάγονται στο σύστημα (δηλαδή προέρχονται εκτός του συστήματος). Ο κύριος σκοπός μιας EI είναι να διατηρήσει ένα ή περισσότερα ILFs και/ή να αλλάξει τη συμπεριφορά του συστήματος.

4.1.5.2 Εξωτερικές Έξοδοι

Εξωτερικές Έξοδοι (External Outputs - EO) ορίζονται οι στοιχειώδεις διαδικασίες που αποστέλλουν δεδομένα εκτός του συστήματος. Ο κύριος σκοπός μιας EO είναι να παρουσιάσει στο χρήστη πληροφορίες που έχουν υποστεί μια λογική επεξεργασία επιπλέον της ανάκτησής τους. Η λογική επεξεργασία θα πρέπει να περιλαμβάνει τουλάχιστον ένα μαθηματικό υπολογισμό, ή να παράγει τροποποιημένα δεδομένα. Επίσης, μια EO μπορεί να διατηρεί ένα ή περισσότερα ILFs και/ή να αλλάξει τη συμπεριφορά του συστήματος.

4.1.5.3 Εξωτερικές Αναζητήσεις

Εξωτερικές Αναζητήσεις (External Inquiries - EQ) ορίζονται οι στοιχειώδεις διαδικασίες που εξάγουν δεδομένα από το σύστημα. Ο κύριος σκοπός μιας EQ είναι να παρουσιάσει στο χρήστη πληροφορίες που απλώς ανακτήθηκαν χωρίς κάποιου είδους επεξεργασία που να περιέχει κάποιο μαθηματικό υπολογισμό, ή τη δημιουργία τροποποιημένων δεδομένων. Μία EQ δε συντηρεί κανένα ILF, ούτε μπορεί να αλλάξει τη συμπεριφορά του συστήματος.

4.1.6 Στοιχείο Δεδομένων

Στοιχείο Δεδομένων (Data Element Type – DET) ορίζεται κάθε ένα μοναδικό, μη επαναλαμβανόμενο, αντιληπτό από το χρήστη δεδομένο/πεδίο.

4.1.7 Στοιχείο Εγγραφής

Στοιχείο Εγγραφής (Record Element Type – RET) ορίζεται κάθε μια αντιληπτή από το χρήστη υποομάδα δεδομένων μέσα σε ένα ILF ή EIF. Η κάθε υποομάδα μπορεί να είναι:

- Προαιρετική: πρόκειται για υποομάδα την οποία ο χρήστης μπορεί να χρησιμοποιήσει ή να αγνοήσει κατά την εκτέλεση μιας στοιχειώδους διαδικασίας που προσθέτει ή δημιουργεί μία οντότητα δεδομένων.
- Υποχρεωτική: Πρόκειται για υποομάδα την οποία ο χρήστης πρέπει να χρησιμοποιήσει κατά την εκτέλεση μιας στοιχειώδους διαδικασίας που προσθέτει ή δημιουργεί μία οντότητα δεδομένων.

4.1.8 Αναφερόμενο Αρχείο

Αναφερόμενο Αρχείο (File Type Referenced – FTR) ορίζεται:

- Ένα εσωτερικό λογικό αρχείο (ILF) που διαβάζεται ή συντηρείται από μια λειτουργική διαδικασία συναλλαγών (EI, EO και EQ) ή
- Ένα εξωτερικό αρχείο διεπαφών (EIF) που διαβάζεται από μια λειτουργική διαδικασία συναλλαγών (EI, EO και EQ).

4.1.9 Λοιποί Ορισμοί

Χρήστης (User) ορίζεται κάθε πρόσωπο ή οντότητα που επικοινωνεί ή έχει διεπαφή με το λογισμικό σε οποιαδήποτε χρονική στιγμή.

Αναγνωρίσιμες από το Χρήστη (User Recognizable) ορίζονται οι απαιτήσεις για τις διαδικασίες ή/και τα δεδομένα για τα οποία έχει διαμορφωθεί συναντίληψη ανάμεσα στους χρήστες και την ομάδα ανάπτυξης λογισμικού.

Όρια Εφαρμογής (Application Boundary) ορίζεται το νοητό σημείο διαχωρισμού της εφαρμογής/λογισμικού υπό μέτρηση και των χρηστών.

Εμβέλεια (Scope) ορίζει το (υπό)σύνολο του λογισμικού προς μέτρηση, συγκεκριμενοποιώντας τις λειτουργίες που θα εξεταστούν κατά τη μέτρηση των Λειτουργικών Σημείων.

Στοιχεία Ελέγχου (Control information) ορίζονται τα δεδομένα που επηρεάζουν μια Στοιχειώδη Διαδικασία προσδιορίζοντας «τι», «πότε» ή «πως» θα χρησιμοποιηθούν τα δεδομένα

4.2 Μέτρηση Λειτουργικού Μεγέθους Λογισμικού

Το Λειτουργικό Μέγεθος (Functional Size – FS) απεικονίζει το μέγεθος της λειτουργικότητας που παρέχεται στο χρήστη από μια εφαρμογή λογισμικού. Η λειτουργικότητα ως FS εκτιμάται με βάση τις λειτουργικές απαιτήσεις των χρηστών, χωρίς να λαμβάνονται υπόψη οι τεχνικές παράμετροι της υλοποίησης και της λειτουργίας.

Ως εκ τούτου για τη μέτρηση λαμβάνονται υπόψη μόνο τα απαιτούμενα και προσδιορισμένα από το χρήστη χαρακτηριστικά/λειτουργίες και όχι χαρακτηριστικά που δημιουργούνται για τεχνικούς λόγους, για να εξυπηρετήσουν μια τεχνική απαίτηση ή να καλύψουν μια τεχνική ανάγκη.

Η μέτρηση του Λειτουργικού Μεγέθους γίνεται ακολουθώντας τα εξής στάδια:

- Συγκέντρωση της τεκμηρίωσης για το Μετρούμενο Λογισμικό
- Προσδιορισμός του Τύπου της Μέτρησης
- Προσδιορισμός των Ορίων του Μετρούμενου Λογισμικού
- Μέτρηση επιμέρους Λειτουργικών Μεγεθών
- Τελικός υπολογισμός Λειτουργικού Μεγέθους

4.2.1 Συγκέντρωση της τεκμηρίωσης για το Μετρούμενο Λογισμικό

Κατά την προεργασία για την εκτέλεση της Μέτρησης συστήνεται η συγκέντρωση της απαραίτητης τεκμηρίωσης που σχετίζεται με τη λειτουργικότητα του μετρούμενου λογισμικού (ή τη λειτουργικότητα που επηρεάζεται από το μετρούμενο λογισμικό). Επίσης χρήσιμη μπορεί να αποδειχθεί και η πρόσβαση σε άτομα με γνώση του μετρούμενου λογισμικού τα οποία να είναι σε θέση να παρέχουν διευκρινίσεις που θα καλύπτουν τυχόν κενά της τεκμηρίωσης.

Δεν υπάρχει αυστηρά ορισμένη λίστα του υλικού τεκμηρίωσης που απαιτείται να είναι διαθέσιμο. Ενδεικτικά (αλλά όχι περιοριστικά) θα μπορούσε να περιλαμβάνει: απαιτήσεις συστήματος, μοντέλα δεδομένων και αντικειμένων, διαγράμματα τάξεων (class diagrams), διαγράμματα ροών δεδομένων, περιγραφές διαδικασιών, υποδείγματα αναφορών και «οθονών», εγχειρίδια χρήστη καθώς και κάθε άλλο υλικό που σχετίζεται με το λογισμικό.

4.2.2 Προσδιορισμός του Τύπου της Μέτρησης

Η μεθοδολογία IFPUG FPA προβλέπει τρεις Τύπους μέτρησης ενός λογισμικού, για καθέναν από τους οποίους παραλλάσσεται ο μαθηματικός τύπος υπολογισμού του Λειτουργικού Μεγέθους. Για το λόγο αυτό θα πρέπει πριν την διενέργεια της μέτρησης να προσδιορίζεται ο Τύπος που θα ακολουθηθεί ανάμεσα στις εξής επιλογές:

- **Μέτρηση Λειτουργικών Σημείων κατά την Ανάπτυξη Λογισμικού:** αφορά την μέτρηση των λειτουργιών που παρέχονται σε ένα τελικό χρήστη με την πρώτη εγκατάσταση ενός λογισμικού, το οποίο παραδίδεται όταν το έργο ανάπτυξης ολοκληρώνεται.
- **Μέτρηση Λειτουργικών Σημείων κατά τη Βελτίωση Υφιστάμενου Λογισμικού:** αφορά τη μέτρηση των τροποποιήσεων σε υφιστάμενη εφαρμογή που προσθέτουν, τροποποιούν ή αφαιρούν λειτουργίες προς τους τελικούς χρήστες όταν το έργο ανάπτυξης ολοκληρώνεται
- **Μέτρηση Λειτουργικών Σημείων Εφαρμογής:** αφορά τη μέτρηση Λειτουργικών Σημείων μιας εγκατεστημένης εφαρμογής και συγκεκριμένα τις τρέχουσες λειτουργίες που μια εφαρμογή προσφέρει στον τελικό χρήστη. Κατά την πρώτη εγκατάσταση ενός λογισμικού η μέτρηση αυτή ταυτίζεται με τη Μέτρηση «Κατά την Ανάπτυξη». Στο πέρασμα του χρόνου όμως το αποτέλεσμα της μέτρησης επικαιροποιείται όσο γίνονται τροποποιήσεις στο λογισμικό είτε για λόγους συντήρησης είτε για λόγους βελτίωσης.

4.2.3 Προσδιορισμός των Ορίων του Μετρούμενου Λογισμικού

Όπως προκύπτει και από τους Ορισμούς (βλ. παρ 4.1) τα Όρια καθορίζουν τη λειτουργικότητα που θα εξεταστεί κατά τη μέτρηση των Λειτουργικών Σημείων. Πιο συγκεκριμένα προσδιορίζουν:

- Το υποσύνολο ή το σύνολο του λογισμικού που θα μετρηθεί.
- Τις λειτουργίες που θα συμπεριληφθούν στη μέτρηση των Λειτουργικών Σημείων.

Τα Όρια της εφαρμογής/λογισμικού έχουν τα εξής χαρακτηριστικά:

- Προσδιορίζουν τι είναι «εξωτερικό» της εφαρμογής
- Σηματοδοτούν το «σύνορο» μεταξύ του λογισμικού και του χρήστη
- Λειτουργούν ως «μεμβράνη» μέσω της οποίας τα δεδομένα που χρησιμοποιούνται εισέρχονται και εξέρχονται προς/από την εφαρμογή
- Περικλείουν τα Λογικά Αρχεία που συντηρεί η εφαρμογή
- Βοηθούν στην αναγνώριση των Λογικών Αρχείων που χρησιμοποιούνται αλλά δεν συντηρούνται από την εφαρμογή
- Εξαρτώνται από την οπτική του χρήστη προς την εφαρμογή ενώ είναι ανεξάρτητα από τις τεχνικές παραμέτρους και τις επιλογές κατά την υλοποίηση.

Συστήνεται η συμμόρφωση με τους ακόλουθους κανόνες κατά τον προσδιορισμό των Ορίων του μετρούμενου λογισμικού:

- Τα όρια πρέπει να προκύπτουν με βάση το τι αντιλαμβάνονται και μπορούν να περιγράψουν οι χρήστες.
- Τα όρια μεταξύ εφαρμογών/συστημάτων που σχετίζονται πρέπει να βασίζονται στην αυτοτελή λειτουργικότητα όπως αυτή γίνεται αντιληπτή από το χρήστη (και όχι στα «συστημικά» όρια ή άλλες τεχνικές προσεγγίσεις).

- Για τα έργα βελτίωσης υφιστάμενου λογισμικού θα πρέπει να διατηρούνται τα όρια που είχαν τεθεί κατά την αρχική μέτρηση της τροποποιούμενης εφαρμογής.

4.2.4 Μέτρηση επιμέρους Λειτουργικών Μεγεθών

Για τη μέτρηση του Λειτουργικού Μεγέθους απαιτείται η ανάλυση των λειτουργικών διαδικασιών δεδομένων και των λειτουργικών διαδικασιών συναλλαγών, και συγκεκριμένα προσδιορίζοντας τον αριθμό των ILFs, EIFs, EIs, EOs και EQs. Αφού όλες οι διαδικασίες αναγνωριστούν μέσα σε ένα σύστημα ακολουθεί για καθεμία εξ' αυτών ο χαρακτηρισμός τους ως Χαμηλής, Μεσαίας ή Υψηλής λειτουργικής πολυπλοκότητας.

Για τα ILF και EIF η βαθμολόγηση της πολυπλοκότητας γίνεται με βάση τον αριθμό των Στοιχείων Εγγραφών (RETs) και των Στοιχείων Δεδομένων (DETs) που συνδέονται με καθένα από τα ILF και EIF. Για τα EI, EO, και EQ η βαθμολόγηση της πολυπλοκότητας γίνεται με βάση τον αριθμό των Αναφερόμενων Αρχείων (FTRs) και των Στοιχείων Δεδομένων (DETs) που συνδέονται με καθένα από τα EI, EO και EQ.

4.2.4.1 Αποτίμηση των λειτουργικών διαδικασιών δεδομένων (ILFs και EIFs)

Μετά τον προσδιορισμό των λειτουργικών διαδικασιών δεδομένων (ILF και EIF) για την μετρούμενη οντότητα λογισμικού, ακολουθεί η αποτίμησή της πολυπλοκότητάς τους με τα εξής βήματα:

Μέτρηση των DETs (για ILFs και EIFs):

- Μετράται ένα DET για κάθε μοναδικό, μη επαναλαμβανόμενο, αντιληπτό από το χρήστη δεδομένο/πεδίο, το οποίο συντηρείται από ένα ILF ή ανακτάται από ένα EIF μέσω της εκτέλεσης μιας στοιχειώδους διαδικασίας.
- Μετράται ένα DET για κάθε δεδομένο/πεδίο που απαιτείται για να εδραιωθεί μια σχέση με ένα άλλο ILF ή EIF.

Μέτρηση των RETs (για ILFs και EIFs):

- Μετράται ένα RET για κάθε προαιρετική ή υποχρεωτική υποομάδα του ILF ή EIF, ή
- Εάν δεν υπάρχουν υποομάδες για το ILF ή EIF, μετράται ως ένα RET το αντίστοιχο ILF ή EIF.

Κατηγοριοποίηση Πολυπλοκότητας των ILF και EIF

Αφού μετρηθούν τα DET και RET (για κάθε ILF και EIF), ακολουθεί η κατηγοριοποίηση της πολυπλοκότητας του εκάστοτε ILF/EIF με βάση τον ακόλουθο πίνακα:

Αριθμός RETs	Αριθμός DETs		
	1-19	20-50	>50
1	Χαμηλή	Χαμηλή	Μέση
2-5	Χαμηλή	Μέση	Υψηλή
>5	Μέση	Υψηλή	Υψηλή

4.2.4.2 Αποτίμηση των λειτουργικών διαδικασιών συναλλαγών (EI, EO και EQ)

Μετά τον προσδιορισμό των λειτουργικών διαδικασιών συναλλαγών (EI, EO και EQ) για την μετρούμενη οντότητα λογισμικού, ακολουθεί η αποτίμησή της πολυπλοκότητάς τους με τα εξής βήματα:

Μέτρηση των FTRs (για EI):

- Μετράται ένα FTR για κάθε ILF που συντηρείται κατά την εκτέλεση της στοιχειώδους διαδικασίας.
- Μετράται ένα FTR για κάθε ILF ή EIF που διαβάζεται κατά την εκτέλεση της στοιχειώδους διαδικασίας.
- Μετράται ένα μόνο FTR για κάθε ILF που συντηρείται και διαβάζεται κατά την εκτέλεση της στοιχειώδους διαδικασίας.

Μέτρηση των DETs (για EI):

- Μετράται ένα DET για κάθε μοναδικό, μη επαναλαμβανόμενο, αντιληπτό από το χρήστη δεδομένο/πεδίο, το οποίο εισάγεται στο σύστημα ή εξάγεται από αυτό και απαιτείται για να ολοκληρωθεί η EI.
- Δε λαμβάνονται υπόψη στη μέτρηση των DET δεδομένα που ανακτώνται ή παράγονται από το σύστημα και αποθηκεύονται σε ένα ILF κατά τη διάρκεια μιας στοιχειώδους διαδικασίας EI εάν αυτά δεν έχουν εισαχθεί στο σύστημα ή δεν έχουν εξαχθεί από αυτό.
- Μετράται ένα DET για τη δυνατότητα του συστήματος να αποστέλλει μηνύματα έξω από τα όριά του για να ενημερώσει ότι εμφανίστηκε ένα λάθος κατά τη διάρκεια της επεξεργασίας, να πληροφορήσει ότι η επεξεργασία ολοκληρώθηκε ή να ζητήσει επιβεβαίωση για συνέχιση.
- Μετράται ένα DET για την ενεργοποίηση μια λειτουργίας ακόμα κι αν υπάρχουν πολλαπλοί τρόποι για την εκτέλεσή της.

Κατηγοριοποίηση Πολυπλοκότητας των EI

Αφού μετρηθούν τα FTR και τα DET (για κάθε EI), ακολουθεί η κατηγοριοποίηση της πολυπλοκότητας του εκάστοτε EI με βάση τον ακόλουθο πίνακα:

Αριθμός FTRs	Αριθμός DETs		
	1-4	5-15	>15
0-1	Χαμηλή	Χαμηλή	Μέση
2	Χαμηλή	Μέση	Υψηλή
>=3	Μέση	Υψηλή	Υψηλή

Μέτρηση των FTRs (για EO και EQ):

- Μετράται ένα FTR για κάθε ILF ή EIF που διαβάζεται κατά την εκτέλεση της στοιχειώδους διαδικασίας.

Επιπρόσθετοι Κανόνες Μέτρησης των FTRs (για EO):

- Μετράται ένα FTR για κάθε ILF που συντηρείται κατά την εκτέλεση της στοιχειώδους διαδικασίας.
- Μετράται ένα μόνο FTR για κάθε ILF που συντηρείται και διαβάζεται κατά την εκτέλεση της στοιχειώδους διαδικασίας.

Κανόνες Μέτρησης των DETs (για ΕΟ και ΕQ):

- Μετράται ένα DET για κάθε αντιληπτό από το χρήστη, μη επαναλαμβανόμενο δεδομένο/πεδίο, το οποίο εισάγεται στο σύστημα και απαιτείται για να καθοριστεί πότε, τι ή/και πως θα ανακτηθούν ή θα παραχθούν τα δεδομένα από τη στοιχειώδη διαδικασία.
- Μετράται ένα DET για κάθε αντιληπτό από το χρήστη, μη επαναλαμβανόμενο δεδομένο/πεδίο, το οποίο εξάγεται από το σύστημα.
- Μετράται μόνο μία φορά τα DETs που εισάγονται και εξάγονται από το σύστημα.
- Μετράται ένα DET για τη δυνατότητα του συστήματος να αποστέλλει μηνύματα έξω από τα όριά του για να ενημερώσει ότι εμφανίστηκε ένα λάθος κατά τη διάρκεια της επεξεργασίας, να πληροφορήσει ότι η επεξεργασία ολοκληρώθηκε ή να ζητήσει επιβεβαίωση για συνέχιση.
- Μετράται ένα DET για την ενεργοποίηση μια λειτουργίας ακόμα κι αν υπάρχουν πολλαπλοί τρόποι για την εκτέλεσή της.
- Δε λαμβάνονται υπόψη κατά τη μέτρηση των DETs τα δεδομένα που ανακτώνται ή παράγονται από το σύστημα και αποθηκεύονται σε ένα ILF κατά τη διάρκεια μιας στοιχειώδους διαδικασίας εάν αυτά δεν έχουν εισαχθεί στο σύστημα ή δεν έχουν εξαχθεί από αυτό.
- Δε λαμβάνονται υπόψη κατά τη μέτρηση των DETs οι αναφορές (reports), οι οθόνες (screens), οι στήλες και τα πεδία.
- Δε λαμβάνονται υπόψη κατά τη μέτρηση των DETs οι μεταβλητές σελιδοποίησης και οι χρονοσημάνσεις (για παράδειγμα δεν λαμβάνονται υπόψη οι αριθμοί σελίδων, οι πληροφορίες προσδιορισμού θέσης, οι εντολές όπως «προηγούμενη σελίδα», «επόμενη σελίδα», τα βέλη σελιδοποίησης, η εμφάνιση ημερομηνίας και χρόνου).

Κατηγοριοποίηση Πολυπλοκότητας των ΕΟ και ΕQ

Αφού μετρηθούν τα FTR και τα DET (για κάθε ΕΟ και ΕQ), ακολουθεί η κατηγοριοποίηση της πολυπλοκότητας του εκάστοτε ΕΟ και ΕQ με βάση τον ακόλουθο πίνακα:

Αριθμός FTRs	Αριθμός DETs		
	1-5	6-19	>19
0-1	Χαμηλή	Χαμηλή	Μέση
2-3	Χαμηλή	Μέση	Υψηλή
>3	Μέση	Υψηλή	Υψηλή

4.2.4.3 Αποτίμηση του Λειτουργικού Μεγέθους

Έχοντας ολοκληρώσει την παραπάνω διαδικασία για το σκέλος του λογισμικού υπό εξέταση, έχουμε υπολογίσει τον αριθμό των ILFs, EIFs, EIs, EOs και EQs ανά επίπεδο πολυπλοκότητας (low, average, high).

Τελευταίο βήμα της διαδικασίας είναι ο υπολογισμός του Λειτουργικού Μεγέθους που «συνεισφέρει» καθένα από τα ILF, EIF, EI, EO και EQ στη μέτρηση με βάση τον ακόλουθο πίνακα στάθμισης:

Πολυπλοκότητα	Στάθμιση (Weight)				
	ILF	EIF	EI	EO	EQ
Χαμηλή	7	5	3	4	3
Μέση	10	7	4	5	4
Υψηλή	15	10	6	7	6

Στον πίνακα που ακολουθεί παρουσιάζεται ένας συνοπτικός τρόπος για την άθροιση του τελικού Λειτουργικού Μεγέθους του μετρούμενου λογισμικού.

Τύπος Λειτουργίας	Επίπεδο Πολυπλοκότητας			Σύνολα Ανά Λειτουργία
	Χαμηλή	Μέση	Υψηλή	
ILFs	__ * 7=ILF1	__ * 10=ILF2	__ * 15=ILF3	ILF1+ILF2+ILF3
EIFs	__ * 5=EIF1	__ * 7=EIF2	__ * 10=EIF3	EIF1+EIF2+EIF3
EIs	__ * 3=EI1	__ * 4=EI2	__ * 6=EI3	EI1+EI2+EI3
EOs	__ * 4=EO1	__ * 5=EO2	__ * 7=EO3	EO1+EO2+EO3
EQs	__ * 3=EQ1	__ * 4=EQ2	__ * 6=EQ3	EQ1+EQ2+EQ3
Συνολικά Λειτουργικά Σημεία				(EI1+EI2+EI3) + (EO1+EO2+EO3) + (EQ1+EQ2+EQ3) + (ILF1+ILF2+ILF3) + (EIF1+EIF2+EIF3)

4.2.4.4 Αποτίμηση λειτουργικότητας μετατροπής

Η Εμβέλεια μέτρησης ενός έργου ανάπτυξης (ή βελτίωσης) λογισμικού ενδέχεται να περιλαμβάνει και λειτουργικότητα μετατροπής, η οποία προφανώς και θα πρέπει να συνηπολογιστεί στο πλαίσιο της μέτρησης.

Ως «λειτουργικότητα μετατροπής» νοείται το σκέλος ενός λογισμικού που χειρίζεται μετατροπές δεδομένων ή/και άλλες ανάγκες μετατροπής προσδιορίζονται από το χρήστη (όπως ειδικές μετατροπές στο πλαίσιο αναφορών). Σε ένα έργο βελτίωσης λογισμικού αφορά το σκέλος που χειρίζεται τυχόν διαδικασίες μετατροπής που προσδιορίζονται από το χρήστη.

Για τη μέτρηση των λειτουργικότητας μετατροπής ακολουθείται η ίδια διαδικασία (ήτοι προσδιορίζοντας το Λειτουργικό Μέγεθος για τις λειτουργικές διαδικασίες δεδομένων και τις λειτουργικές διαδικασίες συναλλαγών) σύμφωνα με τα αναφερόμενα στις παραγράφους 4.2.4.1 και 4.2.4.2.

4.2.4.5 Αποτίμηση λειτουργικότητας βελτιώσεων

Τα έργα βελτίωσης λογισμικού μπορεί να περιλαμβάνουν προσθήκες, τροποποιήσεις και απαλοιφές λειτουργικότητας σε ένα υφιστάμενο σύστημα. Η μέτρηση της λειτουργικότητας βελτιώσεων θα πρέπει να γίνεται ακολουθώντας τις εξής κατευθυντήριες γραμμές:

- Δεν αλλοιώνονται τα Όρια Μέτρησης που είχαν καθοριστεί κατά την πρώτη μέτρηση λειτουργικότητας
- Προσδιορίζονται και μετρώνται οι λειτουργικές διαδικασίες δεδομένων σύμφωνα με τη διαδικασία που περιγράφεται στην παράγραφο 4.2.4.1
- Προσδιορίζονται και μετρώνται οι λειτουργικές διαδικασίες συναλλαγών σύμφωνα με τη διαδικασία που περιγράφεται στην παράγραφο 4.2.4.2
- Το Λειτουργικό Μέγεθος της μετρούμενης εφαρμογής ενημερώνεται ώστε να απεικονίζει
 - Την λειτουργικότητα λόγω προσθηκών (η οποία αύξησε το λειτουργικό μέγεθος)
 - Την λειτουργικότητα λόγω τροποποιήσεων (η οποία είτε αύξησε, είτε μείωσε, είτε δεν επηρέασε το λειτουργικό μέγεθος)
 - Την λειτουργικότητα λόγω απαλοιφών (η οποία μείωσε το λειτουργικό μέγεθος)

Σημειώνουμε ότι μια αλλαγή σε λειτουργική διαδικασία δεδομένων, ενδέχεται να προσθέτει, να τροποποιεί ή να απαλείφει DETs ή/και RETs. Αντίστοιχα μια αλλαγή σε λειτουργική διαδικασία συναλλαγών ενδέχεται να προσθέτει, να τροποποιεί ή να απαλείφει DETs ή/και FTR. Οι αλλαγές των λειτουργικών διαδικασιών συναλλαγών ενδέχεται να τροποποιούν και την εκτελούμενη αλγοριθμική λογική.

4.2.5 Τελικός υπολογισμός Λειτουργικού Μεγέθους

Η διαδικασία ολοκληρώνεται με τον τελικό υπολογισμό του Λειτουργικού Μεγέθους, χρησιμοποιώντας την κατάλληλη μαθηματική συνάρτηση με βάση τον Τύπο της μέτρησης (όπως προσδιορίστηκε στην παρ. 4.2.2) και τα Όρια του μετρούμενου λογισμικού (όπως προσδιορίστηκαν στην παρ. 4.2.3):

1) Για Μέτρηση Λειτουργικού Μεγέθους κατά την Ανάπτυξη Λογισμικού χρησιμοποιείται η συνάρτηση:

$$DFP = ADD + CFP \quad [\text{Συνάρτηση 1}]$$

όπου:

- DFP είναι η μέτρηση λειτουργικού μεγέθους για το έργο ανάπτυξης λογισμικού.
- ADD είναι το μέγεθος της λειτουργικότητας προς το χρήστη από το έργο λογισμικού.
- CFP είναι το μέγεθος της λειτουργικότητας μετατροπής.

2) Για Μέτρηση Λειτουργικού Μεγέθους Εφαρμογής χρησιμοποιείται η συνάρτηση:

AFP = ADD [Συνάρτηση 2]

όπου:

- AFP είναι η μέτρηση λειτουργικού μεγέθους για την εφαρμογή.
- ADD είναι το μέγεθος της λειτουργικότητας προς το χρήστη από το έργο λογισμικού (χωρίς να λαμβάνονται υπόψη το μέγεθος της λειτουργικότητας μετατροπής) ή το μέγεθος της λειτουργικότητας που διαπιστώνεται κατά τη μέτρηση της εφαρμογής.

3) Για Μέτρηση Λειτουργικού Μεγέθους κατά τη Βελτίωση Υφιστάμενου Λογισμικού χρησιμοποιείται η συνάρτηση:

EFP = ADD + CHGA + CFP + DEL [Συνάρτηση 3]

όπου:

- EFP είναι η μέτρηση λειτουργικού μεγέθους για το έργο Βελτίωσης Υφιστάμενου Λογισμικού
- ADD είναι το μέγεθος της λειτουργικότητας που προστέθηκε από το έργο Βελτίωσης.
- CHGA είναι το μέγεθος της λειτουργικότητας που τροποποιήθηκε από το έργο Βελτίωσης (ως έχει μετά την ολοκλήρωση των αλλαγών).
- CFP είναι το μέγεθος της λειτουργικότητας μετατροπής.
- DEL είναι το μέγεθος της λειτουργικότητας που απαλείφθηκε από το έργο Βελτίωσης

4) Για Μέτρηση Λειτουργικού Μεγέθους Εφαρμογής κατά τη Βελτίωση Υφιστάμενου Λογισμικού χρησιμοποιείται η συνάρτηση:

AFPA = (AFPB + ADD + CHGA) - (CHGB + DEL) [Συνάρτηση 4]

όπου:

- AFPA είναι η μέτρηση λειτουργικού μεγέθους για την εφαρμογή μετά τη Βελτίωση Υφιστάμενου Λογισμικού
- AFBP είναι η μέτρηση λειτουργικού μεγέθους πριν το Βελτίωση του Λογισμικού.
- ADD είναι το μέγεθος της λειτουργικότητας που προστέθηκε από το έργο Βελτίωσης.
- CHGA είναι το μέγεθος της λειτουργικότητας που τροποποιήθηκε από το έργο Βελτίωσης (ως έχει μετά την ολοκλήρωση των αλλαγών).
- CHGB είναι το μέγεθος της λειτουργικότητας που τροποποιήθηκε από το έργο Βελτίωσης (όπως ήταν πριν την διενέργεια αλλαγών).
- DEL είναι το μέγεθος της λειτουργικότητας που απαλείφθηκε από το έργο Βελτίωσης

5 Χρησιμοποιώντας τη μεθοδολογία IFPUG FPA 4.3 για έργα SOA

5.1 Πεδίο εφαρμογής της IFPUG FPA 4.3 σε έργα SOA

Υλοποιώντας δράσεις υλοποίησης έργων σε ένα περιβάλλον Υψηροεισοστραφούς Αρχιτεκτονικής (SOA), είναι δυνατό να δαπανηθούν /αξιοποιηθούν πόροι σε μια σειρά από εργασίες και έργα που είναι απαραίτητα ώστε να υπάρξει λειτουργικό αποτέλεσμα διαθέσιμο σε παραγωγική λειτουργία [14].

Ενδεικτικά, έχουμε τα ακόλουθα αντικείμενα έργων και εργασιών που είναι δυνατό να εκτελεστούν σε ένα οικοσύστημα SOA:

- **Ανεύρεση Υπηρεσιών (Service Mining):** Αφορά την αναγνώριση και τον εντοπισμό υπηρεσιών από υπάρχοντα λειτουργούντα συστήματα.
- **Υλοποίηση Υπηρεσιών (Service Development):** Αφορά νέα υλοποίηση ή τροποποίηση της υφιστάμενης λειτουργίας.
- **Υλοποίηση Εφαρμογών (Application Development):** Αφορά υλοποίηση εφαρμογών χρησιμοποιώντας υπηρεσίες και μπορεί να αποτελεί νέα υλοποίηση ή να είναι τροποποίηση υφιστάμενης λειτουργίας.
- **Ενσωμάτωση Υπηρεσιών (Service Integration):** Αφορά την ενσωμάτωση υπηρεσιών και εφαρμογών σε νέες ολοκληρωμένες υπηρεσίες και εφαρμογές ή την τροποποίηση υφισταμένων.
- **Υποδομή SOA (SOA Infrastructure):** Αφορά την υλοποίηση, ανάπτυξη ή/και προμήθεια υποδομών SOA.
- **Διακυβέρνηση SOA (SOA Governance):** Αφορά την ανάπτυξη πολιτικών και διαδικασιών διακυβέρνησης (συμπεριλαμβανομένων δράσεων για την ένταξη και παρακολούθηση της εκτέλεσης των Συμφωνιών Παροχής Εγγυημένου Επιπέδου Υπηρεσιών - Service Level Agreements).
- **Ανάλυση Αρχιτεκτονικής SOA (SOA Architecture Analysis):** Αφορά την ανάλυση της αρχιτεκτονικής του συστήματος SOA για να διαπιστωθεί η συμμόρφωση με προδιαγραφές ποιότητας λειτουργίας (λ.χ. για την εξέταση ικανοποίησης των επιθυμητών επιπέδων ποιότητας ασφάλειας του συστήματος).

Επισημαίνεται πως η προτεινόμενη μεθοδολογία Ανάλυσης Λειτουργικών Σημείων (Function Point Analysis - FPA) του οργανισμού IFPUG [23] θα πρέπει να εφαρμόζεται για την ποσοτικοποίηση του «μεγέθους» της λειτουργικότητας έργων λογισμικού SOA που εντάσσονται αποκλειστικά στους ακόλουθους τύπους έργων:

- **Υλοποίηση Υπηρεσιών (Service Development).**
- **Υλοποίηση Εφαρμογών (Application Development).**
- **Ενσωμάτωση Υπηρεσιών (Service Integration).**

5.1.1 Δράσεις που δεν συνεισφέρουν Λειτουργικά Σημεία

Ανεξάρτητα από την εφαρμοζόμενη αρχιτεκτονική- μέθοδο υλοποίησης λογισμικού υπογραμμίζεται η διαφοροποίηση μεταξύ έργων ανάπτυξης και βελτίωσης λογισμικού τα οποία συνεισφέρουν Λειτουργικά Σημεία και άλλων δράσεων κατά την ανάπτυξη λογισμικού οι οποίες δεν συνεισφέρουν στη μέτρηση Λειτουργικού Μεγέθους.

Οι παρακάτω δραστηριότητες είναι ενδεικτικά παραδείγματα, δράσεων έργων λογισμικού που τυπικά δεν μπορούν να προσεγγιστούν χρησιμοποιώντας μεθοδολογίες μέτρησης Λειτουργικού Μεγέθους:

- **Ενέργειες υποστήριξης εφαρμογών:**

- Υποστήριξη παραγωγικής διαδικασίας,
- Ανίχνευση και διόρθωση σφαλμάτων (debugging)
- Απόσυρση συστήματος από τη χρήση
- Ενέργειες διαμόρφωσης του υπολογιστικού περιβάλλοντος (πχ. για τη βελτίωση χρόνου απόκρισης ή της ασφάλειας)
- Συμβουλευτική και ad hoc υποστήριξη εμπλεκομένων
- Έναρξη λειτουργίας έργου (deployment)
- Δράσεις μετάπτωσης και εισαγωγής δεδομένων σε βάσεις δεδομένων (η ανάπτυξη διαδικασίας για τη μετάπτωση λογίζεται κατά τη μέτρηση των Λειτουργικών Σημείων – όχι όμως και η καθεαυτή διαδικασία μετάπτωσης η οποία στην πράξη εξαρτάται από το πλήθος των εγγραφών και όχι από τη διαδικασία).

5.2 Μετρώντας επαναχρησιμοποιούμενες μονάδες λογισμικού με την IFPUG FPA 4.3

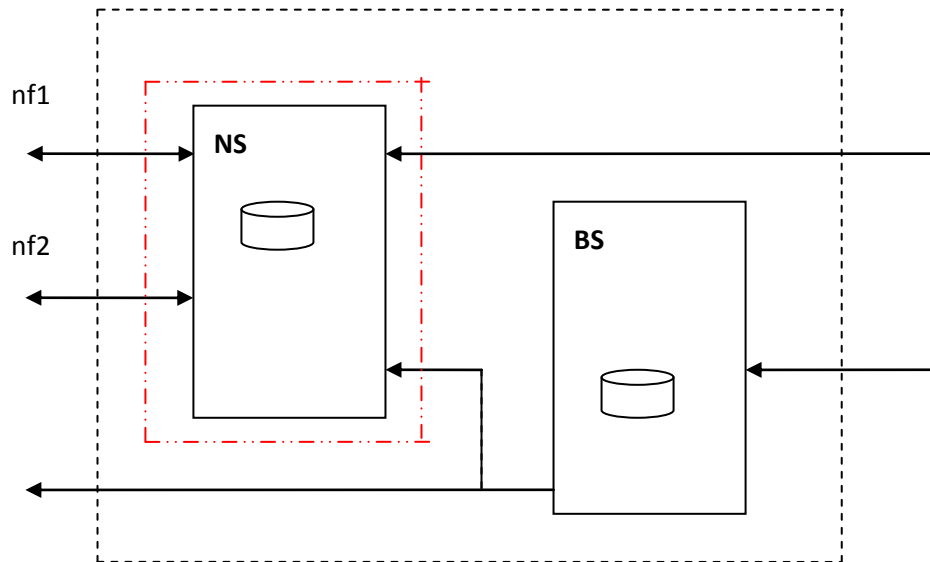
Το πρόβλημα της μέτρησης Λειτουργικού Μεγέθους αποκτά ειδικότερο ενδιαφέρον σε λογισμικό το οποίο βασίζεται σε επαναχρησιμοποιούμενα συνθετικά μέρη. Αναφερόμαστε σε περιβάλλοντα που ο κώδικας και η αρχιτεκτονική του λογισμικού έχει ειδικά σχεδιαστεί ώστε να ενθαρρύνεται η επαναχρησιμοποίηση από άλλες εφαρμογές. Σε κάθε περίπτωση βέβαια μια «καλή» μέθοδος ανάπτυξης λογισμικού εμπεριέχει την πρακτική της επαναχρησιμοποίησης κώδικα ή βιβλιοθηκών που αναπτύσσονται.

Η παρούσα προσέγγιση δεν αφορά περιπτώσεις μιας απλοϊκής επαναχρησιμοποίησης τύπου «copy library», αλλά στη χρήση ολοκληρωμένων συνθετικών μερών λογισμικού που λειτουργούν αυτόνομα σε χαλαρή διασύνδεση με άλλα συνθετικά μέρη ή εφαρμογές, όπως συμβαίνει σε ένα SOA οικοσύστημα. Στην περίπτωση αυτή τα συνθετικά μέρη του περιβάλλοντος είναι σχεδιασμένα ώστε να εκτελούν λειτουργίες σε δεδομένα καθώς και να μπορούν να εννορηστρώνονται συνθέτοντας νέες υπηρεσίες.

Η προτεινόμενη μεθοδολογία από τον IFPUG [25] για τη μέτρηση του Λειτουργικού Μεγέθους σε αυτές τις περιπτώσεις συνίσταται και πάλι στα ακόλουθα τρία βήματα:

- **Καθορισμός του εύρους της μέτρησης,**
- **Αναγνώριση των ορίων των εφαρμογών και των συνθετικών μερών που θα μετρηθούν,**
- **Καθορισμός λειτουργικών διαδικασιών δεδομένων και λειτουργικών διαδικασιών συναλλαγών.**

Για να γίνει πιο κατανοητή η προτεινόμενη μεθοδολογία ας υποθέσουμε ένα πρόβλημα μέτρησης λειτουργικού μεγέθους στο οικοσύστημα που περιγράφηκε, το οποίο αφορά στην υλοποίηση μιας Νέας Υπηρεσίας (New Service - NS) που θα αξιοποιεί μια Υφιστάμενη Μονάδα Λογισμικού (Base Service - BS). Η BS είναι ήδη υλοποιημένη και βρίσκεται σε παραγωγική λειτουργία. Η BS θα αξιοποιηθεί από τη NS και παράλληλα θα συνεχίζει να λειτουργεί και αυτόνομα, σύμφωνα με το ακόλουθο σχήμα.



Σχήμα 2: Επαναχρησιμοποίηση Λογισμικού

Αξιοποιώντας την έξοδο από τη BS η υπηρεσία NS θα παρέχει λειτουργίες (έστω nf1 και nf2) για κλήσεις σε δεδομένα που συντηρεί αυτόνομα η ίδια.

Το Λειτουργικό Μέγεθος της υφιστάμενης μονάδας λογισμικού BS έχει μετρηθεί αυτόνομα και περιλαμβάνει ένα (1) EI, ένα (1) ILF και ένα (1) EO.

Το πρόβλημα εντοπίζεται στον τρόπο μέτρησης του Λειτουργικού Μεγέθους της Νέας Υπηρεσίας (NS), η οποία επαναχρησιμοποιεί την BS, χωρίς όμως κατά τη διαδικασία να διπλομετρηθούν Λειτουργικά Σημεία που παρέχονται και αυτόνομα από την BS.

Σύμφωνα με όσα προβλέπονται στο [23], είναι απαραίτητος προσεκτικός προσδιορισμός του εύρους της μέτρησης και ορθός, καθορισμός των ορίων των εφαρμογών και των συνθετικών μερών που θα μετρηθούν, καθώς και του σκοπού της μέτρησης.

Το υπό εξέταση πρόβλημα μπορεί να προσεγγιστεί με δύο τρόπους [25]:

- Αφενός με μια «επιχειρησιακή προσέγγιση» όπου η νέα υπηρεσία (NS) μετράται για το σύνολο της παρεχόμενης λειτουργίας της ως να ήταν μια εφαρμογή χωρίς να ενδιαφέρει η επαναχρησιμοποίηση της υπηρεσίας BS. Το αποτέλεσμα είναι ότι τα Λειτουργικά Σημεία που αφορούν στην αλληλεπίδραση της BS με την NS παραλείπονται δίδοντας μια αποτίμηση της λειτουργικότητας για τη συνολική λειτουργία της NS που θα είναι δύο (2) EIs, δύο (2) ILFs, δύο (2) EQs.
- Αφετέρου με μια «τεχνική προσέγγιση» όπου υπηρεσία BS λαμβάνεται ως ξεχωριστή εφαρμογή που επαναχρησιμοποιείται από την NS. Τα Λειτουργικά Σημεία που αφορούν στην ανάπτυξη της NS θα είναι ένα (1) EI, ένα (1) EIF, ένα (1) ILF, δύο (2) EQs. όπου μετρώνται επίσης τα Λειτουργικά Σημεία της αλληλεπίδρασης με την BS. Η προσέγγιση είναι χρήσιμη για την μέτρηση της προσπάθειας που απαιτείται για την νέα ανάπτυξη.

Συγκρίνοντας τα αποτελέσματα της μέτρησης για το Λειτουργικό Μέγεθος της NS υιοθετώντας την «τεχνική προσέγγιση» με τα αποτελέσματα της μέτρησης επιλέγοντας την «επιχειρησιακή προσέγγιση», είναι εφικτή η αξιολόγηση της επιλογής να αναπτυχθεί και να συντηρείται η BS ώστε να λειτουργεί ως επαναχρησιμοποιούμενο συνθετικό.

Ο «επιχειρησιακός χρήστης» δεν ενδιαφέρεται να γνωρίζει πως προκύπτει το αποτέλεσμα του λειτουργικού μεγέθους που μετρά γιατί δίνει έμφαση στη μέτρηση της λειτουργικότητας που τελικά αυτός χρησιμοποιεί. Τον ενδιαφέρει «ποιά, πόση» λειτουργικότητα προσφέρεται στο επιχειρησιακό περιβάλλον και όχι «πως» αυτή παρέχεται.

Ο «τεχνικός χρήστης» ενδιαφέρεται να γνωρίζει τα Λειτουργικά Σημεία που αφορούν την αλληλεπίδραση μεταξύ των συνθετικών μερών και του λογισμικού που επαναχρησιμοποιείται καθώς και τα Λειτουργικά Σημεία που αφορούν στην επιπλέον υλοποίηση. Με τον τρόπο αυτό είναι σε θέση να διαστασιολογήσει εύστοχα την προσπάθεια και τον χρόνο που χρειάζεται για την ανάπτυξη ή/και το κόστος κάποιας σχετικής ανάθεσης σε εξωτερικό εργολάβο. Ο «τεχνικός χρήστης» ενδιαφέρεται για το «πως» μπορεί να παρασχεθεί η λειτουργικότητα που ζητείται από τον «επιχειρησιακό χρήστη».

Σύνοψη «τεχνικής προσέγγισης»:

Βασικοί κανόνες με τους οποίους εφαρμόζεται η τεχνική προσέγγιση για τη μέτρηση Λειτουργικού Μεγέθους σε περιβάλλον επαναχρησιμοποιούμενου λογισμικού, είναι οι ακόλουθοι:

- a. Δίνεται έμφαση στην αυτόνομη μέτρηση του Λειτουργικού Μεγέθους για τις υπομονάδες λογισμικού που επαναχρησιμοποιούνται, ειδικότερα για τα συνθετικά μέρη του λογισμικού που έχει αποφασιστεί ότι ευρύτερα θα χρησιμοποιούνται από άλλο λογισμικό και νέες εφαρμογές.
- b. Κατά την υλοποίηση μιας νέας εφαρμογής με επαναχρησιμοποίηση υφιστάμενου λογισμικού (χωρίς αυτό να τροποποιηθεί), αθροίζονται στο Λειτουργικό Μέγεθος της νέας εφαρμογής και τα Λειτουργικά Σημεία της αλληλεπίδρασης της νέας υλοποίησης με το υφιστάμενο λογισμικό.
- c. Αν κατά την υλοποίηση μιας νέας εφαρμογής απαιτηθεί αλλαγή στη λογική της λειτουργίας ενός υφιστάμενου συνθετικού που χρησιμοποιείται από τη νέα εφαρμογή, τότε τα Λειτουργικά Σημεία που το αφορούν θα αθροιστούν (μετά την εκτέλεση των αλλαγών) ως να ήταν νέα υλοποίηση και θα αθροιστούν στο Λειτουργικό Μέγεθος της νέας εφαρμογής.
- d. Τα Λειτουργικά Σημεία που αθροίζονται κατά την εκ νέου υλοποίηση/τροποποίηση υφιστάμενου συνθετικού (ως συνέπεια του (c) για παράδειγμα) δεν λαμβάνονται υπόψη στις υλοποιήσεις που αυτό επαναχρησιμοποιείται μόνο εφόσον δεν προκύπτουν αλλαγές στον τρόπο λειτουργίας των εφαρμογών που το επαναχρησιμοποιούν.

Σημειώνουμε ότι το περιβάλλον ανάπτυξης όπου η επαναχρησιμοποίηση λογισμικού είναι διαδεδομένη (όπως ένα περιβάλλον SOA) πρέπει να θεωρείται ξεχωριστή κατηγορία. Τα όποια συμπεράσματα προκύπτουν από τις μετρήσεις Λειτουργικού Μεγέθους των υλοποιήσεων, σχετικά με την απόδοση και την παραγωγικότητα δεν μπορούν να συγκρίνονται με τις μετρήσεις που προέρχονται από εφαρμογές που υλοποιούνται ακολουθώντας παραδοσιακές μεθόδους ανάπτυξης λογισμικού.

Ασχέτως της επιλογής προσέγγισης, ιδιαίτερη σημασία έχει να τηρηθεί με συνέπεια η απόφαση που θα παρθεί σχετικά με την τακτική που θα ακολουθείται κατά τη μέτρηση Λειτουργικού Μεγέθους. Η απόφαση αυτή θα πρέπει να τηρείται για όλο το εύρος του συστήματος ή των συστημάτων που εξετάζονται και για όλο τον κύκλο ζωής των συστημάτων αυτών.

5.3 Παραδείγματα μέτρησης επαναχρησιμοποιούμενων μονάδων λογισμικού με την IFPUG FPA 4.3

Για την καλύτερη κατανόηση του τρόπου μέτρησης λειτουργικού μεγέθους στις περιπτώσεις επαναχρησιμοποίησης λογισμικού και κλήσης Web Service για υποβολή ή λήψη δεδομένων παρουσιάζουμε δύο σενάρια –παραδείγματα, που βασίζονται στις γενικές κατευθύνσεις της IFPUG [23] [26].

5.3.1 Σενάριο 1 – Κλήση Web Service για την πρόσβαση σε δεδομένα

Έστω ένα Web Service B (WS-B) το οποίο πραγματοποιεί κλήση σε ήδη υλοποιημένο Web Service A (WS-A) για την πρόσβαση σε δεδομένα που το δεύτερο συντηρεί και παρέχει. Συγκεκριμένα υποθέτουμε μια λειτουργία που εκτελείται από το WS-B και για την ολοκλήρωσή της απαιτεί τη λήψη δεδομένων, τα οποία βρίσκονται σε πηγή που συντηρείται από το WS-A.

Για παράδειγμα μπορούμε να θεωρήσουμε ότι το WS-A παρέχει δεδομένα μητρώου φορολογουμένων και ότι το WS-B παρέχει μια υπηρεσία εξατομικευμένης ενημέρωσης για ληξιπρόθεσμες οφειλές, κατά την εκτέλεση της οποίας απαιτείται η επιβεβαίωση της εγκυρότητας των στοιχείων ταυτοποίησης του φυσικού προσώπου που αιτείται την ενημέρωση.

Προσέγγιση Μέτρησης:

Ως προς το WS-A: Από την πλευρά του Web Service A, τα δεδομένα είναι διαθέσιμα «εντός» του Web Service και συντηρούνται από αυτό, οπότε το αρχείο που τηρεί τα δεδομένα αποτελεί Εσωτερικό Λογικό Αρχείο (ILF) για το WS-A. Μέσω κλήσης (EQ) υποσύνολο αυτών διαβιβάζεται προς το WS-B. Άλλες απαιτήσεις λειτουργίας δε σημειώνονται και άλλα Λειτουργικά Σημεία δεν μετρώνται.

Ως προς το WS-B: Από την πλευρά του Web Service B, στη λειτουργία του εμπλέκεται μόνο μια πηγή δεδομένων. Συγκεκριμένα κατά τη μέτρηση των Λειτουργικών Σημείων για το WS-B η πηγή δεδομένων είναι «εντός» του WS-A και λαμβάνεται υπόψη ως Εξωτερικό Αρχείο Διεπαφών (EIF). Σημειώνεται ότι για τη μέτρηση λειτουργικού μεγέθους της διαδικασίας που εκτελεί το WS-B στη συνέχεια, το αρχείο δεδομένων λαμβάνεται υπόψη ως Αναφερόμενο Αρχείο (File Type Referenced – FTR) πχ (EQ).

Καταλήγοντας, στην περίπτωση που θεωρήσουμε το WS-A ως ήδη υλοποιημένο, τότε για το WS-B η διαδικασία επικοινωνίας και λήψης δεδομένων μέσω του WS-A προσθέτει στη μέτρηση λειτουργικού μεγέθους τα Λειτουργικά Σημεία που προκύπτουν λόγω του Εξωτερικού Αρχείου Διεπαφών (EIF). Λειτουργικά Σημεία προκύπτουν στη συνέχεια στο WS-B, λόγω του συνυπολογισμού του ως Αναφερόμενου Αρχείου (FTR), (EQ) κατά την επιβεβαίωση της εγκυρότητας των στοιχείων ταυτοποίησης.

5.3.2 Σενάριο 2 – Κλήση Web Service για την αποστολή δεδομένων

Έστω μια ήδη υλοποιημένη Εφαρμογή A (ΕΦ-A) η οποία παράγει ένα ακριβές αντίγραφο των δεδομένων που συντηρεί (χωρίς να εφαρμόσει λογική επεξεργασία), και το υποβάλλει σε ένα Web Service B (WS-B) το οποίο αναλαμβάνει να το φορτώσει χωρίς να το επεξεργαστεί περεταίρω. Συγκεκριμένα υποθέτουμε ότι το WS-B έχει την ιδιότητα να λαμβάνει ένα Αρχείο (έστω «ΑΡΧ») που θα παράγει η ΕΦ-A με σκοπό να διασταυρώνει δεδομένα και να κάνει αναφορά (validation and reference only).

Η απαίτηση αναφέρει ότι η ΕΦ-A παράγει με βάση τα δεδομένα που τηρεί ένα πλήρες αρχείο (χωρίς να έχει προηγηθεί κάποιου είδους επεξεργασία) το οποίο υποβάλλει στο WS-B. Κάθε φορά που το ΑΡΧ

αποστέλλεται στο WS-B ενεργοποιείται διαδικασία ενημέρωσης που επικαιροποιεί τα τηρούμενα δεδομένα από το WS-B σε συμφωνία με τα περιεχόμενα του ληφθέντος Αρχείου.

Για παράδειγμα μπορούμε να θεωρήσουμε ότι η ΕΦ-A συντηρεί δεδομένα ηλεκτρονικών πληρωμών και στο τέλος κάθε ημέρας παράγει ένα εξαγόμενο αρχείο με το σύνολο των εγγραφών τις οποίες και υποβάλλει καλώντας το WS-B. Στην περίπτωση αυτή το WS-B προβλέπεται να φορτώνει το αρχείο που έλαβε ώστε να εξυπηρετεί εσωτερικές του διαδικασίες διασταύρωσης και αναφοράς.

Προσέγγιση Μέτρησης:

Από την οπτική πλευρά της ΕΦ-A, αυτή η μεταφορά δεδομένων μέσω του APX είναι μια τεχνική επιλογή, για την ικανοποίηση μιας επιχειρησιακής απαίτησης υποβολής των δεδομένων στο WS-B.

Τα δεδομένα εξακολουθούν να διατηρούνται «εντός» της ΕΦ-A και ως εκ τούτου η αντιγραφή τους σε αρχείο, αποτελεί τη λύση σε μια μη λειτουργική απαίτηση (λ.χ. τα δεδομένα της ΕΦ-A να είναι διαθέσιμα στο WS-B ακόμα και αν η Εφαρμογή ΕΦ-A βρίσκεται εκτός λειτουργίας).

Η κύρια ανάγκη του WS-B είναι να χρησιμοποιήσει τα δεδομένα της ΕΦ-A για την αναφορά και τη διασταύρωση των δεδομένων που τηρεί. Ως εκ τούτου τόσο για την δημιουργία του αρχείου όσο και για την εισαγωγή του δεν απαιτείται επεξεργασία από τα αντίστοιχα λογισμικά. Με βάση τα ανωτέρω προκύπτει ότι η διαδικασία «μεταφοράς» του αρχείου, δημιουργία αντιγράφου και φόρτωση –κλήση δηλαδή του WS-B, είναι μέρος μιας τεχνικής λύσης και δεν αθροίζονται στο Λειτουργικό Μέγεθος ούτε της ΕΦ-A ούτε του WS-B. Αντίστοιχα, το φόρτωμα- επικαιροποίηση με τα δεδομένα του APX που συμβαίνει στο WS-B, επίσης δεν έχει λειτουργικό σημείο.

Στην πράξη μετρώντας αυτόνομα το Λειτουργικό Μέγεθος της ΕΦ-A υπάρχει το ενδεχόμενο να μην είναι σαφές ότι η εν λόγω λειτουργικότητα υπάρχει για να ικανοποιεί μια μη λειτουργική απαίτηση, και δεν μπορεί να αποκλειστεί ο κίνδυνος να προσμετρηθεί λανθασμένα ως μια (1) Εξωτερική Έξοδος (EO) ή μια Εξωτερική Αναζήτηση (EQ).

Επί της ουσίας από την πλευρά της Εφαρμογής A το δημιουργούμενο APX λογίζεται ως Εσωτερικό Λογικό Αρχείο (ILF).

Από την πλευρά του Web Service B το λαμβανόμενο αντίγραφο του APX λογίζεται ως Εξωτερικό Αρχείο Διεπαφών (EIF).

Καταλήγοντας, στην περίπτωση που θεωρήσουμε την ΕΦ-A ως ήδη υλοποιημένη, τότε για το WS-B η διαδικασία επικοινωνίας και υποβολής δεδομένων μέσω του WS-B προσθέτει στη μέτρηση λειτουργικού μεγέθους τα Λειτουργικά Σημεία που προκύπτουν αποκλειστικά λόγω του Εξωτερικού Αρχείου (EIF). Οι λειτουργίες διασταύρωσης και αναφοράς που θα λαμβάνουν χώρα στη συνέχεια στο WS-B θα συνυπολογίζουν Λειτουργικά Σημεία για το APX ως Αναφερόμενο Αρχείο (FTR) πχ (EQ).

Παραλλαγή Σεναρίου 2: Υποθέτουμε πως μετά την κλήση του Web Service B (WS-B) για την υποβολή δεδομένων, το WS-B επιστρέφει μήνυμα επιβεβαίωσης (acknowledgment). Στην περίπτωση αυτή το WS-B παρέχει επιπλέον ένα (1) EO, σε Αναφερόμενο Αρχείο ενός DET,

5.4 Διαφοροποίηση στη μέτρηση Λειτουργικού Μεγέθους μεταξύ RESTful και SOAP Web Services

Υλοποιώντας λογισμικό σε ένα περιβάλλον Υπηρεσιοστραφούς Αρχιτεκτονικής, τα βασικά συστατικά της ανάπτυξης είναι οι διαδικτυακές υπηρεσίες (Web Services - WS). Η βιομηχανία και τα πρότυπα παρέχουν δύο τύπους διαδικτυακών υπηρεσιών: τα RESTful και τα SOAP. Τα RESTful WS βασίζονται στην αρχιτεκτονική Representational State Transfer (REST) και τα SOAP στο πρωτόκολλο Simple Object Access Protocol.

Λόγω των ειδικών χαρακτηριστικών που διαθέτει ο κάθε τύπος WS και της πολυμορφίας των αναγκών που θα προκύψουν κατά την ανάπτυξη εφαρμογών στο περιβάλλον της Γ.Γ.Π.Σ., είναι βέβαιο ότι και οι δύο τύποι θα συνυπάρξουν υπηρετώντας συγκεκριμένους στόχους.

Ενδεικτικά σε μια υλοποίηση όπου διαδικτυακές υπηρεσίες (WS) χωρίς μνήμη (stateless) μπορούν να καλύψουν τις απαιτήσεις και παράλληλα το περιβάλλον παραγωγικής λειτουργίας έχει πεπερασμένους υπολογιστικούς πόρους, θα προτιμηθούν RESTful WS. [28]

Σε υλοποίηση για την οποία επιβάλλεται η υποστήριξη λειτουργιών ασύγχρονης κλήσης και απάντησης (asynchronous request-response), ή διαλογική κλήση λειτουργιών και διαβίβαση μηνυμάτων τα SOAP WS είναι μονόδρομος.

Αναφορικά με τη μέτρηση Λειτουργικών Σημείων σε RESTful WS τα πράγματα είναι απλά, αφού στη πράξη οι μέθοδοι υλοποιούν λειτουργίες CRUD (Create, Read, Update, Delete), επιστρέφοντας σύγχρονα κάποια επιβεβαίωση εκτέλεσης (acknowledgment). Τα παραδείγματα της παραγράφου 5.3 (Σενάριο 1,2 και τροποποιημένο Σενάριο 2) καλύπτουν την περίπτωση RESTful WS, αναλύοντας τη λειτουργία για το γενικό παράδειγμα stateless WS [26].

Η περίπτωση των SOAP WS, παρουσιάζει ειδικό ενδιαφέρον στη μέτρηση Λειτουργικού Μεγέθους στην περίπτωση που υλοποιείται ασύγχρονη επικοινωνία, με την πλατφόρμα να παρέχει τη σχετική λειτουργικότητα.

Συγκεκριμένα για την Γ.Γ.Π.Σ. το περιβάλλον ανάπτυξης και λειτουργίας WS αναμένεται να είναι ο Weblogic Server της Oracle (είτε κάποια πλατφόρμα με αντίστοιχη λειτουργικότητα). Ουσιαστικό πλεονέκτημα αυτού του είδους συστημάτων λογισμικού, είναι ότι παρέχουν προηγμένη λειτουργικότητα σύμφωνα με τα διαδεδομένα πρότυπα. Έτσι η συμμόρφωση των υλοποιήσεων των WS με χαρακτηριστικά του SOAP διασφαλίζεται μέσω παραμετροποίησης και διαμόρφωσης του περιβάλλοντος της πλατφόρμας, το οποίο παρέχει έτοιμες κλάσεις και μεθόδους προς ενσωμάτωση κατά την υλοποίηση ή τη λειτουργία των WS και των σχετικών εφαρμογών. Τέτοια προηγμένα χαρακτηριστικά των SOAP WS σύμφωνα με τα [30] [31] [32] [33] τα οποία υποστηρίζονται από Weblogic Server είναι τα ακόλουθα:

- WS-security,
- Reliable Messaging,
- Asynchronous Request-Response,
- Callbacks,
- Conversational,
- Buffered.

Για παράδειγμα έστω ένα SOAP Buffered WS (Buff-WS) όπου κατά την κλήση του, ο καλών (WS consumer) υποβάλει δεδομένα στο WS, χωρίς να περιμένει απόκριση (response, ack). Τα δεδομένα της υποβολής τηρούνται σε ουρά (queue) του παρόχου (WS provider), έως ότου κάποια άλλη εφαρμογή αποκριθεί ή το WS τα επεξεργαστεί. Παρέχονται επίσης αναφορές της κατάστασης της ουράς. Σημειώνεται πως η

πλατφόρμα παρέχει κλάσεις και μεθόδους μέσω των οποίων υλοποιείται η διαχείριση και οι αναφορές κατάστασης της ουράς των μηνυμάτων για το σύνολο των WS του παρόχου.

Σε αναλογία λοιπόν με το παράδειγμα «Σενάριο 2 – Κλήση Web Service για την αποστολή δεδομένων» (βλ. παράγραφο 5.3.2) για τα Λειτουργικά Σημεία του WS-B, για το Buff-WS θα μετράμε ένα (1) EIF λόγω της υποβολής του αρχείου δεδομένων. Από την τήρηση της ουράς των μηνυμάτων θα μετρήσουμε άλλο ένα (1) ILF. Οι αναφορές για την ουρά των μηνυμάτων αποτελούν επιπλέον ένα (1) EQ όπως σημειώνεται και σε σχετικό παράδειγμα στο [29]. Επομένως, ανεξάρτητα από τα Λειτουργικά Σημεία που θα παρέχει το συγκεκριμένο Buff-WS, η πλατφόρμα θα παρέχει ένα (1) ILF και ένα (1) EQ σε κάθε Buffered WS που έχει αναπτυχθεί.

Η πρακτική λοιπόν που θα πρέπει να ακολουθηθεί σε συνέπεια και με το παράδειγμα της παραγράφου «5.2 - Μετρώντας επαναχρησιμοποιούμενες μονάδες λογισμικού» και σύμφωνα με την «τεχνική προσέγγιση», είναι ότι τα Λειτουργικά Σημεία που παρέχονται από συνθετικά λογισμικού που επαναχρησιμοποιούνται, θα πρέπει να μετρηθούν εφάπαξ. Στην περίπτωση μας επαναχρησιμοποιούνται συστατικά που παρέχονται από την πλατφόρμα Weblogic Server της Oracle και προφανώς δε χρειάζεται να μετρηθούν. Θα μετρηθεί μόνο η κλήση από το Buff-WS της σχετικής κλάσης/μεθόδου που παρέχει τη λειτουργικότητα διαχείρισης της ουράς και παρουσιάζεται αναλυτικά στο [34]

Συνοψίζοντας στο παράδειγμά μας, το Buff-WS θα έχει ένα (1) EIF λόγω των δεδομένων που υποδέχεται και ένα (1) επιπλέον EIF λόγω της κλήσης της μεθόδου που παρέχεται από τον Weblogic Server και υποστηρίζει τη λειτουργικότητα τήρησης και αναφορών για την ουρά, συνολικά δηλαδή δύο (2) EIF.

Σε αναλογία με την **Παραλλαγή Σεναρίου 2** της παραγράφου 5.3.2 υποθέτουμε πως μετά την κλήση του Buff-WS για την υποβολή δεδομένων, επιστρέφεται στον καλούντα μήνυμα επιβεβαίωσης (acknowledgment). Στην περίπτωση αυτή κανένα επιπλέον λειτουργικό σημείο δεν μετράται για το Buff-WS γιατί η λειτουργία αυτή παρέχεται από το περιβάλλον του Weblogic Server.

6 Μεθοδολογίες για την προεκτίμηση αξιοποιώντας δεδομένα IFPUG

6.1 Ερευνητικές προσεγγίσεις

Είναι εύλογο το ενδιαφέρον που υπάρχει παγκοσμίως για την καλή προεκτίμηση, της προσπάθειας που απαιτείται για την ανάπτυξη νέων έργου λογισμικού. Στην κατεύθυνση αυτή χρησιμοποιούνται δεδομένα από υλοποιήσεις έργων, με ευρύτητα διαδεδομένη και αξιοποιούμενη τη βάση δεδομένων για έργα ανάπτυξης λογισμικού που τηρείται από το ISBSG [37] [42]. με μετρήσεις Λειτουργικών Μεγεθών κατά IFPUG. Είναι επίσης χαρακτηριστικό ότι γύρω από το πρόβλημα καταγράφεται έντονο ερευνητικό ενδιαφέρον και αποτελεί πρόκληση για τον ακαδημαϊκό χώρο [35] [36]

Συγκεκριμένα πρόσφατα παρουσιάστηκε ανάλυση των δεδομένων της βάσης του ISBSG εφαρμόζοντας ένα μοντέλο πρόβλεψης βασιζόμενο στη χρήση συγκεχυμένων συνόλων δεδομένων σε δένδρα απόφασης (Fuzzy ID3 decision tree) για την πρόβλεψη της προσπάθειας στην ανάπτυξη έργων λογισμικού [36]. Για τη μετρήση της ακρίβειας των προβλέψεων χρησιμοποιήθηκαν η μέση τιμή σχετικού σφάλματος MMRE (Mean Magnitude of Relative Error) και Pred (l) (Prediction at level l).

Σειρά από πειράματα απέδειξε ότι η χρήση δένδρων αποφάσεων (ID3 decision trees) με σύνολα δεδομένων που έχουν γίνει συγκεχυμένα (Fuzzy Sets) επιτρέπει στο μοντέλο να διαχειριστεί την αβεβαιότητα και την μειωμένη ακρίβεια στα δεδομένα των έργων της βάσης του ISBSG.

6.2 Χρήση των μεθόδων του ISBSG

Ακόμα και αν δεν χρησιμοποιούνται προηγμένες τεχνικές από το χώρο της τεχνητής νοημοσύνης η χρήση της βάσης του ISBSG προσφέρει δεδομένα και έτοιμες συναρτήσεις για την εξαγωγή συμπερασμάτων με βάση 2.600 και πλέον έργα που έχουν υλοποιηθεί και καταμετρηθεί κατά IFPUG παγκοσμίως. Στο [39] παρουσιάζεται εκτενώς η μεθοδολογία χρήσης της βάσης του ISBSG και οι βασικές οι δυνατότητες που παρέχονται είναι οι ακόλουθες:

Εκτίμηση με χρήση συναρτήσεων παλινδρόμησης (Regression): Πραγματοποιείται με χρήση χαρακτηριστικών που παρέχονται ως πεδία στη βάση του ISBSG και τα οποία εκτιμάται ότι αφορούν το υπό εκτίμηση έργο. Τα βασικά χαρακτηριστικά είναι τα ακόλουθα:

- Πλατφόρμα (mainframe, midrange, PC, and multiplatform). Η θεώρηση δεν αφορά μόνο το υλικό αλλά και τη συνολική υποδομή με το σκεπτικό ότι ένα έργο που αναπτύσσεται σε mainframe ή multiplatform υποδομή, έχει πολλούς χρήστες και θα αφορά πολλές επιχειρησιακές μονάδες του οργανισμού.
- Τύπος γλώσσας προγραμματισμού (3GL, 4GL, and application generator)
- Τύπος ανάπτυξης (enhancement and new development).
- Συνδυασμός χαρακτηριστικών (platforms, language type, development type)

Οι συναρτήσεις λαμβάνουν ως παράμετρο την εκτίμηση του μεγέθους του έργου σε Λειτουργικά Σημεία και προαιρετικά το μέγεθος της ομάδας έργου και δίδουν τιμές για τα ακόλουθα μεγέθη

- Project delivery rates: Παραγωγικότητα εκφρασμένη σε ώρες ανά λειτουργικό σημείο.

- **Effort:** Ανθρωποχρόνος σε ώρες για την ομάδα ανάπτυξης. Δεν αφορά τον ανθρωποχρόνο για την παρακολούθηση του έργου ή την επιβάρυνση του προσωπικού που διαχειρίζεται υλικό και πλατφόρμες (βλ. παρ 5.1.1 - Δράσεις που δεν συνεισφέρουν Λειτουργικά Σημεία)
- **Duration :** Διάρκεια υλοποίησης σε μήνες.
- **Speed of delivery:** Ταχύτητα παράδοσης εκφρασμένη σε Λειτουργικά Σημεία ανά μήνα για το σύνολο του χρόνου υλοποίησης του έργου.
- **Speed of delivery per person:** Ταχύτητα παράδοσης εκφρασμένη σε Λειτουργικά Σημεία ανά μήνα και ανά άτομο της ομάδας έργου.

Εκτίμηση με σύγκριση: Δημιουργώντας μια σύνοψη από έργα με συγκρίσιμα χαρακτηριστικά με το υπό υλοποίηση έργο και υιοθέτηση των διαμέσων (median) ή τις μέσες τιμές (main average) των τιμών που ενδιαφέρουν για το υπό εκτίμηση έργο.

Εκτίμηση σε αναλογία: Ανακαλύπτοντας από τη βάση ένα ή δύο έργα με χαρακτηριστικά σε πολύ μεγάλη ομοιότητα με το υπό υλοποίηση έργο, υιοθετούνται άμεσα οι τιμές που ενδιαφέρουν για το υπό εκτίμηση έργο.

Εκτίμηση με χρήση ανάλυσης δομής έργου (Work Breakdown Structure): όπου εφαρμόζοντας κοινές παραδοχές, το έργο αναλύεται σε υποέργα που εμφανίζουν συνεκτικότητα και κατά το δυνατό αυτοτέλεια και στη συνέχεια εφαρμόζεται μία από τις προηγούμενες μεθόδους.

Επισημαίνεται πως αποτελεί διεθνή πρακτική, εταιρείες και όμιλοι που αναπτύσσουν λογισμικό για πελάτες τους είτε εσωτερικά για δική τους χρήση, να τηρούν σε βάσεις δεδομένα των έργων τους με μετρήσεις Λειτουργικών Σημείων, γραμμών κώδικα, κόστους και χρόνου εκτέλεσης και προϋπολογισμού. Συνηθίζεται πολλές φορές να διαμοιράζονται και να χρησιμοποιούν από κοινού με τρίτους τις βάσεις δεδομένων αυτές.

6.3 Ορόσημα και μετρούμενα μεγέθη

Ανεξάρτητα της μεθόδου προεκτίμησης που θα εφαρμόζεται, εφόσον αυτή βασίζεται σε ιστορικά δεδομένα, θα πρέπει να τηρούνται δεδομένα μέτρησης μεγεθών σε συγκεκριμένα ορόσημα της εκτέλεσης του έργου. Ορόσημα μπορούν να τεθούν σε οποιαδήποτε φάση υλοποίησης του έργου, ωστόσο, τα ελάχιστα ορόσημα που ακολουθούνται κατά την διεθνή πρακτική για την μέτρηση λειτουργικότητας και προτείνονται από το [35] είναι:

- αρχικοποίηση του έργου και ο ορισμός του εύρους της υλοποίησης ($i=0$),
- η ολοκλήρωση των απαιτήσεων και του σχεδιασμού ($i=1$),
- η θέση του έργου σε παραγωγική λειτουργία ($i=k$).

Επιπρόσθετα, είναι χρήσιμο, να λαμβάνεται ως επιπλέον ορόσημο κάθε αποδοχή σοβαρής απαίτησης αλλαγής ή πρόσθετης απαίτησης που ανακοινώνεται κατά τη διάρκεια υλοποίησης του έργου μετά την ολοκλήρωση της ανάλυσης και του σχεδιασμού και πριν τη θέση του έργου σε παραγωγική λειτουργία ($1 < i < k$). Ορόσημο μπορεί να αποτελέσει επίσης κάθε αλλαγή που επιβάλλεται στο έργο για την αποκατάσταση σφαλμάτων της ανάλυσης και του σχεδιασμού.

Τα μεγέθη που πρέπει να μετρώνται ή να προεκτιμούνται σε κάθε ορόσημο είναι τα ακόλουθα:

- Λειτουργικό Μέγεθος σε FP (κατά IFPUG FPA 4.3),

- Ανθρωποχρόνος σε ανθρωποώρες και ανθρωπομήνες εργασίας (PHs, PMs),
- Παραγωγικότητα σε ανθρωποώρες ανά λειτουργικό σημείο (hs/FP),
- Διάρκεια υλοποίησης σε ημέρες και μήνες (hs,ms)

Επιπρόσθετα μεγέθη που είναι σημαντικό να αποτυπώνονται είναι η παρέκκλιση του εύρους και τα σφάλματα του έργου, όπως παρουσιάζονται ακολούθως.

6.3.1 Μέτρηση της παρέκκλισης του εύρους του έργου

Πολύ σημαντική είναι η αποτύπωση της παρέκκλισης του εύρους του έργου (Project Scope Creep) ως ο λόγος $FPs(i)-FPs(0)/FPs(0)$, $i=1..k$, όπου $FPs(i)$ η λειτουργικότητα που εκτιμάται ότι θα απαιτηθεί από την υλοποίηση του συνολικού έργου στα ορόσημα υλοποίησης $0,1..k$.

6.3.2 Μέτρηση των σφαλμάτων υλοποίησης

Πολύ σημαντική είναι η μέτρηση σφαλμάτων (defects) για κάθε 1000 Λειτουργικά Σημεία για έργο που υλοποιείται ή λειτουργεί (ds/KFPs). Τα σφάλματα είναι σκόπιμο να ταξινομούνται ως προς τη σημαντικότητα τους (severity) σε κλίμακα από 1 (εξαιρετικά σοβαρό - fatal) έως 4 (σφάλμα ενημέρωσης - info), είτε σε αντιστοιχία με την κλίμακα σημαντικότητας που υπαγορεύεται από το SLA του έργου. Τα σφάλματα σταθμίζονται με συντελεστές ως προς τη σοβαρότητα τους και το σταθμισμένο άθροισμά τους αποτελεί μέτρο ποιότητας του παραγόμενου λογισμικού. Επισημαίνεται πως τα σφάλματα μπορεί να διαπιστώνονται κατά την υλοποίηση του έργου ή και μετά από αυτήν, κατά την παραγωγική λειτουργία και συντήρηση του έργου.

Είναι δυνατό να αφορούν σφάλματα στην ανάλυση απαιτήσεων, το σχεδιασμό, την ανάπτυξη του κώδικα, τους μετασχηματισμούς δεδομένων, τη συντήρηση και τις βελτιώσεις.

6.4 Πρόταση εφαρμογής μεθόδου προμέτρησης και επιμέτρησης έργων στη Γ.Γ.Π.Σ.

Από τη μελέτη των δεδομένων της βάσης ISBSG θεωρούμε ότι είναι πολύ ευνοϊκή η συγκυρία ότι η Γ.Γ.Π.Σ. στη παρούσα φάση εμφανίζει κοινά τεχνολογικά χαρακτηριστικά στην υλοποίηση των έργων λογισμικού. Αυτά είναι η ανάπτυξη σε περιβάλλον Oracle RDBMS, η χρήση γλώσσας J2EE σε n-tier αρχιτεκτονική σε εξοπλισμό που ισοδυναμεί με "mainframe" και αξιοποιώντας συχνά εργαλεία ανάλυσης (UpperCASE tools), τεκμηρίωσης, ελέγχων και εκσφαλμάτωσης (LowerCASE tools). Από την επισκόπηση δεδομένων και δημοσιευμένων συμπερασμάτων [40] [41] [42] σχετικά με τα δεδομένα της βάσης ISBSG μπορούν να απαντηθούν τα ακόλουθα ερωτήματα:

- Ποιό είναι το μέγεθος των έργων που εμφανίζουν καλή αρχική εκτίμηση του εύρους τους σε Λειτουργικά Σημεία;

Σύμφωνα με τα δεδομένα της βάσης ISBSG, τα έργα με μεγέθους έως 250-350 FPs, εμφανίζουν καλή προβλεψιμότητα της εκτίμησης σε Λειτουργικά Σημεία που διαμορφώνεται σε πρώιμο στάδιο της ανάπτυξης του έργου σε σχέση με το τελικά υλοποιούμενο αντικείμενο. Αυτό συμβαίνει γιατί το αντικείμενο είναι καταληπτό, τα όρια του συστήματος αναγνώσιμα από την αρχή ώστε να μην υπάρξουν σημαντικές αλλαγές στην ανάλυση των απαιτήσεων κατά την εκτέλεση του. Λόγω του σύντομου σχετικά χρόνου ολοκλήρωσής τους οι ομάδες υλοποίησης και παρακολούθησης είναι πιθανότατα οι ίδιες και το τεχνολογικό και επιχειρησιακό περιβάλλον του έργου πιθανότατα θα παραμείνει σταθερό μέχρι την παράδοση του έργου.

- Ποιό είναι το μέγεθος των έργων που εμφανίζουν τη μικρότερη διακύμανση στην παραγωγικότητα τους;

Τα έργα μεγέθους μεγαλύτερου των 180 FPs, σύμφωνα με τα δεδομένα της βάσης ISBSG, εμφανίζουν μικρότερη διακύμανση στη παραγωγικότητα για δεδομένη τεχνολογία και είδος εφαρμογής από άλλα μικρότερα έργα.

- Πώς μεταβάλλει την παραγωγικότητα το μέγεθος της ομάδας έργου που εκτελεί το έργο;

Οι μικρές ομάδες (έως 4 άτομα) έχουν τη μεγαλύτερη παραγωγικότητα. Ομάδες άνω των 10 ατόμων εμφανίζουν μειωμένη παραγωγικότητα και ομάδες άνω των 30 ατόμων εξαιρετικά χαμηλότερη.

- Ποιό είναι ένα μέσο εύρος τιμών για την παραγωγικότητα σε έργα που εμφανίζουν κοινά χαρακτηριστικά με τα έργα της Γ.Γ.Π.Σ.

Σύμφωνα με τα δεδομένα της βάσης ISBSG, εφαρμόζοντας τεχνολογία DBMS, σε προγραμματιστικό περιβάλλον Java ή .NET, αξιοποιώντας εργαλεία ανάλυσης (UpperCASE tools), τεκμηρίωσης, ελέγχων και εκσφαλμάτωσης (LowerCASE tools) για έργα MIS που είναι μεγαλύτερα των 180 FPs, η παραγωγικότητα είναι μεταξύ 8 και 13 hs/FP. Βεβαιώνοντας πως για τα συγκεκριμένα έργα ίσχυαν είτε ισχύουν οι ανωτέρω προϋποθέσεις, μπορεί να ληφθεί κοινή τιμή για την αρχικοποίηση της παραγωγικότητας οι 10 ή 11 hs/FP.

Από τα παραπάνω προκύπτει ότι η Γ.Γ.Π.Σ. μπορεί να ξεκινήσει άμεσα την καταγραφή και μέτρηση της λειτουργικότητας έργων ανάπτυξης λογισμικού λειτουργικού μεγέθους 180-360 FPs. Τέτοια έργα είναι εκείνα που ενδεικτικά μπορούν να υλοποιηθούν από μικρή ομάδα υλοποίησης (3-4 στελεχών), πλήρως απασχολούμενης, σε χρονική προθεσμία 6-12 ημερολογιακών μηνών από την αρχική ανάλυση των απαιτήσεων έως και την παράδοση του έργου σε παραγωγική λειτουργία με την απαραίτητη τεκμηρίωση. Η Γ.Γ.Π.Σ. έχει την εμπειρία και μπορεί να αναγνωρίσει τέτοια έργα στο πλαίσιο εσωτερικών υλοποιήσεων που έχει εκτελέσει ή εκτελεί το προσωπικό της, είτε στο πλαίσιο συμβάσεων ενδεχομένως μεγαλύτερων έργων, που ολοκλήρωσαν είτε εκτελούν εξωτερικοί ανάδοχοι.

Αξιοποιώντας το δεδομένο ότι το περιβάλλον της Γ.Γ.Π.Σ. είναι τεχνολογικά ομοιογενές ως προς τα χαρακτηριστικά που μας αφορούν, ο στόχος διαμορφώνεται στο να δημιουργηθούν, δεδομένα για μετρήσεις Λειτουργικών Σημείων σε τουλάχιστον 6 έργα με τα χαρακτηριστικά που περιγράφηκαν παραπάνω και τα οποία έχουν ήδη ολοκληρωθεί είτε εκπονούνται. Εφόσον υπάρχουν στοιχεία για το κόστος των έργων και το μέσο κόστος ανθρωπομήνα που αφορά τις συγκεκριμένες υλοποιήσεις από αναδόχους ή δεδομένα χρονοαπασχόλησης της εσωτερικής ομάδας ανάπτυξης, τα δεδομένα αυτά θα αξιοποιηθούν. Σε περίπτωση που δεν είναι διαθέσιμα ή ακριβή μπορούν να ληφθούν οι 11 hs/FP της αναμενόμενης παραγωγικότητας για έργο μεγέθους 180-360FP και μικρή ομάδα υλοποίησης.

Για έργα τα οποία βρίσκονται σε φάση ανάπτυξης οι εκτιμήσεις - προμέτρησεις και οι πραγματοποιημένες μετρήσεις - επιμετρήσεις, θα πρέπει να πραγματοποιούνται σύμφωνα με τα ορόσημα που περιγράφηκαν σε προηγούμενη παράγραφο.

Με τον τρόπο αυτό θα αρχικοποιηθεί, με δεδομένα που κατά το δυνατόν θα συγκλίνουν, ένα σύνολο ικανό να εφαρμοστούν συναρτήσεις παλινδρόμησης για την εξαγωγή εκτιμήσεων σε νέα έργα με διαφοροποιημένα χαρακτηριστικά όπως μεγαλύτερο πλήθος Λειτουργικών Σημείων ή/και μεγαλύτερες ομάδες έργου. Έργα που είναι σημαντικά μεγαλύτερα (πολλαπλάσια) από το μέγεθος των 350FP μπορούν

να αντιμετωπίζονται με χρήση ανάλυσης δομής έργου (Work Breakdown Structure) εφαρμόζοντας κατά περίπτωση τις κατάλληλες παραδοχές.

Έτσι, χωρίς να απαιτούνται συγκεχυμένη λογική και δένδρα αποφάσεων όπως προτείνουν οι σχετικές δημοσιεύσεις από το χώρο της τεχνητής νοημοσύνης, θα είναι εφικτό, χρησιμοποιώντας απλά εργαλεία παλινδρόμησης να δημιουργηθούν καλές εκτιμήσεις σχετικά με:

- την παραγωγικότητα μετρούμενη σε hs/FP ,
- παρέκλιση στο εύρος του έργου σε FP και
- σφάλματα υλοποίησης df/KFP .

6.5 Εναλλακτική προσέγγιση

Εναλλακτικά η Γ.Γ.Π.Σ. μπορεί να ακολουθήσει τη μεθοδολογία που προτείνεται από το ISBSG [37] σύμφωνα και με την παρ. 6.2 (Χρήση των μεθόδων του ISBSG).

7 Εφαρμόζοντας τη μεθοδολογία IFPUG FPA 4.3 στη Γ.Γ.Π.Σ.

7.1 Εσωτερική δομή στη Γ.Γ.Π.Σ. για την εφαρμογή της μεθοδολογίας

Η αποτελεσματική εκτίμηση μεγέθους έργων λογισμικού, αποτελεί μια συνεχή δραστηριότητα στο εσωτερικό ενός Φορέα και θα πρέπει να εκτελείται με τρόπο μεθοδικό και οργανωμένο. Ειδικά σε ένα Φορέα του μεγέθους της Γ.Γ.Π.Σ. όπου κατά κανόνα εξελίσσονται περισσότερα από ένα έργα ανάπτυξης και βελτίωσης λογισμικού (είτε «εσωτερικά» είτε μέσω συμβάσεων με Αναδόχους), οι εργασίες για την προεκτίμηση και την αποτίμηση του Λειτουργικού Μεγέθους των έργων αποτελούν κρίσιμη και απαιτητική δραστηριότητα.

Για το σκοπό αυτό προτείνουμε την δημιουργία μιας εσωτερικής δομής στη Γ.Γ.Π.Σ. με αποκλειστικό αντικείμενο την εκτίμηση του λειτουργικού μεγέθους (και του κόστους) έργων λογισμικού. Για τις ανάγκες αναφοράς στο πλαίσιο του παρόντος, η εν λόγω δομή θα ονομάζεται «Ομάδα Εκτίμησης Μεγέθους Έργων Λογισμικού» (Ο.Ε.Μ.Ε.Λ.).

7.1.1 Αρχές λειτουργίας

Η λειτουργία της Ο.Ε.Μ.Ε.Λ. θα πρέπει να στηρίζεται στις προβλέψεις και τις απαιτήσεις που τίθενται από το διεθνές πρότυπο ISO/IEC 15939, το οποίο προσδιορίζει τις βασικές αρχές για την εκτέλεση μέτρησης λογισμικού. Πρακτικά το πρότυπο αυτό προσφέρει τις κατευθυντήριες γραμμές για την τυποποίηση της εκτελούμενης διαδικασίας μέτρησης, χωρίς να επιβάλλει την εφαρμογή συγκεκριμένης μεθοδολογίας μέτρησης.

Στο πλαίσιο του ISO/IEC 15939 καταγράφονται τέσσερις συγκεκριμένες «δράσεις» (καθεμία εκ των οποίων επιμερίζεται σε ειδικότερες ενέργειες) για την ολοκληρωμένη διαχείριση και τη διενέργεια της διαδικασίας μέτρησης. Με δεδομένο ότι το πρότυπο περιγράφει τη διαδικασία συνολικά, απαιτείται μια προσαρμογή στο πνεύμα των αναγκών λειτουργίας της Ο.Ε.Μ.Ε.Λ. ως εξής:

Δράση Α – Σχεδιασμός της διαδικασίας μέτρησης, περιλαμβάνοντας τις εξής ενέργειες

1. Προσδιορισμός διαδικασιών συλλογής, ανάλυσης και καταγραφής δεδομένων
 - a. Ορισμός των διαδικασιών για τη συλλογή δεδομένων (συμπεριλαμβανομένου του τρόπου αποθήκευσής και πιστοποίησής τους).
 - b. Ορισμός των διαδικασιών για την επεξεργασία των δεδομένων και την αναφορά των αποτελεσμάτων.
 - c. Ορισμός των διαδικασιών διαχείρισης παραμετροποίησης (Configuration Management)
2. Ορισμός κριτηρίων για την αποτίμηση του Λειτουργικού Μεγέθους και της διαδικασίας μέτρησης
 - a. Ορισμός κριτηρίων για την αποτίμηση του μετρούμενου Λειτουργικού Μεγέθους
 - b. Ορισμός κριτηρίων για την αποτίμηση της διαδικασίας μέτρησης
3. Έλεγχος, έγκριση του σχεδιασμού και διάθεση πόρων για εργασίες μέτρησης
 - a. Αναθεώρηση και έγκριση του σχεδιασμού της μέτρησης
 - b. Διάθεση πόρων για διενέργεια της σχεδιασθείσας μέτρησης
4. Λήψη και εγκατάσταση υποστηρικτικών τεχνολογιών
 - a. Αξιολόγηση των διαθέσιμων υποστηρικτικών τεχνολογιών και επιλογή των κατάλληλων

b. Λήψη και θέση σε λειτουργία των επιλεγισών τεχνολογιών υποστήριξης

Δράση Β – Εκτέλεση της διαδικασίας μέτρησης, περιλαμβάνοντας τις εξής ενέργειες

1. Ενοποίηση των διαδικασιών
 - a. Ενσωμάτωση στη διαδικασία των απαιτούμενων ενεργειών για την παραγωγή και τη συλλογή πρωτογενών δεδομένων.
 - b. Γνωστοποίηση των διαδικασιών συλλογής δεδομένων στις δομές του Φορέα που είναι αρμόδιες για την «παραγωγή» τους.
 - c. Ενσωμάτωση στη διαδικασία των απαιτούμενων ενεργειών για ανάλυση των δεδομένων και δημιουργία αναφορών.
2. Συλλογή των δεδομένων
 - a. Συλλογή των απαραίτητων δεδομένων για τη μέτρηση
 - b. Αποθήκευση των δεδομένων και των πληροφοριών που απαιτούνται για την επαλήθευση, κατανόηση και αποτίμησή τους
 - c. Επαλήθευση των συλλεγέντων δεδομένων
3. Ανάλυση δεδομένων και υπολογισμός Λειτουργικού Μεγέθους
 - a. Ανάλυση των δεδομένων και υπολογισμός Λειτουργικού Μεγέθους
 - b. Προσέγγιση/ερμηνεία των αποτελεσμάτων
 - c. Έλεγχος και επαλήθευση του υπολογιζόμενου Λειτουργικού Μεγέθους
4. Γνωστοποίηση αποτελεσμάτων
 - a. Τεκμηρίωση του υπολογισθέντος Λειτουργικού Μεγέθους
 - b. Γνωστοποίηση των αποτελεσμάτων της μέτρησης στους ενδιαφερόμενους.

Δράση Γ – Αξιολόγηση της μέτρησης, περιλαμβάνοντας τις εξής ενέργειες

1. Αξιολόγηση των αποτελεσμάτων και της διαδικασίας μέτρησης
 - a. Αποτίμηση της διαδικασίας μέτρησης με συγκεκριμένα κριτήρια και λήψη συμπερασμάτων για τα υπέρ και τα κατά της.
 - b. Καταγραφή της εμπειρίας που προέκυψε από την αξιολόγηση.
2. Αναγνώριση πιθανών βελτιώσεων της μέτρησης
 - a. Αναγνώριση πιθανών βελτιώσεων στη διαδικασία μέτρησης
 - b. Γνωστοποίηση των καταγεγραμμένων πιθανών βελτιώσεων

Χάριν πληρότητας σημειώνουμε ότι η Δράση «Εδραίωση και διατήρηση της δέσμευσης για μέτρηση» που προβλέπεται από το ISO 15939 **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** απαλείφθηκε από την παραπάνω καταγραφή καθώς δεν εφαρμόζεται στην εν λόγω περίπτωση. Επίσης από τη Δράση «Σχεδιασμός της διαδικασίας μέτρησης» που προβλέπεται από το πρότυπο έχουν απαλειφθεί οι ενέργειες «Προσδιορισμός χαρακτηριστικών της οργανωτικής μονάδας», «Αναγνώριση των αναγκών ενημέρωσης» και «Επιλογή μεθοδολογίας μέτρησης» καθώς υπερκαλύπτονται από την επιλογή για συμμόρφωση με το πρότυπο ISO/IEC 14143 και το ειδικότερο ISO/IEC 20926. Τέλος για τον ίδιο λόγο κατάλληλη προσαρμογή έγινε και σε ενέργειες της Δράσης «Αξιολόγηση της μέτρησης», καθώς μετά την απαίτηση συμμόρφωσης με το ISO/IEC 20926 δεν εφαρμόζονται στην εν λόγω περίπτωση.

7.1.2 Απαιτούμενοι πόροι

Για την οργάνωση και την αποδοτική λειτουργία της Ο.Ε.Μ.Ε.Λ. κρίνεται απαραίτητη η διασφάλιση από το Φορέα των ακόλουθων πόρων:

- 1. Προσωπικό για την ανάληψη των εξής διακριτών ρόλων :**
 - a. Μετρητής Λειτουργικών Σημείων: Τα στελέχη που θα επιφορτιστούν με την εκτέλεση των μετρήσεων και τα οποία θα πρέπει να έχουν λάβει τυπική εκπαίδευση στη μεθοδολογία που περιγράφεται στο ISO/IEC 20926.
 - b. Συντονιστής: Το στέλεχος που θα αναλάβει να οργανώνει και να καθοδηγεί την Ομάδα Μέτρησης.
 - c. Μέντορας: Τα στελέχη που θα αναλαμβάνουν την υποστήριξη των νεοεισερχόμενων μετρητών στην ομάδα.
 - d. Έμπειρος Μετρητής: Το στέλεχος που θα αναλάβει τον έλεγχο των μετρήσεων και την επιβεβαίωση των συμπερασμάτων.
 - e. Επιχειρησιακός Χρήστης: Το στέλεχος του Φορέα που θα έχει εκπαιδευτεί και θα κατανοεί τις έννοιες και τις διαδικασίες εκτέλεσης της μεθοδολογίας μέτρησης.
- 2. Υλικό Αναφοράς**
 - a. Το ISO/IEC 20926
 - b. Το Counting Practices Manual του IFPUG
 - c. Ηλεκτρονικές πηγές και το υλικό στο διαδικτυακό τόπο του IFPUG
 - d. Βιβλία και λοιπές εκδόσεις σχετικές με τη μεθοδολογία μέτρησης και τις μετρικές
- 3. Διαδικασίες**
 - a. Προσαρμοσμένες διαδικασίες μέτρησης, με βάση τις αρχές και τους κανόνες του Φορέα για την τεκμηρίωση των μετρήσεων, τα κριτήρια επιλογής, τους ρόλους και τις αρμοδιότητες για την καταγραφή, αποθήκευση και παρουσίαση των αποτελεσμάτων.
 - b. Προσαρμοσμένο εγχειρίδιο αντιμετώπισης προβλημάτων, που καταγράφει τον τρόπο χειρισμού σε ζητήματα που άπτονται της διαδικασίας μέτρησης εντός του συγκεκριμένου Φορέα
 - c. Μηχανισμός αναθεώρησης της διαδικασίας μέτρησης σε σχέση με τις προβλεπόμενες ενέργειες κατά τη μέτρηση, καθώς και υποδείγματα για την αναφορά θεμάτων που προκύπτουν μετά την ολοκλήρωσή της.
- 4. Εργαλεία**
 - a. Λογισμικό που προσφέρει μηχανισμό για την καταγραφή, τεκμηρίωση και ενημέρωση των εκτελούμενων μετρήσεων λειτουργικού μεγέθους.
 - b. Λογισμικό ή μεθοδολογία για την διαχείριση των παραμέτρων διενέργειας των μετρήσεων.
- 5. Εκπαίδευση**
 - a. Τυποποιημένο πρόγραμμα εκπαίδευσης των μετρητών, μέσω του οποίου θα εξασφαλιστεί η εμπέδωση των βασικών εννοιών της μέτρησης Λειτουργικών Σημείων για όλους τους τύπους εφαρμογών και όλους τους τύπους μέτρησης. Το εκπαιδευτικό υλικό θα πρέπει να περιλαμβάνει πρακτικά παραδείγματα και case studies, πάνω στα οποία οι εκπαιδευόμενοι θα μπορούν να εφαρμόζουν τις θεωρητικές τους γνώσεις (ελάχιστη προτεινόμενη διάρκεια: 2 ημερών).
 - b. Τακτικές εκπαιδεύσεις σε προχωρημένα θέματα, ώστε να διασφαλίζεται ότι οι μετρητές έχουν ενημερωθεί για τις τελευταίες σχετικές εξελίξεις (λ.χ. νέα περιβάλλοντα) καθώς και για τον χειρισμό δύσκολων σεναρίων μέτρησης.

6. Δικτύωση

- a. Πρόσβαση σε υπηρεσίες εξωτερικών ειδικών και συμβούλων για θέματα μετρικών, για τη λήψη ad hoc συμβουλών.
- b. Συμμετοχή σε τυχόν τοπικές ενώσεις/ομάδες με θέμα τη μέτρηση Λειτουργικών Σημείων και τον οργανισμό IFPUG.
- c. Συμμετοχή σε σεμινάρια, συνέδρια και workshop, με αντικείμενο τη μέτρηση λειτουργικού μεγέθους λογισμικού
- d. Δικτύωση και ανταλλαγή τεχνογνωσίας με άλλους φορείς και οργανισμούς που υλοποιούν αντίστοιχες μετρητικές διαδικασίες.

7.1.3 Προγραμματισμός και οργάνωση μετρήσεων

Η Ο.Ε.Μ.Ε.Λ. θα έχει την αποκλειστική ευθύνη για την κατάρτιση πλάνου χρονοπρογραμματισμού των μετρήσεων μεγέθους λογισμικού.

Παράλληλα η Ο.Ε.Μ.Ε.Λ. με βάση την επιστημονική και επιχειρησιακή της γνώση θα προσδιορίσει δύο κομβικές παραμέτρους της διαδικασίας: τον Σκοπό μέτρησης του Λειτουργικού Μεγέθους καθώς τους συγκεκριμένους τρόπους μέσω των οποίων θα αξιοποιηθούν από τον Φορέα τα αποτελέσματα των διενεργούμενων μετρήσεων.

Πρακτικά η Ο.Ε.Μ.Ε.Λ. θα λαμβάνει αποφάσεις για:

- Την στρατηγική για τη διαδικασία μέτρησης (λ.χ. την επιθυμητή ακρίβεια των αποτελεσμάτων και το βαθμό λεπτομέρειας στην τεκμηρίωση της μέτρησης).
- Την εμπέλεια της διαδικασίας μέτρησης, συγκεκριμενοποιώντας ποιες εφαρμογές και ποια έργα θα μετρηθούν.
- Την τοποθέτηση των ορίων της μέτρησης για τις επιμέρους εφαρμογές (λ.χ. διαχωρίζοντας ή/και συνδυάζοντας εφαρμογές με βάση τις ανάγκες του Φορέα και το Σκοπό της μέτρησης).

Η διεθνής πρακτική έχει δείξει ότι μεγάλοι Φορείς και Επιχειρήσεις επιλέγουν να μετρούν μέρος των έργων λογισμικού που υλοποιούν ή/και συντηρούν, με βάση μια προκαθορισμένη λίστα κριτηρίων, όπως για παράδειγμα:

- Έργα με υψηλό ρίσκο αποτυχίας
- Έργα που υπερβαίνουν ένα προαποφασισμένο προϋπολογισμό
- Έργα που επιλέγονται ως αντιπροσωπευτικά στο σκέλος της παραγωγικότητας
- Έργα που αφορούν ένα συγκεκριμένο περιβάλλον ανάπτυξης
- Έργα με Λειτουργικό Μέγεθος που υπερβαίνει τα 50 Λειτουργικά Σημεία.

Ομοίως παρόμοια κριτήρια επιλογής χρησιμοποιούνται για τον προσδιορισμό εκείνων των εφαρμογών που ενδιαφέρουν ένα Φορέα ή μια Επιχείρηση να μετρήσει όπως για παράδειγμα:

- Εφαρμογές που υλοποιούνται μια συγκεκριμένη πλατφόρμα ανάπτυξης
- Εφαρμογές που αναπτύσσονται εξ' ολοκλήρου εσωτερικά στον Φορέα και δεν βασίζονται σε λογισμικό τρίτων
- Εφαρμογές που προβλέπεται να παραμείνουν σε παραγωγική λειτουργία για τουλάχιστον 12 μήνες.
- Εφαρμογές που είναι παραγωγικές και απαιτούν συχνές και σημαντικές βελτιώσεις.

Εκτός από την επιλογή των έργων λογισμικού και των εφαρμογών προς μέτρηση, η Ο.Ε.Μ.Ε.Λ. θα αποφασίζει σε ποια στάδια του κύκλου ζωής κάθε μετρούμενου έργου λογισμικού θα εκτελούνται μετρήσεις λειτουργικού μεγέθους, λαμβάνοντας υπόψη το σκοπό της μέτρησης. Εάν για παράδειγμα σκοπός είναι η διαμόρφωση εκτιμήσεων για την πορεία και η παρακολούθηση της εκτέλεσης ενός έργου, τότε θα απαιτηθεί η διενέργεια αναλυτικών και καλώς τεκμηριωμένων μετρήσεων Λειτουργικού Μεγέθους σε προκαθορισμένα ορόσημα του έργου (λ.χ. φάση ανάλυσης απαιτήσεων, ολοκλήρωση σχεδιασμού, φάση υλοποίησης).

Επιπρόσθετα στοιχεία μετρήσεων μπορούν να συλλέγονται και να συνοψίζονται από την Ο.Ε.Μ.Ε.Λ. για την εξαγωγή συμπερασμάτων όπως στοιχεία που αφορούν στην ποιότητα του παραγόμενου λογισμικού εκφρασμένα σε σφάλματα (Defects) ανά 1.000 Λειτουργικά Σημεία.

Τέλος, εξαιρετικά σημαντική είναι η συλλογή στοιχείων από τις δηλώσεις χρονοαπασχόλησης του προσωπικού των αναδόχων ή των ομάδων εσωτερικής ανάπτυξης, ώστε αυτές να συνοψίζονται ανά έργο από την Ο.Ε.Μ.Ε.Λ. με σκοπό να εξάγονται συμπεράσματα για την παραγωγικότητα στα έργα, εκφρασμένη σε ανθρωπώρες εργασίας ανά Λειτουργικό Σημείο.

7.2 Οφέλη από τη λειτουργία της Ο.Ε.Μ.Ε.Λ.

Ουσιαστικό όφελος για το κάθε έργο από τη λειτουργία της Ο.Ε.Μ.Ε.Λ. θα είναι η εποπτεία από κλιμάκιο που οριζόντια θα βλέπει όλα τα έργα της Γ.Γ.Π.Σ. διαμορφώνοντας αντικειμενική και συγκρίσιμη εικόνα μεταξύ των έργων, των αναδόχων, και των κλιμακίων παρακολούθησης και παραλαβής.

Σε τουλάχιστον τρία κρίσιμα σημεία - ορόσημα του σχεδιασμού και της υλοποίησης των έργων αλλά και σε κάθε απαίτηση αλλαγής που προκύπτει κατά την υλοποίηση θα έχει άποψη η Ο.Ε.Μ.Ε.Λ.:

- Στην αρχική σύλληψη επιβεβαιώνοντας παράλληλα την ρεαλιστικότητα του προϋπολογισμού, της αναμενόμενης προσπάθειας και του χρόνου εκτέλεσης και εκλογικεύοντας τις αισιόδοξες εκτιμήσεις και τον ενθουσιασμό των εισηγητών του έργου.
- Στην ολοκλήρωση της ανάλυσης απαιτήσεων και του τεχνικού σχεδιασμού, πρακτικά της «Μελέτης Εφαρμογής» όπως συνηθίζεται να λέγεται όπου παράλληλα με τη μέτρηση βεβαιώνεται η συνέπεια είτε διαπιστώνεται η απόκλιση από τον αρχικά διατυπωμένη πρόθεση υλοποίησης.
- Κάθε απαίτηση αλλαγής, πρόσθετη απαίτηση είτε αλλαγή λόγω λάθους στο σχεδιασμό, μετά τη μεσολάβηση της Ο.Ε.Μ.Ε.Λ., αποκτά αντικειμενικότερη θεώρηση σε σχέση με την επιτροπή παρακολούθησης και την ομάδα έργου του αναδόχου.

Η εφαρμογή της μεθόδου και η λειτουργία της Ο.Ε.Μ.Ε.Λ. θα έχει ένα ακόμα παράπλευρο όφελος για τη Γ.Γ.Π.Σ. Θα ενισχύσει την ανάγκη για καλή διατύπωση των εγγράφων ανάλυσης, σχεδιασμού και τεκμηρίωσης του έργου. Χωρίς να είναι απαραίτητη η υιοθέτηση συγκεκριμένης μεθοδολογίας ή προϊόντων για την υποστήριξη των σχετικών εργασιών, θα πρέπει να αναμένεται σημαντική βελτίωση στην ποιότητα των εγγράφων που τεκμηριώνουν και περιγράφουν το σύστημα σε όλες τις φάσεις υλοποίησης του.

8 Βιβλιογραφία & Αναφορές

- [1] OASIS. "Reference Model for Service Oriented Architecture", Committee Specification 1, August 2006. [Online: <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf> - Πρόσβαση: Μάρτιος 2012]
- [2] Wikipedia contributors, "Service-oriented architecture", Wikipedia, The Free Encyclopedia, [Online: http://en.wikipedia.org/wiki/Service-oriented_architecture - Πρόσβαση: Μάρτιος 2012]
- [3] T. Erl. "Service-Oriented Architecture – Concepts, Technology, and Design", Prentice Hall/PearsonPTR, 2006
- [4] B. Lublinsky. "SOA Design: Meeting in the Middle", FTPOnline, August 20, 2004
- [5] Haas, H., "Designing the architecture for Web services", W3C Talks, 2003, <http://www.w3.org>
- [6] Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, Ray Madachy, Richard Selby. "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0", 1995
- [7] Computer Science Department, University of Southern California. "COCOMO II Model Definition Manual", 1997.
- [8] F. J. Heemstra. "Software cost estimation". Information and Software Technology, vol. 34, 1992, pages. 627-639, 1992
- [9] L. De Rore, M. Snoeck, G. Poels, G. Dedene. "Cocomo II as productivity measurement: a case study at KB", K.U.Leuven - Faculty of Business and Economics, 2008.
- [10] N. Merlo – Schett. "COCOMO (Constructive Cost Model)", Seminar on Software Cost Estimation WS 2002 / 2003, 2002.
- [11] Z. Stojanovic, A. Dahanayake. "Service-Oriented Software System Engineering: Challenges and Practices", Idea Group Publishing, London, 2005.
- [12] G. Lewis, E. Morris, L. O'Brien, D. Smith, L. Wrage. "SMART: The Service-Oriented Migration and Reuse Technique". CMU/SEI-2005-TN-029, Software Engineering Institute, 2005.
- [13] L. Zheng, J. Keung. "Software Cost Estimation Framework for Service-Oriented Architecture Systems Using Divide-and-Conquer Approach". Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), 2010.
- [14] L. O'Brien. "A Framework for Scope, Cost and Effort Estimation for Service Oriented Architecture (SOA) Projects", Australian Software Engineering Conference, 2009
- [15] ISO/IEC. "14143:2007 'Information technology – Software Measurement – Functional Size Measurement'", 2007
- [16] A. J. Albrecht. "Measuring Application Development Productivity," Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, October 14–17, IBM Corporation (1979), pages 83–92

- [17] ISO/IEC. “20926:2009 ‘Software and systems engineering - Software measurement — IFPUG functional size measurement method 2009’”, 2009
- [18] ISO/IEC. “19761: 2011 ‘Software engineering -- COSMIC: a functional size measurement method’”, 2011
- [19] ISO/IEC. “20968: 2002 ‘Software engineering -- Mk II Function Point Analysis -- Counting Practices Manual’”, 2002
- [20] ISO/IEC. “24570: 2005 ‘Software engineering -- NESMA functional size measurement method version 2.1 -- Definitions and counting guidelines for the application of Function Point Analysis’”, 2005
- [21] E.J.D. Candido, R. Sanches. “Estimating the size of Web applications by using a simplified function point method”, 2004
- [22] ISO/IEC. “29881: 2010 ‘Software Information technology -- Systems and software engineering -- FiSMA 1.1 functional size measurement method’”, 2010
- [23] IFPUG, “Function Points Counting Practices Manual”, Release 4.3.1, 2010
- [24] H. Diab, M. Frappier, R. Denis. “Formalizing cosmic-ffp using room”. ACS/IEEE International Conference on Computer Systems and Applications, June 2001, pages 312 – 318
- [25] IFPUG. “Using Function Points to Measure Reusable Software”, Release 1.0, 2010.
- [26] IFPUG. “Sizing Component-Based Development Using Function Points”, 2009
- [27] C. Symons. “Conversion between ifpug 4.0 and mkii function points”. Software Measurement Services, 1999.
- [28] S. Tyagi. “RESTful Web Services”. Technical article, Sun Microsystems, 2006. [Online: <http://www.oracle.com/technetwork/articles/javase/index-137171.html> - Πρόσβαση: Μάρτιος 2012]
- [29] IFPUG. “Function Points & Counting Middleware Software Applications” Release 1.0, 2009
- [30] OASIS. “Web Service Security Specification”, February 2006. [Online: <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf> - Πρόσβαση: Μάρτιος 2012]
- [31] OASIS. “Web Service Message Reliability Specification”, November 2004. [Online: http://docs.oasis-open.org/wsrn/ws-reliability/v1.1/wsrn-ws_reliability-1.1-spec-os.pdf - Πρόσβαση: Μάρτιος 2012]
- [32] W3C. “Web Service Policy Specification”, April 2006. [Online: <http://www.w3.org/Submission/WS-Policy> - Πρόσβαση: Μάρτιος 2012]
- [33] ORACLE. “Programming Advanced Features of WebLogic Web Services Using JAX-RPC 10g Release 3 (10.3)” - Advanced JWS Programming: Implementing Asynchronous Features, August 2008, pages 6-1 – 6-61 [Online: http://docs.oracle.com/cd/E15051_01/wls/docs103/pdf/webserv_adv_rpc.pdf - Πρόσβαση: Μάρτιος 2012]

- [34] ORACLE. “Programming Advanced Features of WebLogic Web Services Using JAX-RPC 10g Release 3 (10.3)” - Advanced JWS Programming: JMS Transport and SOAP Message Handlers, August 2008, pages 7-1 – 7-9 [Online: http://docs.oracle.com/cd/E15051_01/wls/docs103/pdf/webserv_adv_rpc.pdf - Πρόσβαση: Μάρτιος 2012]
- [35] V. Sharma, H. K. Verma. “Optimized Fuzzy Logic Based Framework for Effort Estimation in Software Development”. Computer Science Issues, Vol. 7, Issue 2, No. 2, 2010, pages 30-38.
- [36] S. Elyassami, A. Idri. “Investigating effort prediction of software projects on the ISBSG dataset”. International Journal of Artificial Intelligence & Applications (IJAI), Vol.3, No.2, March 2012
- [37] <http://www.isbsg.org>
- [38] ISO/IEC. “15939:2007 ‘Systems and software engineering -- Measurement process’”, 2007
- [39] P. R. Hill. “Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration”, McGraw-Hill Professional, 2011.
- [40] P. Morris. “Resources For Implementing A Metrics Program”, © Total Metrics Pty. Ltd, April 2001.
- [41] P. Morris. “Case Study of a Successful Measurement Program”, Version 1.2, © Total Metrics Pty. Ltd, September 2007.
- [42] ISBSG. “Early Lifecycle Software Estimation”, 2011

Π2: Εγχειρίδιο εφαρμογής μεθοδολογίας προμέτρησης και επιμέτρησης έργων SOA

1 Εργαλείο Υπολογισμού Λειτουργικού Μεγέθους Λογισμικού

Για την διευκόλυνση της εφαρμογής της μεθοδολογίας IFPUG FPA 4.3, δημιουργήθηκε εργαλείο που υλοποιεί το βασικό μαθηματικό υπολογισμό της μέτρησης λειτουργικού μεγέθους για μια υπομονάδα λογισμικού.

Το εν λόγω εργαλείο υλοποιείται μέσω ενός προτύπου (template) για την εφαρμογή Microsoft Excel, και με διαδικασία οδηγού (wizard) καθοδηγεί τον χρήστη στα βήματα που χρειάζονται έτσι ώστε να γίνει ο υπολογισμός του Λειτουργικού Μεγέθους.

Χρησιμοποιώντας το πρότυπο αρχείο, ο χρήστης μπορεί να δημιουργήσει ένα νέο βιβλίο εργασίας, ενσωματώνοντας συγκεκριμένα φύλλα εργασίας με ήδη μορφοποιημένους πίνακες, μαθηματικούς τύπους και μακροεντολές Visual Basic for Applications για την διενέργεια της μέτρησης.

2 Οδηγίες Χρήσης Εργαλείου Υπολογισμού

2.1 Εισαγωγή

Το εργαλείο υπολογισμού Λειτουργικού Μεγέθους «FSM Template ΓΓΠΣ.v2.xlt» παράγει βιβλία εργασίας του Microsoft Excel με προδιαγεγραμμένη μορφοποίηση για την εκτέλεση της μέτρησης. Κάθε νέο βιβλίο εργασίας (εφεξής FSMT), όπως και το πρότυπο βιβλίο αποτελείται από επτά (7) φύλλα εργασίας (sheets) όπως περιγράφονται παρακάτω:

- {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ}
- {FPs}
- {ILFs}
- {EIFs}
- {EIs}
- {EOs}
- {EQs}

Τα κελιά με λευκό χρώμα χρησιμοποιούνται για την εισαγωγή δεδομένων ενώ όλα τα άλλα κελιά έχουν γκριζό χρώμα.

Το FSMT επεξεργάζεται αυτόματα τα δεδομένα που εισάγει ο χρήστης και διαγνωστικές πληροφορίες παρουσιάζονται στο κάτω μέρος του κάθε φύλλου εργασίας με κόκκινο χρώμα.

2.2 Καταχώρηση Ταυτότητας Έργου

Το πρώτο φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ} του FSMT (Εικόνα 1) αποτελεί το σημείο έναρξης της διαδικασίας υπολογισμού του Λειτουργικού Μεγέθους για την υπό εξέταση υπομονάδα λογισμικού.

Στο φύλλο αυτό γίνεται η καταγραφή των στοιχείων για την μετρούμενη υπομονάδα λογισμικού (module). Συγκεκριμένα στον πίνακα «Ταυτότητα Μέτρησης Έργου» ο χρήστης εισαγάγει τα βασικά στοιχεία του έργου:

- Κωδικός (id) Έργου
- Ονομασία Έργου
- Φάση του Έργου
- Όνομα Υπευθύνου Έργου
- Κωδικός (id) Μέτρησης
- Περιγραφή Εύρους (Scope) Μέτρησης
- Όνομα Υπεύθυνου Μέτρησης
- Ημερομηνία Μέτρησης

Ειδικά για το πεδίο «Φάση του Έργου», σημειώνουμε ότι μπορεί να λάβει τις παρακάτω τιμές:

- 0 εάν πρόκειται για αρχική εκτίμηση στη φάση αρχικοποίησης του έργου - ΠΡΟΕΚΤΙΜΗΣΗ
- 1 εάν έχει ολοκληρωθεί η σύνταξη της μελέτης εφαρμογής για το έργο
- k (όπου k ακέραιος σύμφωνα και με την παρ. 6.3.1) εάν το έργο βρίσκεται σε παραγωγική λειτουργία

Στο ίδιο φύλλο εργασίας γίνεται και η σύνοψη των εκτιμήσεων μετά την ολοκλήρωση της μέτρησης. Συγκεκριμένα από το φύλλο εργασίας {FPs} μεταφέρεται αυτόματα το πλήθος των υπολογισθέντων Λειτουργικών Σημείων.

Στον πίνακα «**Σύνοψη Εκτιμήσεων**» ο χρήστης καλείται να δώσει δύο παραμέτρους για τον υπολογισμό της ανθρωποπροσπάθειας. Συγκεκριμένα ο χρήστης εισαγάγει:

- Productivity (PR) in hs/FP: Συμπληρώνεται η εκτιμώμενη ανθρωποπροσπάθεια (σε ανθρωποώρες) ανά λειτουργικό σημείο (Έχει προσυμπληρωθεί η τιμή 11 με βάση την προσέγγιση στην παρ. 6.4)
- Hours per Person-Month: Συμπληρώνονται οι εργάσιμες ώρες ανά ανθρωπομήνα (Έχει προσυμπληρωθεί η συνήθης τιμή 160).

Με βάση τις δύο παραπάνω παραμέτρους υπολογίζεται αυτομάτως η εκτίμηση της ανθρωποπροσπάθειας για το έργο σε ανθρωπομήνες στο πεδίο Person-Months (PM).

2.3 Εκτέλεση Μέτρησης Λειτουργικών Σημείων

Μετά τη συμπλήρωση των στοιχείων ταυτότητας της μέτρησης, ο χρήστης καλείται να συμπληρώσει τα φύλλα εργασίας που αναφέρονται στη μέτρηση Λειτουργικών Σημείων για τις Λειτουργικές Διαδικασίες Δεδομένων και τις Λειτουργικές Διαδικασίες Συναλλαγών.

2.3.1 Μέτρηση ILFs

Στο φύλλο εργασίας {ILFs} (Εικόνα 2) βρίσκεται πίνακας όπου ο χρήστης πρέπει να καταγράψει τα στοιχεία των **Εσωτερικών Λογικών Αρχείων (Internal Logical Files)** στην προς μέτρηση υπομονάδα λογισμικού (module).

Στη στήλη **Internal Logical Files** ο χρήστης ονοματίζει όλα τα ILFs που έχουν αναγνωρισθεί, ενώ στις στήλες με όνομα **# of DETs** (Data Element Types) και **# of RETs** (Record Element Types) ο χρήστης συμπληρώνει τα αντίστοιχα δεδομένα.

Οι στήλες με την ένδειξη **Complexity** συμπληρώνονται αυτόματα από το FSMT ανάλογα με τα αριθμητικά δεδομένα που έχει εισάγει ο χρήστης στις στήλες # of DETs και # of RETs και τα αριθμητικά αποτελέσματα κατατάσσονται αυτόματα στις στήλες **Low, Average, High**.

Στη στήλη **Παραδοχές/Σημειώσεις** ο χρήστης εισάγει για μελλοντική αναφορά τυχόν σχόλια και παρατηρήσεις σχετικά με τους υπολογισμούς του κάθε Εσωτερικού Λογικού Αρχείου.

Εάν οι διαθέσιμες γραμμές στον πίνακα δεν επαρκούν, ο χρήστης μπορεί μέσω του πλήκτρου **Εισαγωγή Νέας Γραμμής** για να προσθέσει νέες γραμμές (εισάγοντας έτσι και όλους τους απαραίτητους μαθηματικούς τύπους υπολογισμών στα κελιά).

Μετά την ολοκλήρωση της καταγραφής των Λειτουργικών Σημείων για τα Εσωτερικά Λογικά Αρχεία, ο χρήστης μπορεί είτε να προχωρήσει σε ένα από τα επόμενα φύλλα εργασίας είτε να επιστρέψει στο αρχικό φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ}.

2.3.2 Μέτρηση EIFs

Στο φύλλο εργασίας {EIFs} (Εικόνα 3) βρίσκεται πίνακας όπου ο χρήστης πρέπει να καταγράψει τα στοιχεία των **Εξωτερικών Αρχείων Διεπαφών (External Interface Files)** στην προς μέτρηση υπομονάδα λογισμικού (module).

Στη στήλη **External Interface Files** ο χρήστης ονοματίζει όλα τα EIFs που έχουν αναγνωρισθεί, ενώ στις στήλες με όνομα **# of DETs** (Data Element Types) και **# of RETs** (Record Element Types) ο χρήστης συμπληρώνει τα αντίστοιχα δεδομένα.

Οι στήλες με την ένδειξη **Complexity** συμπληρώνονται αυτόματα από το FSMT ανάλογα με τα αριθμητικά δεδομένα που έχει εισάγει ο χρήστης στις στήλες # of DETs και # of RETs και τα αριθμητικά αποτελέσματα κατατάσσονται αυτόματα στις στήλες **Low, Average, High**.

Στη στήλη **Παραδοχές/Σημειώσεις** ο χρήστης εισάγει για μελλοντική αναφορά τυχόν σχόλια και παρατηρήσεις σχετικά με τους υπολογισμούς του κάθε Εξωτερικού Αρχείου Διεπαφών.

Εάν οι διαθέσιμες γραμμές στον πίνακα δεν επαρκούν, ο χρήστης μπορεί μέσω του πλήκτρου **Εισαγωγή Νέας Γραμμής** για να προσθέσει νέες γραμμές (εισάγοντας έτσι και όλους τους απαραίτητους μαθηματικούς τύπους υπολογισμών στα κελιά).

Μετά την ολοκλήρωση της καταγραφής των Λειτουργικών Σημείων για τα Εξωτερικά Αρχεία Διεπαφών, ο χρήστης μπορεί είτε να προχωρήσει σε ένα από τα επόμενα φύλλα εργασίας είτε να επιστρέψει στο αρχικό φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ}.

2.3.3 Μέτρηση Εξωτερικών Εισόδων

Στο φύλλο εργασίας {EIs} (Εικόνα 4) βρίσκεται πίνακας όπου ο χρήστης πρέπει να καταγράψει τα στοιχεία των **Εξωτερικών Εισόδων (External Inputs)** στην προς μέτρηση υπομονάδα λογισμικού (module).

Στη στήλη **External Inputs** ο χρήστης ονοματίζει όλα τα EIs που έχουν αναγνωρισθεί ενώ στις στήλες με όνομα **# of DETs** (Data Element Types) και **# of FTRs** (File Types Referenced) ο χρήστης συμπληρώνει τα αντίστοιχα δεδομένα.

Οι στήλες με την ένδειξη **Complexity** συμπληρώνονται αυτόματα από το FSMT ανάλογα με τα αριθμητικά δεδομένα που εισάγονται στις στήλες # of DETs και # of FTRs και τα αριθμητικά αποτελέσματα κατατάσσονται αυτόματα στις στήλες **Low, Average, High**.

Στη στήλη **Παραδοχές/Σημειώσεις** ο χρήστης εισάγει για μελλοντική αναφορά τυχόν σχόλια και παρατηρήσεις σχετικά με τους υπολογισμούς του κάθε Εξωτερικής Εισόδου.

Εάν οι διαθέσιμες γραμμές στον πίνακα δεν επαρκούν, ο χρήστης μπορεί μέσω του πλήκτρου **Εισαγωγή Νέας Γραμμής** για να προσθέσει νέες γραμμές (εισάγοντας έτσι και όλους τους απαραίτητους μαθηματικούς τύπους υπολογισμών στα κελιά).

Μετά την ολοκλήρωση της καταγραφής των Λειτουργικών Σημείων για τις Εξωτερικές Εισόδους, ο χρήστης μπορεί είτε να προχωρήσει σε ένα από τα επόμενα φύλλα εργασίας είτε να επιστρέψει στο αρχικό φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ}.

2.3.4 Μέτρηση Εξωτερικών Εξόδων

Στο φύλλο εργασίας {EOs} (Εικόνα 5) βρίσκεται πίνακας όπου ο χρήστης πρέπει να καταγράψει τα στοιχεία των **Εξωτερικών Εξόδων (External Outputs)** στην προς μέτρηση υπομονάδα λογισμικού (module).

Στη στήλη **External Outputs** ο χρήστης ονοματίζει όλα τα EOs που έχουν που έχουν αναγνωρισθεί ενώ στις στήλες με όνομα # of DETs (Data Element Types) και # of FTRs (File Types Referenced) ο χρήστης συμπληρώνει τα αντίστοιχα δεδομένα.

Οι στήλες με την ένδειξη **Complexity** συμπληρώνονται αυτόματα από το FSMT ανάλογα με τα αριθμητικά δεδομένα που εισάγονται στις στήλες # of DETs και # of FTRs και τα αριθμητικά αποτελέσματα κατατάσσονται αυτόματα στις στήλες **Low, Average, High**.

Στη στήλη **Παραδοχές/Σημειώσεις** ο χρήστης εισάγει για μελλοντική αναφορά τυχόν σχόλια και παρατηρήσεις σχετικά με τους υπολογισμούς του κάθε Εξωτερικής Εξόδου.

Εάν οι διαθέσιμες γραμμές στον πίνακα δεν επαρκούν, ο χρήστης μπορεί μέσω του πλήκτρου **Εισαγωγή Νέας Γραμμής** για να προσθέσει νέες γραμμές (εισάγοντας έτσι και όλους τους απαραίτητους μαθηματικούς τύπους υπολογισμών στα κελιά).

Μετά την ολοκλήρωση της καταγραφής των Λειτουργικών Σημείων για τις Εξωτερικές Εξόδους, ο χρήστης μπορεί είτε να προχωρήσει σε ένα από τα επόμενα φύλλα εργασίας είτε να επιστρέψει στο αρχικό φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ}.

2.3.5 Μέτρηση Εξωτερικών Αναζητήσεων

Στο φύλλο εργασίας {EQs} (Εικόνα 6) βρίσκεται πίνακας όπου ο χρήστης πρέπει να καταγράψει τα στοιχεία των **Εξωτερικών Αναζητήσεων (External Queries)** στην προς μέτρηση υπομονάδα λογισμικού (module).

Στη στήλη **External Queries** ο χρήστης ονοματίζει όλες τις EQs που έχουν που έχουν αναγνωρισθεί ενώ στις στήλες με όνομα # of DETs (Data Element Types) και # of FTRs (File Types Referenced) ο χρήστης συμπληρώνει τα αντίστοιχα δεδομένα για τις εισόδους (**Input Side**) και τις εξόδους (**Output Side**) αντίστοιχα.

Οι στήλες με την ένδειξη **Input Complexity** και **Output Complexity** συμπληρώνονται αυτόματα ανάλογα με τα αριθμητικά δεδομένα που εισάγονται στις στήλες # of DETs και # of FTRs αντίστοιχα. Ομοίως οι στήλες

με την ένδειξη **Complexity** συμπληρώνονται αυτόματα από το FSMT και προκύπτουν από το μεγαλύτερο συγκριτικά αποτέλεσμα πολυπλοκότητας ανάμεσα στις εισόδους και τις εξόδους των EQs. Τα αριθμητικά αποτελέσματα κατατάσσονται αυτόματα στις στήλες **Low, Average, High**.

Στη στήλη **Παραδοχές/Σημειώσεις** ο χρήστης εισάγει για μελλοντική αναφορά τυχόν σχόλια και παρατηρήσεις σχετικά με τους υπολογισμούς του κάθε Εξωτερικής Αναζήτησης.

Εάν οι διαθέσιμες γραμμές στον πίνακα δεν επαρκούν, ο χρήστης μπορεί μέσω του πλήκτρου **Εισαγωγή Νέας Γραμμής** για να προσθέσει νέες γραμμές (εισάγοντας έτσι και όλους τους απαραίτητους μαθηματικούς τύπους υπολογισμών στα κελιά).

Μετά την ολοκλήρωση της καταγραφής των Λειτουργικών Σημείων για τις Εξωτερικές Αναζητήσεις, ο χρήστης μπορεί να επιστρέψει είτε σε ένα από τα προηγούμενα φύλλα εργασίας είτε στο αρχικό φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ}

2.4 Υπολογισμός Λειτουργικού Μεγέθους

Στο φύλλο εργασίας {FPs} εκτελείται ο Υπολογισμός του Λειτουργικού Μεγέθους της προς μέτρηση υπομονάδας λογισμικού. Στο φύλλο αυτό (Εικόνα 7) μεταφέρονται αυτόματα από το FSMT τα Λειτουργικά Σημεία, όπως αυτά έχουν υπολογιστεί κατά τα προηγούμενα βήματα της διαδικασίας (μέσω των φύλλων εργασίας {ILFs}, {EIFs}, {EIs}, {EOs}, {EQs}).

Σημειώνουμε ότι σ' αυτό το φύλλο εργασίας δεν απαιτείται καμία ενέργεια από το χρήστη, και παρέχεται μόνο για την καλύτερη εποπτεία της διαδικασίας μέτρησης.

Το τελικό αποτέλεσμα της μέτρησης μεταφέρεται αυτόματα στο αρχικό φύλλο εργασίας {ΤΑΥΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ} όπου και παρουσιάζεται στον χρήστη, ενώ παράλληλα λειτουργεί και ως παράμετρος για τον υπολογισμό της ανθρωποπροσπάθειας.

3 Εικόνες (Screenshots) Χρήσης Εργαλείου Υπολογισμού

ΣΥΝΟΠΤΙΚΑ ΣΤΟΙΧΕΙΑ ΤΑΥΤΟΤΗΤΑΣ ΜΕΤΡΗΣΗΣ ΕΡΓΟΥ		
Ταυτότητα Μέτρησης Έργου		
Κωδικός (id) Έργου		
Ονομασία Έργου		
Φάση του Έργου		"0" Αρχική, "1" Ολοκλ. Μελέτης Εφαρμογής,...., "κ" σε Παραγ. Λειτουργ.
Όνομα Υπευθύνου Έργου		
Κωδικός (id) Μέτρησης		
Περιγραφή Εύρους (Score) Μέτρησης		
Όνομα Υπευθύνου Μέτρησης		
Ημερομηνία Μέτρησης		
Σύνοψη Εκτιμήσεων		
Total Function Point Count (FPs)		Από το φύλλο εργασίας FPs
Productivity (PR) in hs/FP	11	
Hours per Person-Month	160	
Person-Months (PM)		
Διαγνωστικές Πληροφορίες		
<p>1. Παρακαλώ εισάγετε τα στοιχεία ταυτότητας μέτρησης του έργου.</p> <p>2. Ελέγξτε το Φύλλο Εργασίας Υπολογισμού Λειτουργικών Σημείων FPs.</p>		

Εικόνα 1

INTERNAL LOGICAL FILES (ILFs)						
Εισαγωγή Νεας Γραμμής						
Internal Logical Files	# of DETs	# of RETs	Complexity			Παραδοχές/Σημειώσεις
			Low	Average	High	
Summary			0	0	0	

Εικόνα 2

ΥΠΟΛΟΓΙΣΜΟΣ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΗΜΕΙΩΝ - FUNCTION POINT COUNT (FP)

Είδος Λειτουργικής Διαδικασίας	Functional Complexity	Count	Weight	Function Points (FPs)
Internal Logical Files (ILFs)	Low	0	7	0
	Average	0	10	0
	High	0	15	0
External Interface Files (EIFs)	Low	0	5	0
	Average	0	7	0
	High	0	10	0
External Inputs (EIs)	Low	0	3	0
	Average	0	4	0
	High	0	6	0
External Outputs (EOs)	Low	0	4	0
	Average	0	5	0
	High	0	7	0
External Queries (EQs)	Low	0	3	0
	Average	0	4	0
	High	0	6	0
Total Function Point Count				0

Διαγνωστικά μηνύματα

1. Χρησιμοποιήστε το φύλλο εργασίας ILFs για τον καθορισμό και την ανάλυση των Internal Logical Files.
2. Χρησιμοποιήστε το φύλλο εργασίας EIFs για τον καθορισμό και την ανάλυση των External Interface Files.
3. Χρησιμοποιήστε το φύλλο εργασίας EIs για τον καθορισμό και την ανάλυση των External Inputs.
4. Χρησιμοποιήστε το φύλλο εργασίας EOs για τον καθορισμό και την ανάλυση των External Outputs.
5. Χρησιμοποιήστε το φύλλο εργασίας EQs για τον καθορισμό και την ανάλυση των External InQueries.

Εικόνα 7