

**ΕΘΝΙΚΗ ΣΧΟΛΗ ΔΗΜΟΣΙΑΣ ΔΙΟΙΚΗΣΗΣ
ΚΑΙ ΑΥΤΟΔΙΟΙΚΗΣΗΣ**

**ΚΕ΄ ΕΚΠΑΙΔΕΥΤΙΚΗ ΣΕΙΡΑ
ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ**

ΤΙΤΛΟΣ

**«Παράμετροι που Επηρεάζουν την Εκτίμηση του Κόστους
Ανάπτυξης Λογισμικού, Σύγκριση Επικρατέστερων
Μεθόδων, Πρακτική Εφαρμογή και Πρόταση Υιοθέτησης
Μεθοδολογίας από Φορείς της Δημόσιας Διοίκησης»**

ΤΜ. ΕΞΕΙΔΙΚΕΥΣΗΣ: ΨΗΦΙΑΚΗΣ ΠΟΛΙΤΙΚΗΣ

Επιβλέπων:

Δρ. ΔΗΜΗΤΡΙΟΣ ΚΑΠΟΠΟΥΛΟΣ

Σπουδαστής:

ΔΗΜΗΤΡΙΟΣ ΧΟΛΙΑΣΜΕΝΟΣ

ΑΘΗΝΑ - 2018

ΧΟΛΙΑΣΜΕΝΟΣ ΔΗΜΗΤΡΙΟΣ

«Παράμετροι που Επηρεάζουν την Εκτίμηση του Κόστους Ανάπτυξης Λογισμικού,
Σύγκριση Επικρατέστερων Μεθόδων, Πρακτική Εφαρμογή και Πρόταση Υιοθέτησης
Μεθοδολογίας από Φορείς της Δημόσιας Διοίκησης»

ΕΘΝΙΚΗ ΣΧΟΛΗ ΔΗΜΟΣΙΑΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΑΥΤΟΔΙΟΙΚΗΣΗΣ

ΑΘΗΝΑ 2018

ΕΣΔΔΑ

Δημήτρης Χολιασμένος ©

2018 - Με την επιφύλαξη παντός δικαιώματος

ΚΕ' Εκπαιδευτική Σειρά

Δήλωση

«Δηλώνω ρητά ότι, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας, δεν παραβιάζει καθ' οιονδήποτε τρόπο πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής.»

Αθήνα, 12/12/2018

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας εργασίας αποτελεί η εκ των προτέρων εκτίμηση του κόστους ανάπτυξης λογισμικού των έργων Τεχνολογιών Πληροφορίας και Επικοινωνίας - ΤΠΕ (Information & Communications Technology - ICT). Η συγκεκριμένη διαδικασία θεωρείται περίπλοκη, αλλά είναι εξαιρετικά κρίσιμης σημασίας για λόγους που περιλαμβάνουν την αποτελεσματική κατανομή των πόρων και την ακρίβεια των χρονοδιαγραμμάτων. Επιπρόσθετα για τους φορείς του δημοσίου, η εκτίμηση του κόστους λογισμικού μπορεί να είναι καταλυτική για την υποστήριξη της απόφασης για εσωτερική ανάπτυξη (in-house development) ή την ανάθεση σε εξωτερικούς φορείς (outsourcing), αλλά και για το είδος της σύμβασης στη δεύτερη περίπτωση.

Η εργασία εστιάζει στην εκτίμηση του κόστους ανάπτυξης έργων λογισμικού της Δημόσιας Διοίκησης. Αρχικά εξετάζει τις παραμέτρους που επηρεάζουν την εκτίμηση του κόστους. Στη συνέχεια συγκρίνει τις επικρατέστερες μεθόδους εκτίμησης, που είναι οι Γραμμές Κώδικα (Lines of Code), τα Λειτουργικά Σημεία (Function Points), το Κατασκευαστικό Μοντέλο Κόστους (Constructive Cost Model - COCOMO) και η Χρήση Σημείων Περίπτωσης (Use Case Points - UCP), παρουσιάζοντας το πεδίο εφαρμογής τους, τα πλεονεκτήματα και τα μειονεκτήματά τους.

Στόχος της εργασίας είναι να προτείνει την καταλληλότερη μέθοδο προς χρήση από τους φορείς της Δημόσιας Διοίκησης. Σύμφωνα με τα συμπεράσματα της εργασίας αυτή είναι η UCP, για λόγους που περιλαμβάνουν τη δυνατότητα αξιόπιστης εκτίμησης του κόστους από πολύ νωρίς, την αυτοματοποίηση της διαδικασίας, αλλά και τη συμβατότητα της μεθόδου με σχεδόν όλα τα έργα λογισμικού.

Τέλος, με βάση υπάρχουσα σύμβαση έργου ΤΠΕ της Δημόσιας Διοίκησης και τη χρήση της προτεινόμενης μεθόδου, γίνεται εκτίμηση του κατά πόσο το συμβατικό τίμημα υπήρξε ρεαλιστικό. Συμπερασματικά, η UCP μπορεί να βοηθήσει καταλυτικά την εκτίμηση του κόστους ανάπτυξης λογισμικού και κατά συνέπεια να υπάρξει ορθολογική χρήση των διαθέσιμων πόρων, υπό την προϋπόθεση της σωστής χρήσης των παραμέτρων της.

Λέξεις-κλειδιά:

Εκτίμηση Κόστους Λογισμικού, Χρήση Σημείων Περίπτωσης, Γραμμές Κώδικα, Λειτουργικά Σημεία, Σημεία Αντικειμένων, Κατασκευαστικό Μοντέλο Κόστους

ABSTRACT

The subject of this thesis is the ex-ante estimation of the software development costs of the Information and Communication Technology (ICT) projects. This process is considered as complex but it is extremely critical for reasons such as efficient allocation of resources and timing accuracy. In addition, for public administration bodies software cost estimation may be important for supporting the decision of internal development or outsourcing option and the type of the contract in the second case.

Therefore, the thesis focuses on estimating the cost of developing Public Administration software projects. Firstly, the parameters that affect the cost estimate are examined. Then, the main evaluation methods are compared, which are the Lines of Code (LOC), Function Points (FP), COCOMO (Constructive Cost Model) and Use Case Points (UCP), and their scope, advantages and disadvantages are presented.

The aim of the thesis is to propose the most appropriate method for use by the Public Administration Bodies, which according to the conclusions of the thesis is UCP, for reasons such as the possibility of reliable cost estimation from a very early stage, the automation of the process and the compatibility of the method with almost all software projects.

Finally, based on an existing public administration ICT contract and the use of the proposed method, an assessment is made on whether the contractual price has been realistic. In conclusion, UCP can help catalytically to estimate the cost of software development, if there has been proper use of its parameters, and consequently to fulfill the aim of rational use of available resources.

Λέξεις-κλειδιά:

Software Cost Estimation, Lines of Code, Function Points, COCOMO, Use Case Points, UCP

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια της ολοκλήρωσης της εκπαίδευσής μου στην Εθνική Σχολή Δημόσιας Διοίκησης και Αυτοδιοίκησης. Η εργασία ήρθε αντιμέτωπη με ένα νέο σχετικά αντικείμενο, η αξία του οποίου δεν φαίνεται να έχει αναγνωρισθεί όπως θα έπρεπε από τη Δημόσια Διοίκηση.

Θα ήθελα σε αυτό το σημείο να ευχαριστήσω τον Δρ. Δημήτρη Καπόπουλο για την υποστήριξή του στα πλαίσια της παρούσας εργασίας, καθώς και για τις πληροφορίες που μου παρείχε ως Εμπειρογνώμων ΤΠΕ στην Ομάδα Διοίκησης Έργου για την υλοποίηση του έργου «Διαχείριση αλλαγών Ολοκληρωμένου Πληροφοριακού Συστήματος Δημοσιονομικής Πολιτικής», ώστε να γίνει δυνατή η ανάλυση της μελέτης περίπτωσης που εξετάζεται στην παρούσα εργασία.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	4
ABSTRACT	5
ΠΡΟΛΟΓΟΣ.....	6
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
ΠΙΝΑΚΑΣ ΕΙΚΟΝΟΓΡΑΦΗΣΗΣ	8
ΠΙΝΑΚΑΣ ΣΥΝΤΜΗΣΕΩΝ ΚΑΙ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ.....	9
ΕΙΣΑΓΩΓΗ.....	10
ΚΥΡΙΟ ΜΕΡΟΣ.....	12
1. Γιατί είναι σημαντική η εκτίμηση του κόστους ανάπτυξης λογισμικού	12
2. Παράμετροι που επηρεάζουν την εκτίμηση	13
3. Μέθοδοι Εκτίμησης Κόστους Λογισμικού.....	17
3.1 Γραμμές κώδικα (Lines of Code – LOC)	17
3.2 Λειτουργικά Σημεία (Function Points – FP).....	18
3.3 Κατασκευαστικό Μοντέλο Κόστους (COCOMO)	21
3.3.1 COCOMO 81	21
3.3.2 COCOMO II.....	24
3.3.3 Πλεονεκτήματα και μειονεκτήματα της μεθόδου	25
3.4 Χρήση Σημείων Περίπτωσης (Use Case Points)	26
3.4.1 Γενικά για τη μέθοδο	26
3.4.2 Συντελεστής Τεχνικής Πολυπλοκότητας (TCF)	27
3.4.3 Συντελεστής Περιβαλλοντικής Πολυπλοκότητας (ECF)	30
3.4.4 Μη Σταθμισμένα Use Cases και Actors (UUCP).....	33
3.4.5 Συντελεστής παραγωγικότητας	37
3.4.6 Υπολογισμός του αποτελέσματος της τελικής εξίσωσης UCP.....	37
3.4.7 Πλεονεκτήματα – μειονεκτήματα της μεθόδου	37
4. Λογισμικό για τον υπολογισμό των τιμών κάθε μεθόδου	38
5. Πρόταση χρήσης της καταλληλότερης μεθόδου για τη Δημόσια Διοίκηση	39
6. Μελέτη Περίπτωσης: Εκτίμηση με βάση υπάρχουσα σύμβαση.....	41
6.1 Περιγραφή περίπτωσης.....	41
6.2 Εκτίμηση Ανθρωποπροσπάθειας με τη μέθοδο UCP.....	42
ΕΠΙΛΟΓΟΣ	49
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΑΡΑΠΟΜΠΕΣ.....	51

ΠΙΝΑΚΑΣ ΕΙΚΟΝΟΓΡΑΦΗΣΗΣ

Πίνακας 1: Παραγωγικότητα ομάδας ανάπτυξης ανά επίπεδο γλώσσας προγραμματισμού.....	18
Πίνακας 2: Υπολογισμός Function Points (FP).....	19
Πίνακας 3: Εξισώσεις COCOMO 81.....	22
Πίνακας 4: Παράγοντες κόστους COCOMO.....	23
Πίνακας 5: Εξισώσεις COCOMO ανά κατηγορία έργου.....	24
Πίνακας 6: Τεχνικοί παράγοντες της UCP:.....	30
Πίνακας 7: Περιβαλλοντικοί παράγοντες της UCP.....	33
Πίνακας 8: Κατηγορίες Use Cases και UUCW.....	34
Πίνακας 9: Κατηγορίες Use Cases και Actors.....	36
Πίνακας 10: Εκτίμηση Ανθρωποπροσπάθειας με τη UCP.....	42
Πίνακας 11: Υπολογισμός Κόστους στη μελέτη περίπτωσης.....	43
Πίνακας 12: Υπολογισμός Συντελεστή Τεχνικής Πολυπλοκότητας.....	44
Πίνακας 13: Υπολογισμός Συντελεστή Περιβαλλοντικής Πολυπλοκότητας.....	45
Πίνακας 14: Υπολογισμός Μη-Σταθμισμένων Use Cases.....	45
Πίνακας 15: Περιγραφή των Use Cases.....	47
Πίνακας 16: Υπολογισμός Μη-Σταθμισμένων Παραγόντων (UAW).....	48
Πίνακας 17: Μη-Σταθμισμένοι Παράγοντες.....	48

ΠΙΝΑΚΑΣ ΣΥΝΤΜΗΣΕΩΝ ΚΑΙ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

ΤΠΕ	Τεχνολογίες Πληροφορίας και Επικοινωνιών
API	Application Programming Interface
BI	Business Intelligence
COCOMO	Constructive Cost Model
ECF	Environmental Complexity Factor
ERP	Enterprise Resource Planning System
FP	Function Points
ICT	Information and Communications Technology
MM	Man Months
LOC	Lines of Code
OP	Object Points
PF	Productivity Factor
TCF	Technical Complexity Factor
UAW	Unadjusted Actor Weight
UCP	Use Case Points
UFC	Unadjusted Function Point Count
UUCP	Unadjusted Use Case Points
UUCW	Unadjusted Use Case Weight

ΕΙΣΑΓΩΓΗ

Στην εποχή μας τα έργα Τεχνολογιών Πληροφορίας και Επικοινωνιών (ΤΠΕ) θεωρούνται εξαιρετικής σημασίας για την οικονομική ανάπτυξη, καθώς οι ΤΠΕ μπορούν να διαδραματίσουν σημαντικό ρόλο σε ένα βιώσιμο αναπτυξιακό μοντέλο για τη χώρα μας, σύμφωνα με το Σχέδιο Δράσης Υλοποίησης έργων ΤΠΕ για τις Ψηφιακές Υποδομές και την Ανάπτυξη (2016).

Οι ΤΠΕ έχουν οριζόντια επίδραση και μπορούν να υποστηρίξουν θεσμικές και οργανωσιακές πρωτοβουλίες, πλέον των τεχνολογικών καινοτομιών που μπορούν να προσφέρουν σε όλους τους τομείς της οικονομίας τις υπηρεσίες, τη βιομηχανία, αλλά και τον πρωτογενή τομέα (γεωργία, εξόρυξη πρώτων υλών).

Όσον αφορά τη Δημόσια Διοίκηση, σύμφωνα με τον Παρασκευά (2015), ο στόχος είναι να βελτιωθεί η παροχή των δημόσιων υπηρεσιών μέσω των ΤΠΕ, καθώς και να ενισχυθούν οι δημοκρατικές διαδικασίες και η υποστήριξη των πολιτικών που ασκεί ο δημόσιος τομέας προς όφελος των πολιτών και των επιχειρήσεων. Η ηλεκτρονική διακυβέρνηση συγκαταλέγεται μεταξύ των σημαντικών προτεραιοτήτων της Ευρωπαϊκής Ένωσης και μπορεί να υλοποιείται μέσω επιτυχημένων έργων ΤΠΕ.

Σύμφωνα με την Εθνική Ψηφιακή Στρατηγική 2016-2021, όσον αφορά τις ψηφιακές υπηρεσίες του Δημοσίου υπάρχει υστέρηση στην παροχή ολοκληρωμένων υπηρεσιών προς τους πολίτες, τις επιχειρήσεις, αλλά και τους Δημόσιους φορείς. Με δεδομένο το ρόλο που οι ΤΠΕ διαδραματίζουν στην ανάπτυξη και στην ευημερία των πολιτών, η ανάπτυξή τους αποτελεί μονόδρομο για τη χώρα και πρέπει να αποτελέσουν το μοχλό για την επανεκκίνηση και ανάπτυξη της οικονομίας και για την ενίσχυση της απασχόλησης - ιδιαίτερα σε τομείς υψηλής εξειδίκευσης, τον καταλύτη για αποτελεσματικότερη και αποδοτικότερη Δημόσια Διοίκηση, καθώς και μέσο για τη βελτίωση της ποιότητας ζωής των πολιτών και την ενίσχυση της κοινωνικής συνοχής. Ένα σημαντικό συγκριτικό πλεονέκτημα που διαθέτει η Ελλάδα, προς την επίτευξη των συγκεκριμένων στόχων, είναι το ανθρώπινο δυναμικό υψηλής εξειδίκευσης που διαθέτει στον τομέα των ΤΠΕ, σύμφωνα πάντα με την Εθνική Ψηφιακή Στρατηγική 2016-2021.

Σύμφωνα με την προτεραιότητα 5.1 του ίδιου κειμένου, βασικές αρχές που θα πρέπει να διέπουν τα νέα έργα πληροφορικής του δημοσίου είναι ο διαμοιρασμός και η επαναχρησιμοποίηση των λύσεων, ο περιορισμός της σπατάλης, ο έλεγχος των δαπανών και η βελτίωση του τρόπου προμηθειών. Κατά συνέπεια, ένα έργο ΤΠΕ, για να θεωρείται

επιτυχημένο, θα πρέπει να έχει κοστολογηθεί σωστά, εκ των προτέρων και με όσο το δυνατόν ακριβέστερη εκτίμηση του πραγματικού κόστους. Για τα έργα ΤΠΕ που περιλαμβάνουν ανάπτυξη λογισμικού, οι μέθοδοι εκτίμησης του κόστους του μπορούν να συνδράμουν ουσιαστικά.

ΚΥΡΙΟ ΜΕΡΟΣ

1. Γιατί είναι σημαντική η εκτίμηση του κόστους ανάπτυξης λογισμικού

Ο Αντωνόπουλος (2015), ορίζει το λογισμικό θεωρείται ένα σύνολο προγραμμάτων που περιλαμβάνουν δομές δεδομένων (διαχείριση πληροφορίας), εντολές (παρέχουν λειτουργίες) ή τεκμηρίωση (περιγραφή του τρόπου λειτουργίας των προγραμμάτων). Σημαντικό μέρος του κόστους στα έργα ΤΠΕ αποτελεί η ανάπτυξη λογισμικού οπότε καθίσταται κρίσιμης σημασίας η εκτίμηση του κόστους λογισμικού, που συνίσταται στην πρόβλεψη των πόρων (υλικών και άυλων) που απαιτούνται για την επιτυχή ανάπτυξή του.

Μία σημαντική απόφαση που θα πρέπει να λάβουν οι φορείς της Δημόσιας Διοίκησης κατά τον προγραμματισμό έργων λογισμικού, είναι αν το λογισμικό θα αναπτυχθεί από το ανθρώπινο δυναμικό του ίδιου του φορέα ή θα επιλεγεί η λύση της εξωτερικής ανάθεσης σε εταιρίες. Σε κάθε περίπτωση, αλλά ιδιαίτερα στη δεύτερη, είναι σημαντικό να συμφωνηθεί η μέθοδος κοστολόγησης που θα χρησιμοποιηθεί, καθώς προφανώς θα επηρεάσει την εκτίμηση του κόστους.

Κατά συνέπεια, η εκτίμηση του κόστους που απαιτείται για την υλοποίηση ενός έργου λογισμικού είναι πολύ σημαντική για τους φορείς λειτουργίας και γενικά για όλους τους συμμετέχοντες. Το έργο της εκτίμησης δυσχεραίνουν η ύπαρξη πολλών σχετικών παραμέτρων, όπως είναι οι νέες απαιτήσεις των χρηστών κατά τη διάρκεια της ανάπτυξης, η έλλειψη κατανόησης των απαιτήσεων του έργου, η μη ορθή αξιολόγηση των ικανοτήτων της ομάδας ανάπτυξης και η ανάγκη συμμόρφωσης με τα πρότυπα.

Η εσφαλμένη εκτίμηση του κόστους ανάπτυξης λογισμικού για έργα της Δημόσιας Διοίκησης μπορεί να επιφέρει δυσάρεστες συνέπειες που περιλαμβάνουν τη διάθεση επιπρόσθετων και αναίτιων κονδυλίων από τον κρατικό και ενωσιακό προϋπολογισμό, καθώς και τον κίνδυνο της αποτυχίας ή/και της βιωσιμότητας των έργων.

Λόγω του γεγονότος ότι το Δημόσιο ξοδεύει μέσα από τον τακτικό προϋπολογισμό, το πρόγραμμα δημοσίων επενδύσεων αλλά και τα κοινοτικά ταμεία σημαντικά ποσά για την ανάπτυξη λογισμικού, γίνεται κατανοητό ότι χωρίς την ύπαρξη της κατάλληλης μεθόδου είναι εξαιρετικά πιθανό το ενδεχόμενο μίας μη συμφέρουσας συμφωνίας για το δημόσιο με επακόλουθο την κατασπατάληση πόρων. Μάλιστα η μέθοδος καλό είναι να αποτελεί και αντικείμενο της σύμβασης που θα υπογράφεται μεταξύ του δημόσιου φορέα και του οικονομικού φορέα (πωλητή).

Σύμφωνα με έρευνες των Puntham και Hastie (2018), η εκτίμηση του κόστους είναι πιο σημαντική από ποτέ, καθώς τα έργα λογισμικού γίνονται μεγαλύτερα και πιο σύνθετα, καθιστώντας δυσκολότερη την εκτίμησή τους. Κρίνεται οπότε καθοριστικής σημασίας να χρησιμοποιείται από τους φορείς της Δημόσιας Διοίκησης συγκεκριμένη μέθοδος - μοντέλο εκτίμησης, ώστε να είναι σε θέση να εκτιμούν οι φορείς τα κόστη των έργων ΤΠΕ και ιδιαίτερα αυτό της ανάπτυξης λογισμικού. Συνήθως οι παράγοντες που επηρεάζουν τη μέτρηση διαφοροποιούν τα μοντέλα. Τα πιο σύγχρονα μοντέλα είναι αλγοριθμικά και εκτός από το μέγεθος του λογισμικού λαμβάνουν υπόψιν και τις διεπαφές με άλλα πληροφοριακά συστήματα, αλλά και τις γνώσεις, εμπειρίες και ικανότητες που έχει η ομάδα ανάπτυξης. Για ορισμένες μεθόδους, όπως θα αναλυθεί στο κύριο μέρος της εργασίας, υπάρχει ειδικό λογισμικό για την αποτύπωση των τιμών των παραγόντων που συνδέονται με την εκτίμηση του κόστους ανάπτυξης λογισμικού.

2. Παράμετροι που επηρεάζουν την εκτίμηση

Ως συνιστώσες του κόστους λογισμικού σύμφωνα με τον Ulbert (2014), και τον Johns (2015), θεωρούνται:

- τα κόστη χρήσης του υλικού και του λογισμικού (εργαλεία) που χρειάζονται για την ανάπτυξη του λογισμικού
- τα έξοδα μετακίνησης και εκπαίδευσης του προσωπικού
- τα λειτουργικά έξοδα των κτιρίων, των εγκαταστάσεων και τα κόστη δικτύωσης και επικοινωνιών
- τα κόστη «ανθρωποπροσπάθειας», που είναι και ο κυρίαρχος παράγοντας κόστους στα περισσότερα έργα και περιλαμβάνουν τους μισθούς προσωπικού, το κόστος κοινωνικής ασφάλισης και λοιπές παροχές προς τους εργαζόμενους

Σύμφωνα με έρευνα των Rosalina και Mansor (2018), υπάρχουν έξι κρίσιμοι παράγοντες που επηρεάζουν σημαντικά τα αποτελέσματα της εκτίμησης κόστους λογισμικού σε ένα έργο του δημόσιου τομέα: οι ικανότητες της ομάδας ανάπτυξης, η υποστήριξη από την ανώτατη διοίκηση, η κατανόηση της ανώτατης διοίκησης όσον αφορά τους στόχους του έργου και η συμμετοχή της σε αυτό, η διαχείριση κινδύνων, οι γνώσεις και η ικανότητα του υπεύθυνου του έργου (project manager).

Οι θεμελιώδεις ερωτήσεις που πρέπει να γίνουν για την εκτίμηση του κόστους έργων λογισμικού είναι πόση προσπάθεια χρειάζεται για να ολοκληρωθεί μία δραστηριότητα,

ποιες θα είναι οι μονάδες μέτρησης της προσπάθειας (όπως για παράδειγμα πώς ορίζεται η ανθρωποημέρα και ο ανθρωπομήνας), πόσο ημερολογιακό χρόνο θα χρειαστεί για να ολοκληρωθεί, και τέλος ποιο είναι το συνολικό κόστος της δραστηριότητας.

Σύμφωνα με τους Leung και Fan (2002), για να θεωρείται επαρκής μία εκτίμηση κόστους λογισμικού θα πρέπει να έχει τα παρακάτω χαρακτηριστικά:

- σχεδιάζεται και υποστηρίζεται και από τον υπεύθυνο του έργου και από την ομάδα ανάπτυξης
- είναι αποδεκτή από όλους τους ενδιαφερόμενους
- βασίζεται σε ένα καλά καθορισμένο και αξιόπιστο μοντέλο εκτίμησης κόστους λογισμικού
- αντλεί στοιχεία από μια βάση δεδομένων που αφορά τη σχετική εμπειρία σε σχέση με το έργο (παρόμοιες διαδικασίες, παρόμοιες τεχνολογίες, παρόμοια περιβάλλοντα, παρόμοια άτομα και παρόμοιες απαιτήσεις)
- καθορίζεται με αρκετή λεπτομέρεια, ώστε να γίνονται αντιληπτοί οι βασικοί κίνδυνοι και να αξιολογείται αντικειμενικά η πιθανότητα επιτυχίας

Υπάρχουν όμως σύμφωνα με τους Καπόπουλο και Σπηλιωτόπουλο (2018) πολλές παράμετροι που οδηγούν στη δύσκολη εκτίμηση και σε υψηλό κίνδυνο αποτυχίας για ακριβή εκτίμηση, όπως:

- η χρήση πολλών παραμέτρων για την εκτίμηση του κόστους, που ενδεχομένως δεν επηρεάζουν όλες στον ίδιο βαθμό το κόστος
- οι νέες απαιτήσεις των χρηστών κατά τη διάρκεια της ανάπτυξης. Επίσης, ο Χαραλαμπίδης (2018) υποστηρίζει ότι αν είναι πιθανό να αλλάξουν οι απαιτήσεις στο μέλλον, οι υποψήφιοι πωλητές μπορεί να μειώσουν τις τιμές των προσφορών για να κερδίσουν τη σύμβαση και αφού το επιτύχουν, στη συνέχεια να χρεώσουν μεγάλα ποσά για τις αλλαγές στις νέες απαιτήσεις
- η έλλειψη κατανόησης των απαιτήσεων του έργου και η έλλειψη ανάλυσής τους, τόσο από την ομάδα διοίκησης του έργου, όσο και από την ομάδα ανάπτυξης του λογισμικού, είτε λόγω πίεσης χρόνου, είτε λόγω μη επιμελούς ενασχόλησης με την ανάλυσή τους
- η μη ορθή αξιολόγηση των ικανοτήτων της ομάδας ανάπτυξης, κυρίως όταν δεν υπάρχουν επαρκή ιστορικά δεδομένα, ή όταν δεν αξιοποιούνται όπως θα έπρεπε

- η ανάγκη συμμόρφωσης με τα πρότυπα, που δημιουργούν νέες απαιτήσεις και που ειδικότερα όσον αφορά στα έργα του δημοσίου τομέα συναντάται σε μεγάλο βαθμό, αυξάνοντας παράλληλα και το κόστος
- η προβληματική συνεννόηση μεταξύ των ενδιαφερομένων, καθώς η μη αποτελεσματική επικοινωνία επιφέρει εκτός από καθυστερήσεις και κατασπατάληση πόρων, πόσο μάλλον όταν αφορά την ανάπτυξη λογισμικού, που από τη φύση της είναι μια περίπλοκη διαδικασία
- η μη πρόβλεψη απαιτούμενων εργασιών, που ενδεχομένως οφείλεται στο ότι η ομάδα διοίκησης του έργου δεν είχε την απαραίτητη εμπειρία και τεχνογνωσία, ή ακόμα και στο γεγονός ότι η ομάδα ανάπτυξης δεν αντιλήφθηκε ορισμένες ιδιαιτερότητες του έργου

Δύο ακόμα δυσκολίες επισημαίνουν και οι Ken Schwaber και Jeff Sutherland (2018), οι συγγραφείς του “Scrum Guide”¹. Η πρώτη είναι η αποτυχία άντλησης ιστορικών δεδομένων από προηγούμενα έργα, κυρίως όσον αφορά την ανθρωποπροσπάθεια, κάτι που θα ήταν εξαιρετικά χρήσιμο στη σωστή εκτίμηση του μεγέθους και του χρονοπρογραμματισμού του έργου. Η δεύτερη, είναι η τάση να συγχέεται η εκτίμηση με τις προθεσμίες υλοποίησης ή τις μετρήσεις επίτευξης στόχων, που αφορούν τους στόχους του οργανισμού ή του φορέα, καθώς η πρώτη βασίζεται σε ποσοτική ανάλυση του τι είναι πιθανό να συμβεί, ενώ οι δεύτερες θα πρέπει να εξετάζονται ανεξάρτητα, ώστε να εξεταστεί αν ταιριάζουν με την πρώτη. Σύμφωνα με έρευνα των Jeff Sutherland και Ken Schwaber (2018), διαπιστώθηκε ότι όσοι απέφευγαν να κάνουν εκτιμήσεις έδωσαν συνολικά μερικούς από τους βραδύτερους χρόνους παράδοσης, ενώ όσοι χρησιμοποιούσαν εκτιμήσεις βασισμένες στο πεδίο εφαρμογής πέτυχαν τα ταχύτερα αποτελέσματα.

Ένα ακόμα πρόβλημα που σχετίζεται με τον ανθρώπινο παράγοντα είναι η υπερβολική αισιοδοξία των εκτιμητών. Επίσης, το κόστος διόρθωσης των λαθών και ατελειών αυξάνει κατά σημαντικό βαθμό όταν αργεί ο εντοπισμός τους, καθώς αυτά μεταφέρονται και σε επόμενα στάδια ανάπτυξης του λογισμικού και απαιτείται έτσι μεγαλύτερη προσπάθεια για τη διόρθωσή τους.

¹ Το Scrum Guide αποτέλεσε την αρχή χρήσης του Scrum, μίας ευέλικτης (agile) μεθόδου πρακτικής οργάνωσης που στοχεύει στην ολοκλήρωση του project μέσα από τη συντονισμένη και ομαδική συνεργασία. Το Scrum βρίσκει πολύ καλή εφαρμογή σε τμήματα με υψηλές απαιτήσεις και μεγάλο όγκο εργασίας, όπως στην ομάδα των προγραμματιστών μιας εταιρίας, αλλά και στον συντονισμό των διαδικασιών ανάμεσα σε διαφορετικά τμήματα.

Δύο ακόμα παράμετροι σύμφωνα με τους Καπόπουλο και Σπηλιωτόπουλο (2018) που πρέπει να ληφθούν υπόψιν όταν εξετάζεται η ανάπτυξη λογισμικού είναι το γεγονός ότι σε έργα με ολιγομελή ομάδα εργασίας το τελικό κόστος ανάπτυξης λογισμικού εξαρτάται κατά πολύ από τις ατομικές ικανότητες, αλλά και το γεγονός ότι το πλήθος των υπαλλήλων δεν είναι ανάλογο με το χρόνο ανάπτυξης του λογισμικού – υπάρχουν και περιπτώσεις αντίθετα, που μπορεί να είναι και αντιστρόφως ανάλογο. Είναι γεγονός πάντως είναι ότι η ποιότητα της ομάδας ανάπτυξης επηρεάζει το κόστος και το χρονοδιάγραμμα καταλυτικά.

Σημαντική επομένως κατά την εκτίμηση του κόστους θεωρείται και η εκτίμηση της παραγωγικότητας της ομάδας ανάπτυξης, ο ρυθμός με τον οποίο οι εμπλεκόμενοι με το έργο παράγουν λογισμικό ή τεκμηρίωση κατά τη διάρκεια ανάπτυξης του έργου. Στην πράξη, η μεγαλύτερη δυσκολία βρίσκεται στην εκτίμηση της παραγωγικότητας της ομάδας ανάπτυξης λογισμικού, ειδικά όταν δεν υπάρχουν επαρκή ιστορικά στοιχεία όπως αναφέρθηκε, ενώ όσον αφορά τον τρόπο μέτρησης της παραγωγικότητας, υπάρχουν δύο κατηγοριών μετρικές:

- *Μετρικές σχετικές με το μέγεθος*, οι οποίες σχετίζονται με το μέγεθος της εξόδου μιας δραστηριότητας. Μπορεί για παράδειγμα να αναφέρονται σε αριθμό γραμμών παραδοτέου πηγαίου κώδικα ή στον αριθμό εντολών του παραδοτέου αντικειμενικού κώδικα.

- *Μετρικές σχετικές με τις λειτουργίες*, που σχετίζονται με τη συνολική λειτουργικότητα του παραδοτέου λογισμικού. Η πιο γνωστή μετρική αυτού του τύπου είναι τα λειτουργικά σημεία.

3. Μέθοδοι Εκτίμησης Κόστους Λογισμικού

Στη συνέχεια παρουσιάζονται τέσσερις δημοφιλείς μέθοδοι για την εκτίμηση του κόστους ανάπτυξης λογισμικού, καθώς και τα βασικά πλεονεκτήματα και μειονεκτήματά τους.

3.1 Γραμμές κώδικα (Lines of Code – LOC)

Η εκτίμηση της παραγωγικότητας με βάση τη LOC είναι η πιο απλή μέθοδος από τις τέσσερις και ορίζεται ως γραμμές πηγαίου κώδικα ανά μήνα προγραμματιστή και χρησιμοποιείται ευρέως ως μέτρο παραγωγικότητας του λογισμικού. Υπολογίζεται με τη διαίρεση του συνολικού αριθμού γραμμών πηγαίου κώδικα με το συνολικό χρόνο ανάπτυξης σε μήνες προγραμματισμού, που απαιτούνται για την ολοκλήρωση του έργου. Η μετρική αυτή είχε προταθεί για πρώτη φορά όταν ακόμα τα προγράμματα γράφονταν με κάρτες², καθεμιά από τις οποίες περιείχε μία γραμμή κώδικα. Στο μοντέλο γίνεται η παραδοχή ότι υπάρχει ευθεία γραμμική σχέση μεταξύ του μεγέθους του συστήματος και του όγκου της τεκμηρίωσης.

Το βασικό πλεονέκτημα της χρήσης της συγκεκριμένης μεθόδου είναι η ευκολία υπολογισμού του σχετικού κόστους. Υπάρχει όμως κίνδυνος για μεγάλες αποκλίσεις λόγω πολυπλοκότητας των πληροφοριακών συστημάτων στην εποχή μας, ενώ υπάρχει και αδυναμία εκτίμησης εκ των προτέρων.

Επίσης, υπάρχει η ιδιαιτερότητα, όσο πιο χαμηλού επιπέδου είναι η γλώσσα προγραμματισμού, τόσο περισσότερος χρόνος να απαιτείται για το ίδιο τελικό αποτέλεσμα. Ο Johns (2015) υποστηρίζει ότι για την υλοποίηση της ίδιας λειτουργικής δυνατότητας χρειάζεται περισσότερος κώδικας σε μια γλώσσα προγραμματισμού χαμηλού επιπέδου συγκριτικά με την υλοποίηση σε μια γλώσσα υψηλού επιπέδου. Οπότε πρέπει να αναρωτηθούν οι διαχειριστές των έργων και αν είναι λειτουργικό να εκτείνεται το λογισμικό σε πολλές γραμμές.

Ακόμα, σε γλώσσες προγραμματισμού όπως η Java πολλές δηλώσεις μπορεί να επεκτείνονται σε πολλές γραμμές ή και σε μία μόνο γραμμή. Οι μετρικές της παραγωγικότητας που βασίζονται στη LOC οπότε, θεωρούν βάσει των παραπάνω ότι οι προγραμματιστές που χρησιμοποιούν πιο αναλυτικές εκφράσεις στη συγγραφή κώδικα

² Διάτρητες κάρτες Hollerith

είναι πιο παραγωγικοί από όσους γράφουν πιο συμπυκνωμένο κώδικα, γεγονός που μπορεί να οδηγήσει σε λάθος εκτιμήσεις.

Χρόνοι Ανάπτυξης Συστήματος					
	Ανάλυση	Σχεδιασμός	Συγγραφή κώδικα	Δοκιμές	Τεκμηρίωση
Γλώσσα assembly	3 εβδομάδες	5 εβδομάδες	8 εβδομάδες	10 εβδομάδες	2 εβδομάδες
Γλώσσα υψηλού επιπέδου	3 εβδομάδες	5 εβδομάδες	4 εβδομάδες	6 εβδομάδες	2 εβδομάδες
	Μέγεθος	Εργασία	Παραγωγικότητα		
Γλώσσα assembly	5000 γραμμές	28 εβδομάδες	714 γραμμές/μήνα		
Γλώσσα υψηλού επιπέδου	1500 γραμμές	20 εβδομάδες	300 γραμμές/μήνα		

Πίνακας 1: Παραγωγικότητα ομάδας ανάπτυξης ανά επίπεδο γλώσσας προγραμματισμού^{3,4}

3.2 Λειτουργικά Σημεία (Function Points – FP)

Η μέθοδος αυτή βασίζεται σε συνδυασμό χαρακτηριστικών στοιχείων του προγράμματος (Tripathi, 2016). Μετράει εισόδους (συναλλαγές εισόδου που ενημερώνουν εσωτερικά αρχεία), εξόδους (αναφορές, μηνύματα σφάλματος), αλληλεπιδράσεις χρηστών, εσωτερικά αρχεία που χρησιμοποιούνται από το σύστημα, διεπαφές (αρχεία που μοιράζονται με άλλα συστήματα), διαδικτυακές υπηρεσίες και ερωτήσεις. Καθένα από αυτά είναι ένα λειτουργικό σημείο.

Κάθε λειτουργικό σημείο έχει έναν συντελεστή βαρύτητας, που κυμαίνεται από την τιμή 3 για τα πιο απλά, έως την τιμή 15 για τα πιο σύνθετα χαρακτηριστικά, σύμφωνα με τον Johns (2015). Η μέτρηση των σημείων λειτουργίας υπολογίζεται πολλαπλασιάζοντας κάθε αρχική μέτρηση με το συντελεστή βαρύτητας και αθροίζοντας όλες τις τιμές.

$$UFC = \sum(\text{πλήθος στοιχείων δεδομένου τύπου}) \times (\text{συντελεστής στάθμισης})$$

UFC: Unadjusted Function Point Count

³ Χαραλαμπίδης Γ, (2018). Σημειώσεις μαθήματος «Τεχνολογίας Λογισμικού» για το Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου

⁴ Σύμφωνα με τον Τσαπέλα (2018), «Γλώσσα υψηλού επιπέδου (high-level language)» θεωρείται γλώσσα στην οποία κάθε πρόταση του πηγαίου κώδικα δημιουργεί πολλές εντολές στο επίπεδο γλώσσας μηχανής, ενώ με τον όρο «Assembly» αναφερόμαστε στη γλώσσα προγραμματισμού που αναπτύχθηκε τη δεκαετία του 1950 και μοιάζει με τη γλώσσα μηχανής, αλλά σε αυτήν οι αριθμητικοί κωδικοί έχουν αντικατασταθεί με μνημονικούς όρους.

Χαρακτηριστικά	Τιμή	Συντ. Βαρύτητας			=	
		Απλός	Μέσος	Περίπλοκος		
Είσοδοι	<input type="checkbox"/>	*	3	4	6	<input type="checkbox"/>
Έξοδοι	<input type="checkbox"/>	*	4	5	7	<input type="checkbox"/>
Αλληλεπιδράσεις χρηστών	<input type="checkbox"/>	*	3	4	6	<input type="checkbox"/>
Αρχεία	<input type="checkbox"/>	*	7	10	15	<input type="checkbox"/>
Διεπαφές	<input type="checkbox"/>	*	5	7	10	<input type="checkbox"/>
Σύνολο	→					<input type="text"/>
Συντελεστής πολυπλοκότητας	→					<input type="text"/>
FP	→					<input type="text"/>

Πίνακας 2: Υπολογισμός Function Points (FP)⁵

Τέλος, το πλήθος λειτουργικών σημείων τροποποιείται ανάλογα με την πολυπλοκότητα του έργου: υπάρχουν 14 παράγοντες που αφορούν την πολυπλοκότητα του έργου, καθένας εκ των οποίων βαθμολογείται με «0» αν δεν είναι σημαντικός ή αν δεν μπορεί να υπολογιστεί και «5» αν είναι εξαιρετικά σημαντικός (Sangeetha 2015).

Οι 14 αυτοί παράγοντες είναι οι παρακάτω:

1. Δημιουργία αντιγράφων ασφαλείας και ανάκτηση
2. Επικοινωνία συστημάτων και βάσεων δεδομένων
3. Κατανεμημένες λειτουργίες επεξεργασίας
4. Κρισιμότητα απόδοσης
5. Το υπάρχον περιβάλλον λειτουργίας
6. Καταχώριση δεδομένων online
7. Εισαγωγή δεδομένων και αλληλεπίδραση σε πολλές οθόνες
8. Online ενημέρωση κύριων αρχείων
9. Πολυπλοκότητα εισόδων, εξόδων, αρχείων, ερωτήσεων
10. Πολυπλοκότητα της επεξεργασίας
11. Σχεδίαση κώδικα για επαναχρησιμοποίηση
12. Συμπερίληψη μετατροπής/εγκατάστασης στο σχεδιασμό
13. Πολλαπλές εγκαταστάσεις
14. Εφαρμογή που έχει σχεδιαστεί για να διευκολύνει την αλλαγή από τον χρήστη

⁵ Johns (2015). Software Engineering - Software Cost Estimation.

Τέλος, ο υπολογισμός των λειτουργικών σημείων γίνεται με βάση την εξίσωση:

$$FP = UFC * [0.65 + 0.01 * \sum_{i=1}^{i=14} F_i]$$

Τα βασικά πλεονεκτήματα της χρήσης της συγκεκριμένης μεθόδου είναι ότι μπορεί να είναι σε σύντομο χρονικό διάστημα διαθέσιμη με μόνο προαπαιτούμενο μία λεπτομερής περιγραφή των λειτουργικών σημείων, ότι δεν περιορίζεται από τον κώδικα και είναι ουσιαστικά ανεξάρτητη από τη γλώσσα προγραμματισμού τα εργαλεία ή τις μεθοδολογίες που χρησιμοποιούνται για την υλοποίηση και ότι είναι σε μεγάλο βαθμό πιο ακριβής μέθοδος από τη LOC. Ακόμα, οι χρήστες που δεν έχουν επαρκείς γνώσεις προγραμματισμού, έχουν καλύτερη κατανόηση του αντικειμένου της μέτρησης, καθώς τα λειτουργικά σημεία βασίζονται στην εξωτερική εικόνα του συστήματος.

Ορισμένα μειονεκτήματα της μεθόδου είναι:

- η υποκειμενικότητα των λειτουργικών σημείων, που εξαρτώνται από τον κάθε εκτιμητή (Satapathy & Rath, 2014)
- η μη συμπερίληψη ενδεχόμενων προβλημάτων ποιότητας της παραγωγής
- η δυσκολία αυτοματοποίησης της μέτρησης των λειτουργικών σημείων
- ότι καθίσταται προβληματικός και ο υπολογισμός έργων με λίγα FPs, αλλά μεγάλη πολυπλοκότητα σε καθένα από αυτά

Σε αυτό το σημείο, καλό θα ήταν να γίνει αναφορά και στη τη μέθοδο των «Σημείων Αντικειμένων» (*Object Points*), που ουσιαστικά αποτελεί βελτίωση της μεθόδου των «Λειτουργικών Σημείων» και είναι η εναλλακτική που προτιμάται για αντικειμενοστραφείς γλώσσες 4^{ης} γενιάς⁶ και συνίσταται στην εκτίμηση του πλήθους των οθονών (forms), των διαδικασιών (procedures) και των αναφορών (reports) του συστήματος, ενώ το πλήθος των αντικειμενικών σημείων ενός προγράμματος είναι μία εκτίμηση των παραπάνω. Αξίζει να σημειωθεί ότι τα OPs δεν ταυτίζονται με τις κλάσεις αντικειμένων (object classes). Σε σύγκριση με τα λειτουργικά σημεία, η εκτίμηση των αντικειμενικών σημείων από τις προδιαγραφές του λογισμικού είναι πιο εύκολη αφού αφορούν μόνο οθόνες, αναφορές, και διαδικασίες (procedures) των συμβατικών γλωσσών προγραμματισμού, άρα το πλήθος τους μπορεί να εκτιμηθεί σε ένα αρκετά

⁶ Οι Γλώσσες 4ης Γενιάς 4GL (4th Generation Languages) είναι γλώσσες προγραμματισμού υψηλού επιπέδου, δηλ. έχουμε την δυνατότητα να κάνουμε πολλές και σύνθετες λειτουργίες με την βοήθειά τους και με λίγες μόνο εντολές.

πρώιμο στάδιο της διαδικασίας ανάπτυξης. Σε ένα τέτοιο στάδιο, όπως έχει ήδη αναφερθεί, η εκτίμηση των γραμμών κώδικα ενός συστήματος για παράδειγμα είναι εξαιρετικά δύσκολη.

3.3 Κατασκευαστικό Μοντέλο Κόστους (COCOMO)

3.3.1 COCOMO 81

Το COCOMO (Constructive Cost Model) αναπτύχθηκε από τον Barry Boehm το 1981, βασίζεται σε αλγοριθμικό μοντέλο κοστολόγησης και θεωρείται το πρώτο σημαντικό άλμα στον τρόπο υπολογισμού κόστους έργων λογισμικού. Σύμφωνα με τον Χαραλαμπίδη (2018), το COCOMO αποτελεί ένα καλά τεκμηριωμένο εμπειρικό μοντέλο που βασίζεται σε εμπειρικά δεδομένα έργων.

Σημαντικοί παράγοντες για τη μέθοδο είναι το μέγεθος του λογισμικού και η ποιότητα της ομάδας ανάπτυξης. Ουσιαστικά αποτελεί ένα καλά τεκμηριωμένο, «ανεξάρτητο» μοντέλο το οποίο δεν δεσμεύεται από συγκεκριμένο προμηθευτή λογισμικού. Το μοντέλο υπολογίζει τη διάρκεια και το κόστους ενός έργου λογισμικού βασιζόμενο στο μέγεθος του προϊόντος και την ποιότητα της ομάδας ανάπτυξης

Οι απλουστευτικές υποθέσεις του είναι:

- 152 ώρες εργασίας ανά εργατομήνα
- ικανή ομάδα διοίκησης του έργου λογισμικού
- το έγγραφο περιγραφής απαιτήσεων δεν τροποποιείται μετά την έγκρισή του
- ο σημαντικότερος παράγοντας κόστους είναι το μέγεθος του έργου λογισμικού
- το μέγεθος του έργου προσδιορίζεται σε χιλιάδες γραμμές παραδοτέου κώδικα (Thousands delivered source instructions - KDSI)

Τα έργα κατατάσσονται σε τρία μοντέλα:

- Βασικό μοντέλο
- Ενδιάμεσο μοντέλο
- Λεπτομερειακό μοντέλο

Πολυπλοκότητα έργου	Τύπος	Περιγραφή
Απλό	$PM = 2,4 (KDSI)^{1,05} \times M$	Κατανοητές εφαρμογές που αναπτύσσονται από μικρές ομάδες
Μέτριο	$PM = 3,0 (KDSI)^{1,12} \times M$	Πιο περίπλοκα έργα στα οποία τα μέλη της ομάδας μπορεί να έχουν περιορισμένη πείρα σε αντίστοιχα συστήματα
Ενσωματωμένο	$PM = 3,6 (KDSI)^{1,20} \times M$	Πολύπλοκα έργα στα οποία το λογισμικό είναι μέρος ενός στενά συνδεδεμένου συμπλέγματος υλικού, λογισμικού, κανονισμών, και λειτουργικών διαδικασιών

Πίνακας 3: Εξισώσεις COCOMO 81⁷

Στο βασικό μοντέλο, υπολογίζεται η απαιτούμενη προσπάθεια σε ανθρωπομήνες:

$$MM = 2.4 (KDSI)^{1.05}$$

Αξίζει να σημειωθεί ότι ο εκθέτης του KDSI είναι σχεδόν ίσος με τη μονάδα, άρα η προσπάθεια ανάπτυξης είναι γραμμική συνάρτηση του μεγέθους του κώδικα.

Απαιτούμενος χρόνος ανάπτυξης του έργου

$$T = 2.5 (MM)^{0.38}$$

Ο αριθμός των απασχολούμενων ατόμων θεωρείται ότι είναι ο «βέλτιστος» για κάθε στάδιο του έργου. Για παράδειγμα, στο μέσο του έργου απασχολούνται περισσότεροι από ότι στη φάση καθορισμού των απαιτήσεων.

Στο ενδιάμεσο μοντέλο εισάγονται δύο κύριες τροποποιήσεις:

- τα έργα λογισμικού διακρίνονται σε τρεις κατηγορίες: οργανικά ή απλά (organic ή simple), ημιαποσπασμένα (moderate ή semi-detached) και ενσωματωμένα (embedded)
- αναγνωρίζονται συγκεκριμένοι παράγοντες που επηρεάζουν το κόστος (αφορούν το προϊόν, τη διαδικασία και το προσωπικό)

Ως προς τις κατηγορίες των έργων:

⁷ Χαραλαμπίδης Γ, (2018). Σημειώσεις μαθήματος «Τεχνολογίας Λογισμικού» για το Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου

- i. *Οργανικά ή απλά*: Το πρόγραμμα είναι σχετικά ανεξάρτητο και έχει μικρή διασύνδεση με το περιβάλλον (π.χ. σύστημα ανάλυσης πληροφοριών πτήσης, που επεξεργάζεται τα δεδομένα μετά το τέλος της πτήσης)
- ii. *Ημιαποσπασμένα*: Υπάρχουν συνδέσεις με το περιβάλλον, αλλά όχι κρίσιμες χρονικές εξαρτήσεις (π.χ. προσομοιωτής πτήσης αεροσκαφών)
- iii. *Ενσωματωμένα*: Κάθε συναλλαγή με το σύστημα είναι κρίσιμη από πλευράς χρόνου και ακρίβειας και υπάρχουν αρκετοί περιορισμοί που εισάγονται από το περιβάλλον (π.χ. λογισμικό αεροσκάφους για την αποφυγή συγκρούσεων με άλλα αεροσκάφη κατά τη διάρκεια της πτήσης)

Εξισώσεις COCOMO: υπάρχει ένα σύνολο από 15 χαρακτηριστικά («παράγοντες κόστους») που συνεισφέρουν στο κόστος και σε κάθε παράγοντα αντιστοιχεί ένας «πολλαπλασιαστής προσπάθειας» (index: q1, q2, ... , q15), οπότε προκύπτει η εξίσωση υπολογισμού παράγοντα προσαρμογής προσπάθειας:

1.	RELY	Απαιτούμενη αξιοπιστία λογισμικού	προϊόν
2.	DATA	Μέγεθος βάσης δεδομένων	προϊόν
3.	CPLX	Πολυπλοκότητα προϊόντος	προϊόν
4.	TIME	Περιορισμός στο χρόνο εκτέλεσης	υπολογιστής
5.	STOR	Περιορισμός στην κύρια μνήμη	υπολογιστής
6.	VIRT	Αλλαγές στο σύστημα HW/SW	υπολογιστής
7.	TURN	Χρόνος απόκρισης υπολογιστή	υπολογιστής
8.	ACAP	Ικανότητα αναλυτών	προσωπικό
9.	AEXP	Εμπειρία αναλυτών σε εφαρμογές	προσωπικό
10.	PCAP	Ικανότητα προγραμματιστών	προσωπικό
11.	VEXP	Εμπειρία με το σύστημα HW/SW (OS etc)	προσωπικό
12.	LEXP	Εμπειρία με τη γλώσσα προγραμματισμού	προσωπικό
13.	MODP	Χρήση μοντέρνων πρακτικών προγραμματ.	έργο
14.	TOOL	Χρήση εργαλείων προγραμματισμού	έργο
15.	SCED	Πίεση από χρονοδιάγραμμα ανάπτυξης	έργο

Πίνακας 4: Παράγοντες κόστους COCOMO⁸

Εξισώσεις στην COCOMO:

Εξίσωση υπολογισμού προσπάθειας ανάπτυξης:

$$MM_{DEV} = q * MM_{NOM}$$

Εξίσωση υπολογισμού κόστους:

$$C = p * MM_{DEV}, \text{ όπου } p: \text{ αξία σε χρήμα ενός ανθρωπομήνα}$$

Εξίσωση υπολογισμού διάρκειας του έργου:

⁸ Αντωνόπουλος Χρ. (2009). Σημειώσεις μαθήματος «Ανάπτυξη και σχεδίαση λογισμικού», Τμήματος Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων Πανεπιστημίου Θεσσαλίας

$$T_{DEV} = R * (MM_{DEV}) * m, \text{ όπου } R, m: \text{ παράμετροι του μοντέλου}$$

Δεδομένα εισόδου: μέγεθος ενός έργου λογιστικού (KDSI), παράγοντες κόστους, τιμή του ανθρωπομήνα εργασίας. Για να λειτουργήσει το μοντέλο απαιτείται βαθμονόμηση (calibration) για την καταχώρηση τιμών στις παραμέτρους του μοντέλου. Ο Boehm βαθμονόμησε το μοντέλο χρησιμοποιώντας δεδομένα από 63 διαφορετικά έργα λογισμικού.

$$MM_{NOM} = C (KDSI)$$

Όπου C παράμετρος του μοντέλου,

KDSI: thousands of delivered source lines,

MM_{NOM}: ονομαστική προσπάθεια σε ανθρωπομήνες

Κατηγορία	ονομαστική προσπάθεια	διάρκεια ανάπτυξης
οργανική	$MM_{NOM} = 3.2(KDSI)^{1.05}$	$T_{DEV} = 2.5(MM_{DEV})^{0.38}$
Ημιαπο- σπασμένη	$MM_{NOM} = 3.0(KDSI)^{1.12}$	$T_{DEV} = 2.5(MM_{DEV})^{0.35}$
Ενσωμα- τωμένη	$MM_{NOM} = 2.8(KDSI)^{1.20}$	$T_{DEV} = 2.5 (MM_{DEV})^{0.3}$

Πίνακας 5: Εξισώσεις COCOMO ανά κατηγορία έργου⁹

3.3.2 COCOMO II

Το μοντέλο COCOMO II λαμβάνει υπόψη διάφορες προσεγγίσεις ανάπτυξης λογισμικού ή επαναχρησιμοποίησης. Ενδείκνυται κυρίως για αντικειμενοστρεφές λογισμικό (object-oriented) και γίνεται χρήση ρουτινών από άλλους κατασκευαστές. Περιλαμβάνει πολλά δευτερεύοντα μοντέλα που μπορούν να παράγουν ολόένα και πιο λεπτομερείς εκτιμήσεις για το λογισμικό (Χαραλαμπίδης 2018).

Είναι πιο κατάλληλη μέθοδος για έργα κατασκευασμένα με σύγχρονα εργαλεία GUI-builder, η εκτίμηση βασίζεται στις τυπικές εκτιμήσεις της παραγωγικότητας των προγραμματιστών σε παραγόμενα σημεία αντικειμένων/μήνα. Υποστηρίζει πρωτότυπα έργα και έργα όπου υπάρχει εκτεταμένη επαναχρησιμοποίηση.

⁹ Αντωνόπουλος Χρ. (2009). Σημειώσεις μαθήματος «Ανάπτυξη και σχεδίαση λογισμικού», Τμήματος Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων Πανεπιστημίου Θεσσαλίας

Τα υπομοντέλα του COCOMO II είναι:

- *Μοντέλο σύνθεσης εφαρμογής.* Χρησιμοποιείται όταν το λογισμικό συντίθεται από υπάρχοντα στοιχεία
- *Μοντέλο πρόιμου σχεδιασμού.* Χρησιμοποιείται όταν οι απαιτήσεις είναι διαθέσιμες, αλλά δεν έχει ξεκινήσει ακόμα ο σχεδιασμός
- *Μοντέλο επαναχρησιμοποίησης.* Χρησιμοποιείται για τον υπολογισμό του φόρτου εργασίας που απαιτείται για την ενοποίηση των επαναχρησιμοποιήσιμων συστατικών στοιχείων.
- *Μετα-αρχιτεκτονικό μοντέλο.* Χρησιμοποιείται αφού έχει οριστικοποιηθεί η αρχιτεκτονική του συστήματος και υπάρχουν περισσότερες διαθέσιμες πληροφορίες για αυτό.

3.3.3 Πλεονεκτήματα και μειονεκτήματα της μεθόδου

Ένα από τα πλεονεκτήματα της COCOMO είναι ότι πρόκειται για μία απλή μέθοδο, αφού απαιτεί ένα μικρό αριθμό δεδομένων για τον προσδιορισμό της προσπάθειας και του κόστους. Οι εξισώσεις προέκυψαν από την ανάλυση ενός μεγάλου αριθμού έργων και συνεπώς μπορεί να ισχυριστεί ότι βασίζεται σταθερά στην πραγματικότητα.

Ένα σχετικό μειονέκτημα όμως είναι ότι αποτελεί ένα εμπειρικό μοντέλο εκτίμησης και τα εμπειρικά δεδομένα που υποστηρίζουν τα μοντέλα της στην πράξη προέρχονται συνήθως από περιορισμένο δείγμα έργων. Ο Abbran (2010) έχει παρατηρήσει ότι πολλοί παράμετροι των μοντέλων COCOMO περιγράφονται με υποκειμενικό τρόπο και η επιρροή τους καθορίζεται από τους εμπειρογνώμονες των έργων και όχι από μία βάση δεδομένων από τα περιγραφικά αποθετήρια που αφορούν έργα που έχουν υλοποιηθεί. Ωστόσο, με τη χρήση βελτιωμένων δεδομένων με αυξημένο αριθμό προηγούμενων έργων, τα αποτελέσματα του COCOMO είναι πιθανό να βελτιωθούν.

3.4 Χρήση Σημείων Περίπτωσης (Use Case Points)

3.4.1 Γενικά για τη μέθοδο

Σύμφωνα με την Bente (2002), η μέθοδος της χρήσης σημείων περίπτωσης (Use Case Points - UCP) αναπτύχθηκε αρχικά από τον Gustav Karner το 1993, αφορά τη μοντελοποίηση των περιπτώσεων χρήσης (use cases στο εξής¹⁰) και αποτελεί μία βελτίωση της μεθόδου των FPs. Στη συνέχεια, η μέθοδος αδειοδοτήθηκε από την Rational Software που αργότερα συγχωνεύθηκε με την IBM.

Η ανάλυση των παραμέτρων των use cases μπορεί να καταλήξει σε πολύτιμα συμπεράσματα σε σχέση με το μέγεθος και την ανθρωποπροσπάθεια που απαιτείται για τη σχεδίαση και υλοποίηση του λογισμικού. Ουσιαστικά η μέθοδος μετράει το μέγεθος του λογισμικού και κατά συνέπεια τον απαιτούμενο χρόνο για την ανάπτυξη του, με τη χρήση των σημείων περίπτωσης. Αναλύει τους συντελεστές χρήσης, τα σενάρια και τους διάφορους τεχνικούς και περιβαλλοντικούς παράγοντες και τους εισάγει σε μια τελική εξίσωση. Η ανάλυση του περιεχομένου των περιπτώσεων παρέχει πολύτιμη εικόνα της προσπάθειας που απαιτείται για τον σχεδιασμό και την υλοποίηση της εφαρμογής (Καπόπουλος, Σπηλιωτόπουλος, 2018).

Ουσιαστικά, ένα use case αποτελεί ένα σύνολο σεναρίων που συνδέονται με ένα κοινό σκοπό για το χρήστη. Συνήθως περιλαμβάνει ένα πρωταρχικό σενάριο (σαν μία ακολουθία από βήματα) και τις εναλλακτικές περιπτώσεις (σαν παραλλαγές της κύριας ακολουθίας βημάτων). Ενώ εκτελείται ένα βήμα, ενδεχομένως συμβούν ορισμένα συμβάντα, οι περιγραφές των οποίων μαζί με τα εναλλακτικά σενάρια που είναι διαθέσιμα περιλαμβάνονται σε ένα use case.

Ανακύπτει οπότε το ζήτημα του τρόπου προσδιορισμού ενός (περισσότερο ή λιγότερο πλήρους) καταλόγου πιθανών γεγονότων, για το οποίο οι σχετικές έρευνες δείχνουν ότι τα μη ταυτοποιημένα γεγονότα αποτελούν τη συνηθέστερη αιτία αλλαγών στις απαιτήσεις και η σημαντικότητα της πληρότητας του καταλόγου πιθανών συμβάντων έχει αναγνωριστεί από πολλούς συγγραφείς συμπεριλαμβανομένων των Carson (1998), Cox (2004), και Wiegers (2003).

¹⁰ Στα πλαίσια αντίστοιχων αναλύσεων στη βιβλιογραφία προτιμάται ο τεχνικός όρος «"use case"».

Σε γενικές γραμμές, οι εφαρμογές με μεγάλα, πολύπλοκα use cases απαιτούν περισσότερη προσπάθεια για το σχεδιασμό και την υλοποίηση από ότι οι μικρές εφαρμογές που έχουν λιγότερο περίπλοκα use cases.

Η εξίσωση υπολογισμού περιλαμβάνει (Silhavy & Prokorova 2018):

- τον Συντελεστή Τεχνικής Πολυπλοκότητας (Technical Complexity Factor - TCF)
- τον Συντελεστή Περιβαλλοντικής (Εξωγενούς) Πολυπλοκότητας (Environmental Complexity Factor - ECF)
- τις Μη Σταθμισμένες Περιπτώσεις Χρήσης και τους Μη Σταθμισμένους Παράγοντες¹¹ (Unadjusted Use Case Points - UUCP)
- τον Συντελεστή Παραγωγικότητας (Productivity Factor - PF)

Κάθε συντελεστής υπολογίζεται με τη χρήση τιμών κατόπιν εκτίμησης αλλά και συγκεκριμένων σταθερών.

Οπότε τα απαραίτητα βήματα ώστε να γίνει ο τελικός υπολογισμός με βάση τη συγκεκριμένη μέθοδο είναι:

1. Προσδιορισμός και υπολογισμός του Συντελεστή Τεχνικής Πολυπλοκότητας (TCF)
2. Προσδιορισμός και υπολογισμός του Συντελεστή Εξωγενούς Πολυπλοκότητας (ECF)
3. Υπολογισμός των Αστάθμητων Use Cases και Actors (UUCP)
4. Προσδιορισμός του Συντελεστή Παραγωγικότητας
5. Υπολογισμός του αποτελέσματος της τελικής εξίσωσης

Καθένα από τα παραπάνω βήματα και ο τρόπος υπολογισμού τους, σύμφωνα με τους Καπόπουλο και Σπηλιωτόπουλο (2018), παρουσιάζεται στη συνέχεια:

3.4.2 Συντελεστής Τεχνικής Πολυπλοκότητας (TCF)

Υπάρχουν 13 συνήθεις τεχνικοί παράγοντες για την εκτίμηση των επιπτώσεων στην παραγωγικότητα, οι οποίοι αξιολογούνται από την ομάδα ανάπτυξης. Τους αποδίδεται μια τιμή από το 0 έως το 5 σύμφωνα με την αντίληψή τους περί πολυπλοκότητας

¹¹ Στο εξής “Actors”, που είναι ο τεχνικός όρος

(complexity). Τιμή ίση με 0 σημαίνει ότι ο τεχνικός παράγοντας είναι άσχετος με το έργο, 5 σημαίνει ότι έχει ισχυρή επιρροή.

Το βάρος κάθε παράγοντα πολλαπλασιάζεται με την πολυπλοκότητά του για να παράγει την τιμή του τεχνικού παράγοντα. Οι επιμέρους παράγοντες αθροίζονται για να παραχθεί ο Συνολικός Παράγοντας Τεχνικής Πολυπλοκότητας. Δηλαδή:

$$\text{TotalFactor} = \sum_{i=1}^{13} W_i * P_i$$

όπου με W_i εκφράζεται το βάρος και με P_i η πολυπλοκότητα

Ο συντελεστής τεχνικής πολυπλοκότητας προκύπτει από την εξίσωση:

$$\text{TCF} = 0.6 + (0.01 \times \text{TotalFactor})$$

Οι 13 τεχνικοί παράγοντες, είναι:

1. *Απαιτούμενο σύστημα διανομής:* Η αρχιτεκτονική της λύσης μπορεί να είναι συγκεντρωτική ή παραμετροποιημένη για κάθε χρήστη (single-tenant), ή μπορεί να διανεμηθεί ή να είναι προσαρμοσμένη σε ομάδες πελατών. Οι υψηλότεροι αριθμοί αντιπροσωπεύουν μια πιο σύνθετη αρχιτεκτονική.
2. *Σημασία του χρόνου απόκρισης:* Η ταχύτητα ανταπόκρισης των χρηστών είναι ένας σημαντικός παράγοντας. Για παράδειγμα, εάν το φορτίο του διακομιστή αναμένεται να είναι πολύ χαμηλό, αυτό μπορεί να είναι ένας ασήμαντος παράγοντας. Υψηλότεροι αριθμοί αντιπροσωπεύουν αυξανόμενη σημασία του χρόνου απόκρισης
3. *Αποτελεσματικότητα Τελικού Χρήστη:* Η εφαρμογή αναπτύσσεται για να βελτιστοποιεί την αποτελεσματικότητα των χρηστών; Υψηλότεροι αριθμοί αντιπροσωπεύουν έργα που βασίζονται περισσότερο στην εφαρμογή για τη βελτίωση της αποδοτικότητας των χρηστών.
4. *Απαίτηση για σύνθετη εσωτερική διαδικασία:* ουσιαστικά εξετάζεται αν υπάρχει πολύ δύσκολη αλγοριθμική εργασία και δοκιμή. Οι σύνθετοι αλγόριθμοι (ισοπέδωση πόρων, ανάλυση συστημάτων χρονικού πεδίου, κύβοι OLAP) έχουν υψηλότερους αριθμούς. Απλά ερωτήματα βάσης δεδομένων θα έχουν χαμηλό αριθμό.

5. *Εστίαση στη χρήση επαναχρησιμοποιούμενου κώδικα:* εξετάζεται αν είναι μέσα στους στόχους η επαναχρησιμοποίηση του κύριου κώδικα. Η επαναχρησιμοποίηση κώδικα μειώνει την προσπάθεια που απαιτείται για την ανάπτυξη ενός έργου και το χρονικό διάστημα που απαιτείται για την εκσφαλμάτωση ενός έργου. Ενδεχόμενη λειτουργία «κοινής βιβλιοθήκης» μπορεί να επαναχρησιμοποιηθεί πολλές φορές και ο καθορισμός του κώδικα σε ένα μέρος μπορεί να επιλύσει πολλά σφάλματα, οπότε όσο υψηλότερο είναι το επίπεδο επαναχρησιμοποίησης, τόσο μικρότερος είναι ο αριθμός.

6. *Ευκολία εγκατάστασης:* εξετάζεται αν η ευκολία εγκατάστασης για τους τελικούς χρήστες αποτελεί βασικό παράγοντα, οπότε όσο υψηλότερο είναι το επίπεδο επάρκειας των χρηστών, τόσο μικρότερος είναι ο αριθμός.

7. *Ευχρηστία:* Αν η ευκολία χρήσης αποτελεί ένα κύριο κριτήριο αποδοχής, που σημαίνει μεγάλη σημασία της χρηστικότητας, τόσο μεγαλύτερος θα είναι ο αριθμός.

8. *Υποστήριξη από πολλαπλές πλατφόρμες:* Εξετάζεται αν απαιτείται υποστήριξη από πολλαπλές πλατφόρμες. Από όσες περισσότερες πλατφόρμες πρέπει να υποστηρίζεται το λογισμικό (όπως για παράδειγμα εκδόσεις προγραμμάτων περιήγησης, κινητές συσκευές ή λειτουργικά συστήματα - Windows/OSX/Unix), τόσο μεγαλύτερη είναι η τιμή.

9. *Ευκολία αλλαγών:* όσο μεγαλύτερη ανάγκη για αλλαγή/προσαρμογή απαιτείται στο μέλλον, τόσο μεγαλύτερη είναι η τιμή.

10. *Ταυτόχρονη επεξεργασία:* Εξετάζεται αν υπάρχει η ανάγκη αντιμετώπισης κλειδώματος της βάσης δεδομένων και άλλων θεμάτων που σχετίζονται με υψηλές ανάγκες για συγχρονισμό των δεδομένων. Όσο περισσότερη προσπάθεια θα πρέπει να καταβληθεί για την επίλυση των συγκρούσεων στα δεδομένα ή την εφαρμογή, τόσο μεγαλύτερη είναι η τιμή.

11. *Προσαρμοσμένη ασφάλεια:* Μπορούν οι υπάρχουσες λύσεις ασφάλειας να αξιοποιηθούν ή πρέπει να αναπτυχθεί προσαρμοσμένος κώδικας; Όσο πιο προσαρμοσμένη εργασία ασφάλειας που πρέπει να γίνει (για παράδειγμα, επίπεδο τομέα, επίπεδο σελίδας ή ασφάλεια βάσει ρόλων), τόσο μεγαλύτερη είναι η τιμή.

12. *Εξάρτηση από κώδικα τρίτων:* Η εφαρμογή απαιτεί τη χρήση ελέγχων ή βιβλιοθηκών από τρίτους; Όπως ο επαναχρησιμοποιήσιμος κώδικας, ο κώδικας τρίτου μέρους μπορεί να μειώσει την προσπάθεια που απαιτείται για την

ανάπτυξη μιας λύσης. Όσο μεγαλύτερος είναι ο κωδικός τρίτου (και όσο πιο αξιόπιστος είναι ο κώδικας του τρίτου μέρους), τόσο μικρότερος είναι ο αριθμός.

13. *Εκπαίδευση χρηστών*: Εδώ εξετάζεται ο απαιτούμενος χρόνος εκπαίδευσης των χρηστών; Είναι η εφαρμογή πολύπλοκη ή υποστηρίζει πολύπλοκες δραστηριότητες; Όσο περισσότερο χρειαστούν οι χρήστες να περάσουν το κατώτατο όριο απορρόφησης (να επιτύχουν ένα επίπεδο γνώσης του προϊόντος), τόσο μεγαλύτερη είναι η τιμή.

Στον παρακάτω πίνακα βλέπουμε τους τεχνικούς παράγοντες με τα βάρη τους:

α/α	Περιγραφή	Βάρος
T1	Απαιτούμενο σύστημα διανομής	2
T2	Σημασία του χρόνου απόκρισης	1
T3	Αποτελεσματικότητα Τελικού Χρήστη	1
T4	Απαίτηση για σύνθετη εσωτερική διαδικασία	1
T5	Εστίαση στη χρήση επαναχρησιμοποιούμενου κώδικα	1
T6	Ευκολία εγκατάστασης	0,5
T7	Ευχρηστία	0,5
T8	Υποστήριξη από πολλαπλές πλατφόρμες	2
T9	Ευκολία αλλαγών	1
T10	Ταυτόχρονη επεξεργασία	1
T11	Προσαρμοσμένη ασφάλεια	1
T12	Εξάρτηση από κώδικα τρίτων	1
T13	Εκπαίδευση χρηστών	1

Πίνακας 6: Τεχνικοί παράγοντες της UCP:

3.4.3 Συντελεστής Περιβαλλοντικής Πολυπλοκότητας (ECF)

Η περιβαλλοντική πολυπλοκότητα εκτιμά τις επιπτώσεις στην παραγωγικότητα που έχουν 8 επιμέρους περιβαλλοντικοί παράγοντες σε μια εφαρμογή. Κάθε περιβαλλοντικός παράγοντας αξιολογείται και σταθμίζεται ανάλογα με την επίδρασή του. Του αποδίδεται μια τιμή μεταξύ 0 και 5. Η βαθμολογία 0 σημαίνει ότι ο περιβαλλοντικός παράγοντας είναι άσχετος με το έργο, 3 είναι μέσος όρος, 5 σημαίνει ότι έχει ισχυρή επιρροή.

Το βάρος κάθε παράγοντα πολλαπλασιάζεται με την πολυπλοκότητά του για να παράγει την τιμή του εξωγενή παράγοντα. Οι επιμέρους παράγοντες αθροίζονται για να

παραχθεί ο Συνολικός Παράγοντας Περιβαλλοντικής (Εξωγενούς) Πολυπλοκότητας. Δηλαδή:

$$\text{TotalFactor} = \sum_{i=1}^8 W_i * P_i$$

όπου με W_i εκφράζεται το βάρος και με P_i η πολυπλοκότητα

Ο συντελεστής περιβαλλοντικής πολυπλοκότητας προκύπτει από την εξίσωση:

$$\text{ECF} = 1.4 + (-0,03 * \text{TotalFactor})$$

Οι 8 περιβαλλοντικοί (εξωγενείς) παράγοντες είναι:

1. *Ο βαθμός εξοικείωσης με το αντικείμενο:* Πόση εμπειρία έχει η ομάδα ανάπτυξης σε αυτό τον τομέα; Το πεδίο ενδιαφέροντος είναι ο στόχος που το λογισμικό καλείται να εξυπηρετήσει, όχι τη γλώσσα προγραμματισμού (για παράδειγμα, για ένα ασφαλιστικό σύστημα αποζημίωσης γραμμένο στη java, το σημείο ενδιαφέροντος είναι η εμπειρία της ομάδας στον ασφαλιστικό χώρο αποζημίωσης - όχι ο βαθμός εξοικείωσης με τη java). Τα υψηλότερα επίπεδα εμπειρίας αποκτούν μεγαλύτερο αριθμό.
2. *Εμπειρία σχετικά με την εφαρμογή:* ο βαθμός εμπειρίας της ομάδας ανάπτυξης με τη συγκεκριμένη εφαρμογή, παράμετρος που θεωρείται σημαντική μόνο όταν απαιτούνται αλλαγές σε μία υπάρχουσα εφαρμογή. Οι υψηλότεροι αριθμοί αντιπροσωπεύουν περισσότερη εμπειρία, για μια νέα εφαρμογή, η εμπειρία όλων θα είναι 0.
3. *Εμπειρία στον αντικειμενοστραφή προγραμματισμό:* ο βαθμός εμπειρίας της ομάδας στον αντικειμενοστραφή προγραμματισμό είναι σαφώς πιο σημαντικός σε έργα με προσανατολισμό στο χρήστη, ή σχεδιασμένα με γνώμονα το χρήστη, καθώς συνήθως θα έχουν εγγενώς δομή αντικειμενοστραφή προγραμματισμού στην υλοποίηση. Υψηλότεροι αριθμοί αντιπροσωπεύουν μεγαλύτερη εμπειρία.
4. *Ικανότητες επικεφαλής αναλυτή:* Πόσο ενημερωμένος και ικανός είναι ο υπεύθυνος για τις απαιτήσεις; Οι άστοχες απαιτήσεις σύμφωνα με έρευνες είναι

ο κύριος παράγοντας αποτυχίας των έργων¹². Οι υψηλότεροι αριθμοί αντιπροσωπεύουν αυξημένες δεξιότητες και γνώσεις.

5. *Κίνητρο/υποκίνηση*: Πόσο υψηλό κίνητρο έχει η ομάδα ανάπτυξης; Οι υψηλότεροι αριθμοί αντιπροσωπεύουν μεγαλύτερο κίνητρο για την ομάδα ανάπτυξης.

6. *Σταθερότητα απαιτήσεων*: Οι αλλαγές στις απαιτήσεις μπορούν να προκαλέσουν αύξηση της εργασίας. Ο τρόπος για να αποφευχθεί αυτό είναι με τον προγραμματισμό της αλλαγής και την καθιέρωση ενός συστήματος χρονοπρογραμματισμού για τη διαχείριση αυτών των αλλαγών. Οι υψηλότεροι αριθμοί αντιπροσωπεύουν περισσότερες αλλαγές (ή λιγότερο αποτελεσματικό σύστημα για τη διαχείριση της αλλαγής).

7. *Προσωπικό μερικής απασχόλησης*: οι υψηλότεροι αριθμοί αντανακλούν τα μέλη της ομάδας που εργάζονται με μερική απασχόληση, εξωτερικούς συμβούλους και προγραμματιστές που χωρίζουν τον χρόνο τους σε διάφορα έργα. Η αλλαγή περιβάλλοντος και άλλοι άυλοι παράγοντες καθιστούν τα μέλη αυτά λιγότερο αποτελεσματικά.

8. *Δύσκολη γλώσσα προγραμματισμού*: οι πιο δύσκολες γλώσσες αντιπροσωπεύουν υψηλότερους αριθμούς, αλλά λαμβάνεται υπόψιν και σε ποιες γλώσσες έχει μεγαλύτερη εμπειρία η ομάδα ανάπτυξης.

¹² Ο όμιλος Standish αναφέρει ότι το 40% έως το 60% των ελαττωμάτων προέρχονται από άστοχες απαιτήσεις, σύμφωνα με τον Blain T. (2007), *Software Cost Estimation With Use Case Points – Environmental Factors*

Στον παρακάτω πίνακα βλέπουμε τους περιβαλλοντικούς παράγοντες με τα βάρη τους:

α/α	Περιγραφή	Βάρος
E1	Βαθμός εξοικείωσης με το αντικείμενο	1,5
E2	Εμπειρία σχετικά με την εφαρμογή	0,5
E3	Εμπειρία στον αντικειμενοστραφή προγραμματισμό	1
E4	Ικανότητες επικεφαλής αναλυτή	0,5
E5	Κίνητρο/υποκίνηση	1
E6	Σταθερότητα απαιτήσεων	2
E7	Προσωπικό μερικής απασχόλησης	-1
E8	Δύσκολη γλώσσα προγραμματισμού	2

Πίνακας 7: Περιβαλλοντικοί παράγοντες της UCP

3.4.4 Μη Σταθμισμένα Use Cases και Actors (UUCP)

3.4.4.1 Πλήθος και πολυπλοκότητα των use cases

Μέχρι τώρα, λήφθηκαν υπόψη οι τεχνικοί και οι περιβαλλοντικοί παράγοντες, αλλά δεν είναι οι μόνοι παράγοντες που επηρεάζουν την ανάλυση στα πλαίσια της UCP. Κάθε use case συμβάλλει στην εκτίμηση του κόστους του έργου και use cases διαφορετικής πολυπλοκότητας έχουν διαφορετική επίδραση στην εκτίμηση του κόστους.

Κατά τη δημιουργία μιας εκτίμησης κόστους του έργου, ξεκινάμε με την αναγνώριση και την κατηγοριοποίηση όλων των use cases που θα υποστηριχθούν στο λογισμικό. Τα use cases ταξινομούνται ανάλογα με την πολυπλοκότητά τους ως απλά, μεσαία και πολύπλοκα. Αυτή η ταξινόμηση περιγράφει το use case, όχι την υποτιθέμενη πολυπλοκότητα της εφαρμογής στο σύνολό της. Το σύστημα ταξινόμησης βασίζεται στον αριθμό των συναλλαγών στη χρήση. Μια συναλλαγή είναι μια ανταλλαγή, όπου πρώτα ο χρήστης κάνει κάτι και στη συνέχεια το σύστημα αποκρίνεται.

- Απλό use case - έως 3 συναλλαγές.
- Μέσο use case - 4 έως 7 συναλλαγές.
- Σύνθετο use case - περισσότερες από 7 συναλλαγές

Για κάθε use case, δίνεται ένας αριθμός σημείων χρήσης (Kress, Hummel, Huq, 2014):

- Απλό use case: 5 μονάδες.
- Μέσο use case: 10 μονάδες.

- Σύνθετο use case: 15 μονάδες.

Τότε αθροίζουμε το σύνολο των μονάδων και το αποτέλεσμα είναι τα μη σταθμισμένα σημεία περιπτώσεων χρήσης (Unadjusted Use Case Weight - UUCW)¹³.

Το UUCW βασίζεται στο συνολικό αριθμό δραστηριοτήτων (ή βημάτων) που περιλαμβάνονται σε όλα τα σενάρια των use cases και υπολογίζεται μετρώντας τον αριθμό των use cases σε κάθε κατηγορία, πολλαπλασιάζοντάς τα με το βάρος και προσθέτοντας όλα μαζί:

$$UUCW = \sum_{i=1}^3 W_i * P_i$$

όπου με W_i εκφράζεται το βάρος και με P_i η πολυπλοκότητα

Κατηγορίες Use Cases	Περιγραφή	Βάρος
Απλό	Απλό user interface, προσπελαύνει μόνο 1 database entity, η κανονική ροή ολοκλήρωσης περιλαμβάνει έως 3 βήματα και η υλοποίησή του αφορά έως 5 classes.	5
Μεσαίο	Πιο σύνθετο user interface, προσπελαύνει 2 ή περισσότερες database entity, η κανονική ροή ολοκλήρωσης περιλαμβάνει έως 4-7 βήματα και η υλοποίησή του αφορά 5-10 classes.	10
Πολύπλοκο	Πολύπλοκο user interface ή επεξεργασία, προσπελαύνει 3 ή περισσότερες database entity, η κανονική ροή ολοκλήρωσης περιλαμβάνει περισσότερα από 7 βήματα και η υλοποίησή του αφορά περισσότερες από 10 classes.	15

Πίνακας 8: Κατηγορίες Use Cases και UUCW

¹³ Στη βιβλιογραφία συναντώνται επίσης και σαν Μη Σταθμισμένα Σημεία Περιπτώσεων Χρήσης (UUCP) και οι Παράγοντες (Actors) σαν AW και όχι UAW όπως στην παρούσα εργασία.

3.4.4.2 Πλήθος και Πολυπλοκότητα των Actors

Αξίζει να σημειωθεί ότι με τον όρο "χρήστης" δεν αναφερόμαστε μόνο σε ανθρώπους. Ενδεχομένως άλλα συστήματα να είναι επίσης χρήστες, αφού το λογισμικό συχνά «επικοινωνεί» με άλλο λογισμικό, ειδικά καθώς μεταβαίνουμε σε όλο και περισσότερες λύσεις που βασίζονται σε αρχιτεκτονικές προσανατολισμένες στις υπηρεσίες και οι ανάγκες επικοινωνίας καλύπτονται μέσω διαλειτουργικότητας. Η ανάλυση των actors αντιστοιχεί στους διάφορους τύπους χρηστών που αναμένεται να αλληλεπιδράσουν με το σύστημα.

Πρέπει επομένως να εντοπιστούν όλοι οι άνθρωποι και τα συστήματα που αλληλεπιδρούν με το λογισμικό που εκτιμάται. Στη συνέχεια θα ταξινομηθούν οι "actors" ως απλοί, μέσοι ή πολύπλοκοι:

Απλοί actors είναι άλλα συστήματα που επικοινωνούν με το λογισμικό σας μέσω ενός προκαθορισμένου API¹⁴. Ένα API θα μπορούσε να λειτουργεί μέσω ενός dll ή ως REST, SOAP ή οποιουδήποτε API web service ή κλήσης απομακρυσμένης διαδικασίας (RPC). Το βασικό στοιχείο είναι ότι υπάρχει αλληλεπίδραση με το λογισμικό μέσω ενός συγκεκριμένου, καλά καθορισμένου μηχανισμού.

Μέσοι actors μπορούν είτε να είναι άνθρωποι που αλληλεπιδρούν με ένα καλά καθορισμένο πρωτόκολλο, είτε θα μπορούσαν να είναι συστήματα που αλληλεπιδρούν μέσω ενός πιο περίπλοκου ή ευέλικτου API. Όταν οι χρήστες του ανθρώπινου δυναμικού είναι περιορισμένοι σε ένα πρωτόκολλο, υπάρχει πολύ μικρή διαφορά μεταξύ των αλληλεπιδράσεών τους και των αλληλεπιδράσεων των συστημάτων μέσω ενός καθορισμένου API. Αυτό το πρωτόκολλο μπορεί να είναι μια διασύνδεση γραμμής εντολών, χρησιμοποιώντας το ftp για απόκτηση αρχείων ή χρησιμοποιώντας html για είσοδο και έξοδο. Βασικό στοιχείο είναι μια σαφώς καθορισμένη σημασιολογία συμπεριφοράς. Αυτό διαφέρει από την αλληλεπίδραση συστήματος με σύστημα, επειδή οι ανθρώπινες ενέργειες είναι πιο δύσκολο να προβλεφθούν από εκείνες συστημάτων λογισμικού. Αν επιτρέπεται σε άλλα συστήματα να αλληλεπιδρούν άμεσα με τα αποθηκευτικά μέσα που χρησιμοποιεί το λογισμικό (π.χ. άμεση ανάγνωση και τροποποίηση της βάσης δεδομένων), τότε θεωρούνται επίσης μέσοι παράγοντες. Χρειάζεται περισσότερη προσπάθεια για να διασφαλιστεί η ποιότητα όταν επιτρέπεται

¹⁴ Διεπαφή Προγραμματισμού Εφαρμογών

απεριόριστη πρόσβαση σε αποθηκευτικούς χώρους δεδομένων από ό, τι θα ήταν να περιοριστεί η πρόσβαση σε προβλέψιμες λειτουργίες.

Ο αρχικός ορισμός *πολύπλοκων actors* αναφέρεται σε χρήστες που αλληλεπιδρούν με το λογισμικό μέσω γραφικής διεπαφής χρήστη. Η ίδια ταξινόμηση θα πρέπει να ισχύει και για χρήστες που αλληλεπιδρούν με το σύστημα με απρόβλεπτες μεθόδους. Αν οι χρήστες αλληλεπιδρούν μέσω μιας διεπαφής "rich html", μπορεί να ταξινομηθούν επίσης ως πολύπλοκοι actors.

Σε κάθε ταξινόμηση, προστίθεται ο παρακάτω αριθμός πόντων για κάθε actor:

- Απλός Actor - 1 πόντος
- Μέσος Actor - 2 πόντοι
- Σύνθετος Actor - 3 πόντοι

Αυτό αντιπροσωπεύει την πολυπλοκότητα του παράγοντα. Δηλαδή:

$$UAW = \sum_{i=1}^3 W_i * P_i$$

όπου με W_i εκφράζεται το βάρος και με P_i η πολυπλοκότητα

Κατηγορίες Use Cases	Περιγραφή	Βάρος
Απλός	Ο Actor αντιστοιχεί σε σύστημα με καθορισμένο API	1
Μεσαίος	Ο Actor αντιστοιχεί σε σύστημα που αλληλεπιδρά μέσω πρωτοκόλλου (π.χ. TCP/IP)	2
Πολύπλοκος	Ο Actor αντιστοιχεί σε πρόσωπο που αλληλεπιδρά μέσω User Interface	3

Πίνακας 9: Κατηγορίες Use Cases και Actors

Το UUCP είναι το άθροισμα των Μη Σταθμισμένων Use Cases και των Μη Σταθμισμένου Βάρους Παραγόντων:

$$UUCP = UUCW + UAW$$

3.4.5 Συντελεστής παραγωγικότητας

Τέλος, απαραίτητος είναι και ο παράγοντας παραγωγικότητας, που είναι ο λόγος του αριθμού των ανθρωποωρών ανά use case. Βασίζεται σε δεδομένα από προηγούμενα έργα, ενώ αν δεν έχουν συλλεχθεί ιστορικά στοιχεία, προτείνεται ένας αριθμός μεταξύ 15 και 30. Τυπική τιμή σύμφωνα με τον Gustav Karner είναι το 20.

3.4.6 Υπολογισμός του αποτελέσματος της τελικής εξίσωσης UCP

$$UCP = TCF * ECF * UUCP$$

$$\text{Ανθρωποπροσπάθεια} = UCP * PF$$

3.4.7 Πλεονεκτήματα – μειονεκτήματα της μεθόδου

Τα βασικά πλεονεκτήματα της μεθόδου είναι ότι αποτελεί σήμερα μία διεθνώς αποδεκτή και ευρέως διαδεδομένη τεχνική για την καταγραφή των επιχειρησιακών διαδικασιών και των απαιτήσεων μιας εφαρμογής λογισμικού και ότι είναι εύκολη στη χρήση και δεν απαιτεί πρόσθετη ανάλυση. Βασίζεται σε use cases και μπορεί να εκτιμηθεί από πολύ νωρίς στον κύκλο ζωής του έργου, ενώ είναι ανεξάρτητη από το μέγεθος, την ικανότητα και την εμπειρία της ομάδας υλοποίησης. Τέλος οι εκτιμήσεις που βασίζονται στο UCP βρίσκονται πολύ κοντά στην πραγματικότητα όταν πραγματοποιούνται από έμπειρα στελέχη. Τα use cases χρησιμοποιούνται ευρέως ως μέθοδος για την περιγραφή των απαιτήσεων και σε τέτοιες περιπτώσεις, η UCP είναι η καλύτερη κατάλληλη τεχνική εκτίμησης κόστους¹⁵.

Υπάρχουν όμως και ορισμένα μειονεκτήματα της μεθόδου, και ακολουθούν όπως περιγράφονται από τους Καπόπουλο και Σπηλιωτόπουλο (2018). Καταρχάς χρησιμοποιείται μόνο όταν οι απαιτήσεις έχουν γραφτεί με τη μορφή use cases. Ακόμα, εξαρτάται από περιπτώσεις ορθής χρήσης γραπτού λόγου και αν τα use cases δεν είναι επαρκώς ή ομοιόμορφα δομημένα, το UCP που προκύπτει ενδεχομένως να μην είναι ακριβές. Ένα άλλο μειονέκτημα είναι ότι τεχνικοί και περιβαλλοντικοί παράγοντες έχουν μεγάλη επίδραση στην UCP. Πρέπει να ληφθεί σημαντική μέριμνα κατά την ανάθεση τιμών σε TCF και ECF. Σε γενικές γραμμές θα μπορούσε να υποστηριχθεί ότι είναι μία πολύ χρήσιμη μέθοδος για την αρχική εκτίμηση του συνολικού μεγέθους του έργου, και όχι τόσο χρήσιμη όταν εφαρμόζεται σε επαναληπτικές εργασίες.

¹⁵ Καπόπουλος Δ., Σπηλιωτόπουλος, Γ. (2018). Σημειώσεις μαθήματος «Διαχείριση Έργων» 25ης σειράς ΕΣΔΔΑ, Τμήμα Ψηφιακής Πολιτικής

4. Λογισμικό για τον υπολογισμό των τιμών κάθε μεθόδου

Οι μέθοδοι αυτοματοποίησης της μέτρησης μειώνουν την αντίσταση που προκύπτει ενδεχομένως από απροθυμία να επενδύσουν οι φορείς σε χειροκίνητη λειτουργική αποτύπωση και επιτρέπουν την ευρεία υιοθέτηση στη Δημόσια Διοίκηση. Τα αυτοματοποιημένα εργαλεία μέτρησης σημείων λειτουργίας βοηθούν τους οργανισμούς να αυξήσουν την ποιότητα, να μειώσουν τον κίνδυνο και να περιορίσουν την προσπάθεια που απαιτείται για τη διαχείριση των έργων. Αυτά τα εργαλεία συμβάλλουν στη μείωση του χρόνου και των πόρων που απαιτούνται για την εκτίμηση του κόστους ανάπτυξης λογισμικού.

Όσον αφορά τη μέθοδο UCP, σύμφωνα με τους Kusumoto, Matukawa, Inoue (2004), υπάρχουν ορισμένα εργαλεία που χρησιμοποιούνται για την υποστήριξη του υπολογισμού του συνόλου των use cases, όπως για παράδειγμα το «Enterprise Architect» και το «Estimate Easy Use». Τα εργαλεία εξαγάγουν τα use cases και τους actors από διαγράμματα σχετικά με τα use cases. Εισάγοντας οι ενδιαφερόμενοι τις τιμές σχετικά με την πολυπλοκότητα των use cases και των actors, και των τεχνικών και περιβαλλοντικών συντελεστών, υπολογίζονται η τιμή του UCP καθώς και η εκτιμώμενη ανθρωποπροσπάθεια.

Αναφορικά με τη χρήση των λειτουργικών σημείων, οι πρόοδοι στα εργαλεία λογισμικού καθιστούν δυνατή την αυτόματη μέτρηση των σημείων λειτουργίας μέσω της χρήσης της δομικής και αρχιτεκτονικής ανάλυσης του πηγαίου κώδικα.

Όσον αφορά την COCOMO, υπάρχει η δυνατότητα αυτόματου υπολογισμού σε ιστοσελίδες αξιόπιστων οργανισμών (π.χ. NASA)¹⁶ να εισάγει ο ενδιαφερόμενος τις τιμές και του παρουσιαστεί σαν έξοδος το αποτέλεσμα. Το ίδιο ισχύει και για την COCOMO II (π.χ. USC Center for Systems and Software Engineering)¹⁷.

¹⁶ <https://strs.grc.nasa.gov/repository/forms/cocomo-calculation/>

¹⁷ <http://csse.usc.edu/tools/cocomoii.php>

5. Πρόταση χρήσης της κατάλληλότερης μεθόδου για τη Δημόσια Διοίκηση

Λαμβάνοντας υπόψιν να παραπάνω, θα πρέπει αρχικά να επισημανθεί ότι οι ιδιαιτερότητες κάθε έργου ΤΠΕ και του λογισμικού που απαιτείται για την υλοποίησή του καθορίζουν εν μέρει την πιο κατάλληλη μέθοδο με την οποία θα εκτιμηθεί το κόστος του. Θα ήταν εξαιρετικά χρήσιμη όμως η υιοθέτηση ενός συγκεκριμένου μοντέλου εκτίμησης κόστους λογισμικού, δεδομένης και της τάσης να γίνονται πολλά έργα λόγω του επείγοντος χαρακτήρα τους με απευθείας ανάθεση σε εταιρίες λογισμικού ή μέσα από «συμφωνίες-πλαίσιο».

Τα οφέλη που μπορεί να έχει το δημόσιο από την υιοθέτηση τέτοιων μεθόδων είναι πολλαπλά και περιλαμβάνουν την ενίσχυση της διαπραγματευτικής του θέσης έναντι των εταιριών υλοποίησης λογισμικού. Αξίζει δε να σημειωθεί ότι δεν είναι μόνο το ελάχιστο δυνατό κόστος το ζητούμενο, αλλά πρέπει αν λαμβάνεται υπόψιν και το επίπεδο ποιότητας των ηλεκτρονικών υπηρεσιών προς τους εσωτερικούς χρήστες (πολίτες, επιχειρήσεις), αλλά και ο χρόνος υλοποίησης. Ειδικά στην πρόσφατη περίοδο των προγραμμάτων δημοσιονομικής προσαρμογής λόγω μνημονιακών υποχρεώσεων, όπου έπρεπε να υλοποιηθούν τα έργα ΤΠΕ που αποτελούσαν δεσμεύσεις της χώρας σε ορισμένες προθεσμίες, η ταχύτητα υλοποίησης των έργων διαδραματίζει σημαντικό ρόλο. Θεωρείται σημαντικό να γίνεται πρώτα μία ικανοποιητική ανάλυση των προδιαγραφών και καταγραφή των απαιτήσεων, προτού συζητηθεί το κόστος, ενώ θα πρέπει να αποφεύγεται κατά το δυνατόν η αλλαγή των απαιτήσεων στην πορεία των έργων.

Το ζητούμενο οπότε είναι να επιλεγεί το πιο ικανοποιητικό μοντέλο, καθώς όταν αναφερόμαστε στην εκτίμηση έργων λογισμικού κανένα μοντέλο δε μπορεί να μας δώσει εγγυημένα σωστές εκτιμήσεις. Ήδη από ευρήματα έρευνας το 2002, σύμφωνα με τους Bente και Dag (2002)¹⁸, η UCP έδωσε μια εκτίμηση που ήταν πιο κοντά στην πραγματική προσπάθεια που δαπανήθηκε για την εφαρμογή του εξεταζόμενου συστήματος από ό, τι οι περισσότερες εκτιμήσεις 37 εμπειρών κατασκευαστών λογισμικού. Ο Scott Sehlhorst (2007), υποστηρίζει ότι η παραγωγή μιας εκτίμησης με τη μέθοδο UCP θα έχει ως αποτέλεσμα μεγαλύτερη πιθανότητα επιτυχίας του έργου από οποιαδήποτε άλλη εκτίμηση που μπορεί να εκτελεστεί στο ίδιο στάδιο του έργου με το ίδιο ποσό δεδομένων,

¹⁸ Bente A., Dag I. (2002). "Towards an inspection technique for use case models. In Proceedings of the 14th international conference on Software engineering and knowledge engineering"

ενώ η πράξη υπολογισμού του UCP στην αρχή ενός έργου θα βοηθήσει στην εύρεση και αποφυγή δαπανηρών λαθών στη συνέχεια του έργου. Αλλά και πρόσφατες έρευνες, όπως των Che Ani, Basri, Sarlan (2017), καταλήγουν στο συμπέρασμα ότι μέσα από την εξέλιξη της μεθόδου, οι τεχνικές που βασίζονται στη UCP εξακολουθούν να είναι αποτελεσματικές και στις σημερινές προσπάθειες εκτίμησης κόστους λογισμικού. Καταδεικνύεται επίσης η ανάγκη συστηματικής υποστήριξη εργαλείων για να εξασφαλιστεί η αξιοπιστία της μεθόδου επιτυγχάνοντας πιο ακριβή αποτελέσματα. Επίσης, η δυνατότητα αυτοματοποίησης του υπολογισμού των τιμών της μεθόδου μέσω λογισμικού που αναφέρθηκε σε προηγούμενο κεφάλαιο είναι εξαιρετικά σημαντική, αφού θα μειωθεί με αυτό τον τρόπο ο χρόνος εκτίμησης σημαντικά.

Με βάση τη σύγκριση των πλεονεκτημάτων και των μειονεκτημάτων των μεθόδων που προαναφέρθηκαν, τις απόψεις των ερευνητών που σχολιάστηκαν και το περιβάλλον ανάθεσης έργων ανάπτυξης λογισμικού της Δημόσιας Διοίκησης, η πιο κατάλληλη μέθοδος προς χρήση από τους φορείς της Δημόσιας Διοίκησης είναι η UCP.

6. Μελέτη Περίπτωσης: Εκτίμηση με βάση υπάρχουσα σύμβαση

6.1 Περιγραφή περίπτωσης

Με βάση τη σύμβαση για το έργο «Διαχείριση αλλαγών Ολοκληρωμένου Πληροφοριακού Συστήματος Δημοσιονομικής Πολιτικής (ΟΠΣ-ΔΠ)»¹⁹ και τη χρήση της προτεινόμενης μεθόδου προς τους φορείς της Δημόσιας Διοίκησης, θα γίνει προσπάθεια στα πλαίσια της παρούσας εργασίας να εκτιμηθεί κατά πόσο το συμβατικό τίμημα ήταν ρεαλιστικό.

Σύμφωνα με την παράγραφο Α.1, με τη μορφή των αιτημάτων ζητήθηκαν από το Υπουργείο Οικονομικών παρεμβάσεις/υλοποιήσεις σε τέσσερις κατηγορίες δράσεων:

1. Λογιστική μεταρρύθμιση στην Κεντρική Κυβέρνηση
2. Λογιστική μεταρρύθμιση και μεταφορά αρμοδιοτήτων από τις ΥΔΕ στις ΓΔΟΥ των Υπουργείων
3. Διεπαφές με άλλα πληροφοριακά συστήματα ή/και άλλους φορείς
4. Πρόσθετες τροποποιήσεις υφιστάμενων λειτουργιών του ΟΠΣ-ΔΠ λόγω των νέων αλλαγών στις επιχειρησιακές διαδικασίες

Η εκτίμηση κόστους στα πλαίσια της παρούσας εργασίας θα αφορά την πρώτη παρέμβαση, τη «Λογιστική μεταρρύθμιση στην Κεντρική Κυβέρνηση», όπου σύμφωνα με τη σύμβαση οι αναγκαίες προσαρμογές θα μπορούσαν να διακριθούν:

- i. σε απαιτήσεις που αφορούν την εισαγωγή της νέας ταξινόμησης στο ΟΠΣ-ΔΠ με βάση τα διεθνή πρότυπα και ειδικότερα:
 - αντικατάσταση της υφιστάμενης Οικονομικής Ταξινόμησης με πλήρη ευθυγράμμιση του Προϋπολογισμού και της Λογιστικής στη βάση Διεθνών Προτύπων (ESA2010, GFSM και IPSAS)
 - αντικατάσταση της υπάρχουσας Διοικητικής Ταξινόμησης του Προϋπολογισμού
 - ένταξη στο σύστημα της Λειτουργικής Ταξινόμησης σύμφωνα με το πρότυπο COFOG
- ii. σε απαιτήσεις που σχετίζονται με την αλλαγή λογιστικών διαδικασιών και κανόνων
- iii. παράλληλη λειτουργία και εκτέλεση προϋπολογισμών των ετών 2017 και 2018

¹⁹ Η σύμβαση υπογράφηκε μεταξύ του Υπουργείου Οικονομικών και της ένωσης εταιριών που ανέλαβε της υλοποίηση του έργου («Ανάδοχος», όπως αναφέρεται και στη σύμβαση) την 30-01-2018

- iv. διατήρηση της ιστορικότητας των στοιχείων στο BI και στο ERP

6.2 Εκτίμηση Ανθρωποπροσπάθειας με τη μέθοδο UCP

Η αρχική εκτίμηση της ανθρωποπροσπάθειας για τη «Λογιστική Μεταρρύθμιση στην Κεντρική Κυβέρνηση» όπως αναφερόταν στη σύμβαση ήταν 120 ανθρωπομήνες.

Με βάση τη μέθοδο UCP, υπολογίστηκε από τον «Ανάδοχο» ότι χρειάζονται 181,48 ανθρωπομήνες για τη συγκεκριμένη παρέμβαση, εκ των οποίων οι 108,48 υπολογίζονται με βάση τη μεθοδολογία UCP και οι 73,00 εργασίες που δεν υπολογίζονται βάσει της συγκεκριμένης μεθοδολογίας. Με την παραδοχή ότι ο κάθε ανθρωπομήνας κοστίζει 4.000€, το συνολικό κόστος ήταν 725.920,00€.

Δύο διαφοροποιήσεις σε σχέση με τις εκτιμήσεις του «Ανάδοchu», στα πλαίσια του επαναυπολογισμού των συντελεστών που καθορίζουν την τελική τιμή των UCP είναι ότι οι ανθρωπομήνες υπολογίζουμε ότι έχουν 22 ημέρες και όχι 20, όπως έγινε η αρχική πρόβλεψη και ότι ο PF είναι 20 και όχι 22, όπως ήταν η πραγματική εκτίμηση του συντελεστή που έγινε από τον Ανάδοχο.

Ακολουθούν οι πίνακες με την εκτίμηση της ανθρωποπροσπάθειας και του κόστους μέσω της μεθοδολογίας UCP, όπως επαναυπολογίστηκαν στα πλαίσια της παρούσας εργασίας:

Υπολογισμοί συντελεστών²⁰		
TCF	Συντελεστής τεχνικής πολυπλοκότητας	0,96
ECF	Συντελεστής εξωγενούς πολυπλοκότητας	0,64
UUCW	Μη-σταθμισμένο βάρος περιπτώσεων χρήσης (Use Cases)	690,00
UAW	Μη-σταθμισμένο βάρος παραγόντων (Actors)	32,00
Υπολογισμός βάρους περιπτώσεων χρήσης (Use Case Points)		
UCP	Συνολικό βάρος περιπτώσεων χρήσης (Use Case Points)	443,60
Υπολογισμός απαιτούμενης ανθρωποπροσπάθειας		
PF	Συντελεστής Παραγωγικότητας (ανθρωποώρες ανά μονάδα βάρους περιπτώσεων χρήσης)	20,00
	Συνολικές ανθρωποώρες	8.872,00
	Συνολικές ανθρωποημέρες	1.109,00
	Συνολικοί ανθρωπομήνες	50,40

Πίνακας 10: Εκτίμηση Ανθρωποπροσπάθειας με τη UCP

²⁰ Στους πίνακες που ακολουθούν παρουσιάζεται αναλυτικά ο τρόπος υπολογισμού των τιμών των επιμέρους παραγόντων των συντελεστών

Παρατηρούμε ότι υπάρχει διαφορά στην εκτίμηση του κόστους, αφού ο Ανάδοχος υπολόγισε ότι χρειάζονται 108,48 ανθρωπομήνες για την υλοποίηση της παρέμβασης με βάση τη μεθοδολογία UCP, ενώ στα πλαίσια της εργασίας υπολογίστηκε 50,40 ανθρωπομήνες. Ακολουθεί ο υπολογισμός του κόστους με βάση τα στοιχεία της δεύτερης εκτίμησης:

Εργασία	Ανθρωπομήνες	Περιγραφή
Λογιστική Αναμόρφωση		
Ανάλυση-υλοποίηση-έλεγχος με χρήση της UCP	50,01	Αφορά εργασίες στις λειτουργικές περιοχές που αξιολογούνται με UCP
Υπηρεσίες προετοιμασίας και μετάβασης σε παραγωγική λειτουργία	4,00	Integration testing, data migration, κλπ.
Καθορισμός εφικτότητας, αντικειμένου, χρονοδιαγράμματος, κ.α. (Scoring)	2,00	Προηγείται της υλοποίησης
Υποστήριξη UAT - εκπαίδευση εκπαιδευτών - τεκμηρίωση	3,00	
Υποστήριξη συστημάτων SAP	3,00	Υποστήριξη ανάπτυξης SAP, παραμετροποίηση διεπαφών, Dev/Test support, κλπ.
Υποστήριξη Πιλοτικής και Παραγωγικής λειτουργίας	5,00	Αφορά υπηρεσίες on-site υποστήριξης
Project Management	8,00	
Σύνολο (Ανθρωπομήνες)	75,01	
Σύνολο (Κόστος Λογισμικού)	300.040,00	

Πίνακας 11: Υπολογισμός Κόστους στη μελέτη περίπτωσης

Στον πίνακα που ακολουθεί παρουσιάζεται ο τρόπος υπολογισμού της τιμής του TCF (=0,96) του πίνακα 10.

Τεχνικοί παράγοντες		Συντελεστής Βαρύτητας	Σχετική τιμή (Από 0 έως 5)	Περιγραφή
1	Απαιτούμενο Σύστημα Διανομής	2	3	Η αρχιτεκτονική της λύσης μπορεί να είναι συγκεντρωτική ή παραμετροποιημένη για κάθε χρήστη (single-tenant), ή μπορεί να διανεμηθεί ή να είναι προσαρμοσμένη σε ομάδες πελατών.
2	Σημασία του Χρόνου Απόκρισης	1	4	Η ταχύτητα ανταπόκρισης των χρηστών είναι σημαντικός παράγοντας.
3	Αποτελεσματικότητα Τελικού Χρήστη	1	4	Η εφαρμογή αναπτύσσεται για να βελτιστοποιεί την αποτελεσματικότητα των χρηστών;.
4	Απαιτήση για Σύνθετη Εσωτερική Εργασία	1	3	Απαιτείται πολύ δύσκολη αλγοριθμική εργασία και δοκιμή;
5	Εστίαση στη Χρήση Επαναχρησιμοποιούμενου Κώδικα	1	3	Είναι μέσα στους στόχους η επαναχρησιμοποίηση του κύριου κώδικα;
6	Ευκολία Εγκατάστασης	0,5	0	Η ευκολία εγκατάστασης για τους τελικούς χρήστες αποτελεί βασικό παράγοντα;
7	Ευχρηστία	0,5	4	Η ευκολία χρήσης αποτελεί ένα κύριο κριτήριο αποδοχής του έργου;
8	Υποστήριξη από Πολλαπλές Πλατφόρμες	2	0	Απαιτείται υποστήριξη από πολλαπλές πλατφόρμες;
9	Ευκολία Αλλαγών	1	4	Πόσο μεγαλύτερη ανάγκη για αλλαγή/προσαρμογή απαιτείται στο μέλλον;
10	Ταυτόχρονη Επεξεργασία	1	3	Υπάρχει η ανάγκη αντιμετώπισης κλειδώματος της βάσης δεδομένων και άλλων θεμάτων που σχετίζονται με υψηλές ανάγκες για συγχρονισμό των δεδομένων;
11	Προσαρμοσμένη Ασφάλεια	1	2	Μπορούν οι υπάρχουσες λύσεις ασφάλειας να αξιοποιηθούν ή πρέπει να αναπτυχθεί προσαρμοσμένος κώδικας;
12	Εξάρτηση από Κώδικα Τρίτων	1	5	Η εφαρμογή απαιτεί τη χρήση ελέγχων ή βιβλιοθηκών από τρίτους;
13	Εκπαίδευση Χρηστών	1	0	Εδώ εξετάζεται ο απαιτούμενος χρόνος εκπαίδευσης των χρηστών; Είναι η εφαρμογή πολύπλοκη ή υποστηρίζει πολύπλοκες δραστηριότητες;
Τιμή TCF			0,96	

Πίνακας 12: Υπολογισμός Συντελεστή Τεχνικής Πολυπλοκότητας

Στον Πίνακα 13 παρουσιάζεται ο τρόπος υπολογισμού της τιμής του ECF (=0,64) του πίνακα 10.

	Περιβαλλοντικοί παράγοντες	Συντελεστής Βαρύτητας	Σχετική τιμή (Από 0 έως 5)	Περιγραφή
1	Βαθμός Εξοικείωσης με το έργο	1,5	5	Πόση εμπειρία έχει η ομάδα ανάπτυξης σε αυτό τον τομέα;
2	Εμπειρία Σχετικά με την Εφαρμογή	0,5	5	Ποιος είναι ο βαθμός εμπειρίας της ομάδας ανάπτυξης με τη συγκεκριμένη εφαρμογή;
3	Εμπειρία στον Αντικειμενοστραφή Προγραμματισμό	1	4	Ποιος είναι ο βαθμός εμπειρίας της ομάδας στον αντικειμενοστραφή προγραμματισμό;
4	Ικανότητα Επικεφαλής Αναλυτή	0,5	5	Πόσο ενημερωμένος και ικανός είναι ο υπεύθυνος για τις απαιτήσεις του έργου;
5	Κίνητρο/Υποκίνηση	1	5	Πόσο υψηλό κίνητρο έχει η ομάδα ανάπτυξης;
6	Σταθερότητα Απαιτήσεων	2	3	Πόσο συχνές προβλέπονται οι αλλαγές στις απαιτήσεις;
7	Προσωπικό Μερικής Απασχόλησης	-1	1	Οι υψηλότεροι αριθμοί αντανακλούν τα μέλη της ομάδας που εργάζονται με μερική απασχόληση, εξωτερικούς συμβούλους και προγραμματιστές που χωρίζουν τον χρόνο τους σε διάφορα έργα.
8	Δύσκολη Γλώσσα Προγραμματισμού	-1	1	Οι πιο δύσκολες γλώσσες αντιπροσωπεύουν υψηλότερους αριθμούς, αλλά λαμβάνεται υπόψιν και σε ποιες γλώσσες έχει μεγαλύτερη εμπειρία η ομάδα ανάπτυξης.
Τιμή ECF			0,64	

Πίνακας 13: Υπολογισμός Συντελεστή Περιβαλλοντικής Πολυπλοκότητας

Στη συνέχεια, στους πίνακες 14 και 15 αντίστοιχα, παρουσιάζεται ο τρόπος υπολογισμού της τιμής UUCW (=690) και οι τιμές τους βάρους για κάθε use case.

a/a	Κατηγορίες Use Cases	Βάρος	Use Cases / κατηγορία	Περιγραφή
1	Απλό	5	14	Απλό user interface, προσπελαίνει μόνο 1 database entity, η κανονική ροή ολοκλήρωσης περιλαμβάνει έως 3 βήματα και η υλοποίησή του αφορά έως 5 classes.
2	Μεσαίο	10	29	Πιο σύνθετο user interface, προσπελαίνει 2 ή περισσότερες database entity, η κανονική ροή ολοκλήρωσης περιλαμβάνει έως 4 - 7 βήματα και η υλοποίησή του αφορά 5 - 10 classes.
3	Πολύπλοκο	15	22	Πολύπλοκο user interface ή επεξεργασία, προσπελαίνει 3 ή περισσότερες database entity, η κανονική ροή ολοκλήρωσης περιλαμβάνει περισσότερα από 7 βήματα και η υλοποίησή του αφορά περισσότερες από 10 classes.
UUCW			690	

Πίνακας 14: Υπολογισμός Μη-Σταθμισμένων Use Cases

Use Cases	Βάρος	Όνομα Use Case ²¹	
1	Πολύπλοκο	15	Καθορισμός λογαριασμών εσόδων, εξόδων, απαιτήσεων και υποχρεώσεων της διεπαφής Δ23 βάσει της λογικής του νέου σχεδίου
2	Πολύπλοκο	15	Λογιστικοποίηση της διεπαφής Δ23 με βάση τα νέα άρθρα
3	Μεσαίο	10	Αλλαγές αρχείων για εθνικό/συγχρηματοδοτούμενο
4	Μεσαίο	10	Αλλαγές αρχείων στις διεπαφές Taxis_1&6, ΕΛΔΑΠ, ΔΔΕ και ΤτΕ_6 βάσει της νέας οικονομικής & διοικητικής ταξινόμησης
5	Πολύπλοκο	15	Παράλληλη λειτουργία υφιστάμενης και νέας οικονομικής / διοικητικής ταξινόμησης στις διεπαφές Taxis_1&6, ΕΛΔΑΠ, ΔΔΕ και ΤτΕ_6 για το έτος 2019
6	Πολύπλοκο	15	Νέοι κανόνες ελέγχου στις διεπαφές Taxis_1&6, ΕΛΔΑΠ, ΔΔΕ και ΤτΕ_6 με βάση την νέα οικονομική & διοικητική ταξινόμηση
7	Μεσαίο	10	Αλλαγές αρχείων επαναβεβαίωσης της διεπαφής Taxis_1 βάσει της νέας οικονομικής & διοικητικής ταξινόμησης
8	Μεσαίο	10	Καθορισμός λογαριασμών απαιτήσεων και εσόδου της διεπαφής Taxis_1 επαναβεβαίωση βάσει της λογικής του νέου σχεδίου
9	Πολύπλοκο	15	Λειτουργία μέρους των σεναρίων βάσει του αρχικού ΚΑΕ (εκτός ανακαταλογισμών)
10	Πολύπλοκο	15	Διαδικασίες επαναβεβαίωσης για το 2019 και μετά
12	Πολύπλοκο	15	Παράλληλη λειτουργία υφιστάμενης και νέας οικονομικής / διοικητικής ταξινόμησης στη διεπαφή Taxis_1, για το έτος 2019
13	Μεσαίο	10	Καθορισμός budget address βάσει της νέας οικονομικής και διοικητικής ταξινόμησης
14	Μεσαίο	10	Καθορισμός λογαριασμών εσόδων, εξόδων, απαιτήσεων και υποχρεώσεων Λογιστικής (εκτός Δ24) βάσει της λογικής του νέου σχεδίου
15	Πολύπλοκο	15	Παράλληλη λειτουργία υφιστάμενης και νέας οικονομικής / διοικητικής ταξινόμησης στη Λογιστική (εκτός Δ24) για το έτος 2019 (εκτός τακτικών ενταλμάτων)
16	Πολύπλοκο	15	Νέοι κανόνες ελέγχου στη Λογιστική (εκτός Δ23) με βάση την νέα οικονομική & διοικητική ταξινόμηση
17	Πολύπλοκο	15	Λογιστικοποίηση με βάση το νέο σχέδιο και τα νέα άρθρα: Τακτικά εντάλματα και εντάλματα αποκατάστασης πάγιας προκαταβολής, Χρηματικά εντάλματα προπληρωμής (ΧΕΠ) / Τακτοποιητικό ΧΕΠ και Αδιάθετα, Εντολές ΚΔ, Εντολές ΥΔΕ, Μισθοδοσία, Λογιστική μέσω διεπαφών (πλην διεπαφής Taxis_1 Έσοδα)
25	Απλό	5	Αποτίμηση πάγιας προκαταβολής προκαταβολής βάσει της νέας οικονομικής και διοικητικής ταξινόμησης
26	Απλό	5	Διαχείριση ΚΥΑ προκαταβολής βάσει της νέας οικονομικής και διοικητικής ταξινόμησης
27	Απλό	5	Διαχείριση πάγιας προκαταβολής βάσει της νέας οικονομικής και διοικητικής ταξινόμησης
28	Μεσαίο	10	Διαχείριση Χ.Ε. παγίων προκαταβολών βάσει της νέας οικονομικής και διοικητικής ταξινόμησης
29	Πολύπλοκο	15	Αλλαγές αρχείων μισθοδοσίας βάσει της νέας οικονομικής / διοικητικής ταξινόμησης
30	Μεσαίο	10	Νέοι κανόνες ελέγχου στη Μισθοδοσία με βάση την νέα οικονομική & διοικητική ταξινόμηση
31	Μεσαίο	10	Διαχείριση αρχείων βάσει της νέας οικονομικής & διοικητικής ταξινόμησης
32	Απλό	5	Παράλληλη λειτουργία υφιστάμενης και νέας οικονομικής / διοικητικής ταξινόμησης στα αρχεία Μισθοδοσίας για το έτος 2019
33	Απλό	5	Δημιουργία και οριστικοποίηση κύριας ενοποιημένης εντολής βάσει της βάσει της νέας οικονομικής & διοικητικής ταξινόμησης
35	Πολύπλοκο	15	Δημιουργία και οριστικοποίηση δευτερεύουσας ενοποιημένης εντολής μισθοδοσίας

²¹ Η χρήση των ονομασιών των Use Cases έγινε μόνο ως παράδειγμα εύστοχης ονοματοδότησής τους και τρόπου υπολογισμού τους, οπότε η επεξήγηση των ορολογιών λογιστικού χαρακτήρα που περιέχονται σε αυτές δεν αποτελεί αντικείμενο της παρούσας εργασίας

36	Μεσαίο	10	Παράλληλη λειτουργία υφιστάμενης και νέας οικονομικής / διοικητικής ταξινόμησης στο Άρθρο 60 της Μισθοδοσίας για το έτος 2019
37	Μεσαίο	10	Λογιστικοποίηση Μισθοδοσίας Άρθρο 60 με βάση την λογική του νέου σχεδίου και των νέων άρθρων
39	Απλό	5	Αλλαγές στην εσωτερική διαλειτουργικότητα με το υποσύστημα εντολών βάσει της νέας οικονομικής & διοικητικής ταξινόμησης
40	Απλό	5	Αλλαγές στην εσωτερική διαλειτουργικότητα με το υποσύστημα Π/Υ και Ενταλμάτων βάσει της νέας οικονομικής & διοικητικής ταξινόμησης
41	Πολύπλοκο	15	Δημιουργία,αλλαγή,εμφάνιση και αντιλογισμός ΔΠ01, ΔΠ02, ΔΠ03, ΔΠ04
49	Μεσαίο	10	Κατάρτιση Προϋπολογισμού ΠΔΕ από Δ20-Φορείς
50	Μεσαίο	10	Κατάρτιση προϋπολογισμού Δαπανών-Φορείς
51	Απλό	5	Κατάρτιση προϋπολογισμού Δαπανών-Εισηγητές
52	Απλό	5	Κατάρτιση προϋπολογισμού Εσόδων - Δ20
53	Μεσαίο	10	Συνδημοσιευόμενοι Προϋπολογισμοί - Κατάρτιση-φορείς
54	Μεσαίο	10	Διαμόρφωση Μηνιαίου Προγράμματος Προβλεπόμενων Εσόδων
55	Απλό	5	Διαχείριση υποσυστήματος κατάρτισης π/υ (BI)
56	Πολύπλοκο	15	εντολές εσόδων
58	Μεσαίο	10	εντολές που ακολουθούνται από συμψηφιστικό ένταλμα
59	Μεσαίο	10	εντολές ενταλμάτων ΤτΕ και εντολές Ζ λογιστικού κυκλώματος
61	Απλό	5	δημιουργία εντολών-λογ.εγγραφών
62	Μεσαίο	10	Δ25 Χρηματικός κατάλογος - Τριπλότυπη Κατάσταση ΦΠ και ΔΕΚΟ
64	Μεσαίο	10	Δ25 Πληρωμή μέσω εντολής ΦΠ και ΔΕΚΟ
66	Απλό	5	Δ25 Διαβιβαστικό Χρηματικού Εντάλματος (word template) και Έκδοση Χρηματικού Εντάλματος
68	Απλό	5	Δ25 ΑΦΕΚ (Διαβιβαστικό και XL)
70	Πολύπλοκο	15	ΠΔΕ εμφάνιση ΣΑΕ και έργου ανά έτος
72	Μεσαίο	10	ΠΔΕ έλεγχος ύπαρξης πληρωμών ΣΑΕ/Εργων
73	Απλό	5	ΠΔΕ Φόρτωση excel για έργα ΔΔΕ-προκαταβολές (ενάριθμα)
74	Πολύπλοκο	15	Δημιουργία νέου προϋπολογισμού
75	Μεσαίο	10	Αλλαγή ελέγχου για την έκδοση εκκαθαριστικού ΧΕΠ για το έργο που δεν είναι ενεργό στο υφιστάμενο Οικ. Έτος
76	Μεσαίο	10	Αλλαγή προγραμμάτων καταχώρισης ΠΔΕ για την προσθήκη του νέου πεδίου Τομέα και για την ανάγνωση των νέων fund center και του νέου προϋπολογισμού
77	Πολύπλοκο	15	Δημιουργία ελέγχων για την επιλογή έργων ανά προϋπολογισμό (εθνικό συγχρηματοδοτούμενο)
78	Μεσαίο	10	Αλλαγή πινάκων καταχώρισης ΠΔΕ για την προσθήκη του νέου πεδίου Τομέα
79	Πολύπλοκο	15	Παραμετροποίηση για τον νέο προϋπολογισμό (π.χ. Workflow, τύποι ενταλμάτων κλπ)
80	Πολύπλοκο	15	Αλλαγή αναφορών και αναζητήσεις ΠΔΕ για την προσθήκη του νέου πεδίου Τομέα
81	Μεσαίο	10	Αλλαγή εκτυπωτικών ΠΔΕ
82	Μεσαίο	10	Αλλαγή Ενταλμάτων ΠΔΕ
84	Μεσαίο	10	Αλλαγή Τακτικών και λοιπών Ενταλμάτων (πχ Αποκατάστασης Πάγιας Προκαταβολής)
86	Μεσαίο	10	Αλλαγή Αναλήψεων
87	Πολύπλοκο	15	Αλλαγή Εκτυπωτικών Ενταλμάτων εκτός ΧΕΠ
88	Πολύπλοκο	15	Αλλαγή λοιπών Εκτυπωτικών
89	Μεσαίο	10	Αλλαγή αναφορών ΧΕΠ
90	Πολύπλοκο	10	Αλλαγή προγράμματος εκκαθάρισης ΧΕΠ για την Εκκαθάριση ΧΕΠ προηγούμενων ετών (αλλαγή ΚΑΕ)
91	Μεσαίο	15	Αλλαγή ΧΕΠ και εκκαθάρισης ΧΕΠ και αλλαγή Εκτυπωτικών

Πίνακας 15: Περιγραφή των Use Cases

Τέλος, στους πίνακες 16 και 17 αντίστοιχα, παρουσιάζεται ο τρόπος υπολογισμού της τιμής UAW (=32) και οι τιμές τους βάρους για κάθε actor.

a/a	Κατηγορίες Actors	Βάρος	Παράγοντες / κατηγορία	Περιγραφή
1	Απλό	1	2	Ο παράγοντας αντιστοιχεί σε σύστημα με καθορισμένο API
2	Μεσαίο	2	3	Ο παράγοντας αντιστοιχεί σε σύστημα που αλληλεπιδρά μέσω πρωτοκόλλου (π.χ. TCP/IP)
3	Πολύπλοκο	3	8	Ο παράγοντας αντιστοιχεί σε πρόσωπο που αλληλεπιδρά μέσω User Interface
UAW			32	

Πίνακας 16: Υπολογισμός Μη-Σταθμισμένων Παραγόντων (UAW)

Actor		Βάρος	Όνομα Actor
1	Πολύπλοκο	3	Χρήστης Ενταλμάτων
2	Πολύπλοκο	3	Χρήστης Εντολών
3	Πολύπλοκο	3	Χρήστης Λογιστικής
4	Πολύπλοκο	3	Χρήστης Προϋπολογισμού
5	Πολύπλοκο	3	Χρήστης ΠΔΕ
6	Πολύπλοκο	3	Χρήστης Δ25
7	Πολύπλοκο	3	Χρήστης Μισθοδοσίας
8	Πολύπλοκο	3	Διαχειριστής
8	Μεσαίο	2	Διεπαφή με ΤτΕ
9	Απλό	1	Web service με TAXIS
10	Μεσαίο	2	Διεπαφή με ΕΛΔΑΠ
11	Μεσαίο	2	Διεπαφή με ΔΔΕ
12	Απλό	1	Web service με ΕΑΠ

Πίνακας 17: Μη-Σταθμισμένοι Παράγοντες

ΕΠΙΛΟΓΟΣ

Η εργασία ήρθε αντιμέτωπη με ένα νέο σχετικά αντικείμενο, η αξία του οποίου δεν φαίνεται να έχει αναγνωρισθεί όπως θα έπρεπε από τη Δημόσια Διοίκηση. Σύμφωνα με τα όσα αναφέρθηκαν στο κύριο μέρος της εργασίας, γίνεται κατανοητή η σημασία που έχει για τους φορείς της Δημόσιας Διοίκησης που είναι υπεύθυνοι να υλοποιήσουν έργα λογισμικού, να προβαίνουν σε εκτίμηση του κόστους του εκ των προτέρων. Η πλέον κατάλληλη μέθοδος που προτείνεται στους φορείς, είναι η UCP, όπως προκύπτει από την παράθεση των πλεονεκτημάτων και μειονεκτημάτων τους, που έγινε στην παρούσα εργασία. Προτεραιότητα όμως πρέπει να αποτελεί από τους φορείς και η συγκέντρωση ιστορικών στοιχείων σχετικά με το κόστος και τους τύπους των έργων, καθώς η μέθοδος βασίζεται σε αυτά για την εκτίμηση των συντελεστών της. Φυσικά όσο προχωράει το κάθε έργο είναι σημαντικό οι διαχειριστές του σε συνεργασία με τους αρμόδιους εκτίμησης κόστους λογισμικού να επικαιροποιούν σε τακτική βάση την εκτίμηση του κόστους, όπως άλλωστε και του χρονοπρογραμματισμού, ώστε να γίνεται όσο το δυνατόν πιο αποτελεσματική χρήση των πόρων.

Άλλη σημαντική δραστηριότητα που θα πρέπει να μην παραμελούν οι δημόσιοι φορείς είναι να καταγράφουν τις διαφορές στις αρχικές και τελικές εκτιμήσεις του κόστους. Με αυτόν τον τρόπο ανάδρασης βελτιώνεται η εκτίμηση του κόστους λογισμικού, αφού μπορεί να στηρίζεται και σε ακριβή ιστορικά στοιχεία. Προτείνεται η δημιουργία μιας κεντρικής βάσης δεδομένων όπου θα αποθηκεύονται για κάθε έργο οι αρχικές και οι τελικές τιμές των παραγόντων που υπεισέρχονται στην εκτίμηση κόστους λογισμικού.

Ιδανικά, προτείνεται οι φορείς να ετοιμάζουν μία εκτίμηση κόστους λογισμικού με τη χρήση της μεθόδου UCP και να υπάρχει η απαίτηση εκτίμησης και από τους υποψήφιους ανάδοχους, ώστε να υπάρχει μία σταθερή, αντικειμενική και κοινά αποδεκτή βάση διαπραγμάτευσης.

Στη μελέτη περίπτωσης που εξετάστηκε για παράδειγμα, οι τιμές του Συντελεστή Τεχνικής Πολυπλοκότητας δεν διαφοροποιούνται από την εκτίμηση του Ανάδοχου, σε αντίθεση με τις τιμές που καθορίζουν τον Συντελεστή Περιβαλλοντικής Πολυπλοκότητας. Διαφορετικές είναι και οι τιμές των συντελεστών Μη-Σταθμισμένου Βάρους Περιπτώσεων Χρήσης και Μη-Σταθμισμένου Βάρους Παραγόντων. Είναι σημαντικό να γίνεται αναλυτική καταγραφή των use cases που προέρχονται από τις

απαιτήσεις των έργων, ώστε να είναι κατανοητές και οι Μη-Σταθμισμένου Βάρους Περιπτώσεις Χρήσης και οι Μη-Σταθμισμένου Βάρους Παράγοντες.

Οι διαφορές στους υπολογισμούς των ανωτέρων συντελεστών οδήγησαν σε διαφορά στην εκτίμηση του κόστους, αφού ο Ανάδοχος υπολόγισε ότι χρειάζονται 108,48 ανθρωπομήνες για την υλοποίηση της παρέμβασης με βάση τη μεθοδολογία UCP, ενώ στα πλαίσια της εργασίας υπολογίστηκαν 50,40 ανθρωπομήνες. Ακόμα και αν δεν συμπίπτουν οι εκτιμήσεις των δημόσιων φορέων με εκείνες των εξωτερικών συνεργατών, σίγουρα η UCP θα μπορέσει να αποτελέσει μία εξαιρετικά ωφέλιμη βάση συζήτησης, ώστε να υπάρξει επιστημονική τεκμηρίωση και σύγκλιση στην εκτίμηση του κόστους ανάπτυξης λογισμικού.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΑΡΑΠΟΜΠΕΣ

Α. Ξενόγλωσσες

- [1] Abran A. (2015). “*Software Metrics and software methodology*”. Available at: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470606834.fmatter>
[Downloaded: 25 Νοεμβρίου 2018]
- [2] Bente A., Dag I. (2002). “*Towards an inspection technique for use case models*”. In Proceedings of the 14th international conference on Software engineering and knowledge engineering. Available at: https://www.researchgate.net/publication/221390632_Towards_an_inspection_technique_for_use_case_models [Downloaded: 25 Νοεμβρίου 2018]
- [3] Bente, A. (2002), “*Comparing Effort Estimates Based on Use Case Points with Expert Estimates*”. Available at: https://www.researchgate.net/publication/228597738_Comparing_Effort_Estimates_Based_on_Use_Case_Points_with_Expert_Estimates [Downloaded: 5 Δεκεμβρίου 2018]
- [4] Che Ani Z., Basri S., Sarlan A. (2017), “*A Reusability Assessment of UCP-Based Effort Estimation Framework using Object-Oriented Approach*”. Available at: <http://journal.utem.edu.my/index.php/jtec/article/viewFile/2971/2103> [Downloaded: 25 Νοεμβρίου 2018]
- [5] Johns C. (2015). “*Software Engineering - Software Cost Estimation*”. Available at: <https://slideplayer.com/slide/7458274/> [Downloaded: 5 Νοεμβρίου 2018]
- [6] Kress C.F., Hummel O., Huq M. (2014). “*A Practical Approach for Reliable Pre-Project Effort Estimation*”. Available at: <http://ceur-ws.org/Vol-1138/re4p21.pdf>
[Downloaded: 5 Νοεμβρίου 2018]
- [7] Kusumoto S., Matukawa F., Inoue K. (2004). “*Estimating Effort by Use Case Points: Method, Tool and Case Study*”. Available at: <https://people.eecs.ku.edu/~hossein/811/Papers/use-case-points-kusmoto.pdf>
[Downloaded: 15 Νοεμβρίου 2018]

- [8] Leung H., Fan Z. (2002). “*Software Cost Estimation in Information and Software Technology*”. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.4489&rep=rep1&type=pdf> [Downloaded: 2 Δεκεμβρίου 2018]
- [9] Putnam L. Jr., Hastie S. (2018). “*Why Software Estimation Is More Important Now Than Ever*”. Available at: <https://www.infoq.com/articles/software-estimation-important> [Downloaded: 15 Νοεμβρίου 2018]
- [10] Rozalina R., Mansor Z. (2018). “*Validated Software Cost Estimation Factors for Government Projects using Rasch Measurement Model*”. Available at: <file:///C:/Users/NB%20HP/Downloads/6386-15161-1-PB.pdf> [Downloaded: 2 Δεκεμβρίου 2018]
- [11] Sangeetha K. (2015). “*Analysis of Software Estimation Method: Function point and Use case point*”. Available at: http://ijiset.com/vol2/v2s11/IJISSET_V2_I11_112.pdf [Downloaded:14 Νοεμβρίου 2018]
- [12] Satapathy S.M., Rath S.K. (2014). “*Use Case Point Approach Based Software Effort Estimation using Various Support Vector Regression Kernel Methods*”. International Journal of Information Processing
- [13] Schwaber, K., Sutherland J. (2014). “*The Scrum Guide*”. Available at: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf> [Downloaded: 20 Νοεμβρίου 2018]
- [14] Sehlhorst S. (2007). “*Software Cost Estimation With Use Case Points - Final Calculations*”. Available at: <tynerblain.com/blog/2007/02/19/software-cost-estimation-ucp-6/> [Downloaded: 20 Νοεμβρίου 2018]
- [15] Silhavy R., Silhavy P., Prokopova Z. (2018). “*Evaluating subset selection methods for use case points estimation*”. Available at: https://ac.els-cdn.com/S0950584917305153/1-s2.0-S0950584917305153-main.pdf?_tid=61487ab6-9731-478c-abb353195168f55&acdnat=1543891728_e6ffc37482ee2ccc8f074327bada651c [Downloaded: 20 Νοεμβρίου 2018]

- [16] Tripathi R. (2016). “*Comperative Study of Software Cost Estimation Techiques*”. Available at:
<file:///C:/Users/NB%20HP/Downloads/ComparativeStudyofSoftwareCostEstimationTechniques.pdf> [Downloaded:15 Νοεμβρίου 2018]
- [17] Ulbert, Z. (2014). “*Software development process and software quality assurance. Chapter 13: Software Cost Estimation*”. Διαθέσιμο στο: http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/szoftverfejlesztési_folyamatok_angol/index.html
 [Προσπελάστηκε: 16 Νοεμβρίου 2018]

B. Ελληνόγλωσσες

- [18] Αντωνόπουλος Χ., Μπίμπη Μ. (2015). “*Παρουσιάσεις μαθήματος: Εισαγωγή στην Τεχνολογία Λογισμικού Τμήμα Μηχανικών Η/Υ Τηλεπικοινωνιών & Δικτύων*”, Πανεπιστήμιο Θεσσαλίας. Διαθέσιμες στο: <https://courses.e-ce.uth.gr/CE420/Fall14/lectures/1-Introduction.pdf> [Τελευταία πρόσβαση: 30 Οκτωβρίου 2018]
- [19] ΓΓ Τηλεπικοινωνιών και Ταχυδρομείων (2016). Σχέδιο Δράσης Υλοποίησης έργων ΤΠΕ για τις Ψηφιακές Υποδομές και την Ανάπτυξη, , Φεβρουάριος 2016, έκδοση 1.4.
- [20] Καπόπουλος Δ., Σπηλιωτόπουλος, Γ. (2018). *Σημειώσεις μαθήματος «Διαχείριση Έργων» 25ης σειράς ΕΣΔΔΑ, Τμήματος Ψηφιακής Πολιτικής*
- [21] Παρασκευάς Μ. (2018). *Η Δημόσια Διοίκηση στην Κοινωνία της Πληροφορίας*. Διαθέσιμο στο: <https://repository.kallipos.gr/handle/11419/412> [Τελευταία πρόσβαση: 30 Νοεμβρίου 2018]
- [22] Σύμβαση για το έργο «*Διαχείριση αλλαγών Ολοκληρωμένου Πληροφοριακού Συστήματος Δημοσιονομικής Πολιτικής (ΟΠΣ-ΔΠ)*», που υπογράφηκε μεταξύ του Υπουργείου Οικονομικών και της Ανάδοχου εταιρίας την 30-01-2018
 ΑΔΑΜ: 18SYMV002601383 2018-01-30
- [23] Τσάπελας Θ, (2018). Σημειώσεις μαθήματος «*Πληροφοριακά Συστήματα Διοίκησης*» Πανεπιστήμιο Πειραιώς. Διαθέσιμες στο: <http://www.unipi.gr/faculty/tsapelas/mis4.pdf> [Τελευταία πρόσβαση: 29 Νοεμβρίου 2018]

- [24] Υπουργείο Ψηφιακής Πολιτικής, Τηλεπικοινωνιών και Ενημέρωσης, Γενική Γραμματεία Ψηφιακής Πολιτικής (2016). *Εθνική Ψηφιακή Στρατηγική*. Διαθέσιμη στο:
http://www.opengov.gr/digitalandbrief/wp-content/uploads/downloads/2016/11/digital_strategy.pdf [Τελευταία πρόσβαση: 30 Οκτωβρίου 2018]
- [25] Χαραλαμπίδης Γ, (2018). Σημειώσεις μαθήματος «Τεχνολογίας Λογισμικού» για το Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου. Διαθέσιμες στο:
http://www.icsd.aegean.gr/website_files/proptyxiako/979899442.pdf [Τελευταία πρόσβαση: 19 Νοεμβρίου 2018]



Εθνική Σχολή Δημόσιας Διοίκησης και Αυτοδιοίκησης (ΕΣΔΔΑ)
Πειραιώς 211, ΤΚ 177 78, Ταύρος
τηλ: 2131306349 , fax: 2131306479
www.ekdd.gr