



Πανεπιστήμιο Αιγαίου

Τεχνολογία Λογισμικού

ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΙΣΜΙΚΟΥ

Γιάννης Χαραλαμπίδης (yannisx@aegean.gr)

Αναπληρωτής Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Στόχοι

- Παρουσίαση των βασικών στοιχείων της κοστολόγησης και της τιμολόγησης του λογισμικού
- Αναφορά τριών μετρικών που χρησιμοποιούνται για την αξιολόγηση της παραγωγικότητας του λογισμικού
- Ερμηνεία των λόγων για τους οποίους κατά την εκτίμηση του κόστους του λογισμικού πρέπει να χρησιμοποιείται φάσμα μεθόδων
- Περιγραφή των αρχών του μοντέλου COCOMO 2 για την αλγοριθμική εκτίμηση του κόστους

Περιεχόμενα

- Παραγωγικότητα λογισμικού
- Τεχνικές εκτίμησης
- Αλγοριθμική μοντελοποίηση κόστους
- Διάρκεια και στελέχωση έργων

Βασικά ζητήματα σχετικά με την εκτίμηση του κόστους

- Πόσος φόρτος εργασίας απαιτείται για την ολοκλήρωση κάθε δραστηριότητας;
- Πόσος ημερολογιακός χρόνος χρειάζεται για την ολοκλήρωση κάθε δραστηριότητας;
- Ποιο είναι το συνολικό κόστος κάθε δραστηριότητας;
- Σε ένα έργο, η εκτίμηση του κόστους του και ο χρονοπρογραμματισμός του είναι αλληλένδετες δραστηριότητες διαχείρισης.

Παράμετροι κόστους λογισμικού

- Κόστος υλικού και λογισμικού
- Έξοδα μετακινήσεων και εκπαίδευσης
- Κόστος της εργασίας (ο κυρίαρχος παράγοντας στα περισσότερα έργα)
 - Οι αμοιβές των μηχανικών που εμπλέκονται στο έργο
 - Έξοδα ασφάλισης
- Στο κόστος εργασίας πρέπει να συνυπολογιστούν και οι διάφορες επιβαρύνσεις
 - Κόστος εργασιακού χώρου, θέρμανσης και φωτισμού
 - Κόστος δικτύωσης και επικοινωνιών
 - Κόστος κοινόχρηστων εγκαταστάσεων (όπως η βιβλιοθήκη, το εστιατόριο για το προσωπικό, κ.λπ.)

Κοστολόγηση και τιμολόγηση

- Οι εκτιμήσεις πραγματοποιούνται με σκοπό τον προσδιορισμό του κόστους ανάπτυξης ενός συστήματος λογισμικού για τον προγραμματιστή.
- Η σχέση του κόστους ανάπτυξης και της τιμής που προτείνεται στον πελάτη δεν είναι απλή.
- Η τιμή που χρεώνεται επηρεάζεται από ευρύτερους εταιρικούς, οικονομικούς, πολιτικούς και επιχειρηματικούς παράγοντες.

Παράγοντες τιμολόγησης λογισμικού

Συγκυρίες της αγοράς

Μια εταιρεία ανάπτυξης λογισμικού μπορεί να προσφέρει χαμηλή τιμή γιατί θέλει να εισχωρήσει σε ένα νέο τομέα της αγοράς λογισμικού. Αν η εταιρεία συμβιβαστεί με χαμηλό κέρδος σε ένα έργο, μπορεί να έχει την ευκαιρία να αποκομίσει αργότερα μεγαλύτερο κέρδος. Επίσης, η εμπειρία που θα αποκτηθεί μπορεί να βοηθήσει στην ανάπτυξη νέων προϊόντων.

Αβεβαιότητα εκτίμησης κόστους

Αν μια εταιρεία δεν είναι σίγουρη για τις εκτιμήσεις κόστους που έχει κάνει, μπορεί να αυξήσει την τιμή της κατά ένα ποσοστό πάνω από το κανονικό κέρδος.

Ζητήματα σύμβασης

Ο πελάτης μπορεί να είναι πρόθυμος να επιτρέψει στο δημιουργό να διατηρήσει την ιδιοκτησία του πηγαίου κώδικα και να τον χρησιμοποιήσει ξανά σε άλλα έργα. Σε αυτή την περίπτωση, η τιμή χρέωσης μπορεί να είναι μικρότερη από την τιμή για την περίπτωση στην οποία ο κώδικας του λογισμικού παραδίδεται στον πελάτη.

Μεταβλητότητα απαιτήσεων

Αν είναι πιθανό να αλλάξουν οι απαιτήσεις, η εταιρεία μπορεί να μειώσει την τιμή για να κερδίσει τη σύμβαση. Αφού γίνει αυτό, μπορεί κατόπιν να χρεώσει μεγάλα ποσά για τις αλλαγές στις απαιτήσεις.

Οικονομική ευρωστία

Εταιρείες ανάπτυξης λογισμικού που βρίσκονται σε οικονομική δυσχέρεια μπορεί να χαμηλώσουν την τιμή τους για να κερδίσουν κάποια σύμβαση. Είναι καλύτερο να υπάρχει μικρότερο κέρδος από το κανονικό ή ακόμα και να ισοσκελίζονται μόνο τα έσοδα με τα έξοδα από το ενδεχόμενο να κλείσει η εταιρεία.

Παραγωγικότητα λογισμικού

- Ο ρυθμός με τον οποίο κάθε μηχανικός που εμπλέκεται στην ανάπτυξη του λογισμικού παράγει τμήματα του προγράμματος και τη σχετική τεκμηρίωση.
- Δεν σχετίζεται ευθέως με την ποιότητα, αν και ο ποιοτικός έλεγχος είναι παράγοντας που επηρεάζει την αξιολόγηση της παραγωγικότητας.
- Στην ουσία, θέλουμε να μετρήσουμε τις χρήσιμες λειτουργικές δυνατότητες που παράγονται σε μία μονάδα χρόνου.

Μετρικές παραγωγικότητας

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

Προβλήματα που αφορούν τις μετρήσεις

- Εκτίμηση του μεγέθους της μετρικής (για παράδειγμα, πόσα λειτουργικά σημεία)
- Εκτίμηση του συνολικού αριθμού ανθρωπομηνών των προγραμματιστών που έχουν παρέλθει
- Εκτίμηση της παραγωγικότητας των αναδόχων (για παράδειγμα, της ομάδας που έχει αναλάβει την τεκμηρίωση) και ενσωμάτωση της εκτίμησης αυτής στη συνολική εκτίμηση.

Γραμμές κώδικα

- Τι είναι μια γραμμή κώδικα;
 - Η μετρική αυτή είχε προταθεί για πρώτη φορά όταν ακόμα τα προγράμματα γράφονταν σε κάρτες, καθεμιά από τις οποίες περιείχε μία γραμμή κώδικα
 - Ποια είναι λοιπόν η αντιστοιχία με εντολές γλωσσών προγραμματισμού όπως είναι η Java, όπου κάθε εντολή μπορεί να καταλαμβάνει πολλές γραμμές ή σε κάθε γραμμή μπορεί να υπάρχουν πολλές εντολές;
- Ποια προγράμματα πρέπει να προσμετρώνται ότι ανήκουν στο σύστημα;
- Στο μοντέλο γίνεται η παραδοχή ότι υπάρχει ευθεία σχέση μεταξύ του μεγέθους του συστήματος και του όγκου της τεκμηρίωσης.

Συγκρίσεις παραγωγικότητας

- Όσο πιο χαμηλού επιπέδου είναι η γλώσσα, τόσο πιο παραγωγικός είναι ο προγραμματιστής
 - Για την υλοποίηση της ίδιας λειτουργικής δυνατότητας χρειάζεται περισσότερος κώδικας σε μια γλώσσα προγραμματισμού χαμηλού επιπέδου απ' ότι σε μια γλώσσα υψηλού επιπέδου.
- Όσο πιο αναλυτικός είναι ο προγραμματιστής στην έκφρασή του, τόσο υψηλότερη είναι η παραγωγικότητα
 - Σύμφωνα με τις μετρικές παραγωγικότητας που βασίζονται στο πλήθος γραμμών κώδικα, οι προγραμματιστές που χρησιμοποιούν πιο αναλυτικές εκφράσεις είναι πιο παραγωγικοί από εκείνους που γράφουν συνεπτυγμένο κώδικα.

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

Λειτουργικά σημεία

- Βασίζονται σε συνδυασμό χαρακτηριστικών του προγράμματος
 - Εξωτερικές εισοδοι και έξοδοι
 - Αλληλεπιδράσεις χρηστών
 - Εξωτερικές διασυνδέσεις
 - Αρχεία που χρησιμοποιούνται από το σύστημα
- Με κάθε χαρακτηριστικό σχετίζεται ένας συντελεστής στάθμισης, οπότε το πλήθος λειτουργικών σημείων υπολογίζεται με τον πολλαπλασιασμό κάθε αρχική μέτρηση με τον συντελεστή στάθμισης και την πρόσθεση όλων των τιμών.

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

Λειτουργικά σημεία

- Το πλήθος λειτουργικών σημείων τροποποιείται ανάλογα με την πολυπλοκότητα του έργου
- Για κάθε γλώσσα προγραμματισμού, μπορούμε να χρησιμοποιήσουμε τα λειτουργικά σημεία για να υπολογίσουμε γραμμές κώδικα (LOC) με βάση το μέσο πλήθος γραμμών κώδικα (AVC) ανά λειτουργικό σημείο της συγκεκριμένης γλώσσας
 - $LOC = AVC * \text{πλήθος λειτουργικών σημείων}$
 - Το AVC είναι ένας συντελεστής που εξαρτάται από τη γλώσσα προγραμματισμού και κυμαίνεται από 200-300 για συμβολική γλώσσα μέχρι 2-40 για γλώσσες τέταρτης γενιάς
- Τα λειτουργικά σημεία είναι πολύ υποκειμενικά Εξαρτώνται από τον εκτιμητή
 - Δεν είναι εφικτή η αυτόματη μέτρηση των λειτουργικών σημείων

Αντικειμενικά σημεία

- Τα **αντικειμενικά σημεία** (τα οποία ονομάζονται και **σημεία εφαρμογής**) είναι μια μετρική που μπορεί να χρησιμοποιηθεί αντί των λειτουργικών σημείων όταν στην ανάπτυξη χρησιμοποιούνται γλώσσες 4ης γενιάς ή άλλες παρόμοιες.
- Τα αντικειμενικά σημεία ΔΕΝ ταυτίζονται με τις κλάσεις αντικειμένων.
- Το πλήθος των αντικειμενικών σημείων ενός προγράμματος είναι μια σταθμισμένη εκτίμηση
 - Του αριθμού των ξεχωριστών οθονών που εμφανίζονται
 - Του αριθμού των αναφορών που παράγονται από το σύστημα
 - Του αριθμού των υπομονάδων που πρέπει να αναπτυχθούν για την υποστήριξη του κώδικα προγραμματισμού βάσεων δεδομένων

Εκτίμηση αντικειμενικών σημείων

- 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 υπομονάδες των συμβατικών γλωσσών προγραμματισμού.
- Άρα το πλήθος τους μπορεί να εκτιμηθεί σε ένα αρκετά πρώιμο στάδιο της διαδικασίας ανάπτυξης.
- Σε ένα τέτοιο στάδιο, η εκτίμηση των γραμμών κώδικα ενός συστήματος είναι πολύ δύσκολη.

Εκτιμήσεις παραγωγικότητας

- Ενσωματωμένα συστήματα πραγματικού χρόνου, 40-160 γραμμές κώδικα/ανθρωπομήνα προγραμματιστή.
- Προγράμματα συστημάτων, 150-400 γραμμές κώδικα/ανθρωπομήνα προγραμματιστή.
- Εμπορικές εφαρμογές, 200-900 γραμμές κώδικα/ανθρωπομήνα προγραμματιστή.
- Με τα αντικειμενικά σημεία, η παραγωγικότητα κυμαίνεται μεταξύ 4 και 50 αντικειμενικών σημείων/μήνα ανάλογα με την υποστήριξη εργαλείων και τις δυνατότητες των προγραμματιστών.

Παράγοντες που επηρεάζουν την παραγωγικότητα

Πείρα στο συγκεκριμένο πεδίο εφαρμογής

Η γνώση του πεδίου εφαρμογής είναι ουσιώδης για την αποτελεσματική ανάπτυξη λογισμικού. Μηχανικοί που γνωρίζουν ένα πεδίο είναι πιθανό να είναι πιο παραγωγικοί.

Ποιότητα διαδικασιών

Η διαδικασία ανάπτυξης που χρησιμοποιείται μπορεί να έχει σημαντική επίδραση στην παραγωγικότητα. Αυτό περιγράφεται στο Κεφάλαιο 28.

Μέγεθος έργου

Όσο μεγαλύτερο είναι ένα έργο, τόσο περισσότερος χρόνος απαιτείται για την επικοινωνία της ομάδας. Έτσι διατίθεται λιγότερος χρόνος για ανάπτυξη, οπότε ελαττώνεται η ατομική παραγωγικότητα.

Τεχνολογική υποστήριξη

Η καλή τεχνολογία υποστήριξης, όπως τα εργαλεία CASE και τα συστήματα διαχείρισης διευθετήσεων, μπορεί να βελτιώσει την παραγωγικότητα.

Εργασιακό περιβάλλον

Όπως είδαμε στο Κεφάλαιο 25, ένα ήσυχο εργασιακό περιβάλλον με ιδιωτικούς χώρους εργασίας συμβάλλει στη βελτίωση της παραγωγικότητας.

Ποιότητα και παραγωγικότητα

- Όλες οι μετρικές που βασίζονται στον όγκο ανά μονάδα χρόνου είναι ελλιπείς επειδή δεν συνυπολογίζουν την ποιότητα.
- Γενικά η παραγωγικότητα μπορεί να αυξηθεί, αλλά αυτό γίνεται σε βάρος της ποιότητας.
- Η σχέση των μετρικών παραγωγικότητας και ποιότητας δεν είναι ξεκάθαρη.
- Αν οι απαιτήσεις συνεχώς μεταβάλλονται, τότε μια προσέγγιση που βασίζεται στην απαρίθμηση γραμμών κώδικα δεν έχει νόημα αφού το ίδιο το πρόγραμμα δεν είναι στατικό

Τεχνικές εκτίμησης

- Δεν υπάρχει απλός τρόπος επακριβούς εκτίμησης του φόρτου εργασίας που απαιτείται για την ανάπτυξη ενός συστήματος λογισμικού
 - Οι αρχικές εκτιμήσεις βασίζονται στις ανεπαρκείς πληροφορίες που περιλαμβάνει ο ορισμός των απαιτήσεων του χρήστη
 - Το λογισμικό ίσως πρέπει να εκτελείται σε υπολογιστές με τους οποίους δεν είστε εξοικειωμένοι ή να χρησιμοποιεί νέα τεχνολογία.
 - Τα άτομα που θα εμπλακούν στο έργο ίσως είναι άγνωστα.
- Οι εκτιμήσεις του κόστους του έργου ίσως να είναι αυτοεκπληρούμενες
 - Η εκτίμηση καθορίζει τον προϋπολογισμό και το προϊόν αναπροσαρμόζεται ώστε να ταιριάζει σε αυτόν.

Αλλαγή της τεχνολογίας

- Ενδεχόμενη αλλαγή στην τεχνολογία μπορεί να συνεπάγεται ότι η προηγούμενη συσσωρευμένη πείρα στον τομέα των εκτιμήσεων δεν μεταφέρεται στα νέα συστήματα
 - Συστήματα κατανεμημένων αντικειμένων αντί για συστήματα κεντρικών υπολογιστών
 - Χρήση υπηρεσιών Ιστού
 - Χρήση συστημάτων ERP ή συστημάτων που στηρίζονται σε βάσεις δεδομένων
 - Χρήση έτοιμων εφαρμογών λογισμικού
 - Ανάπτυξη με σκοπό την επαναχρησιμοποίηση και ανάπτυξη με επαναχρησιμοποίηση
 - Ανάπτυξη με χρήση γλωσσών σεναρίων
 - Χρήση εργαλείων CASE και γεννητριών προγραμμάτων

Τεχνικές εκτίμησης

- Αλγοριθμική μοντελοποίηση κόστους
- Κρίση ειδικών
- Εκτίμηση κατ' αναλογία
- Νόμος Parkinson
- Τιμολόγηση με σκοπό την ανάληψη του έργου

Τεχνικές εκτίμησης

Αλγοριθμική μοντελοποίηση κόστους

Με βάση πληροφορίες άλλων έργων σχετικά με το κόστος, αναπτύσσεται ένα μοντέλο που συσχετίζει κάποια παράμετρο του λογισμικού (συνήθως το μέγεθός του) με το κόστος του έργου. Γίνεται μια εκτίμηση αυτής της παραμέτρου και το μοντέλο προβλέπει την απαιτούμενη εργασία.

Κρίση ειδικών

Ζητείται η γνώμη πολλών ατόμων που ειδικεύονται στην προτεινόμενη τεχνική ανάπτυξης του λογισμικού και στο συγκεκριμένο πεδίο εφαρμογής. Καθένας τους εκτιμά το κόστος του έργου. Γίνεται σύγκριση και ανάλυση αυτών των εκτιμήσεων. Η διαδικασία εκτίμησης επαναλαμβάνεται μέχρι να προκύψει μια εκτίμηση στην οποία συμφωνούν όλοι.

Εκτίμηση κατ' αναλογία

Αυτή η τεχνική μπορεί να εφαρμοστεί όταν έχουν ολοκληρωθεί άλλα έργα στο ίδιο πεδίο εφαρμογής. Γίνεται εκτίμηση του κόστους ενός νέου έργου κατ' αναλογία με αυτά τα ολοκληρωμένα έργα. Ο Myers (Myers, 1989) δίνει μια πολύ σαφή περιγραφή αυτής της προσέγγισης.

Ο νόμος του Parkinson

Σύμφωνα με το νόμο του Parkinson, ο φόρτος εργασίας κατανέμεται έτσι ώστε να καταλάβει όλο το διαθέσιμο χρόνο. Το κόστος καθορίζεται από τους διαθέσιμους πόρους και όχι από την αξιολόγηση των αντικειμενικών στόχων. Αν το λογισμικό πρέπει να παραδοθεί σε 12 μήνες και υπάρχουν διαθέσιμοι 5 άνθρωποι, εκτιμάται ότι η απαιτούμενη εργασία ανέρχεται σε 60 ανθρωπομήνες.

Τιμολόγηση με σκοπό την ανάληψη του έργου

Το κόστος του λογισμικού εκτιμάται στο ύψος που μπορεί να διαθέσει ο πελάτης για το έργο. Η εκτιμώμενη εργασία εξαρτάται από τον προϋπολογισμό του πελάτη και όχι από τις λειτουργίες του λογισμικού.

Τιμολόγηση με σκοπό την ανάληψη του έργου

- Το κόστος του λογισμικού εκτιμάται στο ύψος που μπορεί να διαθέσει ο πελάτης για το έργο.
- Πλεονεκτήματα:
 - Η σύμβαση ανατίθεται σε εσάς.
- Μειονεκτήματα:
 - Είναι μικρή η πιθανότητα ο πελάτης να λάβει το σύστημα που επιθυμεί. Το κόστος δεν ανταποκρίνεται στον απαιτούμενο φόρτο εργασίας.

Αναλυτική και συνθετική εκτίμηση

- Οποιαδήποτε από τις παραπάνω προσεγγίσεις μπορεί να χρησιμοποιηθεί είτε με συνθετικό είτε με αναλυτικό τρόπο.
- Αναλυτικά
 - Ξεκινάμε από το επίπεδο του συστήματος και εξετάζουμε τη συνολική λειτουργικότητα του προϊόντος και τον τρόπο που αυτή παρέχεται μέσα από τα υποσυστήματα.
- Συνθετικά
 - Ξεκινάμε από το επίπεδο των συστατικών στοιχείων και εκτιμούμε το φόρτο εργασίας που απαιτείται για την ανάπτυξη κάθε στοιχείου. Αθροίζουμε το φόρτο εργασίας κάθε στοιχείου για να διατυπώσουμε μια τελική εκτίμηση.

Αναλυτική εκτίμηση

- Μπορεί να χρησιμοποιηθεί χωρίς να είναι γνωστή η αρχιτεκτονική του συστήματος και τα συστατικά στοιχεία που το απαρτίζουν.
- Λαμβάνει υπόψη έξοδα ενοποίησης, τεκμηρίωσης και διαχείρισης διευθετήσεων.
- Μπορεί να υποεκτιμήσει το κόστος επίλυσης δύσκολων τεχνικών προβλημάτων σε γλώσσες χαμηλού επιπέδου.

Συνθετική εκτίμηση

- Χρησιμοποιείται όταν είναι γνωστή η αρχιτεκτονική του συστήματος και έχουν προσδιοριστεί τα συστατικά στοιχεία που το απαρτίζουν.
- Είναι μια μέθοδος μεγάλης ακρίβειας αν το σύστημα έχει σχεδιαστεί πλήρως.
- Μπορεί να υποεκτιμήσει το κόστος δραστηριοτήτων που αναφέρονται στο επίπεδο ολόκληρου του συστήματος, όπως είναι η ενοποίηση και η τεκμηρίωση.

Μέθοδοι εκτίμησης

- Κάθε μέθοδος έχει πλεονεκτήματα και αδυναμίες.
- Η τελική εκτίμηση πρέπει να βασίζεται σε πολλές μεθόδους.
- Αν αυτές δεν δίνουν κατά προσέγγιση το ίδιο αποτέλεσμα, τότε μάλλον δεν έχετε αρκετές πληροφορίες για την πραγματοποίηση της εκτίμησης.
- Πρέπει να αναζητήσετε περισσότερες πληροφορίες για να προχωρήσετε σε πιο ακριβή εκτίμηση.
- Μερικές φορές, η "τιμολόγηση με σκοπό την ανάληψη του έργου" είναι η μοναδική στρατηγική που μπορεί να εφαρμοστεί.

Τιμολόγηση με σκοπό την ανάληψη του έργου

- Αυτή η προσέγγιση μπορεί να φαίνεται αντιδεοντολογική και αντιεπαγγελματική.
- Ωστόσο, όταν δεν υπάρχουν λεπτομερείς πληροφορίες, μπορεί να είναι η στρατηγική που ενδείκνυται.
- Η συμφωνία για το κόστος ενός έργου γίνεται με βάση μια περιληπτική προσφορά και η ανάπτυξη του έργου περιορίζεται από αυτό το κόστος.
- Στη συνέχεια μπορεί να γίνουν διαπραγματεύσεις για πιο λεπτομερείς προδιαγραφές ή μπορεί να χρησιμοποιηθεί μια πιο επαναστατική προσέγγιση για την ανάπτυξη του συστήματος.

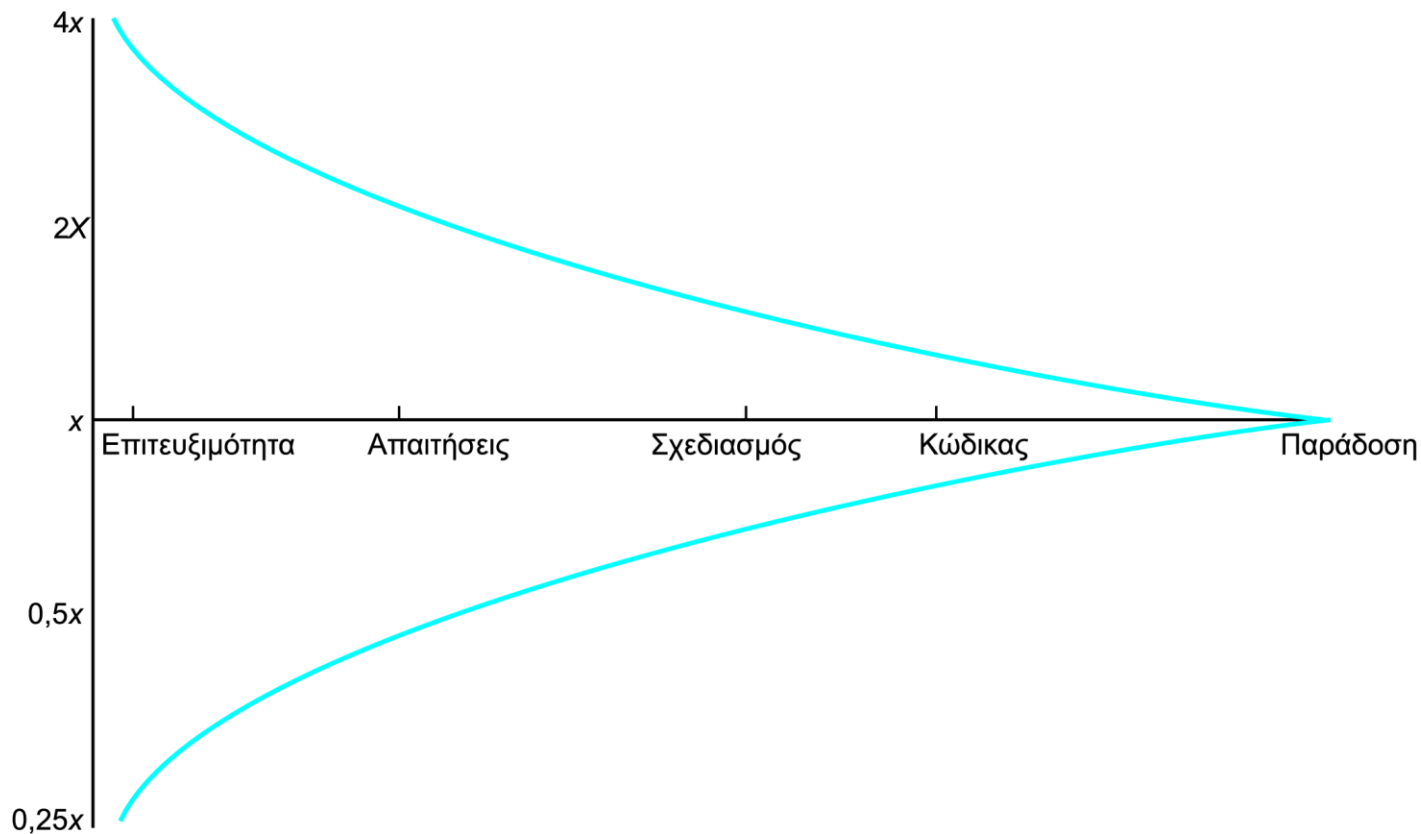
Αλγοριθμική μοντελοποίηση κόστους

- Η πρόβλεψη του κόστους ενός έργου γίνεται με τη χρήση ενός μαθηματικού τύπου που περιλαμβάνει παράγοντες σχετικούς με το προϊόν, το έργο και τη διαδικασία, των οποίων οι τιμές εκτιμώνται από τους διευθυντές του έργου:
 - $\text{Εργασία} = A \times \text{Μέγεθος}^B \times M$
 - Το A είναι σταθερά της οποίας η τιμή εξαρτάται από την κάθε εταιρεία, το B αντιπροσωπεύει το δυσανάλογο φόρτο εργασίας που πρέπει να καταβληθεί για τα μεγάλα έργα και το M είναι ένας πολλαπλασιαστής που προκύπτει από το συνδυασμό των ιδιοτήτων της διαδικασίας, του προϊόντος και των εμπλεκόμενων ατόμων.
- Η ιδιότητα που χρησιμοποιείται συνήθως για την εκτίμηση του κόστους είναι το μέγεθος του κώδικα.
- Τα περισσότερα μοντέλα μοιάζουν μεταξύ τους, αλλά χρησιμοποιούν διαφορετικές τιμές A, B και M.

Ακρίβεια εκτιμήσεων

- Το μέγεθος ενός συστήματος λογισμικού είναι γνωστό με ακρίβεια μόνο μετά την ολοκλήρωση της ανάπτυξης του συστήματος.
- Υπάρχουν πολλοί παράγοντες που επηρεάζουν το τελικό μέγεθος
 - Η χρήση εμπορικών εφαρμογών και έτοιμων συστατικών στοιχείων
 - Η γλώσσα προγραμματισμού
 - Η κατανομή του συστήματος
- Καθώς η διαδικασία της ανάπτυξης προχωρά, η εκτίμηση του μεγέθους του συστήματος γίνεται πιο ακριβής.

Αβεβαιότητα εκτίμησης



Το μοντέλο COCOMO

- Είναι ένα εμπειρικό μοντέλο που βασίζεται σε εμπειρικά δεδομένα έργων.
- Είναι ένα καλά τεκμηριωμένο, "ανεξάρτητο" μοντέλο το οποίο δεν δεσμεύεται από κάποιο συγκεκριμένο προμηθευτή λογισμικού.
- Έχει μακρόχρονη διαδρομή αφού ξεκίνησε από την πρώτη του υλοποίηση το 1981 (COCOMO-81) και έφτασε, μέσα από διάφορες εκδοχές, στην πιο πρόσφατη έκδοσή του, την COCOMO 2.
- Το COCOMO 2 λαμβάνει υπόψη διάφορες προσεγγίσεις ανάπτυξης λογισμικού, επαναχρησιμοποίησης, κ.λπ.

COCOMO 81

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

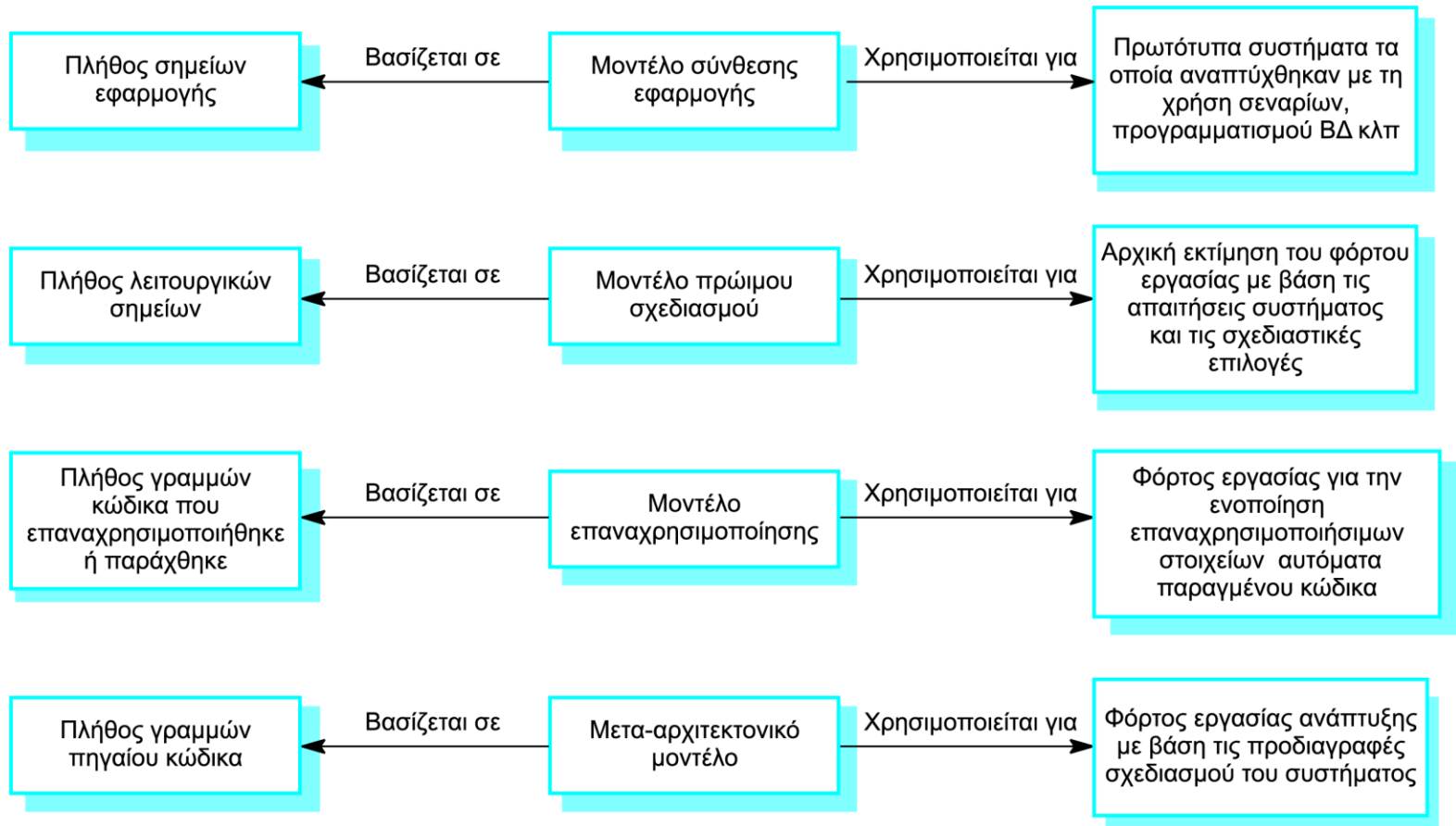
COCOMO 2

- Το COCOMO 81 αναπτύχθηκε με βάση την παραδοχή ότι θα χρησιμοποιούνταν διαδικασίες παραγωγής λογισμικού με βάση το μοντέλο καταρράκτη και ότι όλα τα προγράμματα θα αναπτύσσονταν εξαρχής.
- Από τη διατύπωση του μοντέλου αυτού, έχουν επέλθει πολλές αλλαγές στην τεχνολογία λογισμικού, οπότε το μοντέλο COCOMO 2 είναι σχεδιασμένο έτσι ώστε να μπορεί να εξυπηρετεί διάφορες προσεγγίσεις ανάπτυξης λογισμικού.

Μοντέλα COCOMO 2

- Το COCOMO 2 περιλαμβάνει πολλά δευτερεύοντα μοντέλα που μπορούν να παράγουν ολοένα και πιο λεπτομερείς εκτιμήσεις για το λογισμικό.
- Τα υπομοντέλα του COCOMO 2 είναι:
 - **Ένα μοντέλο σύνθεσης εφαρμογής.** Χρησιμοποιείται όταν το λογισμικό συντίθεται από υπάρχοντα στοιχεία.
 - **Ένα μοντέλο πρώιμου σχεδιασμού.** Χρησιμοποιείται όταν οι απαιτήσεις είναι διαθέσιμες, αλλά δεν έχει ξεκινήσει ακόμα ο σχεδιασμός.
 - **Ένα μοντέλο επαναχρησιμοποίησης.** Χρησιμοποιείται για τον υπολογισμό του φόρτου εργασίας που απαιτείται για την ενοποίηση των επαναχρησιμοποιήσιμων συστατικών στοιχείων.
 - **Ένα μετα-αρχιτεκτονικό μοντέλο.** Χρησιμοποιείται αφού έχει οριστικοποιηθεί η αρχιτεκτονική του συστήματος και υπάρχουν περισσότερες διαθέσιμες πληροφορίες για αυτό.

Χρήση των υπομοντέλων του COCOMO 2



Μοντέλο σύνθεσης εφαρμογής

- Υποστηρίζει έργα στα οποία απαιτείται δημιουργία πρωτοτύπων και έργα με εκτεταμένη επαναχρησιμοποίηση.
- Βασίζεται σε τυποποιημένες εκτιμήσεις της παραγωγικότητας των προγραμματιστών εκπεφρασμένες σε σημεία εφαρμογής (αντικειμενικά σημεία ανά μήνα).
- Προβλέπει τη χρήση εργαλείων CASE.
- Ο μαθηματικός τύπος είναι:
 - $PM = (NAP \times (1 - \%reuse/100)) / PROD$
 - PM είναι ο φόρτος εργασίας σε ανθρωπομήνες, NAP είναι το πλήθος των σημείων εφαρμογής και PROD είναι η παραγωγικότητα.

Παραγωγικότητα βάσει αντικειμενικών σημείων

Πείρα και ικανότητες προγραμματιστών	Πολύ χαμηλή	Χαμηλή	Μέση	Υψηλή	Πολύ υψηλή
Ωριμότητα και δυνατότητες εργαλείων CASE	Πολύ χαμηλή	Χαμηλή	Μέση	Υψηλή	Πολύ υψηλή
PROD (NAP/μήνα)	4	7	13	25	50

Μοντέλο πρώιμου σχεδιασμού

- Αφού συμφωνηθούν οι απαιτήσεις, μπορούν να γίνουν οι εκτιμήσεις.
- Βασίζεται σε έναν τυπικό μαθηματικό τύπο αλγοριθμικών μοντέλων
 - Εργασία = $A \times \text{Μέγεθος}^B \times M$ όπου
 - $M = \text{PERS} \times \text{RCPX} \times \text{RUSE} \times \text{PDIF} \times \text{PREX} \times \text{FCIL} \times \text{SCED}$
 - $A = 2,94$ για την αρχική βαθμονόμηση, Μέγεθος σε KLOC, B κυμαίνεται από 1,1 έως 1,24 ανάλογα με το βαθμό καινοτομίας του έργου, την ευελιξία ανάπτυξης, τις χρησιμοποιούμενες διαδικασίες αντιμετώπισης κινδύνων και το επίπεδο ωριμότητας διαδικασιών.

Πολλαπλασιαστές

- Οι πολλαπλασιαστές αντιπροσωπεύουν τις δυνατότητες των προγραμματιστών, τις μη λειτουργικές απαιτήσεις, το βαθμό εξοικείωσης με την πλατφόρμα ανάπτυξης, κ.λπ.
 - RCPX - αξιοπιστία και πολυπλοκότητα του προϊόντος
 - RUSE - απαιτούμενη επαναχρησιμοποίηση
 - PDIF - δυσκολία της πλατφόρμας
 - PREX - πείρα προσωπικού
 - PERS - δυνατότητες προσωπικού
 - SCED - χρονοδιάγραμμα
 - FCIL - λειτουργίες υποστήριξης της ομάδας ανάπτυξης

Μοντέλο επαναχρησιμοποίησης

- Λαμβάνει υπόψη τον κώδικα μαύρου κουτιού που επαναχρησιμοποιείται χωρίς να απαιτείται η τροποποίησή του και τον κώδικα που πρέπει να προσαρμοστεί για να ενοποιηθεί με νέο κώδικα.
- Υπάρχουν δύο εκδοχές:
 - Επαναχρησιμοποίηση κώδικα μαύρου κουτιού, όπου ο κώδικας δεν τροποποιείται. Υπολογίζεται μια εκτίμηση του φόρτου εργασίας (PM).
 - Επαναχρησιμοποίηση κώδικα λευκού κουτιού, όπου ο κώδικας τροποποιείται. Υπολογίζεται μια εκτίμηση για το μέγεθος η οποία είναι ισοδύναμη με το πλήθος γραμμών νέου πηγαίου κώδικα. Με τον τρόπο αυτό προσαρμόζεται η εκτίμηση μεγέθους που αφορά το νέο κώδικα.

Εκτιμήσεις του μοντέλου επαναχρησιμοποίησης (1)

- Για κώδικα που παράγεται αυτόματα:
 - $PM = (ASLOC * AT/100)/ATPROD$
 - ASLOC είναι το πλήθος γραμμών κώδικα που παράγεται αυτόματα
 - AT είναι το ποσοστό του κώδικα που παράγεται αυτόματα
 - ATPROD είναι η παραγωγικότητα των μηχανικών κατά την ενοποίηση αυτού του κώδικα

Εκτιμήσεις του μοντέλου επαναχρησιμοποίησης (2)

- Όταν απαιτείται κατανόηση και ενοποίηση του κώδικα:
 - $ESLOC = ASLOC * (1-AT/100) * AAM$
 - Οι ASLOC και AT όπως παραπάνω
 - AAM είναι ο ρυθμιστικός πολλαπλασιαστής προσαρμογής που υπολογίζεται με βάση το κόστος της πραγματοποίησης αλλαγών στον επαναχρησιμοποιούμενο κώδικα, το κόστος του τρόπου ενοποίησης του κώδικα και το κόστος της λήψης αποφάσεων επαναχρησιμοποίησης.

Μετα-αρχιτεκτονικό επίπεδο

- Χρησιμοποιείται ο ίδιος μαθηματικός τύπος όπως και στο μοντέλο πρώιμου σχεδιασμού, αλλά με 17 πολλαπλασιαστές και όχι μόνο 7.
- Η εκτίμηση του μεγέθους του κώδικα γίνεται με βάση:
 - Μια εκτίμηση του συνολικού πλήθους γραμμών νέου κώδικα που πρόκειται να αναπτυχθεί
 - Μια εκτίμηση του ισοδύναμου πλήθους γραμμών πηγαίου κώδικα που υπολογίζεται με το μοντέλο επαναχρησιμοποίησης
 - Μια εκτίμηση του πλήθους γραμμών κώδικα που πρέπει να τροποποιηθούν λόγω αλλαγών στις απαιτήσεις

Ο εκθετικός όρος

- Εξαρτάται από 5 παράγοντες κλίμακας (δείτε την επόμενη διαφάνεια). Το άθροισμά τους, διαιρεμένο με 100, προστίθεται στην τιμή 1,01
- Μια εταιρεία αναλαμβάνει ένα έργο σε ένα νέο πεδίο εφαρμογής. Ο πελάτης δεν έχει ορίσει τη διαδικασία που θα χρησιμοποιηθεί και δεν έχει προβλέψει χρόνο για να γίνει ανάλυση κινδύνων. Η εταιρεία έχει βαθμολογηθεί ως εταιρεία Επιπέδου 2 σύμφωνα με το μοντέλο CMM.
 - Πείρα - νέο έργο (4)
 - Ευελιξία ανάπτυξης - δεν εμπλέκεται ο πελάτης - πολύ υψηλή (1)
 - Αρχιτεκτονική ανάλυση/ανάλυση κινδύνων - δεν διενεργείται ανάλυση κινδύνων - πολύ χαμηλή (5)
 - Συνεκτικότητα ομάδας - νέα ομάδα - μέση (3)
 - Ωριμότητα διαδικασιών - υπάρχει ένας βαθμός ελέγχου των διαδικασιών - μέση (3)
- Επομένως ο συντελεστής κλίμακας είναι 1,17.

Συντελεστές κλίμακας που χρησιμοποιούνται κατά τον υπολογισμό του εκθέτη

Πείρα	Αντικατοπτρίζει την προϋπάρχουσα πείρα της εταιρείας σε αυτό τον τύπο έργων. Πολύ χαμηλός βαθμός σημαίνει ότι δεν υπάρχει προϋπάρχουσα πείρα· Εξαιρετικά υψηλός σημαίνει ότι η εταιρεία είναι πλήρως εξοικειωμένη με αυτό τον τομέα εφαρμογών.
Ευελιξία ανάπτυξης	Αντικατοπτρίζει το βαθμό ευελιξίας στη διαδικασία ανάπτυξης. Πολύ χαμηλός βαθμός σημαίνει ότι χρησιμοποιείται μια προκαθορισμένη διαδικασία· Εξαιρετικά υψηλός σημαίνει ότι ο πελάτης καθορίζει μόνο γενικούς στόχους.
Αρχιτεκτονική ανάλυση/ανάλυση κινδύνων	Αντικατοπτρίζει το βαθμό της ανάλυσης κινδύνων που διενεργείται. Πολύ χαμηλός βαθμός σημαίνει λίγη ανάλυση· Εξαιρετικά υψηλός συνεπάγεται πλήρη και διεξοδική ανάλυση κινδύνων.
Συνεκτικότητα ομάδας	Αντικατοπτρίζει πόσο καλά γνωρίζονται και συνεργάζονται τα μέλη της ομάδας ανάπτυξης. Πολύ χαμηλός βαθμός σημαίνει πολύ προβληματικές συναναστροφές· Εξαιρετικά υψηλός αντιστοιχεί σε ολοκληρωμένη και αποτελεσματική ομάδα χωρίς προβλήματα επικοινωνίας.
Ωριμότητα διαδικασιών	Αντικατοπτρίζει την ωριμότητα των διαδικασιών της εταιρείας. Ο υπολογισμός αυτής της τιμής εξαρτάται από το Ερωτηματολόγιο Ωριμότητας του μοντέλου CMM, αλλά μπορεί να γίνει μια εκτίμησή της με την αφαίρεση του επίπεδου ωριμότητας διαδικασιών CMM από την τιμή 5.

Πολλαπλασιαστές

- **Ιδιότητες του προϊόντος**
 - Αφορούν τα απαιτούμενα χαρακτηριστικά του προϊόντος λογισμικού που αναπτύσσεται.
- **Υπολογιστικές ιδιότητες**
 - Οι περιορισμοί που επιβάλλονται στο λογισμικό από την πλατφόρμα υλικού.
- **Ιδιότητες του προσωπικού**
 - Πολλαπλασιαστές που λαμβάνουν υπόψη την πείρα και τις ικανότητες των ατόμων που απασχολούνται στο έργο.
- **Ιδιότητες του έργου**
 - Αφορούν τα ιδιαίτερα χαρακτηριστικά του έργου ανάπτυξης λογισμικού.

Η επίδραση των κατευθύνσεων κόστους

Τιμή εκθέτη

1,17

Μέγεθος συστήματος (συμπεριλαμβανομένων των 128.000 DSI
συντελεστών για επαναχρησιμοποίηση και
μεταβλητότητα απαιτήσεων)

Αρχική εκτίμηση COCOMO χωρίς κατευθύνσεις κόστους

730 ανθρωπομήνες

Αξιοπιστία

Πολύ υψηλή, πολλαπλασιαστής = 1,39

Πολυπλοκότητα

Πολύ υψηλή, πολλαπλασιαστής = 1,3

Περιορισμοί μνήμης

Υψηλοί, πολλαπλασιαστής = 1,21

Χρήση εργαλείων

Χαμηλή, πολλαπλασιαστής = 1,12

Χρονοδιάγραμμα

Επισπευσμένο, πολλαπλασιαστής = 1,29

Προσαρμοσμένη εκτίμηση COCOMO

2306 ανθρωπομήνες

Αξιοπιστία

Πολύ χαμηλή, πολλαπλασιαστής = 0,75

Πολυπλοκότητα

Πολύ χαμηλή, πολλαπλασιαστής = 0,75

Περιορισμοί μνήμης

Κανένας, πολλαπλασιαστής = 1

Χρήση εργαλείων

Πολύ υψηλή, πολλαπλασιαστής = 0,72

Χρονοδιάγραμμα

Κανονικό, πολλαπλασιαστής = 1

Προσαρμοσμένη εκτίμηση COCOMO

295 ανθρωπομήνες

Ποια επιλογή θα προτιμηθεί;

- Η επιλογή Δ (χρήση πιο πεπειραμένου προσωπικού) φαίνεται να είναι η καλύτερη εναλλακτική
 - Ωστόσο, ο σχετικός κίνδυνος είναι υψηλός επειδή το πεπειραμένο προσωπικό είναι δυσεύρετο.
- Η επιλογή Γ (αναβάθμιση μνήμης) επιφέρει μικρότερη εξοικονόμηση, αλλά ενέχει πολύ μικρό κίνδυνο.
- Σε γενικές γραμμές, το μοντέλο αποκαλύπτει τη σημασία της πείρας του προσωπικού που εμπλέκεται στην ανάπτυξη λογισμικού.

Διάρκεια και στελέχωση έργων

- Εκτός από την εκτίμηση του φόρτου εργασίας, οι διευθυντές έργων πρέπει να εκτιμήσουν επίσης πόσος ημερολογιακός χρόνος θα χρειαστεί για την ολοκλήρωση του λογισμικού και τότε θα χρειαστούν προσωπικό για να απασχοληθεί στο έργο.
- Εκτίμηση του ημερολογιακού χρόνου μπορεί να γίνει με τη χρήση ενός μαθηματικού τύπου του μοντέλου COCOMO 2
 - $TDEV = 3 \times (PM)(0,33+0,2*(B-1,01))$
 - PM είναι η τιμή του φόρτου εργασίας και B είναι ο υπολογισμένος εκθέτης, όπως αναφέρθηκε παραπάνω (το B είναι ίσο με 1 για το πρώιμο μοντέλο πρωτοτύπων). Αυτός ο υπολογισμός προβλέπει το ονομαστικό χρονοδιάγραμμα του έργου.
- Ο απαιτούμενος χρόνος δεν εξαρτάται από το πλήθος των ατόμων που απασχολούνται στο έργο.

Κύρια σημεία

- Δεν υπάρχει μια απλή σχέση μεταξύ της τιμής με την οποία χρεώνεται ένα σύστημα και του κόστους ανάπτυξής του.
- Στους παράγοντες που επηρεάζουν την παραγωγικότητα συγκαταλέγονται οι ατομικές δεξιότητες, η πείρα στο πεδίο εφαρμογής, η διαδικασία ανάπτυξης, το μέγεθος του έργου, η υποστήριξη εργαλείων και το εργασιακό περιβάλλον.
- Συχνά η τιμολόγηση του λογισμικού στοχεύει στη σύναψη μιας σύμβασης και η λειτουργικότητά του προσαρμόζεται με βάση την τιμή που συμφωνήθηκε.

Κύρια σημεία

- Κατά την εκτίμηση του κόστους πρέπει να χρησιμοποιούνται διάφορες τεχνικές.
- Κατά την πρόβλεψη του απαιτούμενου φόρτου εργασίας, το μοντέλο COCOMO συνυπολογίζει ιδιότητες του έργου, του προϊόντος, του υλικού και του προσωπικού.
- Τα αλγοριθμικά μοντέλα κόστους υποστηρίζουν την ποσοτική ανάλυση επιλογών καθώς επιτρέπουν τη σύγκριση του κόστους των διαφόρων επιλογών.
- Ο χρόνος που απαιτείται για την ολοκλήρωση ενός έργου δεν είναι απλώς ανάλογος του αριθμού των ατόμων που απασχολούνται στο έργο.