ΚΕΦΑΛΑΙΟ 2ο

ΣΧΕΔΙΑΣΜΟΣ ΕΡΓΟΥ ΚΑΙ ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ

Παράγοντες που πρέπει να ληφθούν υπόψη στο σχεδιασμό του έργου:

- Τεχνικές προϋπολογισμού (χρειάζεται ακρίβεια).
- Μοντέλο κύκλου-ζωής. (π.χ. Μοντέλο του Καταρράκτη)
- Δομή οργάνωσης.
- Επίπεδο αυστηρότητας στην παρουσίαση των προδιαγραφών.
- Επίπεδο επαλήθευσης και επικύρωσης.
- Επίπεδο Διασφάλισης ποιότητας.
- Ευθύνες συντήρησης που θα ακολουθήσουν.
- Εργαλεία που πρέπει να αναπτυχθούν και να χρησιμοποιηθούν.
- Πρόσληψη προσωπικού και εκπαίδευση.

Παράγοντες που πρέπει να ληφθούν υπόψη στην τοποθέτηση στόχων (goals):

- Νέες ικανότητες που πρέπει να παραχθούν.
- Παλιές ικανότητες που πρέπει να διατηρηθούν.
- Επίπεδο πολυπλοκότητας χρήσης.
- Απαιτήσεις αποδοτικότητας.
- Απαιτήσεις αξιοπιστίας.
- Πιθανές αλλαγές.
- Προτεραιότητες υλοποίησης.
- Απαιτήσεις μεταφερσιμότητας.
- Προβλήματα ασφαλείας

Υπολογισμός κόστους λογισμικού(1).

- Είναι μια από τις πιο δύσκολες δουλειές με μεγάλο κίνδυνο να γίνουν λάθη.
- Είναι δύσκολο να υπάρξει **ακρίβεια** κατά τη διάρκεια του σχεδιασμού της ανάπτυξης λογισμικού γιατί υπάρχουν πολλοί **άγνωστοι παράγοντες** σ' αυτή τη φάση.
- Παρ' όλ' αυτά ο υπολογισμός κόστους είναι αναγκαίος για να γίνει **ανάληψη έργου** και είναι μέρος της **μελέτης εφικτότητας.**

Υπολογισμός κόστους λογισμικού(2).

- Το πρόβλημα του υπολογισμού κόστους είναι δύσκολο, γι'αυτό το λόγο μερικοί οργανισμοί χρησιμοποιούν μια **σειρά από τέτοιους υπολογισμούς.**

- Για παράδειγμα:

  Ένας πρωταρχικός υπολογισμός κατά τη διάρκεια του σχεδιασμού του έργου.

  Ένας βελτιωμένος υπολογισμός όταν έχουν ετοιμαστεί οι Απαιτήσεις του Λογισμικού.

  Ένας τελικός υπολογισμός στο τέλος του πρωταρχικού σχεδιασμού του συστήματος.

Παράγοντες που επηρεάζουν το κόστος λογισμικού.

- Η ικανότητα των προγραμματιστών.
- Η πολυπλοκότητα του προϊόντος.
- Το μέγεθος του προϊόντος.
- Ο διαθέσιμος χρόνος.
- Η απαιτούμενη αξιοπιστία.
- Το επίπεδο Τεχνολογίας

Η ΙΚΑΝΟΤΗΤΑ ΤΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΩΝ

Στα μεγάλα έργα η ικανότητα του κάθε προγραμματιστή δε φαίνεται, σε αντίθεση, με τα έργα πέντε ή λιγότερων προγραμματιστών.

Η πολυπλοκότητα του προϊόντος.

Σε αυτή τη φάση θα θεωρήσουμε 3 μεγάλες κατηγορίες λογισμικού:

- **Προγράμματα εφαρμογών:**

  Επεξεργασία δεδομένων και επιστημονικά προγράμματα.

- **Προγράμματα χρησιμότητας (utility):**

  π.χ. μεταγλωττιστές, linkage editors.

- **Προγράμματα συστήματος:**

  π.χ. λειτουργικά συστήματα, συστήματα real time

ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΛΟΓΙΣΜΙΚΟΥ ΚΑΤΑ BROOKS

- Ο Brooks (Bro 74) αναφέρει ότι τα **προγράμματα χρησιμότητας** είναι τρεις φορές πιο δύσκολο να γραφούν απ' ότι τα **προγράμματα εφαρμογών** και τα **προγράμματα συστήματος** τρεις φορές πιο δύσκολα από τα **προγράμματα χρησιμότητας.**

- Τα επίπεδα πολυπλοκότητας κατά Brooks είναι 1 - 3 - 9 για προγράμματα εφαρμογών - χρησιμότητας - συστήματος αντίστοιχα.

ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΛΟΓΙΣΜΙΚΟΥ ΚΑΤΑ BOEHM(1).

Κατά το Boehm υπάρχουν **τρία επίπεδα πολυπλοκότητας:**

- **Οργανικά προγράμματα (organic),**
- **Ημιεφαπτόμενα προγράμματα (semidetached),**
- **Ενσωματωμένα προγράμματα (embedded),**

ΥΠΟΛΟΓΙΣΜΟΣ ΤΩΝ ΕΠΙΠΕΔΩΝ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

Ο **Boehm** δίνει **εξισώσεις** για να υπολογιστούν τα τρία επίπεδα πολυπλοκότητας:

1. Οι μήνες προγραμματιστή της προσπάθειας (PM) programmer-months.

Ένας μήνας προγραμματιστή=19 ημέρες * 8 ώρες / ημέρα = 152 ώρες

2. Ο χρόνος ανάπτυξης για ένα πρόγραμμα (TDEV) development time.

   Εξαρτάται από τον αριθμό των εντολών πηγαίου προγράμματος: (KDSI) Thousands of Delivered Source Instructions. Ένα DSI θεωρείται ότι είναι μια γραμμή.

ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΛΟΓΙΣΜΙΚΟΥ ΚΑΤΑ ΒΟΕΗΜ(2).

- **Υπολογισμός μηνών προγραμματιστή**

Προγράμματα εφαρμογών	PM = 2.4 * (KDSI) **1.05
Προγράμματα χρησιμότητας	PM = 3.0 * (KDSI) ** 1.12
Προγράμματα συστημάτων	PM = 3.6 * (KDSI) ** 1.20

Από τον πίνακα προκύπτει ότι χρειάζεται **διπλή προσπάθεια** για ένα πρόγραμμα χρησιμότητας και **τριπλή προσπάθεια** για ένα πρόγραμμα συστήματος απ' ότι για ένα πρόγραμμα εφαρμογών.

ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΛΟΓΙΣΜΙΚΟΥ ΚΑΤΑ ΒΟΕΗΜ(2)(συν).

- **Υπολογισμός χρόνου ανάπτυξης για ένα πρόγραμμα:**

Πρόγραμμα εφαρμογών: **TDEV = 2.5 * (PM) * * 0.38**
Πρόγραμμα χρησιμότητας: **TDEV = 2.5 * (PM) * * 0.35**
Πρόγραμμα συστήματος: **TDEV = 2.5 * (PM) * * 0.32**

- **ΣΗΜΕΙΩΣΕΙΣ:**

1. Αν γνωρίζουμε το χρόνο ανάπτυξης προσδιορίζουμε τους μήνες προγραμματιστή και αντίστροφα.

2. Αν δοθούν οι μήνες προγραμματιστή και ο χρόνος ανάπτυξης μπορούμε να βρούμε πόσο προσωπικό χρειαζόμαστε με μία διαίρεση:

Για 60 KDSI (60.000 γραμμές)

Εφαρμογών: 176.6 PM / 17.85 MO = 9.9 προγραμματιστές
Χρησιμότητας: 294 PM / 18.3 MO = 16 προγραμματιστές
Συστήματος: 489.6 PM / 18.1 MO = 27 προγραμματιστές

ΛΟΙΠΟΙ ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΕΠΗΡΕΑΖΟΥΝ ΤΟ ΚΟΣΤΟΣ ΛΟΓΙΣΜΙΚΟΥ.

3. Το μέγεθος του προϊόντος.

Το μεγάλο έργο είναι πιο ακριβό από ένα μικρό.

4. Διαθέσιμος χρόνος.

Ένα έργο λογισμικού χρειάζεται μεγαλύτερη συνολική προσπάθεια αν συμπιέσουμε το χρόνο ανάπτυξης ή τον μεγαλώσουμε περισσότερο από τον κανονικό.

5. Απαιτούμενο επίπεδο αξιοπιστίας.

Η αξιοπιστία ορίζεται σαν την πιθανότητα να εκτελέσει ένα πρόγραμμα μια απαιτούμενη λειτουργία κάτω από προκαθορισμένες συνθήκες για ένα προκαθορισμένο διάστημα χρόνου.

Επηρεάζεται από την ακρίβεια, την ευρωστία, την πληρότητα, την συνέπεια.

Πολλαπλασιαστές προσπάθειας ανάπτυξης για αξιοπιστία λογισμικού (Boehm).

Κατηγορία αξιοπιστίας	Κόστος αποτυχίας	Πολλαπλασιαστής προσπάθειας
1) Πολύ χαμηλή	Μικρή ενόχληση	0.75
2) Χαμηλή	Η ζημιά επανορθώνεται εύκολα	0.88
3) Κανονική	Η ζημιά επανορθώνεται μετρίως δύσκολα	1.00
4) Υψηλή	Μεγάλο οικονομικό κόστος	1.15
5) Πολύ υψηλή	Κίνδυνος σε ανθρώπινη ζωή	1.40

ΕΠΙΠΕΔΟ ΤΕΧΝΟΛΟΓΙΑΣ

Το επίπεδο Τεχνολογίας επηρεάζεται από:

- τη γλώσσα προγραμματισμού
- τη νοητή μηχανή (hardware/software)
- τις τεχνικές προγραμματισμού
- τα εργαλεία λογισμικού που χρησιμοποιούνται

Το Αλγοριθμικό μοντέλο κόστους COCOMO

- Το κυριότερο αλγοριθμικό μοντέλο κόστους και περιγράφτηκε από τον Boehm 81.
- Χρησιμοποιεί εξισώσεις που υπολογίζουν μήνες προγραμματιστή και χρόνο ανάπτυξης και προσαρμόζουν το αποτέλεσμα ανάλογα με τους πολλαπλασιαστές προσπάθειας.
- Οι εξισώσεις αυτές προϋποθέτουν κάποιες παραδοχές.

  Παράδειγμα:

  Οι εξισώσεις για **κανονικά οργανικά** προγράμματα (εφαρμογών) ισχύουν για τις εξής περιπτώσεις:

  1) Για μικρά έως μέτρια έργα στο μέγεθος (2K μέχρι 32K DSI).
  2) Για οικεία περιοχή εφαρμογών.
  3) Σταθερή νοητή μηχανή και κατανοητή.
  4) "In-house" (εντός-οίκου) προσπάθεια ανάπτυξης
- Χρησιμοποιεί συστηματικές τεχνικές της Τεχνολογίας Λογισμικού καθόλη τη διάρκεια της ανάπτυξης.

COCOMO ΠΟΛΛΑΠΛΑΣΙΑΣΤΕΣ ΠΡΟΣΠΑΘΕΙΑΣ

Πολλαπλασιαστής	Περιοχή τιμών
Χαρακτηριστικά προϊόντος	
Απαιτούμενη αξιοπιστία	0.75 μέχρι 1.40
Μέγεθος data-base	0.94 μέχρι 1.16
Πολυπλοκότητα	0.70 μέχρι 1.65
Χαρακτηριστικά υπολογιστή	
Περιορισμός χρόνου εκτέλεσης	1.00 μέχρι 1.66
Περιορισμός κύριας μνήμης	1.00 μέχρι 1.56
Χαρακτηριστικά προσωπικού	
Ικανότητα Αναλυτή	1.46 μέχρι 0.71
Ικανότητα Προγραμματιστή	1.42 μέχρι 0.70
Εμπειρία εφαρμογών	1.29 μέχρι 0.82
Εμπειρία νοητής μηχανής	1.21 μέχρι 0.90
Εμπειρία γλώσσας προγραμματισμού	1.14 μέχρι 0.95
Χαρακτηριστικά έργου	
Χρήση σύγχρονων τεχνικών προγραμματισμού	1.24 μέχρι 0.82
Χρήση εργαλείων λογισμικού	1.24 μέχρι 0.83

Παράδειγμα αλγοριθμικού υπολογισμού κόστους με τη χρήση του COCOMO

- **Πρόβλημα:**

Το προϊόν που πρέπει να αναπτυχθεί είναι μεγέθους 10.000 γραμμών, είναι τύπου ενσωματωμένου λογισμικού για επεξεργασία τηλεπικοινωνιών σε έναν μικροεπεξεργαστή.

Λύση:

Ισχύει 10000 γραμμές=10KDSI

Κανονικές εξισώσεις: PM=3.6*(10)**1.20=57.05

TDEV=2.5*(57)**0.32=9.11

Οι πολλαπλασιαστές προσπάθειας χρησιμοποιούνται για να προσαρμόσουν τον υπολογισμό στις μη-κανονικές πλευρές του έργου.

Συντελεστής προσαρμοσμένης προσπάθειας =1.00*0.94*1.30…=1.17

Άρα,Προσαρμοσμένοι: PM=57.05*1.17=66.74

TDEV=9.11*1.17=10.65

Το χρηματικό κόστος σε προσωπικό:

Συνολικό κόστος = (66.74 PM) * (κόστος ανά PM)

Κόστος ανά PM = μισθός προγραμματιστή ή αναλυτή

Πολλαπλασιαστές προσπάθειας για ενσωματωμένο λογισμικό τηλεπικοινωνιών.

Πολλαπλασιαστής Αξιοπιστία	**Σκεπτικό** Μόνο για τοπική χρήση. Όχι σοβαρά προβλήματα επανόρθωσης.	**Τιμή** 1.00
Βάση Δεδομένων	20.000 bytes (χαμηλή).	0.94
Πολυπλοκότητα	Επεξεργασία τηλεπικοινωνιών (πολύ υψηλή).	1.30
Χρόνος	Θα χρησιμοποιήσει 70% του χρόνου επεξεργασίας (υψηλό).	1.11
Μνήμη	Θα χρησιμοποιήσει 70% της διαθέσιμης κύριας μνήμης (υψηλό).	1.06
Μηχανή	Σταθερή. Ο μικροεπεξεργαστής είναι διαθέσιμος στο εμπόριο (κανονικό).	1.00
.Αναλυτές	Πεπειραμένοι (υψηλό).	0.86
Προγραμματιστές	Πεπειραμένοι (υψηλό).	0.86
Πείρα	3 χρόνια στις τηλεπικοινωνίες (κανονικό).	1.00
Πείρα	6 μήνες στον μικροεπεξεργαστή (χαμηλό).	1.10
Τεχνικές	Περισσότερο από ένα χρόνο πείρα σε σύγχρονες τεχνικές (υψηλό).	0.91
Εργαλεία	Βασικό λογισμικό για μικροεπεξεργαστή.	1.10
Χρονοδιάγραμμα	Εννέα μήνες όταν TDEV = 8.4 μήνες (κανονικό).	1.00

ΥΠΟΛΟΓΙΣΜΟΣ ΚΟΣΤΟΥΣ ΜΕ ΤΟ COCOMO

1. Καθορίζουμε ποια είναι τα υποσυστήματα και κομμάτια του προϊόντος.
2. Υπολογίζουμε το μέγεθος κάθε κομματιού και υπολογίζουμε το μέγεθος ολόκληρου του συστήματος
3. Καθορίζουμε τους πολλαπλασιαστές προσπάθειας για κάθε κομμάτι. Οι πολλαπλασιαστές σε επίπεδο κομματιού είναι:

   πολυπλοκότητα,

   ικανότητα προγραμματιστή,

   εμπειρία στη μηχανή,

   εμπειρία στη γλώσσα προγραμματισμού.
4. Υπολογίζουμε προσπάθεια και χρόνο ανάπτυξης για κάθε κομμάτι υπό κανονικές συνθήκες και μετά χρησιμοποιώντας τους πολλαπλασιαστές σε επίπεδο κομματιού.

ΥΠΟΛΟΓΙΣΜΟΣ ΚΟΣΤΟΥΣ ΜΕ ΤΟ COCOMO(συν.)

5. Καθορίζουμε τους υπόλοιπους πολλαπλασιαστές για κάθε υποσύστημα
6. Λαμβάνοντας υπόψη τα αποτελέσματα από τα βήματα 4 και 5 υπολογίζουμε την **προσπάθεια** και **χρόνο** ανάπτυξης για κάθε υποσύστημα.
7. Από το 6 υπολογίζουμε **συνολική** προσπάθεια και **χρόνο**.
8. Κάνουμε μια **ανάλυση ευαισθησίας** στον υπολογισμό για να δούμε που να δώσουμε έμφαση.
9. Προσθέτουμε κάθε άλλο κόστος ανάπτυξης που δεν συμπεριλαμβάνεται στον υπολογισμό (π.χ. η ανάλυση).
10. Συγκρίνουμε τον υπολογισμό με άλλον υπολογισμό που βγαίνει με άλλο τρόπο. Ομαλοποιούμε τις διαφορές.

Τα (+) και (-) του COCOMO

- **Πλεονέκτημα**

  Το μοντέλο χρησιμοποιείται για να καταλάβουμε περισσότερο τους παράγοντες κόστους σε έναν οργανισμό.

- **Μειονέκτημα**

  Προϋποθέτει ότι οι πολλαπλασιαστές προσπάθειας είναι ανεξάρτητοι μεταξύ τους ενώ στην πραγματικότητα ο ένας εξαρτάται από τον άλλο.

ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΚΑΤΑΡΡΑΚΤΗ

Έγγρφα που χρειάζονται σε κάθε φάση του μοντέλου του καταρράκτη

Εργασία	Έγγραφα
Ανάλυση Απαιτήσεων	1) Μελέτη εφικτότητας. 2) Σκελετός απαιτήσεων.
Ορισμός Απαιτήσεων	Προσδιορισμός Απαιτήσεων (Specification).
Καθορισμός Συστήματος	1) Λειτουργικός καθορισμός. 2) Προσδιορισμός του ελέγχου αποδοχής προϊόντος. 3) Πρόχειρο εγχειρίδιο χρήστη (user manual).
Αρχιτεκτονικός Σχεδιασμός	1) Προσδιορισμός αρχιτεκτονικού σχεδίου. 2) Προσδιορισμός ελέγχων συστήματος.
Σχεδιασμός Interface	1) Προσδιορισμός interface. 2) Προσδιορισμός ελέγχου ακεραιότητας.
Αναλυτική Σχεδίαση	1) Προσδιορισμός σχεδιασμού. 2) Προσδιορισμός ελέγχου ενοτήτων.
Κωδικοποίηση	Πρόγραμμα.
Έλεγχος ενότητας (για κάθε ενότητα).	Αναφορά αποτελέσματος ελέγχου ενότητας.
Έλεγχος ακεραιότητας	1) Αναφορά αποτελέσματος ελέγχου. 2) Τελικό **εγχειρίδιο χρήστη**.
Έλεγχος συστήματος	Αναφορά ελέγχου συστήματος.
Έλεγχος αποδοχής	Τελικό σύστημα.