



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

# Ανάλυση Απαιτήσεων Εκτίμηση Κόστους Λογισμικού

Μάρα Νικολαΐδου

# 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

# 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

	<b>Ανάλυση</b>	<b>Σχεδιασμός</b>	<b>Συγγραφή κώδικα</b>	<b>Δοκιμές</b>	<b>Τεκμηρίωση</b>
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

	<b>Μέγεθος</b>	<b>Εργασία</b>	<b>Παραγωγικότητα</b>
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 κατανομημένων 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.



# 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
- Ο απαιτούμενος χρόνος δεν εξαρτάται από το πλήθος των ατόμων που απασχολούνται στο έργο.

# Απαιτήσεις στελέχωσης

- Το απαιτούμενο προσωπικό δεν υπολογίζεται από τη διαίρεση του χρόνου ανάπτυξης με το απαιτούμενο χρονοδιάγραμμα.
- Το πλήθος των ατόμων που απασχολούνται σε ένα έργο ποικίλλει ανάλογα με τη φάση του έργου.
- Όσο περισσότερα άτομα απασχολούνται σε ένα έργο, τόσο περισσότερος είναι ο συνολικός φόρτος εργασίας που απαιτείται συνήθως.
- Ενδεχομένη εξαιρετικά ταχεία αύξηση των απασχολούμενων ατόμων σχετίζεται με αποκλίσεις στο χρονοδιάγραμμα.

# 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.

# Ενδεικτική Βιβλιογραφία

- Κεφάλαιο 28<sup>ο</sup> «Εκτίμηση Κόστους Λογισμικού» του βιβλίου «*Βασικές Αρχές Τεχνολογίας Λογισμικού*» του Ian Sommerville
- Κεφάλαιο 1<sup>ο</sup> *Εισαγωγή στην Ανάλυση και το Σχεδιασμό Συστημάτων – Μελέτη Σκοπιμότητας* (σελ. 25-65) του βιβλίου «*Ανάλυση και Σχεδιασμός Συστημάτων με τη UML 2.0*» των Davis, Wixon, Tegarden. ΚΑΙ το Παράρτημα 1