

ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ ΕΡΓΩΝ

Π. ΦΙΤΣΙΛΗΣ

Διαχείριση Έργων Λογισμικού

- ▶ Η διαχείριση έργων λογισμικού περιλαμβάνει διαφορετικά θέματα όπως:
 - ▶ Υπολογισμός κόστους και όγκου εργασίας για τη ανάπτυξη συστημάτων λογισμικού (Cost and Effort estimation)
 - ▶ Υπολογισμός απαιτούμενων πόρων ανθρώπινου δυναμικού (Staffing)
 - ▶ Επιλογή, και διαχείριση της διαδικασίας ανάπτυξης συστημάτων λογισμικού
 - ▶ Χρονικός προγραμματισμός των εργασιών που είναι σχετικές με τη ανάπτυξη συστημάτων λογισμικού (Scheduling activities)
 - ▶ Παρακολούθηση και έλεγχος ποιοτικών χαρακτηριστικών του έργου και του συστήματος υπό ανάπτυξη (Monitoring quality)
 - ▶ ...
- ▶ Σε επίπεδο οργανισμού, τόσο οι διαδικασίες για τη διαχείριση έργων λογισμικού, όσο και η δομή του ίδιου του οργανισμού θα πρέπει συνεχώς να εξελίσσονται και να αναβαθμίζονται



Εκτίμηση παραγωγικότητας

- ▶ Μέτρο του ρυθμού με τον οποίον οι εμπλεκόμενοι με το έργο παράγουν λογισμικό ή τεκμηρίωση κατά την διάρκεια ανάπτυξης του έργου.
- ▶ Παραγωγικότητα =
$$\frac{\text{Μέγεθος}}{\text{Προσπάθεια}}$$
- ▶ Γίνεται μέτρηση της λειτουργικότητας που παράγεται ανά μονάδα χρόνου.
- ▶ Η παραγωγικότητα μετράται σε:
 - Γραμμές κώδικα ανά μονάδα χρόνου.
 - Βαθμοί λειτουργίας ανά μονάδα χρόνου.
 - Αντικειμενοστραφής κώδικας ανά μονάδα χρόνου.



Παραγωγικότητα Προγραμματιστών

- ▶ Μια μετρική του ρυθμού με το οποίο οι προγραμματιστές παράγουν λογισμικό και το συναφές με αυτό υλικό τεκμηρίωσης
- ▶ Η ποιότητα του παραγόμενου έργου θα πρέπει τελικά με κάποιο τρόπο να λαμβάνεται υπόψη στον υπολογισμό της παραγωγικότητας
- ▶ Επηρεάζεται και εξαρτάται από:
 - ▶ Την πολυπλοκότητα του προγράμματος, μοντέλων, αλγορίθμων
 - ▶ Το μέγεθος του προγράμματος / εφαρμογής
 - ▶ Την ευκολία συνεννόησης ανάμεσα στους προγραμματιστές
 - ▶ Χρονικούς περιορισμούς
 - ▶ Κοινωνικούς παράγοντες
- ▶ Ένας τρόπος μέτρησης της παραγωγικότητας των προγραμματιστών είναι η παραγόμενη προδιαγραμμένη λειτουργικότητα του συστήματος όπως αυτή υλοποιείται στην μονάδα του χρόνου και ανά προγραμματιστή



Mythical Man-Month

- ▶ Book: “[The Mythical Man-Month](#)”
 - ▶ Author: Fred Brooks
- ▶ “*The* classic book on the human elements of software engineering”



Τεχνικές Εκτίμησης Κόστους

- ▶ Εμπειροτεχνικά (Expert judgement)
- ▶ Κατά αναλογία (Estimation by analogy)
- ▶ Με το νόμο του Parkinson (Parkinson's Law)
- ▶ Με σκοπό την κατοχύρωση του έργου (Pricing to win)
- ▶ Εκτίμηση Top-down (Top-down estimation)
- ▶ Εκτίμηση Bottom-up (Bottom-up estimation)
- ▶ Εκτίμηση με τη χρήση μετρικής Function Point (Function point estimation)
- ▶ Εκτίμηση με τη χρήση αλγοριθμικών μοντέλων κόστους (Algorithmic cost modelling)



Εμπειροτεχνία - Expert judgement

- ▶ Ένας ή περισσότεροι εμπειρογνώμονες στις περιοχές πεδίου εφαρμογής και στις περιοχές ανάπτυξης συστημάτων εκτιμούν το κόστος ανάπτυξης. Η διαδικασία επαναλαμβάνεται όσες φορές χρειάζεται μέχρι που να βρεθεί μια κοινή γνώμη και όλοι οι εμπειρογνώμονες είναι ικανοποιημένοι από το αποτέλεσμα.
- ▶ Κάθε εμπειρογνώμονας εκφράζει και μια διαφορετική σκοπιά για την τελική εκτίμηση του κόστους
- ▶ Πλεονεκτήματα της μεθόδου: Σχετικά ολιγοέξοδη μέθοδος υπολογισμού / εκτίμησης κόστους Relatively cheap estimation method.
- ▶ Μειονεκτήματα της μεθόδου: Αναξιόπιστη μέθοδος εάν δεν υπάρχουν καλοί εμπειρογνώμονες

7



Εκτίμηση Κόστους Κατά Αναλογία

- ▶ Το κόστος της εφαρμογής εκτιμάται κατά αναλογία με το κόστος άλλων γνωστών και παρόμοιων εφαρμογών
- ▶ Πλεονεκτήματα της μεθόδου: Σχετικά ακριβής και ορθή μέθοδος εάν υπάρχουν ιστορικά δεδομένα για την κατά αναλογία εκτίμηση του κόστους[∞]
- ▶ Μειονεκτήματα της μεθόδου: Δεν είναι εφαρμόσιμη μέθοδος εάν δεν έχουμε στοιχεία. Η μέθοδος χρειάζεται συστηματική καταγραφή δεδομένων στοιχείων κόστους για κάθε έργο που έχει τελειώσει.



Parkinson's Law

- ▶ Το έργο θα κοστίσει όσο κοστίζουν οι διαθέσιμοι πόροι
- ▶ Πλεονεκτήματα της μεθόδου: Ποτέ δεν βγαίνουμε εκτός προϋπολογισμού
- ▶ Μειονεκτήματα της μεθόδου: Η εφαρμογή συνήθως δεν παραδίδεται τελειωμένη ή ποτέ δεν τελειώνει



Κατοχύρωση του Έργου

Pricing to win

- ▶ Το έργο θα κοστίσει όσο έχει να πληρώσει ο πελάτης / χρήστης
- ▶ Πλεονεκτήματα της μεθόδου: Κατοχυρώνουμε το έργο
- ▶ Μειονεκτήματα της μεθόδου: Η πιθανότητα ο πελάτης να παραλάβει τελικά την εφαρμογή έτσι όπως θα την ήθελε είναι χαμηλή. Σε αυτή τη μέθοδο το κόστος ανάπτυξης δεν αντικατοπτρίζει σωστά την εργασία που απαιτείται για την εκτέλεση του έργου.



Εκτίμηση Top-down

- ▶ Για την εκτίμηση του κόστους ανάπτυξης της εφαρμογής ξεκινάμε από το σύστημα σαν σύνολο και προσδιορίζουμε τα επιμέρους υποσυστήματα που πρέπει να υλοποιηθούν, και υπολογίζουμε τα επί μέρους κόστη.
- ▶ Καταλήγουμε τη διαδικασία όταν φτάσουμε σε απλές μονάδες που δεν απαιτούν άλλα υποσυστήματα ψηφίδες συστήματος για την υλοποίησή τους
- ▶ Η μέθοδος λαμβάνει υπ' όψη της το κόστος ενοποίησης των ψηφίδων / υποσυστημάτων
- ▶ Η μέθοδος υποφέρει από το πρόβλημα ότι μπορεί να υπο-εκτιμήσουμε το κόστος της ανάπτυξης των τελικών μονάδων της εφαρμογής



Εκτίμηση Bottom-up

- ▶ Για την εκτίμηση του κόστους ανάπτυξης της εφαρμογής ξεκινάμε από τις επιμέρους μονάδες του συστήματος (απαιτείται να έχει γίνει η σχεδίαση). Το κόστος ανάπτυξης της κάθε μονάδας υπολογίζεται ανεξάρτητα και αθροίζεται με το κόστος των άλλων μονάδων, για να μας δώσει το τελικό κόστος ανάπτυξης της εφαρμογής
- ▶ Η μέθοδος είναι ακριβής και ορθή εάν έχουμε μια πολύ καλή εικόνα της σχεδίασης του συστήματος
- ▶ Η μέθοδος μπορεί να υπο-εκτιμήσει το κόστος ανάπτυξης εάν υπο-εκτιμήσουμε το κόστος ενοποίησης και τεκμηρίωσης

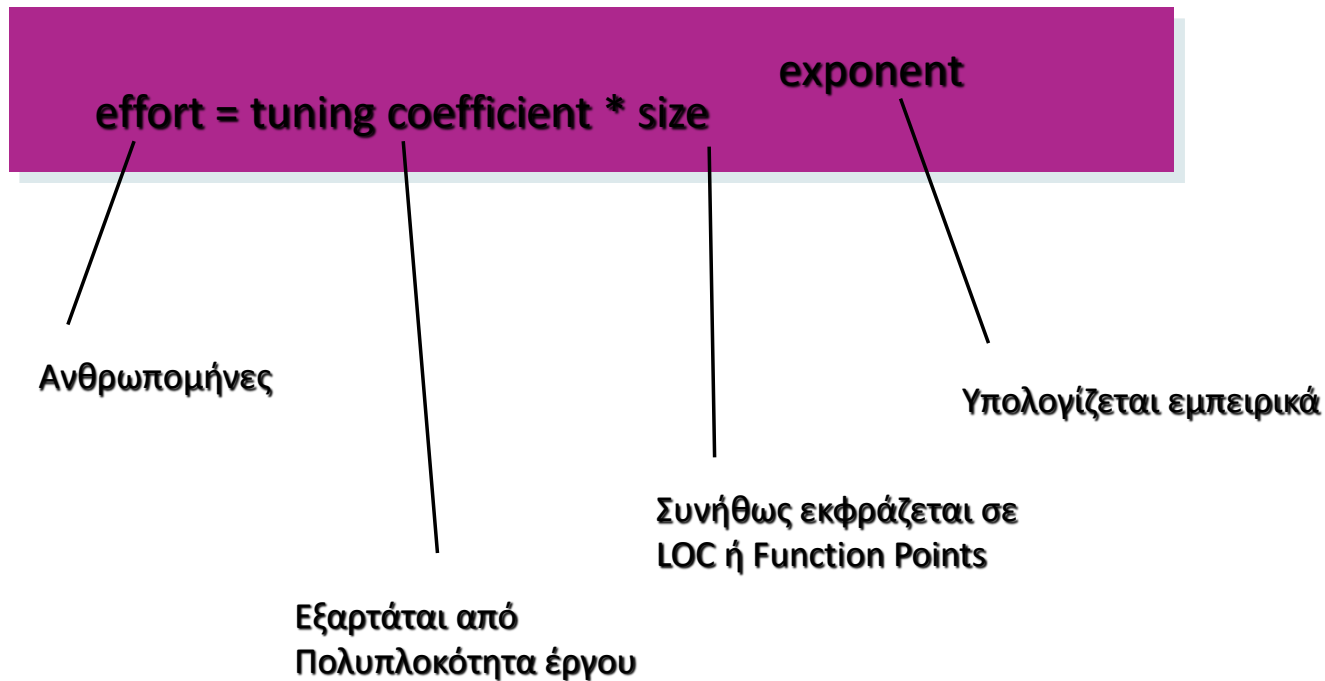


Παραμετρικά μοντέλα

COCOMO

Παραμετρικά (Αλγοριθμικά) Μοντελα

Γενική μορφή:



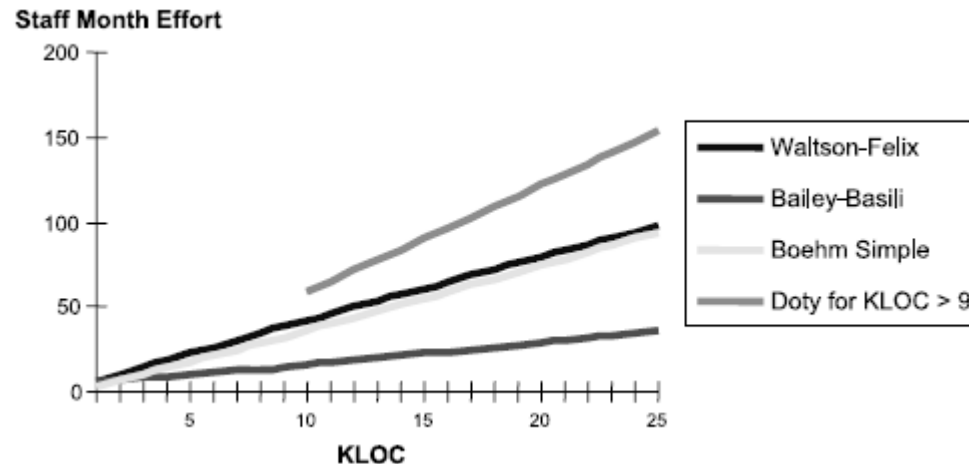
Διαφορετικά Μοντέλα βασισμένα σε KLOC

Walston-Felix: $\text{Effort} = 5.2 * (\text{KLOC})^{0.91}$

Bailey-Basili: $\text{Effort} = 5.5 + 0.73 * (\text{KLOC})^{1.16}$

Boehm Simple: $\text{Effort} = 3.2 * (\text{KLOC})^{1.05}$

Doty: $\text{Effort} = 5.288 * (\text{KLOC})^{1.047}$



Διαφορετικά Μοντέλα βασισμένα σε FP

Albrect–Gaffney: $\text{Effort} = 13.39 + 0.0545 * \text{FP}$

Kemerer: $\text{Effort} = 60.62 + 7.728 * (10^{-8}) * \text{FP}^3$

Matson–Barret–Meltichamp: $\text{Effort} = 585.7 + 15.12 * \text{FP}$



COCOMO

- ▶ COnstructive COst MOdel
- ▶ Πρώτο σημαντικό άλμα στον τρόπο υπολογισμού κόστους έργων λογισμικού
- ▶ Αναπτύχθηκε από τον **Barry Boehm**: “ Software Engineering Economics ”
- ▶ Είναι από τα πιο ευρέως χρησιμοποιούμενα μοντέλα
- ▶ Το μοντέλο COCOMO υπολογίζει:
 - ▶ Τη διάρκεια και το κόστος ενός έργου λογισμικού βασιζόμενο στο μέγεθος του προϊόντος και την ποιότητα της ομάδας ανάπτυξης
- ▶ http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html



Παραδοχές

- ▶ Θεωρούμε 152 ώρες εργασίας ανά ανθρωπομήνα
- ▶ Θεωρούμε ικανό management
- ▶ Θεωρούμε ότι το έγγραφο περιγραφής απαιτήσεων δεν τροποποιείται μετά την έγκρισή του
- ▶ Θεωρούμε ότι ο σημαντικότερος παράγοντας κόστους είναι το μέγεθος του έργου λογισμικού
- ▶ Το μέγεθος του έργου προσδιορίζεται σε χιλιάδες γραμμές παραδοτέου κώδικα (KDeliveredSourceInstructions - KDSI)
- ▶ Τρία μοντέλα:
 - ▶ Βασικό μοντέλο
 - ▶ Ενδιάμεσο μοντέλο
 - ▶ Λεπτομερειακό μοντέλο



Βασικό Μοντέλο

- Απαιτούμενη προσπάθεια σε ανθρωπομήνες :

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

- Απαιτούμενος χρόνος ανάπτυξης του έργου

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

- Ενδιαφέροντα σημεία: ο εκθέτης του KDSI είναι σχεδόν ίσος με τη μονάδα, άρα η προσπάθεια ανάπτυξης είναι γραμμική συνάρτηση του μεγέθους του κώδικα
- Ο αριθμός των ατόμων θεωρείται "βέλτιστος" για κάθε στάδιο του έργου. Π.χ. στο μέσο του έργου απασχολούνται περισσότεροι από ότι στη φάση καθορισμού των απαιτήσεων



Ενδιάμεσο μοντέλο

- ▶ **Οργανικά**
 - ▶ Το πρόγραμμα είναι σχετικά ανεξάρτητο και έχει μικρή διασύνδεση με το περιβάλλον, Σύστημα ανάλυσης πληροφοριών πτήσης, που επεξεργάζεται τα δεδομένα μετά το τέλος της πτήσης, Compilers, Επιστημονικά μοντέλα.
- ▶ **Ημιαποσπασμένα**
 - ▶ Υπάρχουν συνδέσεις με το περιβάλλον, αλλά όχι κρίσιμες χρονικές εξαρτήσεις, Προσομοιωτής πτήσης αεροσκαφών, Λειτουργικά Συστήματα, Βάσεις δεδομένων,
- ▶ **Ενσωματωμένα**
 - ▶ Κάθε συναλλαγή με το σύστημα είναι κρίσιμη από πλευράς χρόνου και ακρίβειας. Υπάρχουν αρκετοί περιορισμοί που εισάγονται από το περιβάλλον, Λογισμικό Αεροσκάφους για την αποφυγή συγκρούσεων με άλλα αεροσκάφη κατά τη διάρκεια της πτήσης, Συστήματα που υλοποιούν νέες τεχνολογίες, συστήματα διαχείρισης κρίσιμων καταστάσεων.



Κατηγορίες έργων λογισμικού

Κατηγορία	ονομαστική προσπάθεια	διάρκεια ανάπτυξης
οργανική	$MM_{NOM} = 3.2(KDSI)^{1.05}$	$TDEV = 2.5 (MM_{DEV})^{0.38}$
ημιαποσπ.	$MM_{NOM} = 3.0(KDSI)^{1.12}$	$TDEV = 2.5 (MM_{DEV})^{0.35}$
ενσωματωμ	$MM_{NOM} = 2.8(KDSI)^{1.20}$	$TDEV = 2.5 (MM_{DEV})^{0.32}$



COCOMO (Εξισώσεις)

- Εξίσωση υπολογισμού απαιτούμενης ονομαστικής προσπάθειας

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

C, K: παράμετροι, KDSI: thousands of delivered source lines

MM_{NOM} : **ονομαστική** προσπάθεια σε ανθρωπομήνες



COCOMO (Εξισώσεις)

- Ένα σύνολο από 15 χαρακτηριστικά ("παράγοντες κόστους") που θεωρούνται ότι συνεισφέρουν στο κόστος.
- Σε κάθε παράγοντα κόστους αντιστοιχεί ένας "πολλαπλασιαστής προσπάθειας" (q_1, q_2, \dots, q_{15})
- Εξίσωση υπολογισμού παράγοντα προσαρμογής προσπάθειας

$$q = q_1 q_2, \dots, q_{15}$$



COCOMO (Εξισώσεις)

- Εξίσωση υπολογισμού προσπάθειας ανάπτυξης

$$MM_{DEV} = q \cdot MM_{NOM}$$

- Εξίσωση υπολογισμού κόστους

$$C_t = p \cdot MM_{DEV}$$

p: αξία σε χρήμα ενός ανθρωπομήνα

- Εξίσωση υπολογισμού διάρκειας του έργου

$$T_{DEV} = R \cdot (MM_{DEV})^m$$

R, m: παράμετροι



COST DRIVERS

- ▶ **Product attributes**
 - ▶ Required software reliability
 - ▶ Size of application database
 - ▶ Complexity of the product
 - ▶ **Hardware attributes**
 - ▶ Run-time performance constraints
 - ▶ Memory constraints
 - ▶ Volatility of the virtual machine environment
 - ▶ Required turnabout time
 - ▶ **Personnel attributes**
 - ▶ Analyst capability
 - ▶ Software engineering capability
 - ▶ Applications experience
 - ▶ Virtual machine experience
 - ▶ Programming language experience
 - ▶ **Project attributes**
 - ▶ Use of software tools
 - ▶ Application of software engineering methods
 - ▶ Required development schedule
-



COST DRIVERS – ΠΑΡΑΓΟΝΤΕΣ ΚΟΣΤΟΥΣ

1.	RELY	Απαιτούμενη αξιοπιστία λογισμικού	προϊόν
2.	DATA	Μέγεθος βάσης δεδομένων	προϊόν
3.	CPLX	Πολυπλοκότητα προϊόντος	προϊόν
4.	TIME	Περιορισμός στο χρόνο εκτέλεσης	H/Y
5.	STOR	Περιορισμός στην κύρια μνήμη	H/Y
6.	VIRT	Αλλαγές στο σύστημα HW/SW	H/Y
7.	TURN	Χρόνος απόκρισης υπολογιστή	H/Y
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



Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of software tools	1.24	1.10	1.00	0.91	0.82	
Application of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	



COCOMO

- Δεδομένα εισόδου: Μέγεθος ενός έργου λογισμικού (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



Παράγοντας κόστους	Χαμηλό-τερη	Χαμηλή	Ονομαστική	Υψηλή	Υψηλότερη	Υψηλότερη
RELY	Μικρή ενόχληση	Μικρές, ευκόλα ανακτήσιμες απώλειες	Μέτριες, ανακτήσιμες απώλειες	Μεγάλες, οικονομικές απώλειες	Κίνδυνος για ανθρώπινες ζωές	
DATA		L<10	10<L<100	100<L<1000	L>1000	
CPLX						
TIME			<50% ΧΔΧΕ	70% ΧΔΧΕ	85% ΧΔΧΕ	95%ΧΔΧΕ
STOR			<50% ΧΔΜ	70% ΧΔΜ	85% ΧΔΜ	95%ΧΔΜ
VIRT		ΣΑ: 12 μήνες ΜΑ: 1 το μήνα	ΣΑ: 6 μήνες ΜΑ: 2 εβδομάδες	ΣΑ: 2 μήνες ΜΑ: 1 εβδομάδα	ΣΑ: 2 εβδομάδες ΜΑ: 2 μέρες	
TURN		Διαλογική	ΜΧΚΕ< 4 ώρες	4<ΜΧΚΕ< 8	ΜΧΚΕ> 12	
ACAP	15 ΕΤ	35 ΕΤ	55 ΕΤ	75 ΕΤ	90 ΕΤ	
AEXP	<4 μήνες	1 χρόνο	3 χρόνια	6 χρόνια	12 χρόνια	
PCAP	15 ΕΤ	35 ΕΤ	55 ΕΤ	75 ΕΤ	90 ΕΤ	
VEXP	<1 μήνα	4 μήνες	1 χρόνο	3 χρόνια		
LEXP	<1 μήνα	4 μήνες	1 χρόνο	3 χρόνια		
MODP	Όχι χρήση	Αρχή χρήσης	Μερική χρήση	Γενική χρήση	Αποκλειστική χρήση	
TOOL						
SCED	75%	85%	100%	130%	160%	

Επεξηγήσεις

- ▶ $L = (\text{DBbytes}) / (\text{Prog.DSI})$
- ▶ ΧΔΧΕ = Χρησιμοποίηση Διαθέσιμου Χρόνου Εκτέλεσης
- ▶ ΧΔΜ = Χρησιμοποίηση Διαθέσιμης Μνήμης
- ▶ ΜΧΚΕ = Μέση Διάρκεια Κύκλου Εξυπηρέτησης
- ▶ Χ ΕΤ = άνω Χ τοις εκατό, π.χ. 35 ΕΤ σημαίνει άνω 35% στην κατάταξη ικανότητας των των αναλυτών
- ▶ SCED Χ% = πίεση για αύξηση ή μείωση κατά Χ% επί της αρχικής, ονομαστικής χρονικής διάρκειας του έργου



Παράδειγμα

- ▶ Ανάπτυξη λογισμικού επικοινωνίας βασισμένου σε μικροϋπολογιστή για δίκτυο μεταφοράς κεφαλαίων με ηλεκτρονικό τρόπο.
- ▶ Απαιτήσεις:
 - ▶ Υψηλή αξιοπιστία,
 - ▶ υψηλή επίδοση,
 - ▶ αυστηρό χρονοδιάγραμμα ανάπτυξης,
 - ▶ καλό interface
- ▶ Υποτιθέμενο μέγεθος κώδικα 10 KDSI



Κατηγορίες έργων λογισμικού

Κατηγορία	ονομαστική προσπάθεια	διάρκεια ανάπτυξης
οργανική	$MM_{NOM} = 3.2(KDSI)^{1.05}$	$TDEV = 2.5 (MM_{DEV})^{0.38}$
ημιαποστ.	$MM_{NOM} = 3.0(KDSI)^{1.12}$	$TDEV = 2.5 (MM_{DEV})^{0.35}$
ενσωματωμ	$MM_{NOM} = 2.8(KDSI)^{1.20}$	$TDEV = 2.5 (MM_{DEV})^{0.32}$

$$MM_{NOM} = 2.8(10)^{1.20} = 44 \text{ ανθρωπομήνες}$$



Παράγοντες κόστους - ταξινόμηση

1.	RELY	Σοβαρές οικονομ. επιπτώσεις	υψηλή	1.15
2.	DATA	20000 bytes	χαμηλή	0.94
3.	CPLX	Επεξεργασία επικοινωνιών	υψηλότατ	1.30
4.	TIME	70% χρήση του διαθέσιμου χρόνου	υψηλή	1.11
5.	STOR	70% χρήση της διαθέσιμης μνήμης	υψηλή	1.06
6.	VIRT	Θα βασισθεί σε υπάρχον Η/Υ σύστημα	ονομαστ.	1.00
7.	TURN	Χρόνος απόκρισης	ονομαστ.	1.00
8.	ACAP	Καλοί ηλικιωμένοι αναλυτές	υψηλή	0.85
9.	AEXP	Τρία έτη	ονομαστ.	1.00
10.	PCAP	Καλοί προγραμματιστές (75 %)	υψηλή	0.86
11.	VEXP	Έξι μήνες	χαμηλή	1.10
12.	LEXP	Δώδεκα μήνες	ονομαστ.	1.00
13.	MODP	Οι περισσότερες τεχνικές είναι γνωστές	υψηλή	0.91
14.	TOOL	Βασικά εργαλεία Η/Υ	χαμηλή	1.10
15.	SCED	Εννέα μήνες	ονομαστ	1.00



Παράδειγμα (συνέχεια)

- Υπολογισμός παράγοντα προσαρμογής προσπάθειας

$$q = q_1 q_2, \dots q_{15} = 1.35$$

- Υπολογισμός προσπάθειας ανάπτυξης

$$MM_{DEV} = q \cdot MM_{NOM} = 1.35 * 44 = 59 \text{ ανθρωπομήνες}$$

- Υπολογισμός κόστους, θεωρώντας κόστος ανθρωπομήνα 1000 €

$$C_t = 59.000 \text{ €}$$

- Υπολογισμός διάρκειας ανάπτυξης

$$T_{DEV} = R \cdot (MM_{DEV})^m = 2.5 \cdot (59)^{0.32} = 9 \text{ μήνες}$$



Source Lines of Code (SLOC)

The COCOMO calculations are based on your estimates of a project's size in Source Lines of Code (SLOC). SLOC is defined such that only:

- Source lines that are **DELIVERED** as part of the product are included
 - test drivers and other support software is excluded.
- **SOURCE** lines are created by the project staff
 - code created by applications generators is excluded.
- One SLOC is one logical line of code.
- Declarations are counted as SLOC.
- Comments are not counted as SLOC.

The original COCOMO 81 model was defined in terms of **Delivered Source Instructions**, which are very similar to SLOC. The major difference between DSI and SLOC is that a single Source Line of Code may be several physical lines. For example, an "if-then-else" statement would be counted as one SLOC, but might be counted as several DSI.



Definition Checklist for Source Statements Counts

Definition name: Logical Source Statements Date: _____

(basic definition)

Originator: COCOMO.II

Measurement unit	Physical source lines			
	Logical source statements	<input checked="" type="checkbox"/>		
Statement type	Definition <input checked="" type="checkbox"/>	Data Array		Includes Excludes
<i>When a line or statement contains more than one type, classify it as the type with the highest precedence.</i>				
1 Executable	Order of precedence →		1	<input checked="" type="checkbox"/>
2 Nonexecutable				
3 Declarations			2	<input checked="" type="checkbox"/>
4 Compiler directives			3	<input checked="" type="checkbox"/>
5 Comments				
6 On their own lines			4	<input checked="" type="checkbox"/>
7 On lines with source code			5	<input checked="" type="checkbox"/>
8 Banners and non-blank spacers			6	<input checked="" type="checkbox"/>
9 Blank (empty) comments			7	<input checked="" type="checkbox"/>
10 Blank lines			8	<input checked="" type="checkbox"/>
11				
12				
How produced	Definition <input checked="" type="checkbox"/>	Data array		Includes Excludes
1 Programmed				<input checked="" type="checkbox"/>
2 Generated with source code generators				<input checked="" type="checkbox"/>
3 Converted with automated translators				<input checked="" type="checkbox"/>
4 Copied or reused without change				<input checked="" type="checkbox"/>
5 Modified				<input checked="" type="checkbox"/>
6 Removed				<input checked="" type="checkbox"/>
7				
8				

Origin	Definition	Data array	Includes	Excludes
1	New work, no prior existence	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Prior work, taken or adapted from:			
3	A previous version, build, or release		<input checked="" type="checkbox"/>	
4	Commercial, off-the-shelf software (COTS), other than libraries			<input checked="" type="checkbox"/>
5	Government furnished software (GFS), other than reuse libraries			<input checked="" type="checkbox"/>
6	Another product			<input checked="" type="checkbox"/>
7	A vendor-supplied language support library (unmodified)			<input checked="" type="checkbox"/>
8	A vendor-supplied operating system or utility (unmodified)			<input checked="" type="checkbox"/>
9	A local or modified language support library or operating system			<input checked="" type="checkbox"/>
10	Other commercial library			<input checked="" type="checkbox"/>
11	A reuse library (software designed for reuse)		<input checked="" type="checkbox"/>	
12	Other software component or library		<input checked="" type="checkbox"/>	
13				
14				



Η εξίσωση του κόστους – COCOMO II – Ενδιάμεσο μοντέλο

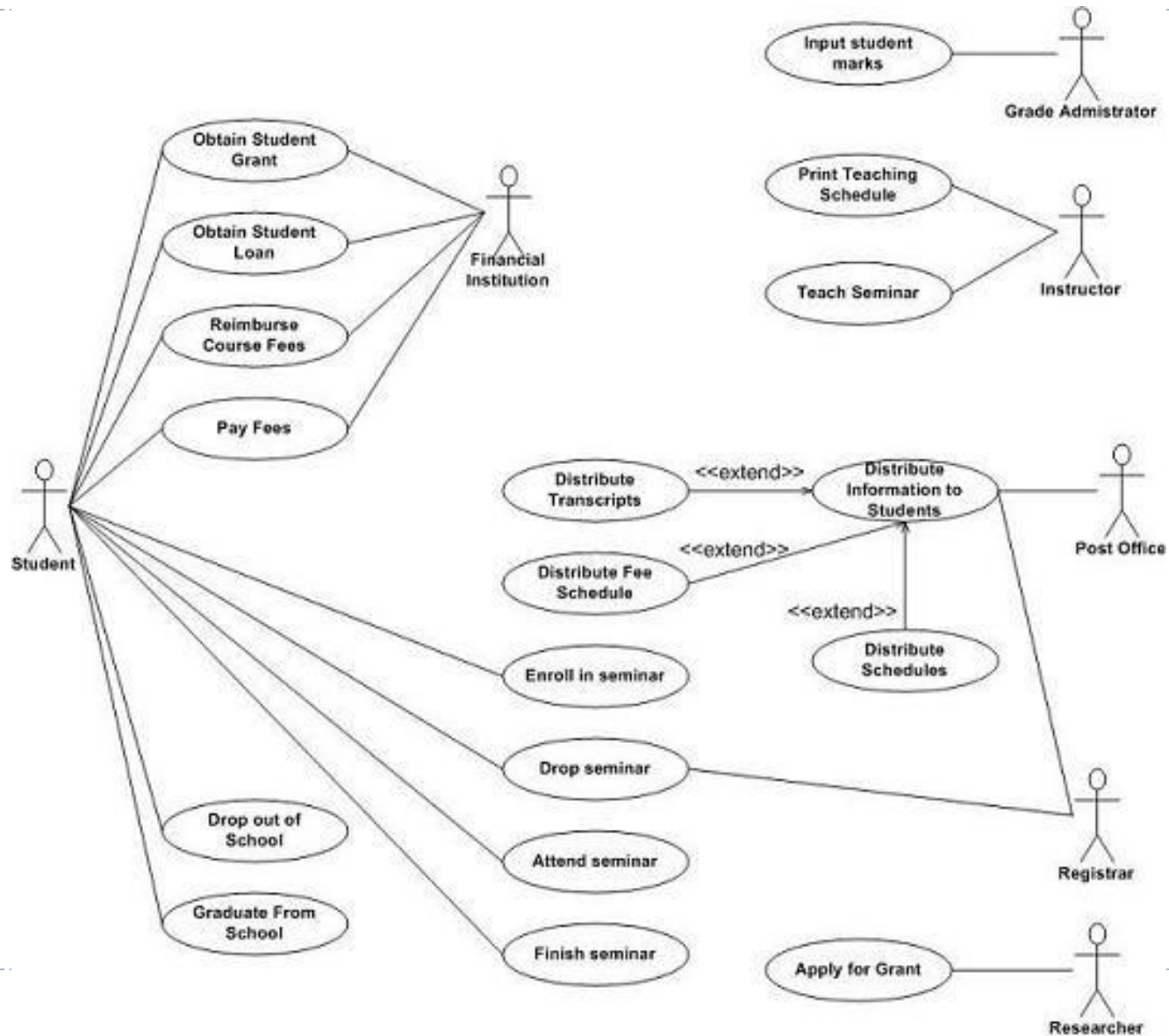
$$\text{Effort} = 2.94 * \text{EAF} * (\text{KSLOC})^E$$

Effort Adjustment Factor (EAF) → cost drivers



Υπολογισμός Προσπάθειας με USE CASE POINTS

USE CASE DIAGRAM



1. Αστάθμητο βάρος χειριστών (Unadjusted Actor Weight - UAW)

Ταξινόμηση	Παράδειγμα χειριστή	Βάρος
Απλός	Ένα API καλά τεκμηριωμένο	1
Μέτριας πολυπλοκότητας	Ένα πρωτόκολλο όπως TCP/IP, HTTP, FTP ή ο χειριστής είναι βάση δεδομένων	2
Σύνθετος	Διεπαφή με άλλο σύστημα ή οποιαδήποτε διεπαφή με το χρήστη	3

2. Υπολογισμός Αστάθμητου Βάρους ΠΧ (Unadjusted Use Case Weight - UUCW)

Ταξινόμηση	Παράδειγμα ΠΧ	Βάρος
Απλή	<4 συναλλαγές	5
Μέτριας πολυπλοκότητας	≥ 4 και ≤ 7 συναλλαγών	10
Σύνθετη	\geq συναλλαγές	15

3. Συνολικός Αριθμός Αστάθμητων σημείων ΠΧ

- ▶ ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ αστάθμητων σημείων ΠΧ (Unadjusted Use Case Points – UUCP).

$$UUCP = UAW + UUCW$$

Βήμα 4

Παράγοντας	Τεχνικοί παράγοντες πολυπλοκότητας	Βάρος
T1	Είναι καταμεμημένο σύστημα	2
T2	Υπάρχουν απαιτήσεις απόδοσης (performance) ή απόκρισης (response)	2
T3	Υπάρχουν απαιτήσεις για την αποδοτικότητα τελικών χρηστών	1
T4	Υπάρχει σύνθετη εσωτερική επεξεργασία	1
T5	Αναπτύσσεται επαναχρησιμοποιήσιμος κώδικας	1
T6	Ευκολία εγκατάστασης	0,5
T7	Ευχρηστία	0,5
T8	Φορητότητα (portability)	2
T9	Ευκολία αλλαγής	1
T10	Ταυτόχρονη λειτουργία	1
T11	Απαιτήσεις για ασφάλεια	1
T12	Παρέχει πρόσβαση σε τρίτους	1
T13	Ειδικές ανάγκες εκπαίδευσης χρηστών	1

Βήμα 5

Περιβαλλοντικοί παράγοντες		
F1	Οικειότητα με τη UML	1,5
F2	Εμπειρία στο πεδίο εφαρμογής	0,5
F3	Εμπειρία σε Object Oriented	1
F4	Ικανότητα βασικού αναλυτή	0,5
F5	Ύπαρξη κινήτρων για την ομάδα ανάπτυξης	1
F6	Σταθερές απαιτήσεις	2
F7	Μερικής απασχόλησης εργαζόμενοι	-1
F8	Δύσκολη γλώσσα προγραμματισμού	-1

Βήμα 6 - ΥΠΟΛΟΓΙΣΜΟΣ

- ▶ 5. Υπολογίζουμε το Tfactor από το άθροισμα όλων των τεχνικών παραγόντων πολυπλοκότητας.
- ▶ 6. Υπολογίζουμε τον παράγοντα τεχνικής πολυπλοκότητας (Technical Complexity Factor - TCF) με βάση τον τύπο

$$TCF = 0,6 + 0,01 * Tfactor$$

- ▶ 7. Υπολογίζουμε το Efactor από το άθροισμα όλων των παραγόντων περιβάλλοντος.

Βήμα 6 - ΥΠΟΛΟΓΙΣΜΟΣ

- ▶ 7. Υπολογίζουμε τον παράγοντα περιβάλλοντος (Environment Factor - EF) με βάση τον τύπο

$$EF = 1,4 - (0,03 * E_{factor})$$

- ▶ 8. Υπολογίζουμε τα σταθμισμένα σημεία ΠΧ (Adjusted Use Case Points - AUCP) με βάση τον τύπο

$$AUCP = UUCP * TCF * EF$$

- ▶ 9. Η συνολική απαιτούμενη προσπάθεια είναι
 $Effort = AUCP * \text{Παράγοντας Παραγωγικότητας}$

Σύμφωνα με τον Karner, ο παράγοντας παραγωγικότητας είναι 20, δηλαδή 20 ώρες εργασίας ανά AUCP.

Παράδειγμα- Απαιτήσεις της εφαρμογής - 1

- × Οι εργαζόμενοι της εταιρείας είναι δυο κατηγοριών:
 - + Πλήρους απασχόλησης και
 - + Μερικής απασχόλησηςΜια ειδική κατηγορία εργαζομένων πλήρους απασχόλησης είναι οι πωλητές.
- × Η πληρωμή των εργαζομένων πλήρους απασχόλησης γίνεται με βάση το μηνιαίο μισθό λαμβάνοντας υπόψη τις ημέρες απουσίας.
- × Η πληρωμή των εργαζομένων μερικής απασχόλησης γίνεται με βάση τις πραγματικές ώρες εργασίας και το ωρομίσθιο του κάθε εργαζομένου.
- × Οι πωλητές, λαμβάνουν εκτός από το μηνιαίο μισθό πρόσθετη αμοιβή (bonus) η οποία εξαρτάται από το συνολικό ποσό των παραγγελιών που έλαβαν για τη δεδομένη χρονική περίοδο. Έτσι ένας πωλητής λαμβάνει
 - + επιπλέον 5% του μισθού του, σε εβδομαδιαία βάση, εάν οι παραγγελίες που έλαβε είναι μικρότερες των 10.000 €.
 - + επιπλέον 7% του μισθού του, σε εβδομαδιαία βάση, εάν οι παραγγελίες που έλαβε είναι περισσότερες των 10.000 € και λιγότερες των 20.000 €.
 - + Για παραγγελίες μεγαλύτερες των 20.000 € λαμβάνουν επιπλέον 10% του μισθού τους σε εβδομαδιαία βάση.



Παράδειγμα- Απαιτήσεις της εφαρμογής - - 2

- ✘ Οι εργαζόμενοι πλήρους απασχόλησης λαμβάνουν υπερωρίες. Ως υπερωρία θεωρείτε η εργασία μετά τις 8 ώρες ημερησίως, η οποία αμείβεται με το ωρομίσθιο του κάθε εργαζομένου προσαυξημένο κατά 50%. Οι υπερωρίες υπολογίζονται σε ημερήσια βάση.
- ✘ Για την ακριβή καταγραφή των ωρών εργασίας των εργαζομένων, η εταιρεία έχει εγκαταστήσει σύστημα «Ελέγχου Προσέλευσης/Αποχώρησης Εργαζομένων», το οποίο καταγράφει την ώρα προσέλευσης και αποχώρησης του κάθε εργαζομένου.
- ✘ Κάθε εργαζόμενος έχει μια κάρτα εισόδου η οποία χρησιμοποιείται για την καταγραφή της ώρας προσέλευσης και αναχώρησης.
- ✘ Ο υπεύθυνος μισθοδοσίας έχει την αρμοδιότητα εισαγωγής, διαγραφής και τροποποίησης των στοιχείων των εργαζομένων.
- ✘ Ο υπεύθυνος μισθοδοσίας έχει την αρμοδιότητα παραγωγής μισθοδοσίας.
- ✘ Το σύστημα θα πρέπει να διαθέτει δυνατότητα παραγωγής εκτυπώσεων τόσο ανά εργαζόμενο όσο και συγκεντρωτικές. Πιο συγκεκριμένα θα πρέπει να παράγει
 - + Φύλλο μισθοδοσίας ανά εργαζόμενο
 - + Συγκεντρωτική μισθοδοσία για όλη την εταιρεία
 - + Παραγγελίες ανά εργαζόμενο
- ✘ Τόσο, τα δεδομένα της μισθοδοσίας όσο και τα στοιχεία των υπαλλήλων θα αποθηκεύονται σε ΒΔ. Όταν ξεκινά η εφαρμογή θα πρέπει να διαβάζεται αυτόματα η πληροφορία των υπαλλήλων από τη ΒΔ και να δημιουργούνται τα κατάλληλα αντικείμενα (υπαλλήλων, κάρτας εισόδου, κ.ο.κ).

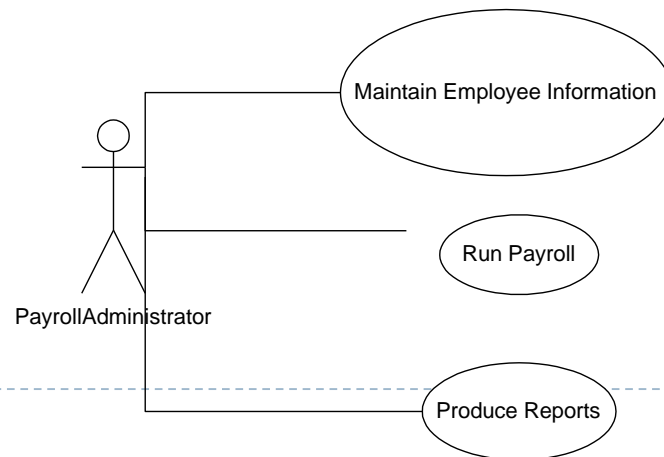
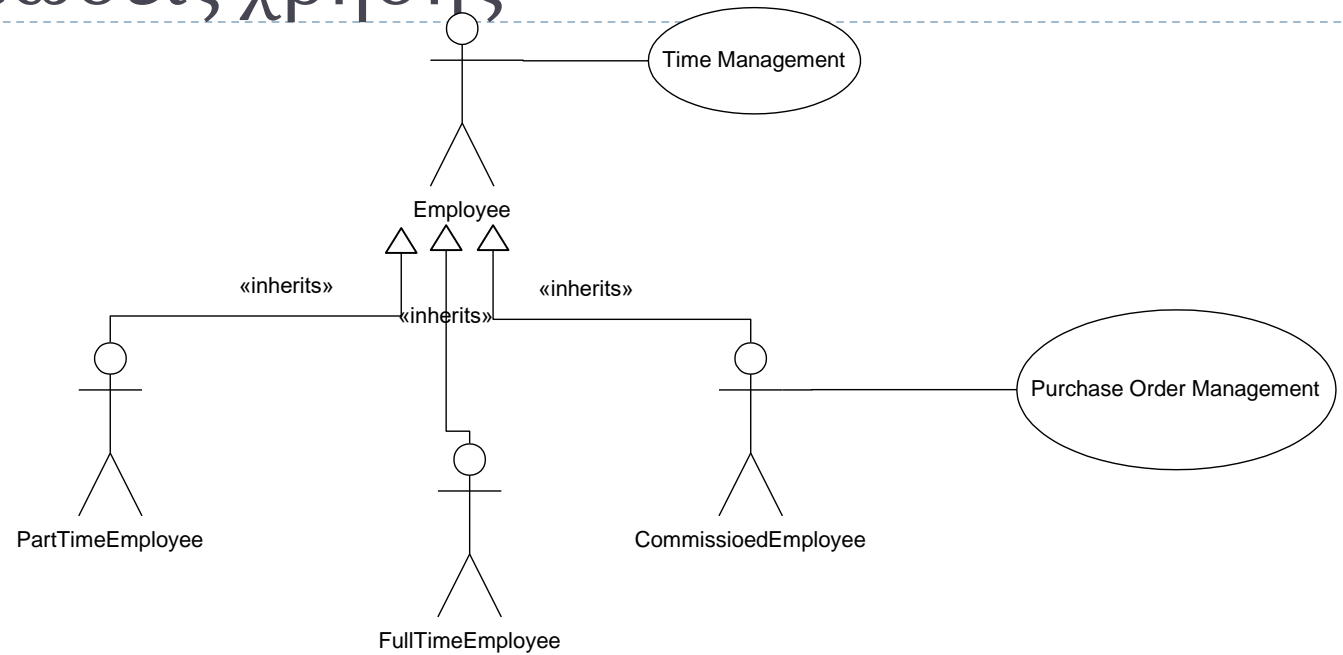


Παραδοχές που κάνουμε

- ✘ Κάθε εργαζόμενος έχει μια κάρτα εισόδου η οποία χρησιμοποιείται για την καταγραφή της ώρας προσέλευσης και αναχώρησης. Για την καταγραφή θα πρέπει να εξομοιωθεί η λειτουργία ενός αναγνώστη κάρτας καθώς και η προσέλευση/αναχώρηση των υπαλλήλων. Ο αναγνώστης κάρτας κάθε ημέρα και για κάθε υπάλληλο δέχεται δύο μηνύματα: Προσέλευση και Αναχώρηση. Κατά την προσέλευση η χρονική στιγμή εισαγωγής της κάρτας θα ορίζεται με τυχαίο τρόπο για το χρονικό διάστημα από 8:00 έως 8:15. Κατά την αναχώρηση η χρονική στιγμή εισαγωγής της κάρτας θα ορίζεται με τυχαίο τρόπο για το χρονικό διάστημα από 16:00 έως 16:30.
- ✘ Το σύστημα «Ελέγχου Προσέλευσης/Αποχώρησης Εργαζομένων» αν και συνεργάζεται με το σύστημα μισθοδοσίας δεν αποτελεί τμήμα του.
- ✘ Η εφαρμογή που παρουσιάζεται καταγράφει τους χρόνους εργασίας όλων των υπαλλήλων για τις εργάσιμες μέρες ενός μήνα. Για την απλοποίηση της λογικής υποθέτουμε πως δεν υπάρχουν απουσίες υπαλλήλων στη διάρκεια του μήνα και πως οι ωρομίσθιοι πληρώνονται στο τέλος κάθε μήνα.
- ✘ Στην παρούσα μελέτη περίπτωσης δε θα ασχοληθούμε με τη διεπαφή του συστήματος με το χρήστη (graphical user interface) στο επίπεδο του κώδικα. Αν και το παράδειγμα, περιλαμβάνει γραφικές οθόνες (διαχείριση παραγγελιών) σκοπός είναι να παρουσιασθεί ο τρόπος με τον οποίο κάνουμε την πρωτοτυποποίηση του συστήματος και αναπτύσσουμε τη διεπαφή με το χρήστη. Για το λόγο αυτό δε δίνεται ιδιαίτερη έμφαση στην επεξήγηση των τεχνικών χαρακτηριστικών που σχετίζονται με την ανάπτυξη του GUI.
- ✘ Η παρούσα μελέτη περίπτωσης δεν έχει περιορισμούς εισόδου (login, password). Συνεπώς η λειτουργικότητα που παρουσιάζεται είναι κοινή για όλους τους χρήστες.



Οι περιπτώσεις χρήσης



1. Αστάθμητο βάρος χειριστών (Unadjusted Actor Weight - UAW)

Χειριστής	Ταξινόμηση	Βάρος
Employee	Σύνθετος	3
PartTimeEmployee	Σύνθετος	3
FullTimeEmployee	Σύνθετος	3
CommissionedEmployee	Σύνθετος	3
PayrollAdministrator	Σύνθετος	3
	Σύνολο UAW	15

2. ΥΠΟΛΟΓΙΣΜΟΣ αστάθμητου βάρους ΠΧ (Unadjusted Use Case Weight - UUCW)

Περίπτωση χρήσης	Ταξινόμηση	Βάρος
Time Management	Απλή	5
Purchase Order Management	Μέτρια (CRUD ΠΧ)	10
Maintain Employee Information	Μέτρια (CRUD ΠΧ)	10
Run Payroll	Μέτρια (CRUD ΠΧ)	10
Produce Reports	Υπάρχουν πολλές αναφορές	10
	Σύνολο	45

3. ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ αστάθμητων σημείων ΠΧ

- ▶ ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ αστάθμητων σημείων ΠΧ (Unadjusted Use Case Points – UUCP).

$$\begin{aligned} \text{UUCP} &= \text{UAW} + \text{UUCW} \\ &= 15 + 45 = \mathbf{60} \end{aligned}$$

Παράγοντας	Περιγραφή τεχνικών παραγόντων πολυπλοκότητας	Βάρος	Πολυπλοκότητα Έργου	Τιμή
T1	Είναι καταναμημένο σύστημα	2	0	0
T2	Υπάρχουν απαιτήσεις απόδοσης (performance) ή απόκρισης (response)	2	3	6
T3	Υπάρχουν απαιτήσεις για την αποδοτικότητα τελικών χρηστών	1	3	3
T4	Υπάρχει σύνθετη εσωτερική επεξεργασία	1	4	4
T5	Αναπτύσσεται επαναχρησιμοποιήσιμος κώδικας	1	0	0
T6	Ευκολία εγκατάστασης	0,5	0	0
T7	Ευχρηστία	0,5	3	1,5
T8	Φορητότητα (portability)	2	0	0
T9	Ευκολία αλλαγής	1	3	3
T10	Ταυτόχρονη λειτουργία	1	0	0
T11	Απαιτήσεις για ασφάλεια	1	0	0
T12	Παρέχει πρόσβαση σε τρίτους	1	0	0
T13	Ειδικές ανάγκες εκπαίδευσης χρηστών	1	0	0
▶ 55			Tfactor	17,5

Περιβαλλοντικοί παράγοντες				
F1	Οικειότητα με το πρόγραμμα	1,5	3	4,5
F2	Εμπειρία πεδίου εφαρμογής	0,5	3	1,5
F3	Εμπειρία ανάπτυξης αντικειμενοστραφών εφαρμογών	1	3	3
F4	Ικανότητα βασικού αναλυτή	0,5	3	1,5
F5	Ύπαρξη κινήτρων για την ομάδα ανάπτυξης	1	3	3
F6	Σταθερές απαιτήσεις	2	3	6
F7	Μερικής απασχόλησης εργαζόμενοι	-1	3	-3
F8	Δύσκολη γλώσσα προγραμματισμού	-1	3	-3
			Efactor	13,5

ΥΠΟΛΟΓΙΣΜΟΙ

▶ Υπολογισμός Tfactor

▶ $Tfactor = 17,5$

▶ Υπολογισμός παράγοντα τεχνικής πολυπλοκότητας (Technical Complexity Factor - TCF) με βάση τον τύπο

▶ $TCF = 0,6 + 0,01 * Tfactor = 0,6 + 0,01 * 17,5 = 0,775$

▶ Υπολογισμός Efactor

▶ $Efactor = 13,5$

▶ Υπολογισμός τον παράγοντα περιβάλλοντος (Environment Factor - EF)

▶ $EF = 1,4 - (0,03 * Efactor) = 1,4 - (0,03 * 13,5) = 0,995$

▶ Υπολογισμός σταθμισμένων σημείων ΠΧ (Adjusted Use Case Points - AUCP)

▶ $AUCP = UUCP * TCF * EF = 60 * 0,775 * 0,995 = 46,26$

▶ Υπολογισμός συνολικής απαιτούμενης προσπάθειας

▶ $Effort = AUCP * \text{Παράγοντας Παραγωγικότητας} = 46,26 * 20 = 925,35$

Υπολογισμός Προσπάθειας με Function Points

Βαθμοί Λειτουργιάς (Function Points)

- ▶ Η ιδέα παρουσιάστηκε από τον Albrecht το 1979
- ▶ Το λογισμικό → μέγεθος και πολυπλοκότητα.
 - ▶ ανάγκη μέτρησης της λειτουργικότητας και της χρησιμότητας του.
- ▶ Λειτουργικότητα ενός προγράμματος
 - ▶ ποσότητα λειτουργίας που παρέχεται από το πρόγραμμα
- ▶ Οι Βαθμοί Λειτουργιάς είναι ένας τρόπος μέτρησης της λειτουργικότητας μίας εφαρμογής.
 - ▶ δομημένη τεχνική.
 - ▶ το σύστημα σπάει σε μικρότερα κομμάτια, έτσι ώστε να κατανοηθούν και να αναλυθούν.



Βαθμοί Λειτουργίας (Function Points)

- ▶ Οι Βαθμοί Λειτουργίας είναι ανεξάρτητοι
 - ▶ της τεχνολογίας.
 - ▶ της γλώσσας.
 - ▶ της μεθόδου ανάπτυξης.
 - ▶ του hardware .
- ▶ Οι βαθμοί λειτουργίας χρησιμοποιούνται για τη σύγκριση της αποτελεσματικότητας εργαλείων, περιβαλλόντων, γλωσσών.
- ▶ Έχουν ευρεία εφαρμογή στη βιομηχανία.

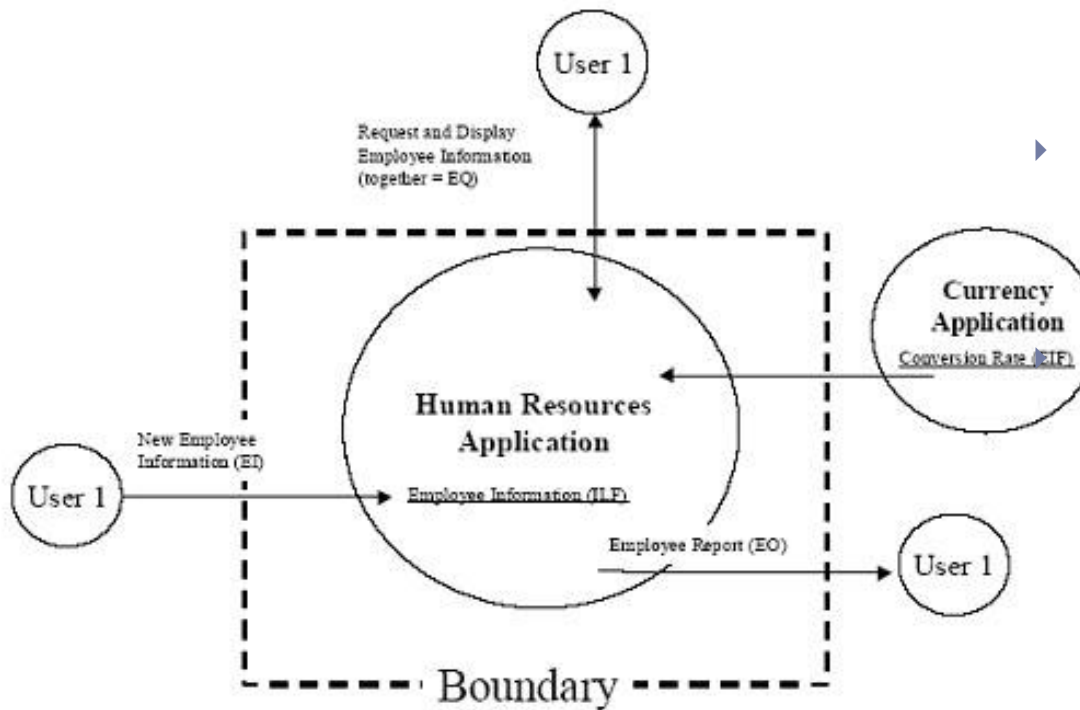


Βήματα

- ▶ Καθορισμός των ορίων (εύρους) της εφαρμογής
- ▶ Αναγνώριση του πεδίου μέτρησης
 - ▶ Ποιες οι απαιτήσεις από το σύστημα ?
 - ▶ Ποιες λειτουργίες θα εκτελεί το σύστημα?
 - ▶ Τι συναλλαγές θα γίνονται?
- ▶ Καταγραφή αμιγούς λειτουργικότητας (Unadjusted Function Points Count)
 - ▶ Λειτουργίες Δεδομένων
 - ▶ Λειτουργίες Συναλλαγών
- ▶ Καθορισμός πολυπλοκότητας λειτουργιών δεδομένων & συναλλαγών
- ▶ Υπολογισμός βαθμών λειτουργίας



Καθορισμός των ορίων της εφαρμογής



- ▶ Πρέπει να σχεδιαστεί ένα όριο για την αλληλεπίδραση του συστήματος με άλλα συστήματα έχοντας υπόψη τον χρήστη σαν οδηγό. Οι λειτουργίες του συστήματος προς ανάπτυξη δεν πρέπει να συγχέονται με αυτά των εξωτερικών συστημάτων.

Σχήμα 1



Παράμετροι Εκτίμησης της Μετρικής Function Points

- ▶ **Ποσοτικοποίηση Δεδομένων Εισόδου από τον Χρήστη** (application oriented data)
- ▶ **Ποσοτικοποίηση Δεδομένων Εξόδου** (application oriented information to the user)
- ▶ **Ποσοτικοποίηση Αναζήτησης Δεδομένων από τον Χρήστη** (ποσοτικοποίηση ερωτημάτων από τον χρήστη που ξεκινούν κάποια διαδικασία)
- ▶ **Ποσοτικοποίηση Χρήσης Αρχείων** (master files)
- ▶ **Ποσοτικοποίηση Εξωτερικών Διαπροσωπειών** (interfaces)




Λειτουργίες δεδομένων

- ▶ Εσωτερικά λογικά αρχεία (Internal Logical Files)


- ▶ ομάδες δεδομένων της εφαρμογής που σχετίζονται λογικά
- ▶ πληροφορίες ελέγχου της εφαρμογής

- ▶ Εξωτερικά αρχεία διαπροσωπείας (External Interface Files EIFs)

- ▶ ομάδες δεδομένων που σχετίζονται λογικά
- ▶ πληροφορίες ελέγχου



Που αναφέρεται η εφαρμογή μας και διατηρούνται μέσα στα όρια της εφαρμογής μας.



Που αναφέρεται η εφαρμογή μας , διατηρούνται μέσα στα όρια μίας άλλης εφαρμογής.



Λειτουργίες Συναλλαγών

- ▶ **Εξωτερικές εισοδοι (External Inputs EIs)**
 - ▶ Επεξεργάζονται δεδομένα ή πληροφορίες ελέγχου που έρχονται έξω από τα όρια της εφαρμογής.
- ▶ **Εξωτερικές έξοδοι (External Outputs EOs)**
 - ▶ Στέλνουν δεδομένα ή πληροφορίες ελέγχου έξω από τα όρια της εφαρμογής.
 - ▶ Παρουσιάζουν πληροφορία στον χρήστη μέσω μίας λογικής επεξεργασίας.
 - ▶ Διατηρούν ένα ή περισσότερα ILF.
- ▶ **Εξωτερικές ερωταποκρίσεις (External Inquiries EQs)**
 - ▶ Στέλνουν δεδομένα και πληροφορίες ελέγχου έξω από τα όρια της εφαρμογής.
 - ▶ Παρουσιάζουν πληροφορία στον χρήστη μέσω της ανάκτησης δεδομένων ή πληροφορίας ελέγχου.



Καθορισμός πολυπλοκότητας

- ▶ Αρχεία αναφοράς (File Types Referenced, FTRs)
 - ▶ Λογικά αρχεία στα οποία γίνεται προσπάθεια προκειμένου να παραχθεί συγκεκριμένο ΕΟ.
 - ▶ 1 FTR για κάθε λογικό αρχείο εισόδου ή εξόδου (ILF, EIF).
- ▶ Στοιχεία τύπου αρχείων (RETs)
 - ▶ υποομάδες στοιχείων δεδομένων μέσα στα ILFs ή EIFs
- ▶ Στοιχεία τύπου δεδομένων (Data Element Types, DETs)
 - ▶ Τα στοιχεία τύπου αρχείου αποτελούνται από στοιχεία τύπου δεδομένων.
 - ▶ Πεδία δεδομένων που εμφανίζονται στην έξοδο, στον χρήστη με βάση τις απαιτήσεις του.
 - ▶ 1 DET για κάθε πεδίο που εμφανίζεται.
 - ▶ Δεν μετρούνται στοιχεία που παράγονται από το σύστημα.



Πινάκες πολυπλοκότητας

	1 έως 19 DETs	20 έως 50 DETs	> 51 DETs
1 RET	Χαμηλή	Χαμηλή	Μέση
2 έως 5 RETs	Χαμηλή	Μέση	Υψηλή
> 6 RETs	Μέση	Υψηλή	Υψηλή

	1 έως 4 DETs	5 έως 15 DETs	16 έως DETs
0 έως 1 FTR	Χαμηλή	Χαμηλή	Μέση
2 FTRs	Χαμηλή	Μέση	Υψηλή
3 έως FTRs	Μέση	Υψηλή	Υψηλή



Υπολογισμός Complexity_i

ΤΥΠΟΣ ΣΤΟΙΧΕΙΟΥ	Συντελεστής C_i		
	Χαμηλή	Μέση	Υψηλή
Εξ. Είσοδος	3	4	6
Εξ. Εξοδος	4	5	7
Ερώτηση	3	4	6
Εξ. Αρχείο	7	10	15
Εσ. Αρχείο	5	7	10



Παράγοντες πολυπλοκότητας

- ▶ Distributed data processing
- ▶ Performance
- ▶ Heavily used configuration
- ▶ Transaction rate
- ▶ On-Line data entry
- ▶ End-user efficiency
- ▶ On-Line update
- ▶ Complex processing
- ▶ Reusability
- ▶ Installation ease
- ▶ Operational ease
- ▶ Multiple sites
- ▶ Facilitate change



Εξισώσεις

- ▶ Υπολογισμός της αμιγούς λειτουργικότητας:
(σε αμιγείς Βαθμούς Λειτουργίας)

$$UFC = \sum_{i=1}^n (\text{αα.στοιχείων } n) \times C_i$$

- ▶ Υπολογισμός του παράγοντα τεχνικής πολυπλοκότητας TCF

$$TCF = 0,65 + 0,01 \sum_{i=1}^{14} F_i$$

- ▶ Υπολογισμός της προσαρμοσμένης λειτουργικότητας:

$$FP = UFC \times TCF$$



Παράδειγμα

- ▶ Εφαρμογή ελέγχου ορθογραφίας
- ▶ Το πρόγραμμα δέχεται στην είσοδο ένα αρχείο - έγγραφο και προαιρετικά ένα αρχείο - προσωπικό λεξιλόγιο.
- ▶ Το πρόγραμμα παράγει μια λίστα από τις λέξεις που δεν βρέθηκαν στο γενικό λεξιλόγιο ή στο προσωπικό λεξιλόγιο
- ▶ Ο χρήστης μπορεί να ζητήσει τον αριθμό των λέξεων που ελέγχθηκαν και τον αριθμό των ορθογραφικών λαθών που βρέθηκαν

- ▶ $A = \# \text{ εξ. εισόδων} =$
- ▶ $B = \# \text{ εξ. εξόδων} =$
- ▶ $C = \# \text{ ερωταποκρίσεων} =$
- ▶ $D = \# \text{ εξ. αρχεία} =$
- ▶ $E = \# \text{ εσ. Αρχείων} =$



Παράδειγμα

- ▶ $A = \# \text{ εξ. εισόδων} = 2$ (όνομα αρχείου εγγράφου, όνομα αρχείου προσ. λεξιλογίου)
- ▶ $B = \# \text{ εξ. εξόδων} = 3$ (έκθεση λαθών, # λέξεων που δεν βρέθηκαν α) στο γενικό β) στο προσωπικό λεξιλόγιο)
- ▶ $C = \# \text{ ερωταποκρίσεων} = 2$ (# ελεγμένων λέξεων, # λαθών)
- ▶ $D = \# \text{ εξ. αρχεία} = 2$ (έγγραφο, προσ. λεξιλόγιο)
- ▶ $E = \# \text{ εσ. αρχείων} = 1$ (γενικό λεξιλόγιο)



Παράδειγμα

- ▶ Μέση πολυπλοκότητα σε όλες τις περιπτώσεις
 - ▶ $UFC = 4A + 5B + 4C + 10D + 7E = 4 \times 2 + 5 \times 3 + 4 \times 2 + 10 \times 2 + 7 \times 1 = 8 + 15 + 8 + 20 + 7 = 58$
- ▶ Υψηλή πολυπλοκότητα για το γενικό λεξιλόγιο και την έκθεση λαθών
 - ▶ $UFC = 4A + (5 \times 2 + 7 \times 1) + 4C + 10D + 10E = 63$
- ▶ Υπολογισμός TCF, π.χ. $TCF = 0,678$ (παίρνει τιμές από 0,65 μέχρι 1,35).
- ▶ $FP = 0,678 \times 58 = 39,3$.

ΤΥΠΟΣ ΣΤΟΙΧΕΙΟΥ	Συντελεστής C_i		
	Χαμηλή	Μέση	Υψηλή
Εξ. Είσοδος	3	4	6
Εξ. Εξοδος	4	5	7
Ερώτηση	3	4	6
Εξ. Αρχείο	7	10	15
Εσ. Αρχείο	5	7	10

ΠΑΡΑΔΕΙΓΜΑ – ΥΠΟΛΟΓΙΣΜΟΣ ΧΡΟΝΟΥ

Εάν τα FP που υπολογίστηκαν είναι

$$\mathbf{FP}_{\text{estimated}} = 39,3$$

η παραγωγικότητα του οργανισμού είναι = 6.5 FP/pm.

Ο Απαιτούμενος Χρόνος είναι 6 ανθρωπομήνες



Παράδειγμα/ Εφαρμογή Α.Δ

- ▶ Σε εφαρμογή ανθρώπινου δυναμικού το σύστημα διατηρεί πληροφορίες σχετικά με τις δουλειές των εργαζόμενων.
- ▶ Όλοι οι ωρομίσθιοι εργαζόμενοι θα πρέπει να πληρώνονται σε δολάρια των Η.Π.Α. Το σύστημα έχει πρόσβαση σε νομισματική εφαρμογή για να ανακτήσει την συναλλαγματική ισοτιμία.
- ▶ Ο χρήστης προσθέτει πληροφορίες εργαζομένων online
- ▶ Online αναφορά εργαζομένων + διάρκεια ανάθεσης εργασίας
- ▶ Παρουσίαση στοιχείων εργαζομένων.



Υπολογισμός ILF

- ▶ Το σύστημα διατηρεί πληροφορίες εργασίας.
- ▶ 2 ILF με πληροφορίες αναφορικά με την εργασία και τον εργαζόμενο.
- ▶ Η εργασία χαρακτηρίζεται από 2 RETs που διατηρούν γενικές και ειδικές πληροφορίες εργασίας
- ▶ Οι γενικές πληροφορίες εργασίας έχουν τα εξής πεδία:
 - ▶ Αριθμό εργασίας
 - ▶ Όνομα εργασίας
 - ▶ Βαθμός μισθού εργασίας
- ▶ Οι ειδικές πληροφορίες εργασίας έχουν τα εξής πεδία :
 - ▶ Αριθμό εργασίας
 - ▶ Διάρκεια ανάθεσης
 - ▶ Περιγραφή εργασίας.
- ▶ Οι πληροφορίες εργαζόμενου έχουν τα εξής πεδία:
 - ▶ Αριθμός εργαζόμενου
 - ▶ Όνομα
 - ▶ Όνομα εργασίας
 - ▶ Περιγραφή δεξιοτήτων
- ▶ Σύνολο 2 ILF με χαμηλή πολυπλοκότητα
- ▶ UFPC = $2 \times 5 = 10$



Υπολογισμός EIF

- ▶ Το σύστημα έχει πρόσβαση σε νομισματική εφαρμογή για να ανακτήσει την συναλλαγματική ισοτιμία.
- ▶ Ένα EIF, 1 RET Συναλλαγματική Πληροφορία
 - ▶ Συναλλαγματική Ισοτιμία
 - ▶ Χώρα
- ▶ Σύνολο 1 EIF, 1 RET, 2 DETs
- ▶ πολυπλοκότητα του EIF είναι χαμηλή
- ▶ UFPC = $1 \times 7 = 7$



Υπολογισμός EI

- ▶ Ο χρήστης προσθέτει πληροφορίες εργαζομένου online
- ▶ Μετράμε ένα FTR για κάθε ILF,EIF που διατηρείται.
- ▶ 1 ILF εργαζόμενος
 - ▶ Αριθμός εργαζόμενου
 - ▶ Όνομα
 - ▶ Όνομα εργασίας
 - ▶ Περιγραφή δεξιοτήτων
- ▶ Σύνολο 1 FTR, 4 DETs
- ▶ πολυπλοκότητα του EI είναι χαμηλή
- ▶ UFPC = $1 \times 3 = 3$



Υπολογισμός ΕΟ

- ▶ Online Αναφορά σχετικά με τη διάρκεια ανάθεσης εργασίας σε εργαζόμενο.
- ▶ ILF εργαζόμενος, εργασία διαβάζονται και εμφανίζονται τα πεδία:
 - ▶ αριθμός εργαζόμενου
 - ▶ όνομα εργαζόμενου
 - ▶ αριθμός εργασίας
 - ▶ όνομα εργασίας
 - ▶ διάρκεια ανάθεσης
 - ▶ σύνολα των εργαζομένων πάνω από 24 και 12 μήνες.
- ▶ Σύνολο 2 FTR, 6 DETs
- ▶ πολυπλοκότητα του ΕΟ είναι μέση.
- ▶ UFPC = $2 \times 5 = 10$



Υπολογισμός ΕQ

- ▶ Παρουσίαση στοιχείων εργαζομένων.
- ▶ ILF εργαζόμενοι:
 - ▶ Ονοματεπώνυμο
 - ▶ Θέση
- ▶ Σύνολο 1 FTR, 2 DETs
- ▶ πολυπλοκότητα του ΕQ είναι χαμηλή.
- ▶ UFPC = 3



Τελικοί υπολογισμοί

- ▶ $UFC = 2 \times 5 + 1 \times 7 + 1 \times 3 + 2 \times 5 + 1 \times 3 = 32$
- ▶ $TCF = 0,678$ (παίρνει τιμές από 0,65 μέχρι 1,35, αποδώσαμε τυχαία μια τιμή)
- ▶ $FP = 32 \times 0,678 = 21.69$



Πλεονεκτήματα

- ▶ Η εκτίμηση γίνεται στις πρώιμες φάσεις του κύκλου ζωής
- ▶ Εκτιμάται η λειτουργικότητα και όχι το μέγεθος του λογισμικού
- ▶ Το τελικό αποτέλεσμα προέρχεται από ανάλυση και όχι από καθαρή εκτίμηση
- ▶ Μεγάλη συσχέτιση με το μέγεθος – μπορεί να γίνει εκτίμηση του μεγέθους με βάση ιστορικά στοιχεία



Απεικόνιση FPs to LOC

- ▶ Χρήση εμπειρικής μεθόδου
 - ▶ Function point = count total $\times [0.65 + 0.01 \times (\text{sum of the } 14 F_i)]$
 - ▶ Companies may want to refine their own version
- ▶ Σύμφωνα με μια μελέτη για την υλοποίηση κάθε Function Point απαιτούνται οι παρακάτω γραμμές πηγαίου κώδικα ανά γλώσσα προγραμματισμού

▶ Assembly	320
▶ C	128
▶ COBOL	106
▶ C++	64
▶ Java	53
▶ Visual Basic	32
▶ SQL	12
- ▶ <http://www.qsm.com/resources/function-point-languages-table>
- ▶ Βλ. **www.ifpug.org** για περισσότερες πληροφορίες για τη μετρική FP3

Μειονεκτήματα

- ▶ Σε πολλά σημεία υπεισέρχεται η υποκειμενική κρίση του αναλυτή
- ▶ Δεν είναι δυνατό να γίνει ακριβής μέτρηση των Β.Λ. με αυτόματα εργαλεία ακόμη και μετά την ολοκλήρωση του λογισμικού
- ▶ Η μέθοδος δίνει καλά αποτελέσματα σε λογισμικό που χειρίζεται δεδομένα – δεν συνίσταται για επιστημονικές εφαρμογές ή συστήματα πραγματικού χρόνου



Εκτίμηση Κόστους με Στατιστικά Στοιχεία

Εκτίμηση Κόστους με Στατιστικά Στοιχεία

	A	B	C	D	E	F	G
1	This is a template for improving one's estimating skills						
2							
3	MAD = $\text{SUM}(A(t) - F(t))/n$ The Average Absolute Error						
4	Tracking Signal = $\text{SUM}(A(t) - F(t))/\text{MAD}$ A Measure of Bias						
5							Tracking
6	Period	Estimate	Actual	$A(t) - F(t)$	$ A(t) - F(t) $	MAD	Signal
7	=====	=====	=====	=====	=====	=====	=====
8							
9	1	155	163	8	8		
10	2	242	240	-2	2	5.00	1.20
11	3	46	67	21	21	10.33	2.61
12	4	69	78	9	9	10.00	3.60
13	5	75	71	-4	4	8.80	3.64
14	6	344	423	79	79	20.50	5.41
15	7	56	49	-7	7	18.57	5.60
16	8	128	157	29	29	19.88	6.69
17							
18				133	159		

Εκτίμηση Κόστους με Στατιστικά Στοιχεία

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							Tracking
6	Period	Estimate	Actual	$A(t) - F(t)$	$ A(t) - F(t) $	MAD	Signal
7	=====	=====	=====	=====	=====	=====	=====
8							
9	1	155	163	$= C9 - B9$	$= ABS(C9 - B9)$		
10	2	242	240	$= C10 - B10$	$= ABS(C10 - B10)$	$= (SUM($E$9:E10))/A10$	$= (SUM($D$9:D10))/F10$
11	3	46	67	$= C11 - B11$	$= ABS(C11 - B11)$	$= (SUM($E$9:E11))/A11$	$= (SUM($D$9:D11))/F11$
12	4	69	78	$= C12 - B12$	$= ABS(C12 - B12)$	$= (SUM($E$9:E12))/A12$	$= (SUM($D$9:D12))/F12$
13	5	75	71	$= C13 - B13$	$= ABS(C13 - B13)$	$= (SUM($E$9:E13))/A13$	$= (SUM($D$9:D13))/F13$
14	6	344	423	$= C14 - B14$	$= ABS(C14 - B14)$	$= (SUM($E$9:E14))/A14$	$= (SUM($D$9:D14))/F14$
15	7	56	49	$= C15 - B15$	$= ABS(C15 - B15)$	$= (SUM($E$9:E15))/A15$	$= (SUM($D$9:D15))/F15$
16	8	128	157	$= C16 - B16$	$= ABS(C16 - B16)$	$= (SUM($E$9:E16))/A16$	$= (SUM($D$9:D16))/F16$
17							
18				$= SUM(D9:D17)$	$= SUM(E9:E17)$		



Εκτίμηση Κόστους με Στατιστικά Στοιχεία

	A	B	C	D	E	F	G
1	This is a template for improving one's estimating skills						
2							
3	MAR = SUM $\{ (A(t) / F(t)) - 1 \} / n$						
4	Tracking Signal = SUM $\{(A(t) / F(t)) - 1\} / \text{MAR}$						
5							Tracking
6	Period	Estimate	Actual	$(A(t)/F(t)) - 1$	$ (A(t)/F(t)) - 1 $	MAR	Signal
7	=====	=====	=====	=====	=====	=====	=====
8							
9	1	155	163	0.052	0.052		
10	2	242	240	-0.008	0.008	0.030	1.448
11	3	46	67	0.457	0.457	0.172	2.904
12	4	69	78	0.130	0.130	0.162	3.898
13	5	75	71	-0.053	0.053	0.140	4.120
14	6	344	423	0.230	0.230	0.155	5.205
15	7	56	49	-0.125	0.125	0.151	4.523
16	8	128	157	0.227	0.227	0.160	5.670
17							
18				0.908	1.281		

Εκτίμηση Κόστους με Στατιστικά Στοιχεία

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							Tracking
6	Period	Estimate	Actual	$(A(t)/F(t)) - 1$	$ (A(t)/F(t)) - 1 $	MAR	Signal
7	=====	=====	=====	=====	=====	=====	=====
8							
9	1	155	163	$= (C9/B9) - 1$	$= ABS((C9/B9) - 1)$		
10	2	242	240	$= (C10/B10) - 1$	$= ABS((C10/B10) - 1)$	$= (SUM($E$9:E10))/A10$	$= (SUM($D$9:D10))/F10$
11	3	46	67	$= (C11/B11) - 1$	$= ABS((C11/B11) - 1)$	$= (SUM($E$9:E11))/A11$	$= (SUM($D$9:D11))/F11$
12	4	69	78	$= (C12/B12) - 1$	$= ABS((C12/B12) - 1)$	$= (SUM($E$9:E12))/A12$	$= (SUM($D$9:D12))/F12$
13	5	75	71	$= (C13/B13) - 1$	$= ABS((C13/B13) - 1)$	$= (SUM($E$9:E13))/A13$	$= (SUM($D$9:D13))/F13$
14	6	344	423	$= (C14/B14) - 1$	$= ABS((C14/B14) - 1)$	$= (SUM($E$9:E14))/A14$	$= (SUM($D$9:D14))/F14$
15	7	56	49	$= (C15/B15) - 1$	$= ABS((C15/B15) - 1)$	$= (SUM($E$9:E15))/A15$	$= (SUM($D$9:D15))/F15$
16	8	128	157	$= (C16/B16) - 1$	$= ABS((C16/B16) - 1)$	$= (SUM($E$9:E16))/A16$	$= (SUM($D$9:D16))/F16$
17							
18				$= SUM(D9:D16)$	$= SUM(E9:E16)$		

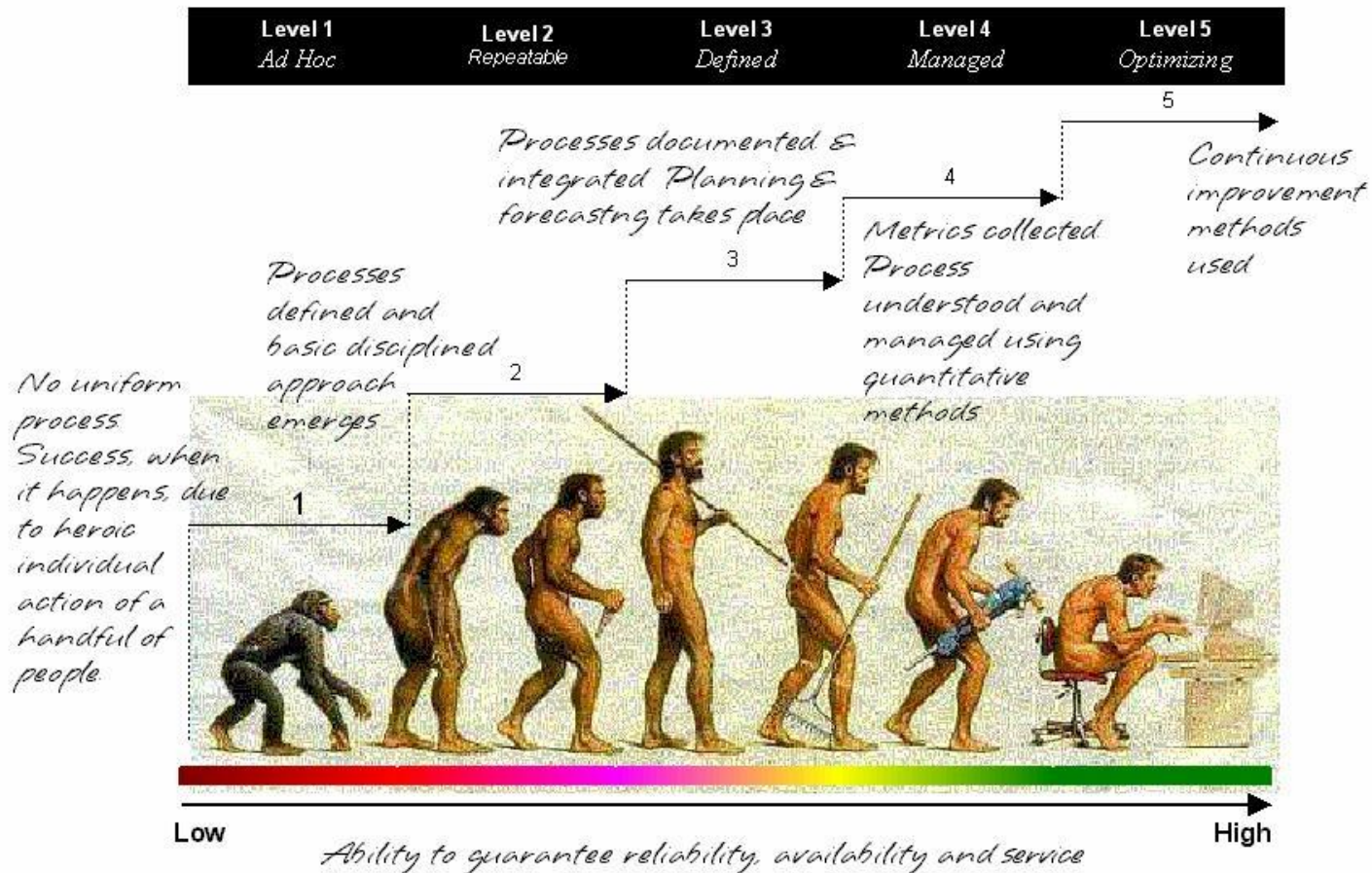




Process Maturity

Capability Maturity Model CMM

Evolution of Process Maturity

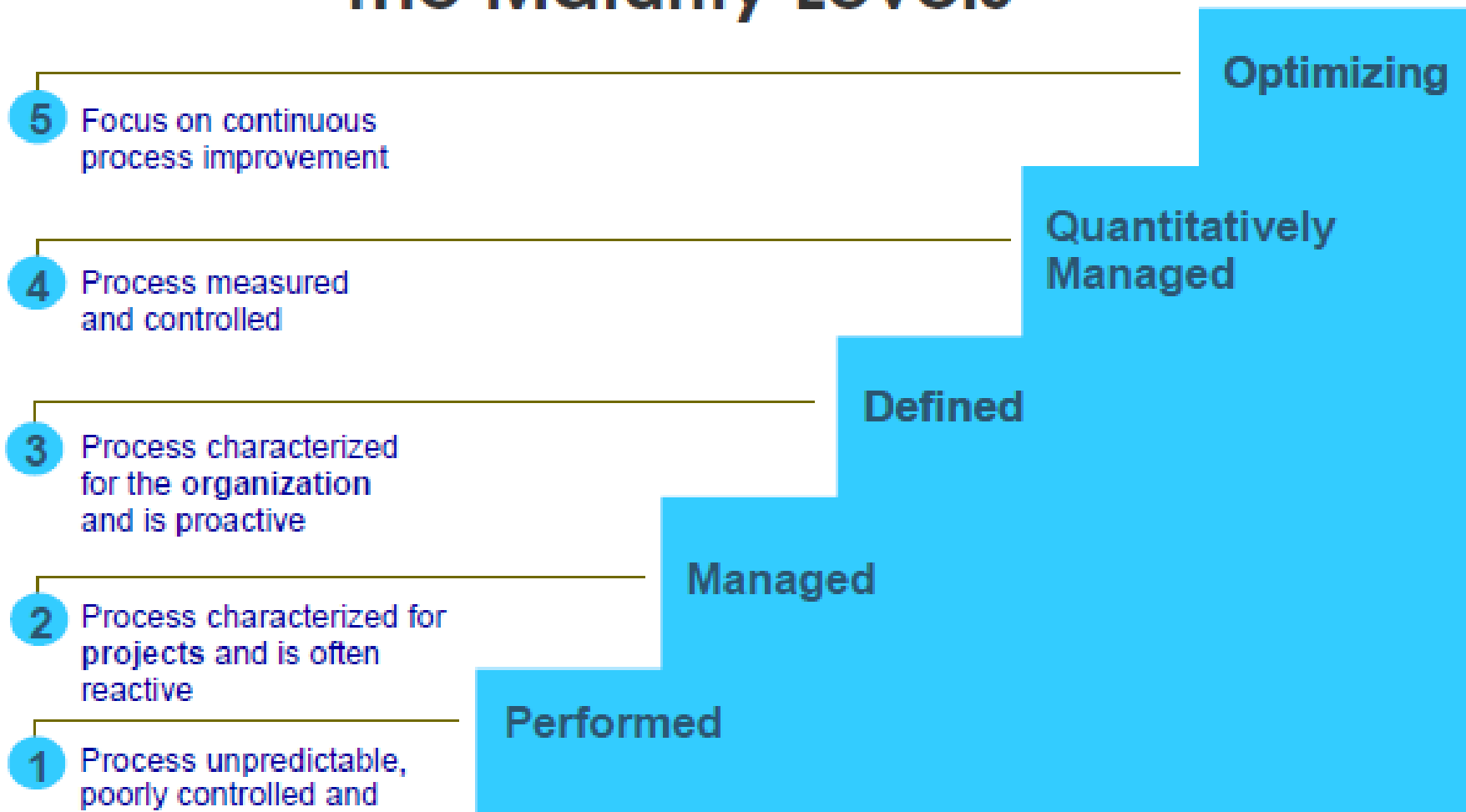


Το μοντέλο CMM – Capability maturity model

- ▶ Το Carnegie Mellon Software Engineering Institute αναπτύσσει το CMM στο τέλος της δεκαετίας του 1980(1986)
- ▶ Η επιτυχία του προκαλεί την δημιουργία πολλών CMM μοντέλων
- ▶ Το CMMI σκοπεύει στο να ενοποιήσει μερικά από τα CMM μοντέλα
- ▶ Αποτέλεσμα: Ένα μοντέλο-πλαίσιο, μία δέσμη ολοκληρωμένων μοντέλων



The Maturity Levels



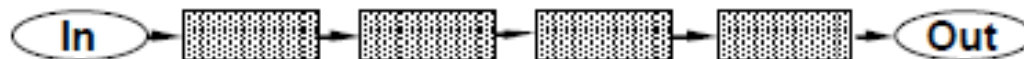
Maturity Level 1 “Performed”



- ▶ Requirements flow in.
- ▶ A product is produced by some undefined process
- ▶ The product flows out and hopefully it works.



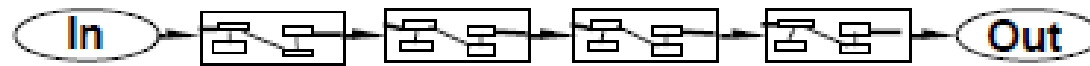
Maturity Level 2 “Managed”



- ▶ Requirements flow in.
- ▶ Plans are developed in accordance with policies
- ▶ Activities are performed in accordance with plans
- ▶ Measurements and reviews occur at defined points
- ▶ The product flows out and usually works



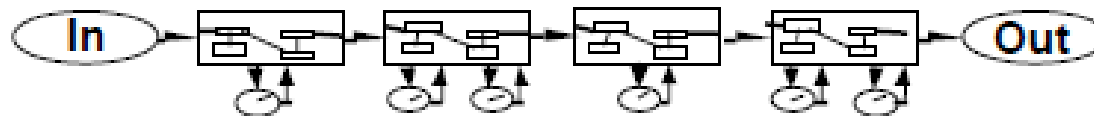
Maturity Level 3 “Defined”



- ▶ Requirements flow in
- ▶ Processes are tailored from standard process assets
- ▶ Plans are made and activities performed according to the tailored process.
- ▶ The product flows out with somewhat predictable quality .



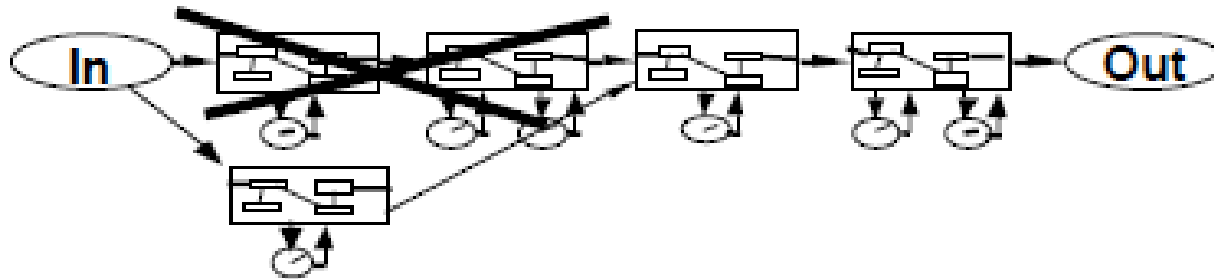
Maturity Level 4 “Quantitatively Managed”



- ▶ The behavior of the process is predictable and quantitatively understood.
- ▶ A quantitative basis exists for decisions to achieve established product quality, service quality, and process performance goals



Maturity Level 5 ‘Optimizing’



- ▶ Continual and measurable process improvement (while managing process stability) is a way of life.

