

# ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πάνος Φιτσιλής  
Γιάννης Σταμέλος

2007

## Περιεχόμενα

<b>ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ .....</b>	<b>5</b>
ΕΝΟΤΗΤΑ 1.1 ΕΙΣΑΓΩΓΗ.....	6
ΕΝΟΤΗΤΑ 1.2 ΑΠΑΙΤΟΥΜΕΝΕΣ ΙΚΑΝΟΤΗΤΕΣ ΓΙΑ ΤΗ ΔΙΑΧΕΙΡΙΣΗ ΈΡΓΩΝ.....	16
ΕΝΟΤΗΤΑ 1.3 ΟΙ ΦΑΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΝΟΣ ΈΡΓΟΥ.....	17
ΕΝΟΤΗΤΑ 1.4 ΟΙ ΣΥΜΜΕΤΕΧΟΝΤΕΣ ΣΕ ΕΝΑ ΈΡΓΟ.....	19
ΕΝΟΤΗΤΑ 1.5 ΓΙΑΤΙ ΤΑ ΈΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΔΙΑΦΕΡΟΥΝ; .....	22
ΕΝΟΤΗΤΑ 1.6 ΣΗΜΑΝΤΙΚΟΙ ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΕΠΗΡΕΑΖΟΥΝ ΤΗ ΔΙΑΧΕΙΡΙΣΗ ΈΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ. ....	24
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	30
<b>ΚΕΦΑΛΑΙΟ 2. ΟΡΓΑΝΩΤΙΚΕΣ ΔΟΜΕΣ – ΚΥΚΛΟΣ ΖΩΗΣ ΛΟΓΙΣΜΙΚΟΥ .....</b>	<b>33</b>
2.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΟΡΓΑΝΩΤΙΚΕΣ ΔΟΜΕΣ .....	33
2.2 ΟΡΓΑΝΩΤΙΚΕΣ ΔΟΜΕΣ ΕΡΓΩΝ .....	34
2.3 ΟΡΓΑΝΩΣΗ ΟΜΑΔΑΣ .....	48
2.4 ΣΥΝΟΨΗ – ΒΙΒΛΙΟΓΡΑΦΙΑ.....	55
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	56
<b>ΚΕΦΑΛΑΙΟ 3: ΔΙΑΧΕΙΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΟΥ ΕΡΓΑΣΙΩΝ.....</b>	<b>57</b>
ΕΝΟΤΗΤΑ 3.1 ΕΙΣΑΓΩΓΗ.....	58
ΕΝΟΤΗΤΑ 3.2 Ο ΟΡΙΣΜΟΣ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ ΕΡΓΑΣΙΩΝ ΕΝΟΣ ΈΡΓΟΥ.....	59
ΕΝΟΤΗΤΑ 3.3 ΔΙΑΧΕΙΡΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΛΟΓΙΣΜΙΚΟΥ.....	61
ΕΝΟΤΗΤΑ 3.4 ΔΟΜΗ ΑΝΑΛΥΣΗΣ ΕΡΓΑΣΙΩΝ.....	62
ΕΝΟΤΗΤΑ 3.5 ΔΙΑΧΕΙΡΙΣΗ ΠΑΡΑΔΟΤΕΩΝ.....	70
ΕΝΟΤΗΤΑ 3.6 ΈΛΕΓΧΟΣ ΑΛΛΑΓΩΝ.....	74
ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ .....	80
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	80
<b>ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΕΡΓΩΝ .....</b>	<b>82</b>
ΕΝΟΤΗΤΑ 4.1 ΕΙΣΑΓΩΓΗ.....	83
ΕΝΟΤΗΤΑ 4.2 ΑΝΑΠΤΥΞΗ ΠΛΑΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΈΡΓΟΥ.....	84
ΕΝΟΤΗΤΑ 4.3 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΈΡΓΟΥ.....	89
4.3.1 Το ΔΙΑΓΡΑΜΜΑ GANTT.....	90
4.3.2 ΑΝΑΛΥΣΗ ΚΡΙΣΙΜΟΥ ΜΟΝΟΠΑΤΙΟΥ.....	95
4.3.3 ΥΠΟΛΟΓΙΣΜΟΣ ΚΡΙΣΙΜΗΣ ΔΙΑΔΡΟΜΗΣ.....	102
4.3.4 Η ΑΒΕΒΑΙΟΤΗΤΑ ΣΤΟ ΧΡΟΝΟ ΟΛΟΚΛΗΡΩΣΗΣ ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ ΚΑΙ Η ΜΕΘΟΔΟΣ PERT.....	109

4.3.5 ΣΥΝΤΟΜΕΥΣΗ ΔΙΑΡΚΕΙΑΣ ΈΡΓΟΥ.....	113
4.3.6 ΣΧΟΛΙΑΣΜΟΣ ΤΩΝ ΜΕΘΟΔΩΝ CPM ΚΑΙ PERT.....	117
ΕΝΟΤΗΤΑ 4.4 ΔΙΑΧΕΙΡΙΣΗ ΠΟΡΩΝ.....	117
4.4.1 ΕΞΙΣΟΡΡΟΠΗΣΗ ΠΟΡΩΝ.....	119
ΕΝΟΤΗΤΑ 4.5 ΔΙΑΧΕΙΡΙΣΗ ΈΡΓΩΝ ΜΕ ΤΗ ΜΕΘΟΔΟ ΤΗΣ ΚΡΙΣΙΜΗΣ ΑΛΥΣΙΔΑΣ.....	126
ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ.....	131
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	131
<b>ΚΕΦΑΛΑΙΟ 5. ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ.....</b>	<b>133</b>
5.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ.....	135
5.2. ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ ΜΕ ΠΑΡΑΜΕΤΡΙΚΑ ΜΟΝΤΕΛΑ.....	141
5.3 ΕΚΤΙΜΗΣΗ ΜΕ ΒΑΣΗ ΤΙΣ ΑΝΑΛΟΓΙΕΣ.....	163
ΣΥΜΠΕΡΑΣΜΑΤΑ – ΒΙΒΛΙΟΓΡΑΦΙΑ.....	169
<b>ΑΠΑΝΤΗΣΕΙΣ ΣΤΙΣ ΑΣΚΗΣΕΙΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ.....</b>	<b>172</b>
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 1.1.....	172
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 1.2.....	172
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 2.1.....	173
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 2.2.....	174
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 2.3.....	175
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 2.4.....	175
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 3.1.....	176
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 3.2.....	176
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 4.1.....	177
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 4.2.....	178
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 5.1.....	180
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 5.2.....	180
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 5.3.....	181
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 5.4.....	182
ΑΠΑΝΤΗΣΗ ΆΣΚΗΣΗΣ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ 5.5.....	183
<b>ΕΝΔΕΙΚΤΙΚΕΣ ΑΠΑΝΤΗΣΕΙΣ ΣΤΙΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ.....</b>	<b>183</b>
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1.1.....	183
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1.2.....	185
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2.1.....	185
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2.2.....	188
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2.3.....	189
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2.4.....	190

ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3.1 .....	191
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3.2 .....	191
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4.1 .....	192
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4.2 .....	193
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4.3 .....	194
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5.1 .....	195
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5.2 .....	195
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5.4 .....	198
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5.5 .....	198
ΕΝΔΕΙΚΤΙΚΗ ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5.6 .....	199
<b>ΠΑΡΑΡΤΗΜΑ Α - ΑΘΡΟΙΣΤΙΚΗ ΚΑΝΟΝΙΚΗ ΚΑΤΑΝΟΜΗΣ ΠΙΘΑΝΟΤΗΤΑΣ (ΑΠΟ <math>-\infty</math> ΩΣ</b>	
<b>Z) .....</b>	<b>199</b>

## ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ

### Συγγραφέας: Πάνος Φιτσιλής

#### Σκοπός:

Σκοπός του κεφαλαίου 1 είναι να παρουσιάσουμε τις βασικές αρχές της διαχείρισης έργων λογισμικού. Θα παρουσιάσουμε τα βασικά χαρακτηριστικά των έργων λογισμικού, αλλά και πως αυτά διαφοροποιούνται από άλλων ειδών έργα. Στη συνέχεια θα παρουσιάσουμε τα βασικά πρότυπα που χρησιμοποιούνται για τη διαχείριση έργων λογισμικού.

#### Προσδοκώμενα Αποτελέσματα:

Στο κεφάλαιο αυτό:

- Θα κατανοήσετε τα βασικά χαρακτηριστικά ενός έργου.
- Θα μάθετε τι σημαίνει διαχείριση έργων λογισμικού.
- Θα κατανοήσετε πως διαφοροποιείται η διαχείριση έργων λογισμικού από τη διαχείριση άλλων έργων.
- Θα μελετήσετε τους παράγοντες επιτυχίας και αποτυχίας ενός έργου και τη σημασία της διαχείρισης έργων.
- Θα κατανοήσετε τις τέσσερις φάσεις διαχείρισης ενός έργου.
- Θα προσδιορίσετε τους διαφορετικούς ρόλους των συμμετεχόντων στα έργα.
- Θα δείτε τα πρότυπα που ακολουθούνται στη διαχείριση έργων λογισμικού, τις μεθοδολογίες και τις διαφορετικές προσεγγίσεις.
- Θα αναπτύξετε θετική στάση ως προς τη σημασία της σωστής διαχείρισης των έργων λογισμικού.

#### Έννοιες κλειδιά:

Έργο (Project).

Διαχείριση ενοποίησης έργου (project integration management).

Διαχείριση αντικειμένου εργασιών έργου (project scope management).

Διαχείριση χρόνου έργου (project time management).

Διαχείριση κόστους έργου (project cost management).

Διαχείριση ποιότητας έργου (project quality management).

Διαχείριση ανθρωπίνων πόρων έργου (project human resource management).

Διαχείριση επικοινωνίας έργου (project communication management).

Διαχείριση κινδύνου έργου (project risk management).

Διαχείριση προμηθειών έργου (project procurement management).

Σχέδιο διαχείρισης έργου (project management plan).

Σχέδιο ποιότητας έργου (project quality plan).

Διευθυντής έργου (project manager).

Χρονοδιάγραμμα έργου (schedule).

Κύκλος ζωής λογισμικού (software life cycle).

## **ΕΝΟΤΗΤΑ 1.1 Εισαγωγή.**

---

*Η εργοκεντρική διοίκηση θα αντικαταστήσει τη λειτουργική διοίκηση και σχεδόν όλοι οι οργανισμοί θα υιοθετήσουν επίπεδες και εύκαμπτες δομές οργάνωσης στη θέση των παλιών γραφειοκρατικών δομών... Η εργοκεντρική δομή θα χρησιμοποιηθεί σαν εργαλείο στον στρατηγικό σχεδιασμό και στην κατάκτηση και διατήρηση ανταγωνιστικού πλεονεκτήματος.*

*Rodney Turner*

*International Journal of Project Management*

Η διαχείριση έργων αναπτύχθηκε ως ξεχωριστό γνωστικό πεδίο σαν αποτέλεσμα της εφαρμογής των αρχών της διοίκησης και της επιχειρησιακής έρευνας σε διάφορους εφαρμοσμένους τομείς, όπως είναι οι κατασκευές, η μηχανολογία, τα μεγάλα στρατιωτικά προγράμματα, κ.α.

Πατέρας του γνωστικού πεδίου της διαχείρισης έργων θεωρείται ο Henry Gantt, που εισήγαγε τις αρχές του προγραμματισμού και ελέγχου στη διαχείριση έργων. Το γνωστό διάγραμμα Gantt, ένα ραβδόγραμμα που παρουσιάζει τις δραστηριότητες του έργου ονομάστηκε έτσι από αυτόν. Ο Gantt μαζί με τον Taylor, ο οποίος έθεσε τις αρχές της επιστημονικής διαχείρισης (scientific management), έθεσαν τις θεμελιώδεις αρχές της διαχείρισης έργων.

Οι βασικές αρχές της διαχείρισης έργων που έκαναν τη διαχείριση έργων ένα διακριτό γνωστικό αντικείμενο και την ανέδειξαν σε επάγγελμα αναπτύχθηκαν στη δεκαετία του '50. Τη δεκαετία αυτή αναπτύχθηκαν δύο βασικά μαθηματικά μοντέλα χρονοπρογραμματισμού δραστηριοτήτων, οι μέθοδοι PERT (Program Evaluation and Review Technique) και CPM (Critical Path Method) οι οποίες αποτέλεσαν σταθμό στη διαχείριση έργων. Η μέθοδος PERT αναπτύχθηκε από το Ναυτικό των Ηνωμένων

Πολιτειών για το έργο της ανάπτυξης των πυραυλικών συστημάτων Polaris. Αντίστοιχα, η μέθοδος CPM , γνωστή στα ελληνικά και ως μέθοδος Κρίσιμης Διαδρομής, αναπτύχθηκε από τις εταιρείες DuPont Corporation και Remington Rand Corporation με σκοπό τη διαχείριση έργων συντήρησης. Η διάδοση και αποδοχή των μεθόδων αυτών υπήρξε ταχύτατη, έτσι ώστε σήμερα να αποτελούν τις βασικές μεθόδους για τη διαχείριση έργων.

Σήμερα, ο χώρος της διαχείρισης έργων θεωρείται ιδιαίτερα αναπτυγμένος και προσελκύει ιδιαίτερο ενδιαφέρον, τόσο στον ιδιωτικό / δημόσιο τομέα, όσο και στην ακαδημαϊκή κοινότητα. Απόδειξη του γεγονότος αυτού αποτελεί η ύπαρξη πολλών και ιδιαίτερα δραστήριων διεθνών οργανισμών που έχουν ως στόχο την ανάπτυξη του γνωστικού πεδίου της διαχείρισης έργων.

### 1.1.1 Τι Είναι Έργο;

*Έργο είναι ένα προσωρινό εγχείρημα που στοχεύει στη δημιουργία ενός μοναδικού προϊόντος ή υπηρεσίας.*

Στον ορισμό αυτό:

- Προσωρινό σημαίνει ότι κάθε έργο έχει καθορισμένη έναρξη και λήξη.
- Μοναδικό σημαίνει ότι το προϊόν ή η υπηρεσία διαφέρει κατά διακριτό τρόπο από όλα τα παρόμοια προϊόντα ή υπηρεσίες.

Ο ορισμός αυτός αναφέρεται γενικότερα σε έργα και όχι μόνο σε έργα πληροφορικής. Οι παραπάνω ιδιότητες των έργων, να είναι δηλαδή προσωρινά, αλλά και μοναδικά εγχειρήματα, έρχονται σε αντίθεση με τη δομή των περισσότερων επιχειρήσεων που λειτουργούν βάσει διαδικασιών που έχουν σταθερό και μόνιμο χαρακτήρα. Η διαχείριση αυτών των ιδιοτήτων είναι συχνά δύσκολη επειδή απαιτεί συνδυασμό ιδιαίτερων ικανοτήτων που προέρχονται από διαφορετικά γνωστικά πεδία.

Έτσι, η πρώτη πρόκληση που αντιμετωπίζουμε στη διαχείριση έργων είναι να εξασφαλίσουμε ότι το έργο εκτελείται σωστά και παραδίδεται έγκαιρα λαμβάνοντας υπόψη καθορισμένους περιορισμούς. Οι περιορισμοί αυτοί μπορεί να είναι ο ανεπαρκής διαθέσιμος χρόνος, ο περιορισμένος προϋπολογισμός, κ.ά. Η δεύτερη πρόκληση, που είναι και η πιο φιλόδοξη, είναι η βελτιστοποίηση που απαιτείται να γίνει σε όλους τους παράγοντες που επηρεάζουν την εκτέλεση ενός έργου. Αυτό σημαίνει ότι σε ένα έργο επιλέγουμε τις δραστηριότητες που απαιτούνται με τέτοιο τρόπο ώστε να γίνεται

βέλτιστη χρήση των πόρων (χρόνος, χρήματα, άνθρωποι, υλικά, μηχανήματα, ενέργεια, χώρος, κ.ά.).

Έτσι, καταλήγουμε σε ένα δεύτερο ορισμό για το έργο:

*Έργο είναι ένα εγχείρημα κατά το οποίο ανθρώπινοι πόροι, μηχανές, οικονομικοί πόροι και πρώτες ύλες οργανώνονται κατά καινοφανή τρόπο, με στόχο την ανάληψη συγκεκριμένου αντικειμένου εργασιών που έχουν συγκεκριμένες προδιαγραφές και υπόκεινται σε δεδομένους κοστολογικούς και χρονικούς περιορισμούς, ώστε να παραχθεί μια επωφελής μεταβολή, η οποία ορίζεται μέσω ποσοτικών και ποιοτικών στόχων.*

Στην επόμενη λίστα παρουσιάζουμε τα βασικά χαρακτηριστικά ενός έργου:

- Αποτελείται από μη επαναλαμβανόμενες δραστηριότητες.
- Απαιτείται σχεδιασμός ώστε να επιτύχουμε το τελικό αποτέλεσμα.
- Το τελικό αποτέλεσμα είναι μοναδικό.
- Η εκτέλεση του έργου απαιτεί την ύπαρξη ομάδας.
- Έχει έναρξη και λήξη.
- Υπόκειται σε περιορισμούς διαφόρων ειδών (χρόνου, κόστους, ποιότητας, κ.ά.).
- Οι διαθέσιμοι πόροι είναι περιορισμένοι.

### **Άσκηση Αυτοαξιολόγησης 1.1**

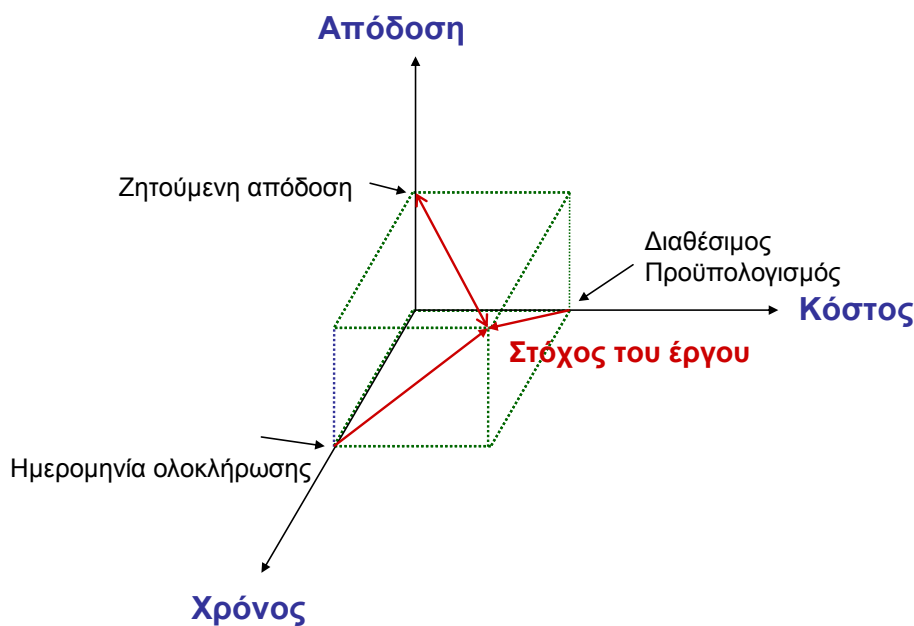
Ποιες από τα παρακάτω εργασίες μπορούν να θεωρηθούν ως έργα και ποια όχι; Τεκμηριώστε την άποψή σας.

	Σωστό	Λάθος
Η μελέτη και η κατασκευή κτιρίου / σπιτιού	<input type="checkbox"/>	<input type="checkbox"/>
Η δημιουργία μιας κινηματογραφικής ταινίας	<input type="checkbox"/>	<input type="checkbox"/>
Η ανάπτυξη μιας εφαρμογής μισθοδοσίας	<input type="checkbox"/>	<input type="checkbox"/>
Η τεχνική υποστήριξη πελατών	<input type="checkbox"/>	<input type="checkbox"/>
Η συντήρηση λογισμικού	<input type="checkbox"/>	<input type="checkbox"/>

Συνήθως, η μέτρηση της προόδου ενός έργου εξαρτάται από έναν αριθμό παραγόντων όπως είναι η τεχνολογία (προδιαγραφές, απόδοση, ποιότητα), ο χρόνος (ορόσημα, ημερομηνίες παράδοσης) και το κόστος (μέγεθος επένδυσης, ταμειακές ροές). Ο



συνδυασμός αυτών των παραγόντων, καθώς και η σχετική τους βαρύτητα αποτελεί βασική απόφαση για τη μέτρηση της απόδοσης, αλλά και της επιτυχίας ενός έργου. Η τριάδα αυτή των περιορισμών συχνά αναφέρεται και ως τρίγωνο διαχείρισης του έργου, όπου κάθε πλευρά αντιπροσωπεύει έναν περιορισμό. Αλλαγή στη μια κορυφή του τριγώνου προκαλεί αλλαγή στους περιορισμούς που σχετίζονται με τους άλλους παράγοντες. Έτσι, αλλαγή στο αντικείμενο των εργασιών του έργου προκαλεί αλλαγή στη διάρκεια του έργου, αλλά και στον προϋπολογισμό (Rosen, 2004). Το Σχήμα 1.1. παρουσιάζει την έννοια της επιτυχίας του έργου, η οποία εξαρτάται από το παραγόμενο αποτέλεσμα (απόδοση) σε σχέση με το ζητούμενο, το κόστος του έργου σε σχέση με το διαθέσιμο προϋπολογισμό και το χρόνο ο οποίος απαιτήθηκε για την ολοκλήρωσή του σε σχέση με το διαθέσιμο. Συνεπώς, η επιτυχία εξαρτάται από ένας συνδυασμός παραγόντων και για το λόγο αυτό θεωρούμε επιτυχημένη κάθε προσέγγιση που βελτιστοποιεί έναν ή περισσότερους από αυτούς.



**Σχήμα 1.1.:** Οι παράγοντες επιτυχίας ενός έργου.

Όλα τα έργα ξεκινούν για να ικανοποιήσουν μια ανάγκη. Για παράδειγμα, στην περίπτωση των έργων λογισμικού, μία τέτοια ανάγκη μπορεί να προέρχεται από:

- Την απαίτηση της αγοράς (μετά από τη μελέτη της αγοράς έγινε σαφές ότι απαιτείται η ανάπτυξη ενός νέου προϊόντος λογισμικού με αυξημένη απόδοση).

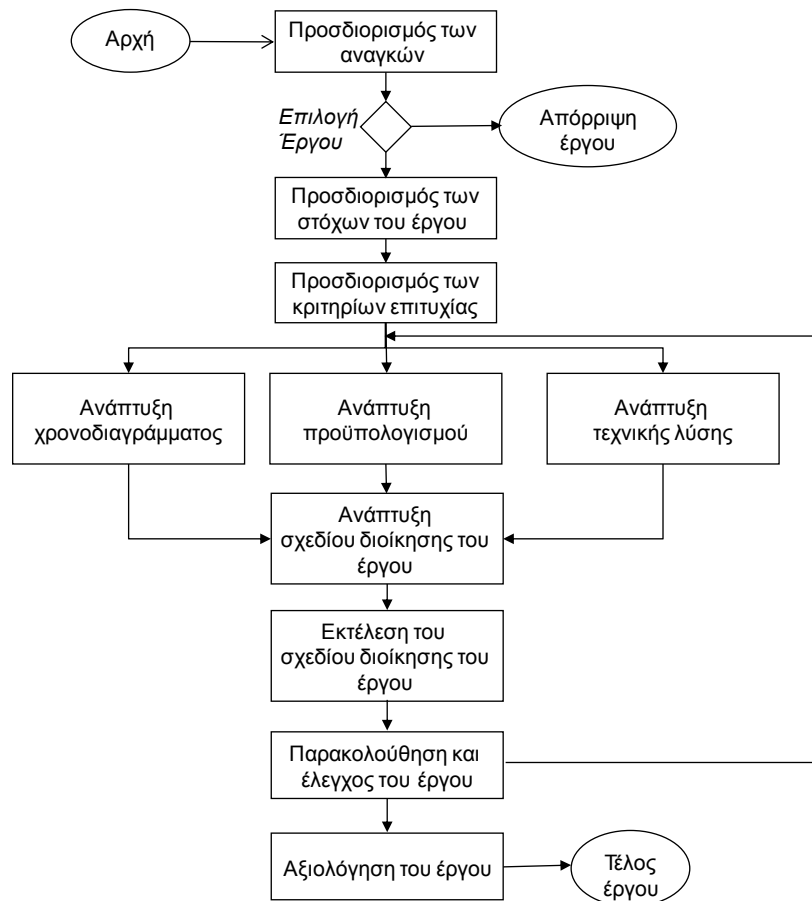
- Τις ανάγκες της επιχείρησης (η επιχείρηση μεγάλωσε σε μέγεθος και το γεγονός αυτό την αναγκάζει να αυτοματοποιήσει την οικονομική διαχείριση).
- Τις απαιτήσεις των πελατών (οι πελάτες επιθυμούν να δίνουν τις παραγγελίες τους ηλεκτρονικά).
- Την πρόοδο της τεχνολογίας (η εισαγωγή ενός καινούργιου συστήματος βάσεως δεδομένων οδηγεί στην εκ νέου συγγραφή τμημάτων ή και όλου του πληροφοριακού συστήματος).
- Νέες νομικές απαιτήσεις (ο νέος νόμος για την προστασία των προσωπικών δεδομένων αναγκάζει την επιχείρηση να ορίσει καινούργιες διαδικασίες ελέγχου).
- Κοινωνικές ανάγκες (η ρύπανση των υδάτων και η λειψυδρία οδηγεί την κυβέρνηση στην απόφαση για ανάπτυξη πληροφοριακού συστήματος με σκοπό τη διαχείριση των υδάτων).

Με βάση τις ανάγκες που έχουν εντοπισθεί γίνεται η επιλογή των συμφερότερων έργων (π.χ. αυτά με τη μεγαλύτερη κερδοφορία ή αυτά που εντάσσονται στη στρατηγική του οργανισμού). Συνεπώς, η επιλογή του έργου αναφέρεται στην αξιολόγηση των διαθέσιμων στοιχείων που μας επιτρέπουν να επιλέξουμε το έργο που θα προκρίνουμε ανάμεσα σε ένα σύνολο υποψηφίων έργων. Τα κριτήρια επιλογής έργων είναι πολλά και ποικίλα. Παρακάτω αναφέρονται ενδεικτικά παραδείγματα κριτηρίων αξιολόγησης (Meredith, 2005; Kerzner, 2003):

- Κριτήρια παραγωγής (πόσος χρόνος χρειάζεται για την παραγωγή, η ενόχληση που θα προκληθεί στο κοινωνικό σύνολο ή στην υφιστάμενη παραγωγή, η ενέργεια που απαιτείται, τα ειδικά χαρακτηριστικά ασφάλειας, κ.λπ.).
- Κριτήρια αγοράς (ποιο είναι το μέγεθος της στοχευόμενης αγοράς, σε ποια φάση ωριμότητας βρίσκεται, υπάρχει ανταγωνισμός, ποιοι είναι οι βασικοί κίνδυνοι, κ.λπ.).
- Οικονομικά κριτήρια (ποιο είναι το κόστος παραγωγής, ποια είναι η αναμενόμενη κερδοφορία, ποιες είναι οι χρηματοοικονομικές ροές, ποια είναι η περίοδος απόσβεσης, κ.λπ.).
- Κριτήρια προσωπικού (ποια είναι η συνολική προσπάθεια που απαιτείται για την κατασκευή του έργου ανά ειδικότητα και ανά φάση, αν είναι το προσωπικό επαρκώς εκπαιδευμένο ή αν θα χρειαστεί συμπληρωματική εκπαίδευση, κ.λπ.).

- Κριτήρια διοικητικά (ποιοι είναι οι κανονισμοί, ποια είναι η νομοθεσία που πρέπει να εφαρμοσθεί, υπάρχουν συγκεκριμένα πρότυπα που πρέπει να ακολουθηθούν κ.λπ.).

Μετά τη λήψη της απόφασης για την επιλογή του έργου θα πρέπει να ορίσουμε επακριβώς τους στόχους του έργου, οι οποίοι θα συνδέονται άμεσα με τους παράγοντες επιτυχίας του έργου. Συνήθως, τα περισσότερα έργα έχουν πολλαπλούς στόχους, οι οποίοι μπορεί να είναι λειτουργικοί, χρονικοί, οικονομικοί, κ.λπ. Λαμβάνοντας υπόψη τους προκαθορισθέντες στόχους αναπτύσσουμε τον προϋπολογισμό και το χρονοδιάγραμμα του έργου. Το επόμενο βήμα είναι να ολοκληρώσουμε το χρονοδιάγραμμα, τον προϋπολογισμό και την τεχνική λύση στο σχέδιο διοίκησης του έργου (project management plan). Καθώς το έργο ξενικά, παρακολουθούμε συνεχώς την εκτέλεσή του, κάνοντας διορθωτικές κινήσεις όπου απαιτείται. Το έργο αξιολογείται στο τέλος του με σκοπό να δούμε αν και κατά πόσο επιτεύχθηκαν οι στόχοι του και να καταγράψουμε τα συμπεράσματα που προέκυψαν. Τα βασικά αυτά στάδια εκτέλεσης ενός έργου περιγράφονται γραφικά στο Σχήμα 1.2. Είναι βέβαιο ότι το Σχήμα 1.2. δεν μπορεί να απεικονίσει με ακρίβεια όλα τα στάδια εκτέλεσης ενός έργου, αλλά προσδιορίζει όμως τα βασικά βήματα ώστε να μπορέσει ο αναγνώστης να κατανοήσει τη γενική εικόνα.



**Σχήμα 1.2.:** Τα βασικά στάδια εκτέλεσης ενός έργου.

### 1.1.2 Τι Είναι η Διαχείριση Έργων;

Σύμφωνα με το PMBOK (Project Management Body of Knowledge) ως διαχείριση έργων ορίζεται η διαδικασία κατά την οποία εφαρμόζουμε γνώσεις (knowledge), δεξιότητες (skills), εργαλεία (tools) και τεχνικές (techniques) κατά την εκτέλεση των δραστηριοτήτων ενός έργου με στόχο να ικανοποιήσουμε τις απαιτήσεις και τις προσδοκίες των συμμετεχόντων (stakeholders)<sup>1</sup>.

Επομένως, διαχείριση έργων είναι η διαδικασία ενσωμάτωσης όλων όσων πρέπει να γίνουν, καθώς το έργο διανύει τον κύκλο ζωής του, ώστε να ικανοποιηθούν οι στόχοι του έργου. Συνήθως οι στόχοι του έργου ορίζονται σε συνάρτηση του αντικείμενου εργασιών, του χρόνου, του κόστους και της ποιότητας, των συμμετεχόντων, οι οποίοι έχουν διαφορετικές ανάγκες και προσδοκίες και των απαιτήσεων.

Είναι σαφές ότι η διαχείριση έργων απαιτεί γνώσεις, ικανότητες αλλά και εμπειρία. Η τυποποίηση των απαιτούμενων γνώσεων ή πιο απλά ο βασικός «κορμός» των

<sup>1</sup> Στην ενότητα 1.4 ορίζουμε τους συμμετέχοντες σε ένα έργο.

απαιτούμενων γνώσεων περιγράφονται με συστηματικό τρόπο στο PMBOK (PMI, 2004), το οποίο έχει αναπτυχθεί από το Project Management Institute (PMI). Το PMBOK έχει αναγνωριστεί παγκοσμίως ως ένα από τα βασικά πρότυπα για τη διαχείριση έργων, τόσο από τις εταιρείες του χώρου, όσο και από διεθνείς οργανισμούς, όπως:

- Το Institute of Electrical and Electronics Engineers (IEEE, 2004), το οποίο υιοθέτησε πλήρως PMBOK ως πρότυπο IEEE (πρότυπο 1490-2003).
- Το American National Standards Institute (ANSI/PMI 99-001-2000).

Το PMBOK είναι δομημένο σε εννέα γνωστικές περιοχές κάθε μια από τις οποίες προσδιορίζει τις διαδικασίες τις οποίες πρέπει να εκτελέσουμε ώστε να ολοκληρωθεί με επιτυχία το έργο. Οι εννέα γνωστικές περιοχές που ορίζονται στο PMBOK είναι οι ακόλουθες (PMI, 2004):

### **1) Διαχείριση ενοποίησης έργου (project integration management).**

Ενοποιεί τις βασικές διαδικασίες που εφαρμόζονται για τη διαχείριση του έργου με σκοπό το συντονισμό των διαδικασιών του έργου. Βασικός στόχος της διαχείρισης ενοποίησης έργου είναι η ανάπτυξη του σχεδίου διαχείρισης έργου (project management plan), το οποίο και αποτελεί το σημαντικότερο παραδοτέο της διαχείρισης του έργου. Το σχέδιο διαχείρισης έργου χρησιμοποιείται για να καθοδηγήσει την εκτέλεση του έργου, να τεκμηριώσει τις παραδοχές του έργου, να τεκμηριώσει τις αποφάσεις που έχουν ληφθεί σε σχέση με τις εναλλακτικές λύσεις, να τεκμηριώσει τις δραστηριότητες που θα εκτελεστούν, να καθορίσει τα πρότυπα που θα χρησιμοποιηθούν, καθώς και τον τρόπο που θα εφαρμοστούν, να καθορίσει τον τρόπο ελέγχου και αποδοχής των παραδοτέων, να καθορίσει τον τρόπο μέτρησης της απόδοσης του έργου κ.ά. Θα μπορούσαμε να παρομοιάσουμε το σχέδιο διαχείρισης έργου με τη «βίβλο» του έργου.

Μια δεύτερη βασική διαδικασία που εκτελείται μέσα στα πλαίσια της διαχείρισης ενοποίησης έργου είναι ο ενοποιημένος έλεγχος αλλαγών (integrated change control). Ως ενοποιημένο έλεγχο αλλαγών ορίζουμε τη διαδικασία που περιλαμβάνει την ανασκόπηση όλων των αιτημάτων αλλαγών (change requests), την έγκριση/απόρριψη των αλλαγών, καθώς και τον έλεγχο των αλλαγών σε παραδοτέα (deliverables).

### **2) Διαχείριση αντικειμένου εργασιών έργου (project scope management).**

Η διαχείριση αντικειμένου εργασιών έργου έχει ως στόχο να προσδιορίσει επακριβώς όλες τις εργασίες που απαιτούνται για την ολοκλήρωση του έργου. Για το σκοπό αυτό αναπτύσσεται μία λεπτομερής έκθεση που αφορά στο αντικείμενο του έργου (project scope statement) και η οποία αποτελεί τη βάση για μελλοντικές αποφάσεις σχετικά με το

έργο, καθώς και η δομή ανάλυσης εργασιών (Work Breakdown Structure, WBS) του έργου, που είναι γνωστότερη ως WBS δομή. Η δομή WBS είναι ιεραρχική και αναλύει το έργο σε φάσεις (phases) και πακέτα εργασίας (work packages), μέχρι να καταλήξει στις συγκεκριμένες δραστηριότητες (activities/tasks) του έργου. Στα έργα ανάπτυξης λογισμικού, η διαχείριση του αντικειμένου εργασιών έργου είναι σε μεγάλο βαθμό ταυτόσημη με τη διαχείριση απαιτήσεων (requirements management) και αποτελεί μια από τις δυσκολότερες δραστηριότητες του έργου.

### **3) Διαχείριση χρόνου έργου (project time management).**

Στόχος των διαδικασιών διαχείρισης χρόνου έργου είναι η εξασφάλιση της έγκαιρης παράδοσης του έργου. Για το σκοπό αυτό χρειάζεται να προσδιορίσουμε τις δραστηριότητες που απαιτούνται για την εκτέλεση του έργου, την αλληλουχία των δραστηριοτήτων, τη διάρκεια της καθεμιάς, τους περιορισμούς που υπάρχουν, κ.ά. Το τελικό αποτέλεσμα είναι το χρονοδιάγραμμα (workplan) του έργου. Το χρονοδιάγραμμα αποτελεί βασικό εργαλείο διαχείρισης και ελέγχου του έργου και χρησιμοποιείται για το συντονισμό του προσωπικού του έργου, των πόρων, των εργασιών, κ.ά. Καθυστερήσεις στην εκτέλεση των επιμέρους δραστηριοτήτων του έργου οδηγούν σε αύξηση του κόστους και σε συνολική καθυστέρηση του έργου.

### **4) Διαχείριση κόστους έργου (project cost management).**

Η διαχείριση κόστους έργου αποτελεί βασική γνωστική περιοχή της διαχείρισης έργων επειδή η οικονομική συνιστώσα είναι παρούσα και σημαντική σε όλα τα έργα. Βασικός στόχος της διαχείρισης κόστους αποτελεί η σύνταξη και η παρακολούθηση του προϋπολογισμού (budget) του έργου. Για τη σύνταξη του προϋπολογισμού απαιτείται η λεπτομερής κοστολόγηση όλων των δραστηριοτήτων, αλλά και των πόρων που απαιτούνται για την εκτέλεση του έργου. Η επιτυχής παρακολούθηση του κόστους συνεπάγεται την όσο δυνατόν γρηγορότερη καταγραφή των αποκλίσεων κόστους με σκοπό τη λήψη διορθωτικών μέτρων.

### **5) Διαχείριση ποιότητας έργου (project quality management).**

Η διαχείριση ποιότητας σε ένα έργο περιλαμβάνει το σχεδιασμό ποιότητας (quality planning), τη διασφάλιση ποιότητας (quality assurance) και τον ποιοτικό έλεγχο (quality control). Στόχος αυτών των τριών διαδικασιών είναι να εξασφαλίσουμε ότι το αποτέλεσμα του έργου ικανοποιεί τις ανάγκες. Για τη μέτρηση της ποιότητας μπορούμε να χρησιμοποιήσουμε διαφορετικές οπτικές γωνίες ανάλογα με τις προτεραιότητες των συμμετεχόντων. Για παράδειγμα, μπορούμε να μετρήσουμε την απόδοση του συστήματος

(performance), τα λειτουργικά του χαρακτηριστικά (features), την αξιοπιστία (reliability), την προσκόλληση στις προδιαγραφές και τα πρότυπα (conformance), κ.ά. Όλα αυτά προσδιορίζονται επακριβώς στη διαδικασία της ανάπτυξης του σχεδίου ποιότητας του έργου (project quality plan), το οποίο μεταξύ άλλων περιλαμβάνει τον ορισμό διαδικασιών διασφάλισης ποιότητας και ποιοτικού ελέγχου.

#### **6) Διαχείριση ανθρωπίνων πόρων έργου (project human resource management).**

Στόχος της διαχείρισης ανθρωπίνων πόρων είναι η εύρεση της βέλτιστης οργανωτικής δομής που μπορεί να φέρει εις πέρας το έργο. Επειδή όμως τα έργα είναι προσωρινά και δεν αντιστοιχούν σε μια σταθερή οργανωτική δομή, θα πρέπει να συμμετέχουν σ' αυτά άτομα από διαφορετικές οργανωτικές δομές, τμήματα, κ.ά. Επομένως, υπάρχει σημαντική ανάγκη για σωστή ανάπτυξη της ομάδας του έργου (team building).

Η θεματική ενότητα της διαχείρισης ανθρωπίνων πόρων περιλαμβάνει εκτός από το σχεδιασμό της οργανωτικής δομής του έργου, τη στελέχωση της ομάδας του έργου, τη διοίκηση του προσωπικού στα πλαίσια του έργου, ενέργειες για την ανάπτυξη του ομαδικού πνεύματος της ομάδας, την οργάνωση της επικοινωνίας κ.λπ.

#### **7) Διαχείριση επικοινωνίας έργου (project communication management).**

Διαχείριση επικοινωνίας έργου είναι η διαδικασία της επιλογής των καναλιών, του τρόπου και του περιεχομένου της επικοινωνίας μεταξύ των συμμετεχόντων σε ένα έργο. Βασικές ερωτήσεις που πρέπει να απαντήσουμε κατά τη σχεδίαση της επικοινωνίας είναι: Τι είδους πληροφορία θα παράγουμε μέσα στο έργο, σε τι μορφή θα παραχθεί, υπάρχουν πρότυπα έγγραφα που θα πρέπει να χρησιμοποιήσουμε, πως θα συλλέξουμε την πληροφορία που περιέχεται στις αναφορές μας, πόσο συχνά θα παράγουμε τις αναφορές μας κ.ά.

#### **8) Διαχείριση κινδύνου έργου (project risk management)<sup>2</sup>.**

Το γεγονός ότι τα έργα αποτελούν στις περισσότερες των περιπτώσεων σύνθετες δραστηριότητες, μοναδικές και μη επαναλαμβανόμενες οδηγεί στο συμπέρασμα ότι η έννοια του έργου είναι συνδεδεμένη με την έννοια του ρίσκου και του κινδύνου. Επίσης, η έννοια του ρίσκου του έργου είναι στενά συνδεδεμένη με το κέρδος που μπορεί να φέρει ένα έργο, αφού, όσο μεγαλύτερος είναι ο κίνδυνος, τόσο μεγαλύτερο είναι και το

---

<sup>2</sup> Ο όρος διαχείριση κινδύνου δεν αποδίδει με απόλυτα σωστό τρόπο τον αγγλικό όρο. Μια καλύτερη απόδοση θα ήταν διαχείριση ρίσκου, όρος που περιλαμβάνει, τόσο την έννοια του κινδύνου, όσο και την έννοια του δυνητικού κέρδους. Ο λόγος που μας οδηγεί στην υιοθέτηση του πρώτου από τους δύο όρους είναι ότι έχει επικρατήσει στην ελληνική βιβλιογραφία. Επιπλέον, στην πράξη ακόμη και όταν αναφερόμαστε στη διαχείριση του ρίσκου εστιάζουμε την προσοχή μας στη διαχείριση των αρνητικών ενδεχομένων του έργου, δηλαδή στη διαχείριση του κινδύνου.

πιθανό κέρδος. Συνεπώς, η διαχείριση κινδύνου δεν είναι ένας τρόπος για να αποφύγουμε τον κίνδυνο, αλλά για να τον ελέγξουμε. Σύμφωνα με το PMBOK, η διαχείριση κινδύνου γίνεται με έξι διαδικασίες, οι οποίες είναι ο σχεδιασμός διαχείρισης κινδύνου (risk management planning), ο προσδιορισμός κινδύνων (risk identification), η ποιοτική ανάλυση κινδύνου (qualitative risk analysis), η ποσοτική ανάλυση κινδύνου (quantitative risk analysis), ο σχεδιασμός απόκρισης στους κινδύνους (risk response planning) και τέλος η παρακολούθηση και ο έλεγχος κινδύνων (risk monitoring and control).

### **9) Διαχείριση προμηθειών έργου (project procurement management).**

Διαχείριση προμηθειών έργου είναι η διαδικασία με την οποία εξασφαλίζεται η προμήθεια αγαθών και υπηρεσιών από πηγές που βρίσκονται εκτός της ομάδας εκτέλεσης του έργου ή εκτός του οργανισμού που εκτελεί το έργο. Η διαχείριση προμηθειών έργου περιλαμβάνει διαδικασίες που ορίζουν πως να αποφασίσουμε ποια τμήματα των παραδοτέων του έργου να προμηθευτούμε και ποια να κατασκευάσουμε μόνοι μας, πως να κάνουμε την προμήθεια ώστε να εξασφαλίσουμε τους καλύτερους όρους, πως να διαχειριστούμε τους προμηθευτές μας, κ.ά.

## **ΕΝΟΤΗΤΑ 1.2 Απαιτούμενες Ικανότητες για τη Διαχείριση Έργων.**

Αντίστοιχα, με το PMI, η Διεθνής Ομοσπονδία Διαχείρισης Έργων (International Project Management Association, IPMA) ανέπτυξε τον οδηγό IPMA Competence Baseline (ICB), όπου περιγράφονται με λεπτομέρεια οι ικανότητες (competences) που απαιτούνται για τη διαχείριση ενός έργου. Αυτές κατατάσσονται σε τρεις κατηγορίες (IPMA, 2006):

- Τεχνικές ικανότητες (technical competences). Οι τεχνικές ικανότητες περιλαμβάνουν τις ικανότητες να ξεκινήσουμε, να σχεδιάσουμε, να παρακολουθήσουμε και να ελέγξουμε ένα έργο.
- Ικανότητες συμπεριφοράς (behavioral competences). Παραδείγματα ικανοτήτων συμπεριφοράς είναι η ηγεσία, η αξιοπιστία, η ικανότητα επίλυσης προβλημάτων, η δημιουργικότητα κ.ά.
- Ικανότητες γνώσης του περιβάλλοντος (contextual competences). Περιλαμβάνουν τη γνώση του οργανισμού που έχει αναλάβει το έργο, καθώς και των υφιστάμενων διαδικασιών και γενικότερα η γνώση του περιβάλλοντος της υλοποίησης του έργου.



Σύμφωνα με τον οδηγό ICB, καθεμία από τις πιο πάνω ικανότητες είναι το άθροισμα της σχετικής γνώσης και εμπειρίας. Επιπλέον, παρατηρούμε μια σημαντική διαφορά στην προσέγγιση σε σχέση με το PMBOK. Η διαχείριση έργων δεν είναι απλά ένα τεχνικό θέμα, αλλά σημαντικό ρόλο έχει, τόσο ο ανθρώπινος παράγοντας, όσο και η γνώση του περιβάλλοντος του έργου.

Μια άλλη προσέγγιση για την τυποποίηση της γνώσης στη διαχείριση έργων αποτελεί η προσέγγιση της APM (Association for Project Management) (<http://www.apm.org.uk>) σύμφωνα με την οποία έχουν εντοπισθεί σαράντα βασικές ικανότητες οι οποίες ομαδοποιούνται στις ακόλουθες κατηγορίες:

- Διαχείριση έργων – καλύπτει όλα τα στοιχεία που διαφοροποιούν τη διαχείριση έργων από τη γενική διοίκηση.
- Οργάνωση και ομάδα έργου – καλύπτει τις οργανωτικές δομές και περιγράφει τις απαιτούμενες ικανότητες για το διευθυντή έργου.
- Τεχνικές και διαδικασίες – καλύπτει όλες τις τεχνικές, τις μεθόδους και τις διαδικασίες που χρησιμοποιούνται στη διαχείριση έργων.
- Γενική διοίκηση – καλύπτει θέματα γενικής διοίκησης.

### **ΕΝΟΤΗΤΑ 1.3 Οι Φάσεις Διαχείρισης ενός Έργου.**

Σύμφωνα με το Maylor (Maylor, 2006) οι φάσεις διαχείρισης ενός έργου είναι οι ακόλουθες τέσσερις (βλ. Σχήμα 1.3.):

- Ορισμός του έργου.
- Σχεδιασμός της διαδικασίας του έργου.
- Παράδοση του έργου.
- Ανάπτυξη της διαδικασίας του έργου.

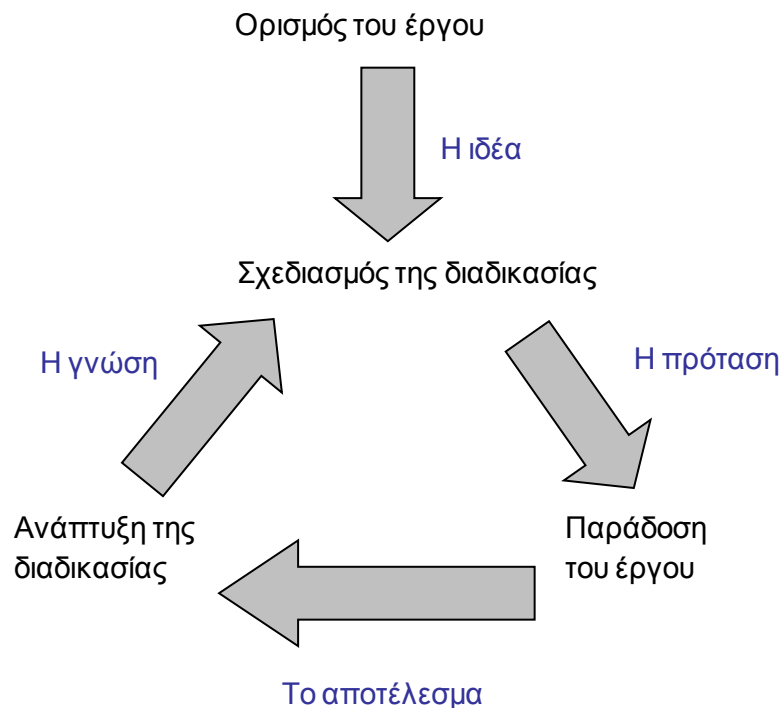
Στη φάση του **ορισμού του έργου** θα πρέπει να καθορίσουμε τους λόγους ύπαρξης και τους στόχους του έργου, καθώς και να εξετάσουμε τις εναλλακτικές λύσεις για τα προβλήματα που προβλέπουμε ότι θα παρουσιασθούν. Δεν θα πρέπει να ξεχνάμε ότι το έργο είναι ένα μέσο υλοποίησης μιας στρατηγικής, είτε αυτή αφορά τον οργανισμό, είτε είναι προσωπική. Αυτό σημαίνει ότι, η στρατηγική και οι στόχοι του έργου θα πρέπει να ενταχθούν στη στρατηγική και στους στόχους του οργανισμού.

Στην επόμενη φάση του **σχεδιασμού της διαδικασίας** του έργου θα πρέπει να κατασκευάσουμε ένα μοντέλο το οποίο θα αποτελεί το βέλτιστο τρόπο εκτέλεσης του έργου λαμβάνοντας υπόψη τους διαθέσιμους πόρους. Θα πρέπει να σημειωθεί ότι ακόμη

και σε αυτό το σημείο δεν έχει δοθεί ακόμη η έγκριση για την έναρξη του έργου μια και η έγκριση του έργου έρχεται αφού πρώτα αξιολογηθεί το χρηματοοικονομικό κόστος και το όφελος του έργου.

Η φάση της **παράδοσης** του έργου ξεκινά με τη συγκρότηση των ομάδων του έργου και τη συγκέντρωση των απαιτούμενων πόρων, ενώ συνεχίζεται με την εκτέλεση των δραστηριοτήτων που έχουν προγραμματισθεί και την παράδοση του τελικού παραδοτέου του έργου.

Η τελική φάση είναι αυτή της **ανάπτυξης της διαδικασίας** του έργου, όπου γίνεται η ανάλυση των αποτελεσμάτων του έργου για όλους τους συμμετέχοντες. Το αποτέλεσμα είναι η εφαρμογή βελτιώσεων σε διαδικασίες και μεθόδους. Επομένως, η φάση της ανάπτυξης της διαδικασίας του έργου είναι φάση αξιολόγησης του έργου και εξαγωγής συμπερασμάτων.

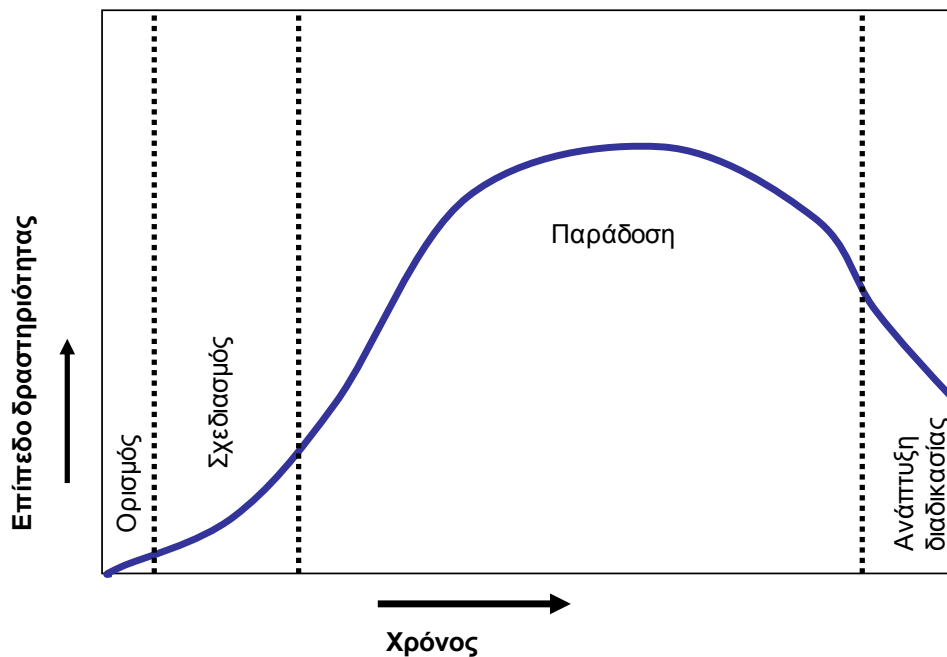


**Σχήμα 1.3.:** Οι φάσεις διαχείρισης ενός έργου.

Οι παραπάνω φάσεις δε γίνονται σειριακά, δηλαδή η μια μετά την άλλη, αλλά αντίθετα στην πράξη υπάρχει πάντα επικάλυψη. Για παράδειγμα, μέρος του ορισμού του έργου γίνεται παράλληλα με το σχεδιασμό της διαδικασίας. Αν και όλες οι φάσεις του έργου είναι σημαντικές και απαραίτητες, δίνουμε ιδιαίτερη έμφαση στο σχεδιασμό της διαδικασίας, η οποία αποτελεί τη βάση της διαχείρισης του έργου, αφού είναι η

φάση κατά την οποία ορίζουμε τον τρόπο με τον οποίο θα εκτελεστεί το έργο. Ο σχεδιασμός της διαδικασίας του έργου, από την ομάδα διαχείρισης του έργου, είναι χρονοβόρα και επίπονη διαδικασία, η οποία μάλιστα πρέπει να γίνει σε περιορισμένο χρονικό διάστημα, έτσι ώστε να μπορέσουν να ξεκινήσουν οι υπόλοιπες δραστηριότητες του έργου.

Αν και ο βασικός όγκος των εργασιών του σχεδιασμού εκτελούνται στην έναρξη του έργου, οι διαδικασίες σχεδιασμού συνεχίζονται σχεδόν μέχρι και τη λήξη αυτού, με την μορφή των αλλαγών και βελτιώσεων στις ήδη εφαρμοζόμενες διαδικασίες. Το Σχήμα 1.4. παρουσιάζει τη μεταβολή του επιπέδου δραστηριότητας (effort) που απαιτείται μια δεδομένη χρονική στιγμή στο έργο..



**Σχήμα 1.4.:** Επίπεδο δραστηριότητας ανά φάση διαχείρισης ενός έργου.

## ΕΝΟΤΗΤΑ 1.4 Οι Συμμετέχοντες σε ένα Έργο.

Οι άνθρωποι ή οι οργανισμοί που συμμετέχουν σε ένα έργο άμεσα ή έμμεσα περιγράφονται με μια λέξη ως *συμμετέχοντες (stakeholders)*. Οι συμμετέχοντες επηρεάζουν τόσο τους στόχους του έργου όσο και τα αποτελέσματα. Οι συμμετέχοντες σε ένα έργο μπορούν να διαχωριστούν στις ακόλουθες γενικές κατηγορίες:

- Σε αυτούς που επωφελούνται από το αποτέλεσμα του έργου ή θα χρησιμοποιούν το προϊόν που παράγει το έργο. Για την κατηγορία αυτή

χρησιμοποιείται ο όρος πελάτης ή/και χρήστης. Σε κάποιες περιπτώσεις ο πελάτης είναι ο ίδιος με το χρήστη, ενώ σε άλλες όχι. Στη γενική περίπτωση όπου πρόκειται για διαφορετικά άτομα, με τον όρο πελάτης (customer) αναφερόμαστε στην οντότητα που παραγγέλνει το έργο, ενώ με τον όρο χρήστης (user) αναφερόμαστε σε αυτόν που χρησιμοποιεί το σύστημα που παράγεται από το έργο.

- Σε αυτούς που χρηματοδοτούν ή διοικούν το έργο. Ο χρηματοδότης του έργου ονομάζεται και χορηγός (sponsor), ενώ η ομάδα διοίκησης του έργου ονομάζεται ομάδα διοίκησης του έργου (project management team).
- Στην ομάδα του έργου (project team) που εκτελεί το έργο. Σε πολλές περιπτώσεις η ομάδα έργου ανήκει σε έναν τρίτο οργανισμό, ο οποίος αναλαμβάνει την εκτέλεση του έργου και ονομάζεται ανάδοχος.

Κάθε έργο είναι μοναδικό και συνεπώς οι ρόλοι που υπάρχουν κάθε φορά είναι διαφορετικοί. Ακόμη και στην περίπτωση που χρησιμοποιείται το ίδιο όνομα για ένα ρόλο, η περιγραφή των καθηκόντων του ρόλου μπορεί να είναι διαφορετική. Επιπλέον, ο αριθμός των διαφορετικών ρόλων που χρησιμοποιούνται διαφοροποιείται ανάλογα με το είδος και την πολυπλοκότητα του έργου. Τα ονόματα των ρόλων που χρησιμοποιούνται στα έργα ποικίλουν. Ενδεικτικά αναφέρονται ο διευθυντής έργου, ο υπεύθυνος ποιότητας, ο υπεύθυνος ομάδας, ο αναλυτής απαιτήσεων, ο σχεδιαστής του έργου, κ.λπ.

Ανάμεσα στους παραπάνω ρόλους ιδιαίτερη βαρύτητα έχει ο ρόλος του διευθυντή του έργου (project manager). Ο διευθυντής του έργου θα πρέπει να ισορροπήσει και να διαχειριστεί τις προσδοκίες όλων των συμμετεχόντων, οι οποίες σε πολλές περιπτώσεις είναι συγκρουόμενες. Για παράδειγμα, ο διευθυντής μιας επιχείρησης, η οποία επιθυμεί να αποκτήσει ένα νέο πληροφοριακό σύστημα διαχείρισης πελατών έχει ως βασικό στόχο τη χαμηλή τιμή, ενώ ο διευθυντής μάρκετινγκ έχει ως βασική προτεραιότητα την πλούσια λειτουργικότητα και την ταχύτητα λειτουργίας.

Ο διευθυντής έργου είναι ο βασικός υπεύθυνος για την επιτυχή ολοκλήρωση ενός έργου. Βασικές υπευθυνότητες του διευθυντή έργου που σχετίζονται άμεσα με την επιτυχημένη ολοκλήρωση ενός έργου είναι:

- Να έχει ορίσει το αντικείμενο εργασιών του έργου σε επαρκή λεπτομέρεια ώστε να αποφεύγονται οι παρανοήσεις. Το αντικείμενο εργασιών πρέπει να είναι υπό έλεγχο και να υπάρχει καλά τεκμηριωμένη διαδικασία διαχείρισης αλλαγών. Συνηθίζεται να λέγεται ότι αν το αντικείμενο εργασιών του έργου

μπορεί να αλλάζει χωρίς περιορισμούς και έλεγχο, ο ρυθμός αλλαγής θα ξεπεράσει το ρυθμό προόδου.

- Να διοικεί αποτελεσματικά την ομάδα έργου.
- Να εξασφαλίσει ότι η απαραίτητη πληροφορία είναι διαθέσιμη στους συμμετέχοντες του έργου.
- Να εξασφαλίσει ότι οι δραστηριότητες του έργου εκτελούνται σύμφωνα με το χρονοδιάγραμμα.
- Να εξασφαλίσει ότι το έργο παραδίδεται σύμφωνα με τον προϋπολογισμό και χωρίς υπερβάσεις.
- Να εξασφαλίσει ότι το έργο παραδίδεται με την αναμενόμενη ποιότητα. Το κριτήριο αυτό είναι γενικό και αναλύεται στο να έχει το αποτέλεσμα του έργου τη σωστή λειτουργικότητα και με τις αναμενόμενες επιδόσεις.

Για να επιτύχει τα παραπάνω ο διευθυντής του έργου θα πρέπει να συγκεντρώνει αυξημένα προσόντα τα οποία συνοψίζονται στα εξής:

- Πρέπει να έχει ιδιαίτερες ικανότητες στην επικοινωνία, να είναι καλός ομιλητής, καλός ακροατής, καλός διαπραγματευτής, ικανός να πείθει, κ.λπ. Επισημαίνεται ότι, η επικοινωνία ως δραστηριότητα και ειδικά στα μεγάλα έργα, απαιτεί σημαντικό ποσοστό του συνολικού χρόνου εργασίας του διευθυντή έργου.
- Πρέπει να διαθέτει οργανωτικές ικανότητες, δηλαδή να μπορεί να σχεδιάζει, να μπορεί να θέτει γενικούς στόχους, ενώ ταυτόχρονα να μπορεί να αναλύει.
- Πρέπει να διαθέτει ικανότητα δημιουργίας ομάδας, εμπνυχώνοντας και κινητοποιώντας τα μέλη της ομάδας. Ο προβληματισμός αν πρέπει να είναι διαχειριστής ή ηγέτης είναι ένας από τους πιο πολυσυζητημένους. Ο ηγέτης είναι αυτός ο οποίος δημιουργεί ένα όραμα και πείθει τα μέλη της ομάδας για τη σπουδαιότητα και ορθότητα αυτού του οράματος, ενώ ο διαχειριστής είναι αυτός που με καλή οργάνωση και με καλό σχεδιασμό επιτυγχάνει τους στόχους.
- Πρέπει να διαθέτει αρετές, όπως προσαρμοστικότητα, δημιουργικότητα, προσήλωση στο στόχο, υπομονή και επιμονή.
- Πρέπει να διαθέτει τεχνικές ικανότητες, οι οποίες είναι απαραίτητες για το ρεαλιστικό και αξιόπιστο σχεδιασμό του έργου, τη σωστή λήψη αποφάσεων και την ορθή αξιολόγηση των αποτελεσμάτων. Οι τεχνικές ικανότητες έχουν

μεγάλη σημασία, ειδικά όταν το μέγεθος του έργου είναι μικρό, ενώ η βαρύτητά τους μειώνεται όταν μεγαλώνει το έργο, αφού σε αυτή την περίπτωση υπάρχει εξειδικευμένο προσωπικό.

### Άσκηση Αυτοαξιολόγησης 1.2

Ο παρακάτω πίνακας παρουσιάζει μια σειρά ικανοτήτων που απαιτείται να έχει ένας διαχειριστής έργων λογισμικού. Ποια κατά τη γνώμη σας είναι η κατάταξη των ικανοτήτων ανάλογα με τη σημασία τους; Χρησιμοποιήστε για την κάθε ικανότητα κατάλληλο ποσοστό που να αντιπροσωπεύει τη σπουδαιότητά της.

Ικανότητα	Κατάταξη (ποσοστό)
Ηγετικές ικανότητες (έχει όραμα, εμπνέει τους συνεργάτες, κ.ά.).	
Ικανότητα δημιουργίας ομάδας (έχει ομαδικό πνεύμα, έχει κίνητρα, κ.ά.)	
Ικανότητα επικοινωνίας (ξέρει να πείθει, είναι καλός ακροατής, έχει ικανότητα έκφρασης)	
Οργανωτικές ικανότητες (γνωρίζει να θέτει στόχους, να δίνει προτεραιότητες, να αναλύει προβλήματα, κ.ά.)	
Χαρακτήρας (είναι ευέλικτος, δημιουργικός, επίμονος, κ.ά.)	
Γνώση της τεχνολογίας (γνωρίζει να αναπτύσσει πληροφοριακά συστήματα, γνωρίζει να διοικεί έργα, έχει εμπειρία, κ.ά.).	

## ΕΝΟΤΗΤΑ 1.5 Γιατί τα Έργα Ανάπτυξης Λογισμικού Διαφέρουν;

Τα έργα ανάπτυξης λογισμικού έχουν αρκετές ομοιότητες με έργα άλλων ειδών, όπως για παράδειγμα τεχνικά έργα. Στις περισσότερες περιπτώσεις οι στόχοι, οι διαδικασίες, και τα εργαλεία που χρησιμοποιούνται είναι παρόμοια. Διαφοροποιείται όμως ο τρόπος εφαρμογής τους λόγω των διαφορών που προκύπτουν από τη φύση του λογισμικού. Όπως έγραψε και ο Brooks (Brooks, 1995) το λογισμικό έχει ορισμένα χαρακτηριστικά που το διαφοροποιούν από άλλα προϊόντα. Πιο συγκεκριμένα, το λογισμικό είναι (Bechtold 1999; Hughes 1999; Κιουντουζής, 1999):

- Άυλο: Μπορούμε πολύ εύκολα να δούμε την πρόοδο που επιτελείται στην κατασκευή μιας γέφυρας, ενώ κάτι τέτοιο είναι αρκετά πιο δύσκολο στην ανάπτυξη λογισμικού.
- Πολύπλοκο: Κατά μέσο όρο το λογισμικό είναι πιο πολύπλοκο από άλλα προϊόντα αντίστοιχης τιμής. Έτσι, ενώ υπάρχει ένα μέγιστο ύψος που μπορεί να φτάσει μια γέφυρα, δεν υπάρχει μέγιστος αριθμός γραμμών κώδικα που μπορεί να έχει ένα σύστημα λογισμικού. Εδώ θα πρέπει να σημειώσουμε ότι η πολυπλοκότητα αυξάνεται με μη γραμμικό τρόπο σε σχέση με το μέγεθος ενός συστήματος λογισμικού.
- Εύπλαστο: Το γεγονός ότι το λογισμικό είναι άυλο, σημαίνει ότι έχει τη δυνατότητα να αλλάζει εύκολα και γρήγορα. Ταυτόχρονα μοντελοποιεί τον τρόπο εργασίας των ατόμων γεγονός που οδηγεί σε πολλές και σύνθετες αλλαγές στο αντικείμενο των έργων. Έτσι, ενώ σε άλλα έργα η διαχείριση αλλαγών αποτελεί μια τετριμμένη διαδικασία, στα έργα ανάπτυξης λογισμικού αποτελεί βασική διαδικασία και πολλές φορές ενέχει κινδύνους.
- Διαθέσιμη τεχνολογία: Κανένας δε θα διαφωνήσει με το γεγονός ότι η τεχνολογία εξελίσσεται με ραγδαίους ρυθμούς έχοντας ως συνέπεια τη δυσκολία διαχείρισης των τεχνολογικών αλλαγών. Αποτελεί συνηθισμένο φαινόμενο στα έργα ανάπτυξης λογισμικού η ύπαρξη προβλημάτων ολοκλήρωσης μεταξύ διαφορετικών εκδόσεων του λογισμικού, η ταυτόχρονη συνύπαρξη εργαλείων διαφορετικών εκδόσεων, κ.λπ. Συνεπώς, η αλλαγή της τεχνολογίας αποτελεί για τα έργα ανάπτυξης λογισμικού ένα σταθερό παράγοντα αστάθειας που θα πρέπει να λαμβάνουμε πάντα σοβαρά υπόψη.

### **Δραστηριότητα 1.1**

Μελετήστε πώς διαφέρει η διαχείριση έργων λογισμικού από τη διαχείριση τεχνικών έργων. Συγκρίνετε την ανάπτυξη ενός λογισμικού οικονομικής διαχείρισης με το κτίσιμο μιας γέφυρας.

---

Δεν αποτελεί έκπληξη λοιπόν το γεγονός ότι ο πιο συνηθισμένος τρόπος ολοκλήρωσης ενός έργου ανάπτυξης λογισμικού είναι η «αποτυχία», δηλαδή η αποτυχία να παραδοθεί η απαιτούμενη λειτουργικότητα, με την απαιτούμενη ποιότητα, στον προκαθορισμένο χρόνο, με τον αρχικό προϋπολογισμό. Σε σχετικές μελέτες που έγιναν στις Ηνωμένες

Πολιτείες τα στοιχεία που παρουσιάζονται είναι συντριπτικά: Το 90% όλων των έργων ανάπτυξης λογισμικού αποτυγχάνει και από τα 100 έργα που ξεκινούν το 94% αποτελεί δεύτερη προσπάθεια ενός αποτυχημένου έργου, ενώ το 31% των έργων, σε σύνολο 175000 έργων, ακυρώνεται πριν τη φυσική του λήξη (Standish Group, 1995).

### **Δραστηριότητα 1.2**

Χρησιμοποιείστε το διαδίκτυο για να βρείτε παραδείγματα αποτυχημένων έργων πληροφορικής. Αναλύστε τους λόγους αποτυχίας.

## **ΕΝΟΤΗΤΑ 1.6 Σημαντικοί Παράγοντες που Επηρεάζουν τη Διαχείριση Έργων Λογισμικού.**

Όπως έχουμε ήδη αναφέρει η διαχείριση έργων λογισμικού αποτελεί γνωστό αντικείμενο που συγκεντρώνει την προσοχή μεγάλου αριθμού διεθνών οργανισμών, μεγάλων εταιρειών, συμβούλων επιχειρήσεων, κ.ά. Ακόμη μεγαλύτερη έμφαση δίνεται στην εφαρμογή μεθοδολογιών ανάπτυξης λογισμικού.

Η διαχείριση έργων λογισμικού θα πρέπει να λάβει υπόψη της και να συνδυάσει αρχές που προέρχονται από τις γενικές αρχές της διοίκησης και της διαχείρισης έργων, αλλά και της τεχνολογίας λογισμικού. Δεδομένου του μεγάλου αριθμού των διαφορετικών προσεγγίσεων θα γίνει μια προσπάθεια στην ενότητα αυτή να παρουσιασθούν οι σημαντικότεροι παράγοντες που επηρεάζουν τη διαχείριση έργων λογισμικού.

### **1.6.1 Οργανισμοί Διαχείρισης Έργων.**

Διεθνείς ή εθνικοί οργανισμοί με στόχο την ανάπτυξη της διαχείρισης έργων είναι οι ακόλουθοι:

- Η Διεθνής Ένωση Διαχείρισης Έργων –(International Project Management Association, IPMA - <http://www.ipma.ch>).
- Το Ινστιτούτο Διαχείρισης Έργων (Project Management Institute, PMI - <http://www.pmi.org>).
- Η Ένωση για τη Διαχείριση Έργων στο Ηνωμένο Βασίλειο –(The Association for Project Management - <http://www.apm.org.uk>).

### **1.6.2 Διεθνή Πρότυπα στη Διαχείριση Έργων.**



Τα σημαντικότερα διεθνή πρότυπα που σχετίζονται με τη διαχείριση έργων είναι τα εξής:

- Ο βασικός κορμός της γνώσης από το Ινστιτούτο Διαχείρισης Έργων –(Project Management Body Of Knowledge, PMBOK).
- Ο οδηγός ικανοτήτων από τη Διεθνή Ομοσπονδία Διαχείρισης Έργων με τίτλο IPMA Competence Baseline (ICB), 5η έκδοση, όπου περιγράφονται με λεπτομέρεια οι ικανότητες που απαιτούνται για τη διαχείριση ενός έργου.
- Το πρότυπο PRINCE2 (PRojects IN a Controlled Environment - <http://www.prince2.org.uk>). Αποτελεί το Βρετανικό πρότυπο για τη διαχείριση έργων. Η μεθοδολογία PRINCE αναπτύχθηκε το 1989 από το CCTA (Central Computer and Telecommunications Agency) ως το Βρετανικό πρότυπο για διαχείριση έργων πληροφορικής. Πολύ σύντομα όμως, λόγω της γρήγορης και μεγάλης αποδοχής της, άρχισε να χρησιμοποιείται και σε έργα διαφορετικών κατηγοριών. Το αποτέλεσμα ήταν να εκδοθεί το 1996 η νέα έκδοση του προτύπου (PRINCE2) που είναι πλέον γενικού σκοπού.
- Το πρότυπο V-Modell (<http://www.v-modell-xt.de>). Αποτελεί το Γερμανικό πρότυπο για τη διαχείριση έργων. Η τελευταία έκδοση ονομάζεται V-Model XT και εκδόθηκε το 2005. Η μεθοδολογία V-Modell εστιάζει σε έργα πληροφορικής και είναι αρκετά αναλυτική αφού παρουσιάζει, όχι μόνο το τι πρέπει να γίνει για τη διαχείριση του έργου, αλλά και τη μέθοδο με την οποία θα γίνει, πότε θα γίνει και από ποιον.
- Το μοντέλο Οργανωσιακής Ωριμότητας στη Διαχείριση Έργων (Organizational Project Management Maturity Model - OPM3) (<http://www.pmi.org>). Το OPM είναι ένα πρότυπο μοντέλο οργανωσιακής ωριμότητας για τη διαχείριση έργων που έχει αναπτυχθεί από το PMI και προσδιορίζει τις απαιτήσεις για την αξιολόγηση των ικανοτήτων ενός οργανισμού στη διαχείριση έργων (project management), προγραμμάτων (programme management) και χαρτοφυλακίων (portfolio management). Με τον τρόπο αυτό μπορούμε να αξιολογήσουμε την ωριμότητα των οργανισμών στη διαχείριση έργων. Το OPM ορίζει 586 καλές πρακτικές και ένα σύνολο 2109 ικανοτήτων.
- Το πρότυπο ISO 10006:2003 (Quality management systems - Guidelines for quality management in projects) για τη διαχείριση ποιότητας σε έργα

(<http://www.iso.org>). Το πρότυπο αυτό για τη διαχείριση έργων παρουσιάζει σημαντικές ομοιότητες με το PMBOK.

- Το πρότυπο BS6079-1:2002 Project Management (British Standards Institution, 2002). Αποτελεί το Βρετανικό πρότυπο για τη διαχείριση έργων, το οποίο προσδιορίζει έξι φάσεις: Τη φάση της σύλληψης (conception), τη φάση της μελέτης σκοπιμότητας (feasibility), τη φάση της υλοποίησης (implementation), τη φάση της λειτουργίας (operation), τη φάση του τερματισμού (termination) και τη φάση της αξιολόγησης (evaluation).

### 1.6.3 Διαδικασίες Ανάπτυξης Λογισμικού.

Μια **διαδικασία ανάπτυξης λογισμικού** (software development process) ορίζει τον τρόπο ανάπτυξης λογισμικού. Ο όρος είναι συνώνυμος του **κύκλου ζωής λογισμικού** (software life cycle). Η διαφορά μεταξύ μιας μεθοδολογίας διαχείρισης έργων και μιας διαδικασίας ανάπτυξης λογισμικού εντοπίζεται στο γεγονός ότι μια μεθοδολογία διαχείρισης έργου εστιάζει στο πώς να οργανώσουμε το έργο, την ομάδα έργου, το πώς να παρακολουθήσουμε τις δραστηριότητες και τον προϋπολογισμό του έργου, κ.ά., ενώ μια διαδικασία ανάπτυξης λογισμικού εστιάζει στο πώς να κατασκευάζουμε το τελικό προϊόν / το λογισμικό. Είναι βέβαιο ότι υπάρχει μεγάλος βαθμός επικάλυψης των μεθοδολογιών διαχείρισης έργων και των διαδικασιών ανάπτυξης λογισμικού. Οι πιο σημαντικές διαδικασίες ανάπτυξης λογισμικού είναι οι ακόλουθες:

- Το πρότυπο IEEE 1074 (<http://www.ieee.org>). Το πρότυπο αυτό αφορά τον κύκλο ζωής λογισμικού. Ορίζει ένα σύνολο δεκαεπτά διαδικασιών που είναι βασικές για την ανάπτυξη και συντήρηση λογισμικού, οι οποίες ομαδοποιούνται σε ομάδες διαδικασιών (process groups). Παραδείγματα ομάδων διαδικασιών είναι η διαχείριση έργων, οι διαδικασίες πριν την ανάπτυξη του συστήματος (pre-development processes), οι διαδικασίες ανάπτυξης και οι διαδικασίες μετά την ανάπτυξη του συστήματος (post-development processes).
- Το μοντέλο ωριμότητας ικανότητας (Capability Maturity Model, CMM), το οποίο αναπτύχθηκε από το Software Engineering Institute (SEI) με σκοπό την αξιολόγηση των διαδικασιών ενός οργανισμού (Chrissis, 2003). Το μοντέλο CMM ορίζει πέντε επίπεδα ωριμότητας που σχετίζονται με το βαθμό υλοποίησης από τον οργανισμό ενός αριθμού βασικών διαδικασιών (key

process areas). Η τρέχουσα έκδοση του μοντέλου είναι η 1.2, η οποία ανακοινώθηκε το 2006 (<http://www.sei.cmu.edu/cmmi/>). Τα πέντε επίπεδα που ορίζονται από το μοντέλο αυτό είναι:

- Επίπεδο 1 – Το αρχικό (initial) επίπεδο, όπου δεν υπάρχει τυποποίηση διαδικασιών.
  - Επίπεδο 2 – Το επαναλαμβανόμενο (repeatable) επίπεδο, όπου οι βασικές διαδικασίες διαχείρισης έργων έχουν ορισθεί και εφαρμόζονται με επαναληπτικό τρόπο.
  - Επίπεδο 3 – Το καθορισμένο επίπεδο (defined), όπου η ανάπτυξη και η διαχείριση λογισμικού είναι τυποποιημένη.
  - Επίπεδο 4 – Το διοικούμενο επίπεδο (managed), όπου γίνονται συστηματικές μετρήσεις, τόσο της απόδοσης των διαδικασιών, όσο και των παραδοτέων του έργου.
  - Επίπεδο 5 – Το βελτιστοποιημένο επίπεδο (optimised), όπου γίνεται βελτιστοποίηση των διαδικασιών με βάση τις μετρήσεις.
- Η ενοποιημένη προσέγγιση (Unified Process, UP) (Γερογιάννης, 2006; Kruchten, 2004), η οποία αποτελεί μια από τις πλέον διαδεδομένες διαδικασίες ανάπτυξης λογισμικού. Τα βασικά στάδια της UP είναι τα εξής:
    - Η σύλληψη (inception) είναι η πρώτη φάση της ενοποιημένης προσέγγισης, όπου παρουσιάζεται η αρχική ιδέα του συστήματος, τουλάχιστον μέχρι του σημείου που είναι αρκετά καλά θεμελιωμένη, έτσι ώστε να επιτρέψει την είσοδο στη φάση επεξεργασίας.
    - Η λεπτομερής επεξεργασία (elaboration) είναι η δεύτερη φάση, όπου περιγράφεται ο σκοπός του συστήματος, καθώς και η υψηλού επιπέδου αρχιτεκτονική του. Σε αυτή τη φάση προσδιορίζονται οι απαιτήσεις του συστήματος.
    - Η κατασκευή (construction) είναι η τρίτη φάση, όπου σχεδιάζεται και κατασκευάζεται το λογισμικό.
    - Η μετάβαση (transition) είναι η τέταρτη φάση της διαδικασίας, όπου το λογισμικό υπόκειται σε έλεγχο και τελικά παραδίδεται στους χρήστες. Η φάση της μετάβασης σηματοδοτεί την έναρξη της φάσης της συντήρησης λογισμικού και όχι το τέλος της διαδικασίας ανάπτυξης.

Εκτός από τις παραπάνω φάσεις η UP περιλαμβάνει ροές εργασιών (workflows). Μια ροή εργασίας είναι μια λογική ομαδοποίηση ενός συνόλου δραστηριοτήτων που λαμβάνουν χώρα κατά την ανάπτυξη ενός συστήματος και περιγράφει το ρόλο καθενός που συμμετέχει στις δραστηριότητες ενός έργου ανάπτυξης λογισμικού, τις δραστηριότητες αυτές καθαυτές, καθώς και τα παραγόμενα (artifacts) των δραστηριοτήτων. Παραδείγματα τυποποιημένων ροών εργασιών είναι η μοντελοποίηση διαδικασιών, η καταγραφή απαιτήσεων, η ανάλυση και σχεδίαση, η διαχείριση έργων, κ.ά.

- Το μοντέλο καταρράκτη (waterfall model), το οποίο αποτελεί ένα από τα παλαιότερα μοντέλα ανάπτυξης λογισμικού. Σύμφωνα με το μοντέλο αυτό η ανάπτυξη λογισμικού είναι μια σειριακή διαδικασία, η οποία περνά από τις φάσεις της ανάλυσης των απαιτήσεων, του σχεδιασμού, της υλοποίησης, του ελέγχου, της ολοκλήρωσης και της συντήρησης. Το μοντέλο αυτό είναι από τα παλαιότερα (Royce, 1970) και αν και είναι ευρέως γνωστό δε χρησιμοποιείται πλέον. Μάλιστα ο Royce το παρουσίασε ως αντιπαράδειγμα και πρότεινε ένα επαναληπτικό μοντέλο ανάπτυξης. Θα πρέπει να σημειωθεί ότι το μοντέλο καταρράκτη δεν αναφέρεται στη διαχείριση των έργων.
- Το μοντέλο σπείρας (spiral model), το οποίο ορίστηκε από τον Barry Boehm το 1988 (Boehm, 1988) είναι μια επαναληπτική μέθοδος ανάπτυξης λογισμικού.
- Η προσέγγιση του ακραίου προγραμματισμού (Extreme Programming, XP), η οποία είναι μια από τις πλέον διαδεδομένες μεθοδολογίες ανάπτυξης λογισμικού (Beck, 2004). Η XP είναι μια εύκαμπτη (agile) μεθοδολογία ανάπτυξης λογισμικού, η οποία ακολουθεί τη δημόσια διακήρυξη αρχών της εύκαμπτης ανάπτυξης λογισμικού (Agile Software Development Manifesto) που εκδόθηκε από μία ομάδα προγραμματιστών και συμβούλων επιχειρήσεων το 2001 (Cockburn, 2006; Anderson, 2003; Highsmith, 2002) (<http://agilemanifesto.org/>) και εστιάζει στην αναγνώριση του ανθρώπινου παράγοντα ως του πρωταρχικού παράγοντα επιτυχίας ενός έργου λογισμικού, καθώς και στη συνεχή έμφαση στην αποτελεσματικότητα, στην προσαρμοστικότητα και στη διαχείριση της αλλαγής (<http://c2.com/cgi/wiki?ExtremeProgramming>). Τα βασικά χαρακτηριστικά της XP είναι:

- Ο περιορισμός του αριθμού των συμμετεχόντων σε κάθε ομάδα ανάπτυξης.
  - Οι μικροί και διακριτοί κύκλοι ανάπτυξης με σαφώς καθορισμένα και απόλυτα λειτουργικά παραγόμενα.
  - Ο διαρκής έλεγχος του λογισμικού.
  - Η μεγαλύτερη δυνατή τμηματοποίηση του έργου σε επίπεδο διαδικασίας ανάπτυξης.
  - Η αφαίρεση όλων των περιττών ενεργειών από ης διαδικασίας ανάπτυξης.
  - Η ανάπτυξη λογισμικού με βάση τη συνεργασία και την επικοινωνία.
  - Η ανάπτυξη μιας πρώτης έκδοσης σε διάστημα μερικών εβδομάδων για τη λήψη άμεσης και γρήγορης ανατροφοδότησης.
  - Η εύρεση και χρήση των απλούστερων λύσεων.
  - Η μείωση της απαραίτητης τεκμηρίωσης του έργου.
  - Η συνεχής επαφή, διαθεσιμότητα και συμμετοχή του πελάτη.
- Η μέθοδος Scrum, η οποία είναι μια εύκαμπτη μέθοδος που δίνει έμφαση στη διαχείριση των έργων ανάπτυξης λογισμικού (Takeuchi, 1986; Beedle 2002). Η μέθοδος αυτή θεωρεί ότι η ανάπτυξη λογισμικού είναι μια σύνθετη και απρόβλεπτη διαδικασία και για το λόγο αυτό την προσομοιώνει με ένα «μαύρο κουτί» και δεν τη θεωρεί ως μια σαφώς ορισμένη διαδικασία. Αντίθετα με τα παραδοσιακά μοντέλα που θεωρούν την ανάπτυξη λογισμικού σαν μια γραμμική διαδικασία, η μέθοδος Scrum θεωρεί ότι η ανάπτυξη μπορεί να ξεκινήσει από οποιαδήποτε φάση και με οποιαδήποτε σειρά. Έτσι, σύμφωνα με την προσέγγιση Scrum, η μέθοδος προσαρμόζεται στο έργο και όχι το αντίστροφο. Άλλα βασικά χαρακτηριστικά της προσέγγισης είναι οι μικρές ομάδες, τα ευέλικτα χρονοδιαγράμματα, τα συχνά σημεία ελέγχου, η καλή συνεργασία κ.λπ. (<http://www.scrumalliance.org/>).

## ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

Στο κεφάλαιο αυτό παρουσιάσαμε τις βασικές έννοιες της διαχείρισης έργων, τι είναι έργο, τι είναι διαχείριση έργων, καθώς και ποια είναι τα βασικά στάδια εκτέλεσης ενός έργου. Στη συνέχεια παρουσιάσαμε το βασικό κορμό της γνώσης ο οποίος αποτελείται από εννέα γνωστικές περιοχές οι οποίες είναι:

- Διαχείριση ενοποίησης έργου (project integration management).
- Διαχείριση αντικειμένου εργασιών έργου (project scope management).
- Διαχείριση χρόνου έργου (project time management).
- Διαχείριση κόστους έργου (project cost management).
- Διαχείριση ποιότητας έργου (project quality management).
- Διαχείριση ανθρωπίνων πόρων έργου (project human resource management).
- Διαχείριση επικοινωνίας έργου (project communication management).
- Διαχείριση κινδύνου έργου (project risk management)
- Διαχείριση προμηθειών έργου (project procurement management).

Ακολουθώντας, δώσαμε εν συντομία τις φάσεις διαχείρισης ενός έργου, ενώ ορίσαμε την έννοια των συμμετεχόντων σε ένα έργο.

Εξηγήσαμε, γιατί τα έργα ανάπτυξης λογισμικού διαφέρουν από άλλα έργα και τέλος παρουσιάσαμε τα πιο γνωστά πρότυπα που χρησιμοποιούνται για τη διαχείριση έργων λογισμικού, καθώς τους πιο γνωστούς κύκλους ζωής λογισμικού.

## **BIBΛΙΟΓΡΑΦΙΑ**

---

### Γενική βιβλιογραφία στη διαχείριση έργων

Brooks, F. P., (1995). No Silver Bullet: Essence and Accidents of Software Engineering. Περιλαμβάνεται στο βιβλίο: The Mythical Man Month, Anniversary Edition. Addison-Wesley.

Charette, R., (2005). Why Software Fails. IEEE Spectrum, Σεπτέμβριος 2005.

Kerzner, H., (2003). Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 8th Edition. John Wiley & Sons.

Kouzes, J. M. και B. Z. Pozner (1987). The leadership challenge. Jossey-Bass, San Francisco, CA.

Maylor, H., (2006). Διαχείριση Έργων. Εκδόσεις Κλειδάριθμος.

Meredith, J., Mantel, S., Mantel, Jr. S., (2005). Project Management: A Managerial Approach, 5th edition. John Wiley & Sons.

The Standish Group. (1995). Charting the Seas of Information Technology—Chaos. West Yarmouth, MA: The Standish Group International.

### Διαχείριση έργων πληροφορικής

Anderson, J., Schragenheim, E., (2003). Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. Prentice Hall.

Bechtold, P., (1999). Essentials of Software Project Management. Management Concepts.

Hughes, B., Cotterell, M., (1999). Software Project Management. McGraw Hill.

Rosen, A., (2004). Effective IT Project Management: Using Teams to Get Projects Completed on Time and Under Budget. AMACOM.

Κιουντουζής, Ε., (1999). Διαχείριση Έργων Πληροφορικής. Εκδόσεις Σταμούλη.

### Μοντέλα κύκλων ζωής λογισμικού

Chrissis, M. B., Konrad, M., και Shrum, S., (2003). CMMI: Guidelines for Process Integration and Product Improvement. Addison-Wesley Professional.

Γερογιάννης, Β., Κακαρόντζας, Γ., Καμέας, Α., Σταμέλος, Γ., Φιτσιλής, Π., (2006). Αντικειμενοστρεφής Ανάπτυξη Λογισμικού με τη UML. Εκδόσεις Κλειδάριθμος.

Kruchten, P., (2004). The Rational Unified Process: An Introduction, 3rd Edition. Εκδότης?

Royce, W., (1970). Managing the Development of Large Software Systems. Proceedings of IEEE WESCON.

Boehm, B., (1988). A Spiral Model of Software Development and Enhancement. IEEE Computer, Vol. 21, No. 5, pp. 61-72.

Cockburn, A., (2006). Agile Software Development: The Cooperative Game, 2nd edition. Addison-Wesley.

Beck, K., (2004). Extreme Programming Explained: Embrace Change, 2nd edition. Addison Wesley..

Takeuchi, H., Nonaka. I., (1986). The New Product Development Game. Harvard Business Review.

Beedle, M., Schwaber, K., (2002). Agile Software Development with SCRUM. Prentice Hall.

Highsmith, J., (2002). Agile Software Development Ecosystems, Addison Wesley..

Διεθνή πρότυπα

IEEE 1490-2003, IEEE Guide Adoption of PMI Standard. A Guide to the Project Management Body of Knowledge. IEEE.

BS6079-1:2002, Project Management, Part 1: Guide to Project Management. British Standards Institution.

PMI Institute, (2004). A Guide to the Project Management Body of Knowledge, PMI Standard Committee.

International Project Management Association, (2006). IPMA Competence Baseline, Version 3.0. Van Haren Publishing.



## **ΚΕΦΑΛΑΙΟ 2. ΟΡΓΑΝΩΤΙΚΕΣ ΔΟΜΕΣ – ΚΥΚΛΟΣ ΖΩΗΣ ΛΟΓΙΣΜΙΚΟΥ**

**Συγγραφέας Ι. ΣΤΑΜΕΛΟΣ**

Στο κεφάλαιο αυτό θα μάθετε:

- Τι εννοούμε με τον όρο οργανωτική δομή
- Ποιες οργανωτικές δομές ενδιαφέρουν τη διαχείριση έργου λογισμικού
- Πως συγκρίνονται μεταξύ τους οι οργανωτικές δομές
- Τι είναι οργάνωση ομάδας
- Ποιοι είναι οι ρόλοι των μελών μιας ομάδας στην ανάπτυξη λογισμικού
- Με ποια κριτήρια επιλέγουμε τα μέλη μίας ομάδας και τους αναθέτουμε ρόλους
- Πως επηρεάζει η προσωπικότητα των μελών τη σύνθεση της ομάδας

### ΠΕΡΙΕΧΟΜΕΝΑ

- 2.1 Εισαγωγή στις οργανωτικές δομές
- 2.2 Οργανωτικές δομές έργων
- 2.3 Οργάνωση ομάδας
- 2.4 Σύνοψη – Βιβλιογραφία
- 2.5 Απαντήσεις Ασκήσεων Αυτοαξιολόγησης – Δραστηριοτήτων

### **2.1 Εισαγωγή στις οργανωτικές δομές**

---

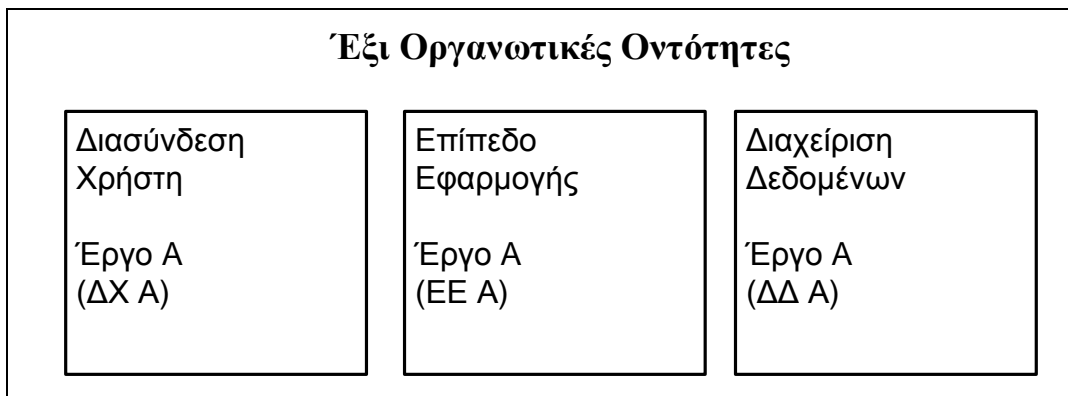
Η επιτυχής εκτέλεση των έργων ενός οργανισμού παραγωγής λογισμικού, δεδομένων των δυσκολιών που αυτά παρουσιάζουν, απαιτεί μία καλή και

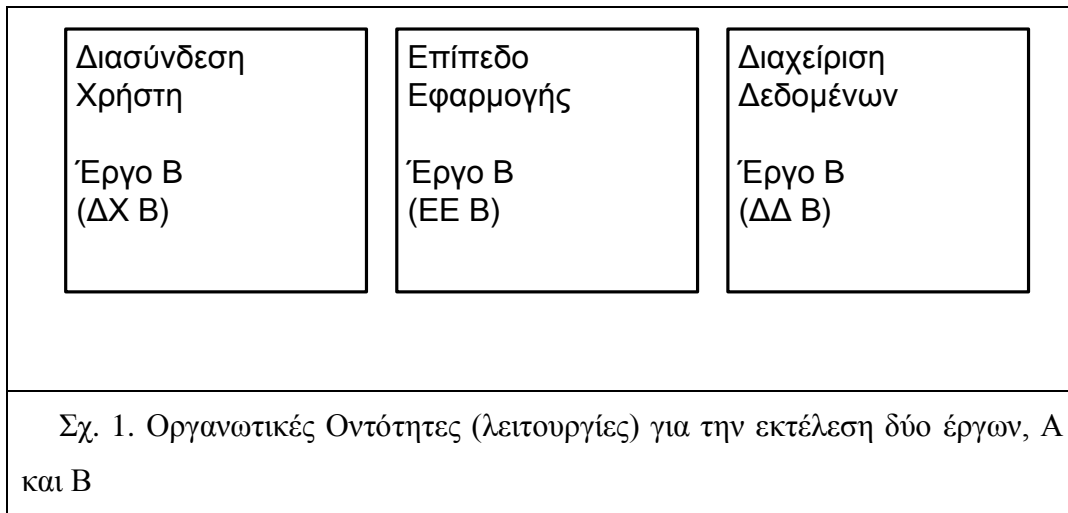
συστηματική οργάνωση. Πρέπει λοιπόν να ορισθούν κάποιες οργανωτικές δομές πάνω στις οποίες θα στηριχθούν και θα αναπτυχθούν τα έργα αυτά. Στο κεφάλαιο αυτό θα ασχοληθούμε με δύο οργανωτικές δομές, την *οργάνωση έργων (project organisation)*, την *οργάνωση ομάδων (team building)*.

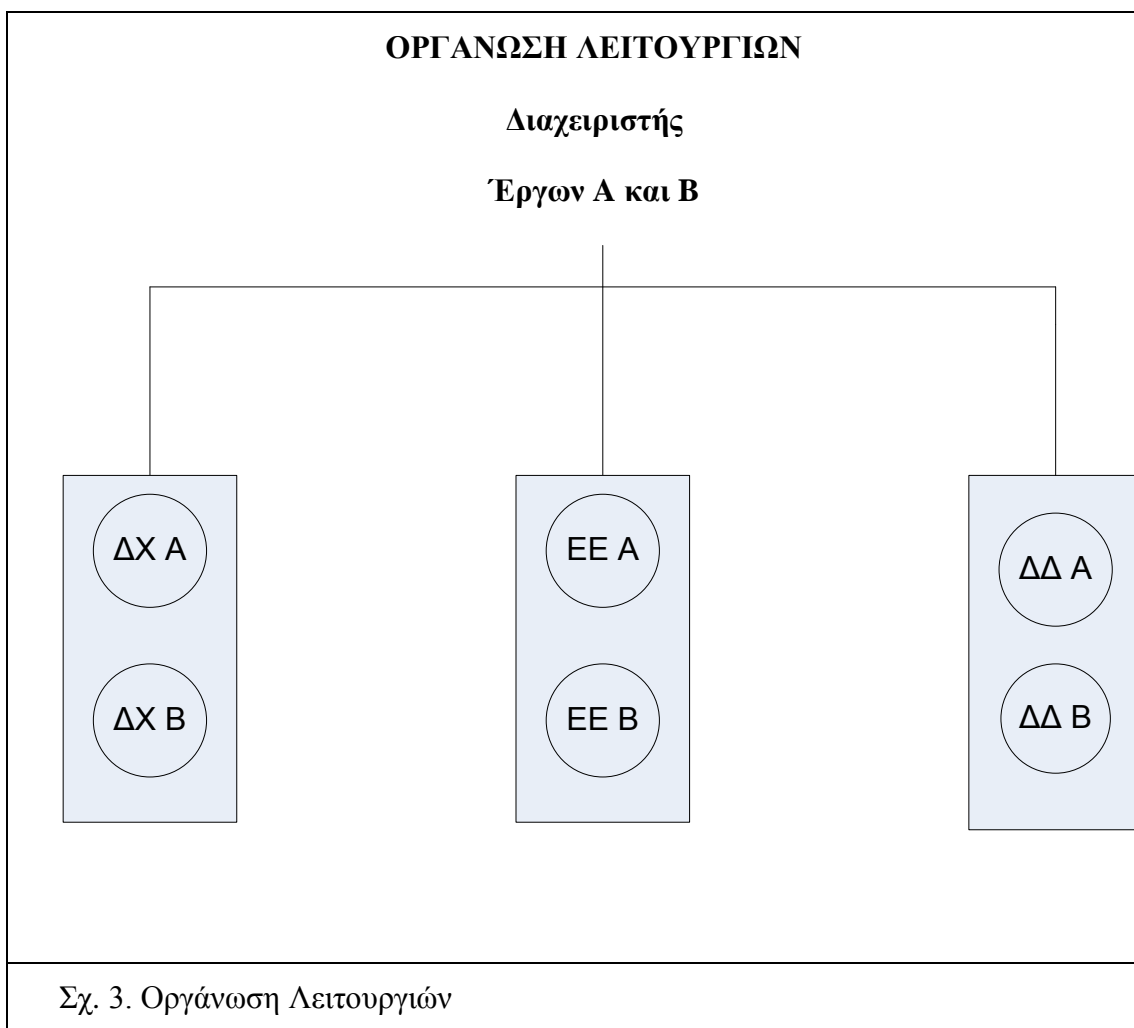
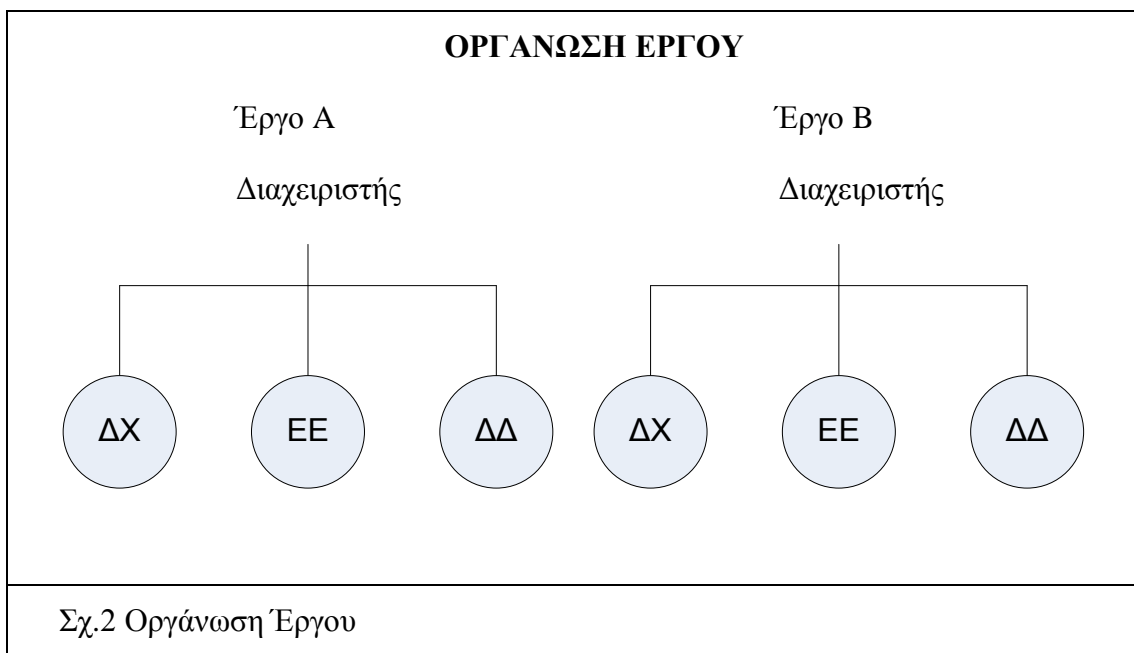
## 2.2 Οργανωτικές δομές έργων

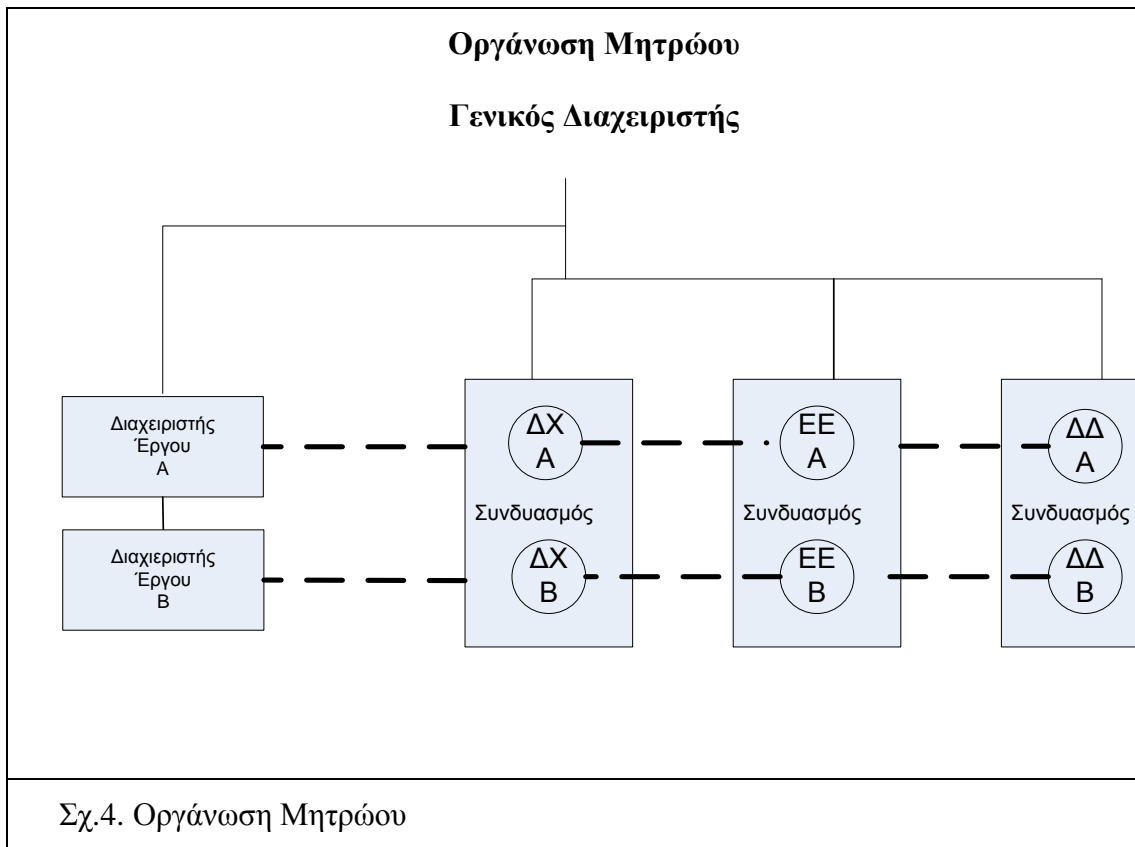
Ας υποθέσουμε ότι ο οργανισμός παραγωγής λογισμικού (π.χ. μία εταιρεία παραγωγής λογισμικού, μία επιχείρηση που αναπτύσσει λογισμικό για τις ανάγκες της κλπ) σκοπεύει να αναπτύξει δύο έργα τα Α και Β. Για κάθε ένα από τα δύο έργα πρέπει να αναπτυχθούν τρεις διαστρωματώσεις: διασύνδεση χρήστη (ΔΧ), επίπεδο εφαρμογής (ΕΕ) και διαχείριση δεδομένων (ΔΔ). Οι απαραίτητες ομάδες για την ανάπτυξη αυτών των έξι λειτουργιών φαίνονται στο σχ. 1.

Ο τρόπος με τον οποίο θα συνδυασθούν οι έξι αυτές ομάδες μπορεί να οδηγήσει σε μία *δομή οργάνωσης έργου (project organization structure, σχ. 2)*, σε μία *δομή οργάνωσης λειτουργιών (functional organization structure, σχ. 3)* ή σε μία *δομή οργάνωσης μητρώου (matrix organization structure, σχ. 4)*. Ανάλογα με τη δομή που θα επιλέξουμε θα έχουμε έναν, δύο ή τρεις διαχειριστές έργου









Είναι προφανές ότι και οι τρεις δομές θα μπορέσουν να αναπτύξουν τα δύο έργα. Το βασικό ερώτημα είναι με ποια κριτήρια και με ποια λογική επιλέγουμε την καταλληλότερη οργανωτική δομή κάθε φορά. Τα σημαντικότερα κριτήρια είναι:

- Μέγεθος του προς ανάπτυξη λογισμικού. Είναι ο κυριότερος παράγοντας που θα επηρεάσει το μέγεθος της ομάδας έργου, δηλ. τον αριθμό των μηχανικών και προγραμματιστών
- Αριθμός των έργων προς υλοποίηση. Π.χ. μπορεί να πρέπει να υλοποιήσουμε λίγα, σχετικά μεγάλα έργα (κάθε ένα με άνω των 20-30 μηχανικών συνολικά) ή πολλά μικρά έργα (κάθε ένα με κάτω από 10 μηχανικούς συνολικά)
- Έκταση της ανάπτυξης, δηλ το σύνολο των διαφορετικών δραστηριοτήτων που πρέπει να αναπτυχθούν. Π.χ. θα χρειασθεί ο χρονοπρογραμματισμός νέων έργων ή η συντήρηση λογισμικού σε κάποια φάση;

- Περιβάλλον ανάπτυξης, δηλ. σε ποιο βαθμό ελέγχονται οι απαιτούμενοι πόροι από τον οργανισμό ανάπτυξης λογισμικού και σε ποιο βαθμό θα χρειασθεί συνδρομή από άλλους, συνεργαζόμενους οργανισμούς (π.χ. για την προμήθεια και ανάπτυξη συστατικών λογισμικού, υπηρεσίες όπως δοκιμή λογισμικού)
- Φυσικοί Περιορισμοί. Τα έργα θα αναπτυχθούν στο ίδιο γεωγραφικό σημείο ή σε διαφορετικά, ίσως και σε διαφορετικές χώρες/ηπείρους (κατανεμημένη ανάπτυξη)
- Οργανωτική κουλτούρα. Τι είδους διαχειριστές έργου έχει στη διάθεσή του ο οργανισμός ανάπτυξης; Σε τι είδους οργανωτικές δομές έχει εμπειρία και προτίμηση ο οργανισμός;

Στη συνέχεια, εξετάζουμε τις τρεις διαθέσιμες δομές ως προς τα πλεονεκτήματα και μειονεκτήματά τους, με βάση τα παραπάνω κριτήρια. Η ανάλυση αυτή θα μας βοηθήσει να κατανοήσουμε τι πρέπει να αποφεύγεται και τι πρέπει να επιδιώκεται κατά τη διαχείριση ενός έργου ή μίας ομάδας έργων.

### *2.2.1 Οργάνωση Έργου*

Ο τύπος αυτός οργάνωσης είναι κατάλληλος για οργανισμούς που έχουν πολλά μικρά έργα να αναπτύξουν και μάλιστα σε ξεχωριστά γεωγραφικά σημεία. Η οργάνωση έργου απαιτεί τουλάχιστον το 70% των απαιτούμενων πόρων να είναι κάτω από τον άμεσο έλεγχο του διαχειριστή του έργου (δηλ. να μην είναι πόροι που θα προσφερθούν από εξωτερικούς οργανισμούς ή από άλλους διαχειριστές έργων). Προϋποθέτει δηλ. μια σημαντική ανεξαρτησία στην εκτέλεση του έργου. Ο διαχειριστής του έργου επιτελεί τόσο το ρόλο του τεχνικού υπεύθυνου όσο και το ρόλο του διαχειριστικού υπεύθυνου του έργου.

Πλεονεκτήματα της οργάνωσης έργου είναι τα ακόλουθα:

- Τεχνικές και διαχειριστικές αποφάσεις λαμβάνονται στο κατώτερο δυνατό επίπεδο διοίκησης του οργανισμού με αποτέλεσμα την ταχύτητα στη λήψη των

αποφάσεων και στον άμεσο έλεγχο (χωρίς να παρεμβάλλονται πολλοί διαχειριστές)

- Εφόσον το έργο είναι υπό τον έλεγχο ενός διαχειριστή, ελαχιστοποιείται η ανάγκη για επικοινωνία με άλλους διαχειριστές ή μέλη ομάδων και είναι δυνατή η σαφής ανάθεση ευθυνών (π.χ. σε περίπτωση που διαπιστωθεί λάθος ενέργεια).
- Ευνοείται η αξιοποίηση μηχανικών που έχουν γενικές γνώσεις και διαχειριστών που δεν έχουν συγκεκριμένες τεχνικές δεξιότητες, καθώς είναι σχετικά εύκολο να τους αποδοθεί ένας γενικός ρόλος.
- Το κίνητρο είναι μεγάλο για τους συμμετέχοντες στα έργα, επειδή ουσιαστικά ταυτίζονται με αυτά και εισπράττουν προσωπικά τις επιτυχίες του έργου.

Μειονεκτήματα της οργάνωσης έργου είναι τα ακόλουθα:

- Τα έργα πρέπει να είναι και να παραμείνουν μικρά, δεν είναι εύκολη η διεύρυνσή τους (π.χ. δεν είναι εύκολη η επέκτασή τους αν είναι πιλοτικής φύσης).
- Η υψηλότερου επιπέδου διοίκηση χάνει την επαφή της με τα έργα, οπότε δεν είναι εύκολη η πληροφόρησή της για την πορεία τους
- Δε διευκολύνεται η βέλτιστη χρήση των πόρων του οργανισμού. Για παράδειγμα δεν είναι εύκολο ένας ειδικός σε διασυνδέσεις χρήστη να απασχοληθεί σε δύο διαφορετικά έργα.
- Το κόστος της εκπαίδευσης του προσωπικού είναι υψηλό, αφού χρειάζονται ειδικευμένοι τεχνολόγοι σε κάθε ξεχωριστό έργο. Συχνά απαιτείται εκπαίδευση και εξειδίκευση που χρησιμεύει για μικρό μόνο χρονικό διάστημα.
- Η μετακίνηση καλών προγραμματιστών μεταξύ διαφορετικών έργων είναι δύσκολη.

- Το υψηλό προσωπικό κίνητρο για συμμετοχή σε ένα έργο που προαναφέρθηκε μπορεί να αποτελέσει πρόβλημα όταν το έργο λήξει και δεν υπάρχει αντίστοιχη, ελκυστική θέση σε άλλο έργο.

- Η απομόνωση των έργων δε βοηθά στην απόκτηση κοινής γνώσης και εμπειρίας στον οργανισμό και δεν υποστηρίζει την ανάπτυξη προτύπων και μεθοδολογιών λογισμικού από τον οργανισμό

### 2.2.2 Οργάνωση Λειτουργιών

Πρόκειται για την παλαιότερη μορφή οργάνωσης η οποία δε χρησιμοποιείται από οργανισμούς μέτριου ή μεγάλου μεγέθους. Μερικές μόνον ιδέες της οργάνωσης λειτουργιών θα μπορούσαν να χρησιμεύσουν στην οργάνωση μητρώου που περιγράφεται παρακάτω. Το βασικό πρόβλημα είναι ότι όλες οι αποφάσεις, τεχνικές και διαχειριστικές, λαμβάνονται από έναν και μόνο διαχειριστή έργου. Δύσκολα ένας διαχειριστής μπορεί να ανταπεξέλθει σε αυτές τις συνθήκες, και συχνά εντοπίζεται στο πρόσωπό του η αιτία για υπερβάσεις στον προϋπολογισμό και στα χρονικά όρια, και σε κακή ποιότητα των έργων. Η εμπειρία έχει δείξει ότι συχνά ο διαχειριστής αναγκάζεται να ασχολείται με καθημερινά προβλήματα και όχι με την υψηλού επιπέδου διαχείριση των έργων.

Πλεονεκτήματα της οργάνωσης λειτουργιών είναι τα ακόλουθα:

- Σε περίπτωση που ο διαχειριστής μπορεί να ανταπεξέλθει στις ανάγκες της οργάνωσης αυτής έχουμε έναν πραγματικά πολύ ισχυρό και αποτελεσματικό έλεγχο
- Η οργάνωση βοηθά στην απόκτηση κοινής γνώσης και εμπειρίας στον οργανισμό καθώς υπάρχει στενή συνεργασία όλων των εξειδικευμένων τεχνολόγων. Είναι εύκολη η ανάπτυξη προτύπων και μεθοδολογιών λογισμικού από τον οργανισμό.



- Το προσωπικό ταυτίζεται με τη λειτουργία στην οποία ασχολείται και εξειδικεύεται. Όταν το έργο λήξει δεν υπάρχει απώλεια κινήτρου, όπως αντιθέτως συζητήθηκε στην ανάλυση της οργάνωσης.
- Ο οργανισμός προσαρμόζεται εύκολα σε μακροπρόθεσμες προκλήσεις, όπως η παρακολούθηση νέων τεχνολογιών και η εκπαίδευση και διακράτηση εξειδικευμένου προσωπικού. Γενικά η δομή αυτή επικεντρώνεται στο άτομο και όχι στο έργο.

Μειονεκτήματα της οργάνωσης λειτουργιών είναι τα ακόλουθα:

- Οποιοδήποτε πρόβλημα διασύνδεσης και επικοινωνίας με το μοναδικό διαχειριστή επιδρά στην πορεία των έργων και πρέπει να επιλυθεί οπωσδήποτε από το διαχειριστή.
- Δεν είναι εύκολη η εκπαίδευση τεχνολόγων με γενικές γνώσεις, οι οποίοι θα μπορούσαν να αναλάβουν πολλαπλούς ρόλους σε ένα έργο ή να αναλάβουν διαφορετικούς ρόλους σε διαφορετικά έργα. Μειώνεται δηλ. η ευελιξία του οργανισμού, ο οποίος διαθέτει μόνον υπερ-εξειδικευμένο προσωπικό στις τάξεις του.
- Αποτυχία του διαχειριστή στο ρόλο του σημαίνει προβλήματα στον προϋπολογισμό, το χρονοδιάγραμμα και στην ποιότητα των έργων

### *2.2.3 Οργάνωση Μητρώου*

Η οργάνωση έργου και η οργάνωση λειτουργιών είναι οργανωτικές δομές που προσπαθούν να βελτιστοποιήσουν ένα μόνο στόχο, τα έργα ή τις λειτουργίες του οργανισμού. Οι δομές της οργάνωσης έργου επικεντρώνονται αναγκαστικά σε σχετικά βραχυπρόθεσμους στόχους, όπως είναι ο προϋπολογισμός, το χρονοδιάγραμμα και το κόστος των έργων. Αντίθετα, η οργάνωση λειτουργιών στρέφει την προσοχή της σε σχετικά μακροπρόθεσμους στόχους, όπως η διαμοίραση γνώσης μεταξύ των έργων, η εξέλιξη της τεχνογνωσίας του οργανισμού, βελτιωμένα

και αποτελεσματικά πρότυπα διαχείρισης, ανάπτυξης και ποιότητας σε επίπεδο οργανισμού και ανάπτυξη στο έπακρο των τεχνολογικών δεξιοτήτων του προσωπικού.

Σε αντίθεση, η οργάνωση μητρώου (σχ. 4) επιχειρεί να βελτιστοποιήσει περισσότερους από ένα οργανωσιακούς στόχους ταυτόχρονα. Μία οργάνωση μητρώου για παράδειγμα, θα μπορούσε να έχει σα στόχο τη βελτιστοποίηση όχι μόνον τα έργα και τις λειτουργίες, αλλά και τη γεωγραφική κατανομή του οργανισμού και τη γενικότερη αναπτυξιακή στρατηγική του. Επομένως, μέσα από μία σχετικά πολύπλοκη οργανωτική δομή, επιχειρείται να ικανοποιηθούν και οι βραχυπρόθεσμοι και οι μακροπρόθεσμοι στόχοι.

Πιθανώς το μεγαλύτερο πρόβλημα στη δομή μητρώου είναι ότι δεν υπάρχει ατομική ευθύνη για αποτυχία ενός έργου. Στην πράξη η επιτυχία ή αποτυχία διαμοιράζεται στον διαχειριστή του έργου και στο διαχειριστή της λειτουργίας.

Στη δομή μητρώου, ο τεχνικός διαχειριστής (υπεύθυνος κάθε λειτουργίας σε επίπεδο οργανισμού) αποφασίζει τον τρόπο με τον οποίο θα αναπτυχθεί το λογισμικό και είναι υπεύθυνος για τη διάθεση των απαραίτητων πόρων για αυτό το σκοπό. Ο διαχειριστής έργου αποφασίζει τι ακριβώς θα αναπτυχθεί. Με άλλα λόγια ο διαχειριστής αποφασίζει το περιεχόμενο της διασύνδεσης χρήστη ανάλογα με τις απαιτήσεις του έργου (ποιους χρήστες θα εξυπηρετεί, τι υπηρεσίες θα τους παρέχει, τι μορφή θα έχει) και ο ειδικός μηχανικός αποφασίζει την τεχνολογία που θα χρησιμοποιηθεί (π.χ. ποια βιβλιοθήκη γραφικών, το περιβάλλον και τα εργαλεία ανάπτυξης κλπ). Η οικονομική διαχείριση γίνεται από το διαχειριστή έργου. Ο γενικός διαχειριστής έχει την ευθύνη της διοίκησης της όλης δομής αλλά και της άμεσης διοίκησης των λειτουργιών. Για κάθε κόμβο του μητρώου, δηλ. για κάθε λειτουργία έργου, υπάρχει ένας τεχνικός υπεύθυνος (*technical manager*). Από το σχ. 4 φαίνεται ότι παράμετροι του μητρώου είναι ο αριθμός των γραμμών (έργα) και ο αριθμός των στηλών (λειτουργίες), ενώ άλλες παράμετροι είναι το μέγεθος κάθε έργου, το ποσοστό των έργων του οργανισμού τα οποία εκτελούνται από τη δομή μητρώου, ο αριθμός των εμπλεκόμενων σε κάθε έργο κλπ.

Ένα άλλο σημαντικό πρόβλημα είναι ότι δεν είναι εύκολο σε όλους τους διαχειριστές να συμμετάσχουν σε μία δομή μητρώου. Ο λόγος είναι ότι συνήθως ο διαχειριστής συνηθίζει να εργάζεται μόνος του και να μη μοιράζεται την ευθύνη ενός έργου με άλλους συναδέλφους του. Η δομή μητρώου δεν είναι κατάλληλη για διαχειριστές με γραφειοκρατική νοοτροπία (απλή εφαρμογή κανόνων) και διαχειριστές με αυστηρή νοοτροπία (οι οποίοι δεν έχουν συνηθίσει να συζητούν τις αποφάσεις τους).

Σε περίπτωση διαφωνίας μεταξύ υπεύθυνου τεχνικού και διαχειριστή, η δομή μητρώου προϋποθέτει την εύρεση μίας συναινετικής λύσης. Αυτό απαιτεί την ύπαρξη αμοιβαίας εμπιστοσύνης, την παρουσίαση και αλληλοενημέρωση για όλα τα σχετικά με τη διαφωνία στοιχεία, και τελικά την επίτευξη μίας συμφωνίας για τον τρόπο συνέχισης του έργου. Σε περίπτωση μη επίτευξης συμφωνίας, η τελική απόφαση λαμβάνεται από το γενικό διευθυντή. Πρόκειται όμως προφανώς για μία αρνητική εξέλιξη για τη δομή μητρώου.

Ένα άλλο πρόβλημα μπορεί να θεωρηθεί το γεγονός ότι οι συμμετέχοντες μηχανικοί στη δομή μητρώου έχουν δύο προϊσταμένους, πράγμα που μπορεί να δημιουργήσει σύγχυση υπό προϋποθέσεις. Θεωρείται όμως ότι αν η καθημερινή λειτουργία της δομής μητρώου είναι ομαλή, αυτό το χαρακτηριστικό μπορεί να μετατραπεί σε πλεονέκτημα καθώς αποφεύγεται η τριβή που μπορεί να προκαλείται σε έναν υπάλληλο όταν υπάρχει μόνον ένας προϊστάμενος.

Από τη στιγμή που ένας οργανισμός θα αποφασίσει την εφαρμογή μίας δομής μητρώου, θα πρέπει να καθορισθούν με λεπτομέρεια οι ευθύνες και οι αρμοδιότητες τόσο της διαχειριστικής όσο και της λειτουργικής πλευράς του μητρώου. Αυτό γίνεται συχνά, παραγνωρίζεται όμως η ανάγκη για διαφορετικό σχεδιασμό των έργων όταν αυτά πρόκειται να αναπτυχθούν με τη μέθοδο του μητρώου. Η δομή αυτή έχει από μόνη της τη δυνατότητα να διαχειρισθεί αποτελεσματικά την ανάπτυξη ενός έργου σε διαφορετικά γεωγραφικά σημεία ή από περισσότερες από μία συνεργαζόμενες εταιρείες. Προτιμάται η ανάθεση εργασίας σε εσωτερικά και εξωτερικά τμήματα του οργανισμού από την ανάθεση εργασίας σε κατώτερα τμήματα της δομής του οργανισμού. Για παράδειγμα, η ανάπτυξη των υποσυστημάτων ενός

μεγάλου συστήματος λογισμικού θα είναι καλύτερο να γίνει σε μεγάλο βαθμό χρησιμοποιώντας συνεργαζόμενες ομάδες στο εσωτερικό του οργανισμού και εξωτερικούς συνεργάτες. Ο κατακερματισμός του έργου σε μικρότερα έργα είναι επομένως μία πολύ βασική δραστηριότητα στη δομή μητρώου.

Ο υπεύθυνος κάθε λειτουργίας βρίσκεται σε μια θέση στην οποία πρέπει να αναπτύξει πολλά, μικρης έκτασης, υποέργα, για την εκτέλεση των οποίων θα πρέπει πιθανώς να συνεργασθεί με άλλες ομάδες, εξειδικευμένες στη συγκεκριμένη λειτουργία. Το κρίσιμο σημείο βρίσκεται στην πολύ καλή προδιαγραφή των απαιτήσεων των υποέργων αυτών για να γίνει δυνατή τόσο η υλοποίησή τους όσο και η διαχείριση της ανάπτυξης στο εξωτερικό του οργανισμού. Στη συνέχεια συνοψίζονται τα πλεονεκτήματα και μειονεκτήματα της δομής μητρώου.

Πλεονεκτήματα της οργάνωσης μητρώου είναι τα ακόλουθα:

- Η οργάνωση μητρώου προσπαθεί να βελτιστοποιήσει ταυτόχρονα πολλούς οργανωσιακούς στόχους: τα έργα, τις λειτουργίες, τη γεωγραφική διασπορά και τη στρατηγική του οργανισμού
- Η λειτουργική διάσταση της δομής μητρώου επιτρέπει τη διαμοίραση γνώσης μεταξύ των έργων, την παρακολούθηση των τεχνολογικών εξελίξεων, την ανάπτυξη προτύπων σε επίπεδο οργανισμού και τη διαμοίραση κρίσιμων πόρων (εξειδικευμένων μηχανικών) μεταξύ των έργων
- Η δομή μητρώου επιτρέπει στους μηχανικούς να ωριμάσουν είτε με την εξειδίκευση σε μία συγκεκριμένη τεχνολογική περιοχή (π.χ. βάσεις δεδομένων) είτε με την απόκτηση ευρείας γνώσης σχετικά με την ανάπτυξη έργων.
- Η δομή μητρώου επιτρέπει τη διαχείριση μεγάλων έργων και την ανάπτυξή τους σε γεωγραφικά απομακρυσμένα σημεία.

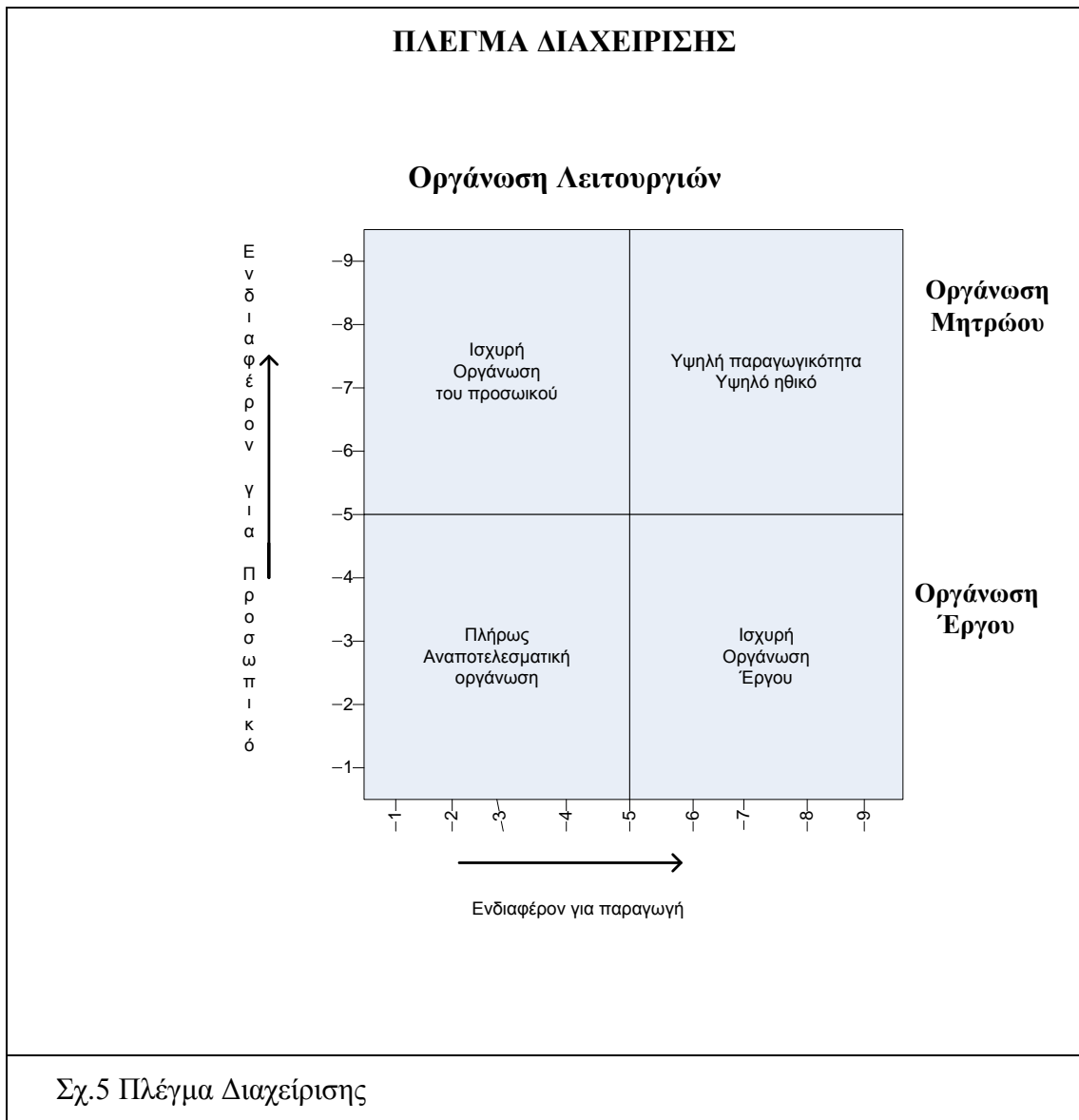
Μειονεκτήματα της οργάνωσης μητρώου είναι τα ακόλουθα:

- Το μητρώο είναι μία περισσότερο πολύπλοκη δομή από τις άλλες δύο που εξετάσαμε και επομένως απαιτεί ώριμη και έμπειρη διαχείριση, η οποία πρέπει να αποφεύγει γραφειοκρατικές ή αυστηρές πρακτικές. Η εμπειρία από την εφαρμογή της δομής μητρώου έχει δείξει ότι θα πρέπει αυτή να εφαρμόζεται σταδιακά.
- Ένα ποσοστό του προσωπικού (υπολογίζεται περίπου στο 15%), δηλ. οι υπεύθυνοι της τεχνική ανάπτυξης για κάθε λειτουργία έργου, θα πρέπει να αναφέρεται σε δύο προϊσταμένους. Μπορεί η κατάσταση αυτή να δημιουργήσει πολλές διαφορετικές απαιτήσεις προερχόμενες από τους δύο προϊσταμένους προς τον τεχνικό υπεύθυνο.
- Η δομή μητρώου απαιτεί περισσότερο σχεδιασμό και προγραμματισμό (δείτε το κεφ. 4 του παρόντος τόμου) αλλά και περισσότερο έλεγχο και παρακολούθηση λόγω της πολυπλοκότητας της δομής. Ένας σημαντικός παράγοντας είναι η ανάθεση μεγάλου μέρους των εργασιών σε ξένες προς τον οργανισμό ομάδες ανάπτυξης.
- Η δομή είναι ακατάλληλη για μικρά έργα εξαιτίας του επιπρόσθετου κόστους και της ανάθεσης αρμοδιοτήτων σε πολλά άτομα. Έτσι μία λογική επιλογή μπορεί να είναι η συνδυασμένη χρήση της δομής μητρώου και της δομής έργου. Ένας γενικός κανόνας μπορεί να είναι η ενσωμάτωση μόνον έργων άνω των 10 συμμετεχόντων στη δομή μητρώου. Εξαιρέση μπορεί να γίνει για μικρότερα έργα στην περίπτωση που αυτά παρουσιάζουν πολλά ισχυρά κοινά στοιχεία (π.χ. κοινές λειτουργίες) με άλλα έργα που ήδη συμμετέχουν στη δομή του μητρώου.

### *Το Πλέγμα Διαχείρισης*

Ένα γνωστό εργαλείο για την καταγραφή του τύπου διαχείρισης είναι το Πλέγμα Διαχείρισης (των R.R.Blake και J.S.Mouton). Το πλέγμα αυτό φαίνεται στο σχ. 5 και δίνει μία δισδιάστατη αναπαράσταση της συμπεριφοράς της διαχείρισης ενός οργανισμού αλλά και ενός μεμονωμένου διαχειριστή. Οι δύο διαστάσεις αφορούν τη συμπεριφορά ως προς το προσωπικό και τη συμπεριφορά ως προς τα έργα που αναλαμβάνει ο οργανισμός. Ο οργανισμός ή διαχειριστής που δείχνει μεγάλο

ενδιαφέρον για το προσωπικό θα πάρει υψηλό βαθμό στον κάθετο άξονα. Ο οργανισμός ή διαχειριστής που δείχνει μεγάλο ενδιαφέρον για την παραγωγή του λογισμικού μέσα από τα έργα θα πάρει υψηλό βαθμό στον οριζόντιο άξονα. Ο ιδανικός οργανισμός ή διαχειριστής θα πάρει υψηλό βαθμό και στις δύο διαστάσεις.



Η εφαρμογή μίας δομής έργου ωθεί έναν οργανισμό προς το κάτω δεξιό τεταρτημόριο του σχήματος, από τη στιγμή που ο οργανισμός θα επιδιώξει κυρίως την ικανοποίηση των στόχων των έργων. Η εφαρμογή μίας δομής λειτουργιών ωθεί έναν οργανισμό προς το άνω αριστερό τεταρτημόριο του σχήματος, από τη στιγμή

που ο οργανισμός επικεντρώνεται κυρίως στο προσωπικό. Η ορθή εφαρμογή της δομής μητρώου, επειδή όπως είδαμε καλύπτει και τις δύο διαστάσεις, προσδίδει στον οργανισμό ένα σημαντικό πλεονέκτημα απέναντι στις άλλες δύο δομές, ωθώντας τον οργανισμό προς το άνω δεξιό τεταρτημόριο.

Η εμπειρία δείχνει ότι ενώ είναι δύσκολο να βρεθούν διαχειριστές που να διακρίνονται και προς τις δύο διαστάσεις, είναι εύκολο να βρεθούν διαχειριστές που διακρίνονται προς μία από τις δύο διαστάσεις. Οι διαχειριστές που είναι κατάλληλοι για διαχείριση προσωπικού μπορούν να συμμετάσχουν στη δομή μητρώου ως διαχειριστές λειτουργιών, ενώ οι διαχειριστές που είναι κατάλληλοι για διαχείριση παραγωγής μπορούν να συμμετάσχουν στη δομή μητρώου ως διαχειριστές έργων.

### **Άσκηση Αυτοαξιολόγησης 2.1**

Να δώσετε παραδείγματα οργανισμών ανάπτυξης λογισμικού διαφόρων τύπων και διαφόρων μεγεθών.

### **Άσκηση Αυτοαξιολόγησης 2.2**

- A) Ποιές είναι οι βασικές οργανωτικές δομές έργων σε έναν οργανισμό;
- B) Ποια είναι τα βασικά κριτήρια για την επιλογή της κατάλληλης οργανωτικής δομής;

### **Άσκηση Αυτοαξιολόγησης 2.3**

Να υποδείξετε τρόπους με τους οποίους η δομή μητρώου μπορεί να εφαρμοσθεί σταδιακά σε έναν οργανισμό.

Υπόδειξη: να χρησιμοποιήσετε τις παραμέτρους της δομής του μητρώου

### **Δραστηριότητα 2.1**

Να προσπαθήσετε να συμπληρώσετε έναν κατάλογο με τα θέματα που πρέπει να απασχολούν την διοίκηση ενός οργανισμού. Βασισθείτε στη μελέτη των πλεονεκτημάτων και μειονεκτημάτων των διαφόρων οργανωτικών δομών. Αιτιολογείστε σύντομα την κάθε απάντησή σας.

## 2.3 Οργάνωση ομάδας

---

Έχοντας αναλύσει διεξοδικά τις οργανωτικές δομές, θα συζητήσουμε τώρα το θέμα της οργάνωσης των ομάδων που θα εκτελέσουν ένα έργο ή ένα τμήμα του (υποέργο).

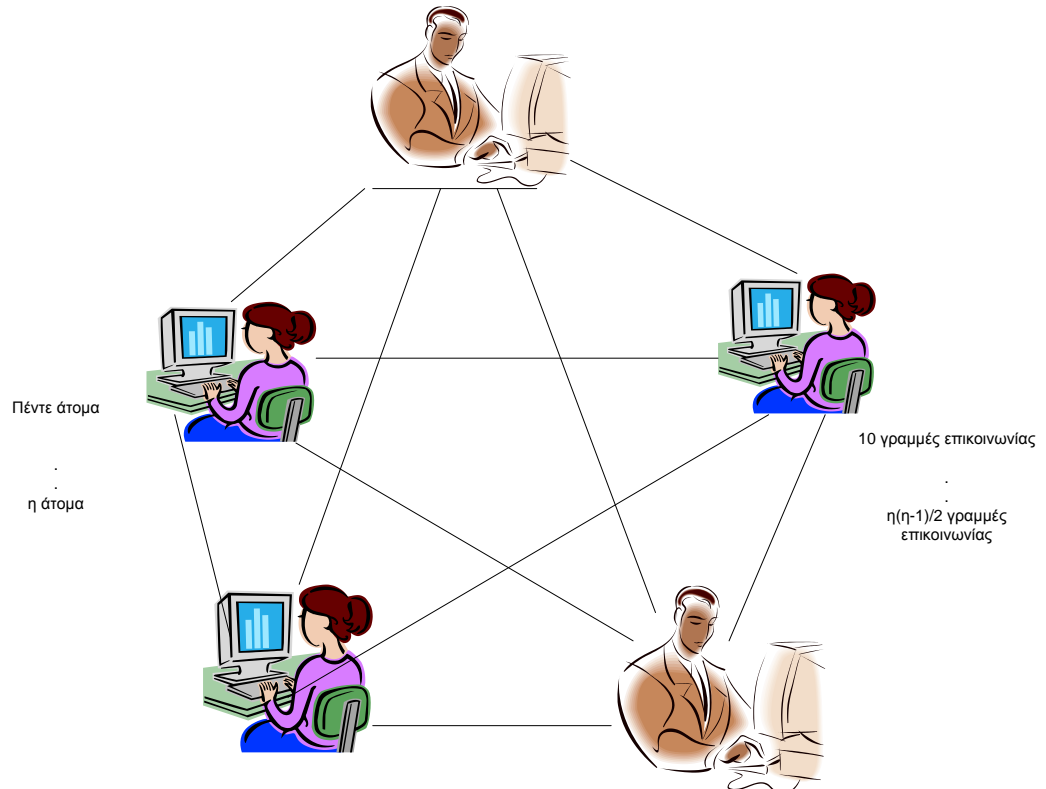
Η δημιουργία αποτελεσματικών ομάδων είναι ιδιαίτερα δύσκολη εργασία και δεν είναι εύκολο να 'διδασθεί'. Σε μεγάλο βαθμό βασίζεται στην εμπειρία και στη διαίσθηση του διαχειριστή / υπεύθυνου για τη δημιουργία της ομάδας. Στο κεφάλαιο αυτό θα προσπαθήσουμε να δώσουμε μερικές μόνο βασικές έννοιες και μερικές κατευθυντήριες γραμμές.

Για ποιο λόγο όμως χρειαζόμαστε ομάδες; Θα μπορούσαμε σε κάθε έργο να έχουμε μία και μόνον ομάδα, ώστε να αποφύγουμε την πολυπλοκότητα του κατακερματισμού του δυναμικού του έργου σε πολλές ομάδες και τη διαχείριση της επικοινωνίας μεταξύ τους. Ένας βασικός λόγος είναι ότι θέλουμε να εκτελέσουμε το έργο παραλληλίζοντας τις διάφορες εργασίες (π.χ. την ανάπτυξη των υποσυστημάτων του), όπως θα δούμε στο κεφάλαιο του προγραμματισμού και σχεδιασμού του έργου. Ένας άλλος όμως λόγος είναι η πολυπλοκότητα της διαχείρισης της επικοινωνίας μεταξύ πολλών ατόμων.

Το σχ. 6 μας δείχνει ότι η αύξηση των γραμμών επικοινωνίας μεταξύ των μελών μίας ομάδας είναι ταχύτερη από την αύξηση του αριθμού των μελών της ομάδας. Το πόσο γρήγορα μπορούν να πολλαπλασιασθούν οι γραμμές επικοινωνίας. Αν μία ομάδα αποτελείται από  $n$  μέλη, δημιουργούνται  $n(n-1)/2$  ζευγη μελών που μπορεί να χρειαστεί να επικοινωνήσουν. Ακόμη υπάρχουν  $2n-1$  διαφορετικές υπο-ομάδες που μπορούν να δημιουργηθούν ώστε να αναλάβουν τμήματα ή συγκεκριμένες εργασίες



του έργου. Είναι ενδεικτικό ότι στην ομάδα των 10 ατόμων της Δραστηριότητας 2.2 που θα δούμε πιο κάτω εντοπίζονται 45 γραμμές επικοινωνίας, και 1023 πιθανές υποομάδες! Είναι προφανές ότι μας συμφέρει να εργαζόμαστε με μικρές σχετικά ομάδες τις οποίες να συντονίζουμε και να διαχειριζόμαστε την επικοινωνία μεταξύ τους. Αυτή συνήθως γίνεται μέσω των τεχνικών διαχειριστών τους.



Σχ. 6. Γραμμές επικοινωνίας σε σχέση με τον αριθμό των μελών μίας ομάδας

Ο όρος οργάνωση ομάδας (*team building*) αφορά την επιτυχημένη σύνθεση ομάδων για την εκτέλεση προκαθορισμένων εργασιών. Κατά τη δημιουργία της σύνθεσης της ομάδας θα πρέπει να διασφαλίζεται ότι τα μέλη της ομάδας θα συνεργάζονται αρμονικά και θα έχουν κοινούς στόχους. Επίσης θα πρέπει να διασφαλίζεται ότι θα ικανοποιούνται σε λογικό βαθμό οι προσωπικές προσδοκίες των μελών της ομάδας. Ένας άλλος σημαντικός στόχος είναι η επίτευξη συνεργιών μεταξύ των γνώσεων και δεξιοτήτων των μελών, ώστε το σύνολο της ομάδας να επιτυγχάνει παραγωγικότητα μεγαλύτερη από το άθροισμα της παραγωγικότητας του κάθε μέλους ξεχωριστά.

*Ρόλοι των μελών της ομάδας*

Τα μέλη μίας ομάδας ανάπτυξης λογισμικού θα κληθούν να αναλάβουν έναν ή περισσότερους *ρόλους*. Οι ρόλοι αυτοί αντιστοιχούν είτε στις φάσεις του μεοντέλου κύκλου ζωής που θα ακολουθηθεί (βλ. Τόμο Α – Τεχνολογία Λογισμικού Ι της Θ.Ε. ΠΛΗ11) είτε σε κάποια από τις παράλληλες δραστηριότητες που συνοδεύουν τις φάσεις αυτές. Επομένως, πιθανοί ρόλοι των μηχανικών σχετιζόμενοι με τις δραστηριότητες των φάσεων μπορεί να είναι:

- Μηχανικός Προδιαγραφών (Requirements Engineer)
- Αναλυτής Συστήματος (System Analyst)
- Σχεδιαστής Συστήματος (System Designer)
- Προγραμματιστής (Programmer)
- Δοκιμαστής (Tester)

Ακόμη, πιθανοί ρόλοι των μηχανικών σχετιζόμενοι με τις παράλληλες δραστηριότητες μπορεί να είναι:

- Τεχνικός Διαχειριστής (Technical Manager)
- Μηχανικός Ποιότητας (Quality Engineer)
- Μηχανικός Τεκμηρίωσης (Documentation Engineer)
- Εκπαιδευτής (Trainer)
- Συντηρητής (Maintainer)

Στην ανάπτυξη λογισμικού συνήθως δεν ανατίθεται ένας μόνο ρόλος σε κάποιο μέλος της ομάδας. Για παράδειγμα, ο μηχανικός ποιότητας θα μπορούσε να εργασθεί και ως δοκιμαστής, καθώς οι δύο δραστηριότητες είναι συγγενείς (οι δοκιμές χρησιμεύουν στη διασφάλιση της ποιότητας). Επίσης μηχανικός προδιαγραφών θα μπορούσε να συμμετάσχει στο έργο ως αναλυτής, καθώς η φάση της ανάλυσης συνήθως ακολουθεί τη φάση των προδιαγραφών. Τέλος, ο αναλυτής θα μπορούσε να συμμετάσχει και ως σχεδιαστής, καθώς η σχεδίαση ακολουθεί την ανάλυση, ενώ πολλές φορές γίνεται σχεδόν ταυτόχρονα, όπως π.χ. στις αντικειμενοστρεφείς μεθοδολογίες.

Σε μεσαία και μεγάλα έργα, ακόμη και σε μικρά, ο ίδιος ρόλος ανατίθεται σε περισσότερα του ενός μέλη μίας ομάδας. Έτσι, ενώ θα μπορούσαμε να έχουμε ένα

μόνο τεχνικό διαχειριστή, είναι λογικό να έχουμε πολλούς προγραμματιστές. Γενικά η δραστηριότητα της υλοποίησης απαιτεί περισσότερους μηχανικούς από τις υπόλοιπες. Η ανάθεση ρόλων εξαρτάται από το μέγεθος της ομάδας ανάπτυξης, το μέγεθος και το είδος της εφαρμογής / συστήματος που αναπτύσσεται αλλά και από το είδος των συμβατικών υποχρεώσεων της ομάδας ανάπτυξης (βλ. Δραστηριότητα 2.2).

Η ανάθεση ρόλων σε διαφορετικά άτομα περιέχει και έναν κίνδυνο: να ανατεθεί υπερβολικό μέρος της ευθύνης της ανάπτυξης σε ένα ή λίγα μόνον άτομα ή σε μία μόνον ομάδα. Για παράδειγμα, θεωρείται λάθος η ανάθεση και του ρόλου του δοκιμαστή στους προγραμματιστές. Ο λόγος είναι ότι ο προγραμματιστής έχει κατανοήσει με συγκεκριμένο τρόπο τις προδιαγραφές και τις ανάγκες του λογισμικού. Αν έχει κάνει λάθος δεν θα μπορέσει να το εντοπίσει ο ίδιος. Επομένως, ο ρόλος του δοκιμαστή θα πρέπει να ανατίθεται σε κάποιον ο οποίος είναι εκτός της ομάδας των προγραμματιστών, δηλ. είτε σε ανεξάρτητο δοκιμαστή ή στο μηχανικό των προδιαγραφών.

Ακόμη και στο εσωτερικό μίας υπο-ομάδας έργου τα άτομα μπορεί να διαφέρουν μεταξύ τους σε πολλά σημεία. Παρατίθεται στη συνέχεια ένας κατάλογος των σημείων στα οποία μπορεί να διαφοροποιούνται δύο οποιαδήποτε μέλη της ομάδας ως προς τις τεχνικές τους ικανότητες:

- Γνώσεις σχετικά με τη λειτουργία ή δραστηριότητα που τους ανατίθεται. Μπορεί π.χ. ο ένας να έχει περισσότερες γνώσεις στην υλοποίηση διασυνδέσεων χρήστη με γραφικά ή στην ανάλυση προδιαγραφών συστήματος. Συνήθως οι γνώσεις προέρχονται από την εκπαίδευση του μηχανικού και από την προσωπική ενασχόληση.
- Ικανότητα σχετικά με τη λειτουργία ή δραστηριότητα. Μπορεί απλά ο ένας να έχει αποδειχθεί ικανότερος στην υλοποίηση διασυνδέσεων χρήστη με γραφικά
- Εμπειρία σχετικά με τη λειτουργία ή δραστηριότητα. Οι δύο μηχανικοί φαίνεται να έχουν περίπου τις ίδιες γνώσεις ή ικανότητες αλλά ο ένας είναι παλαιότερος και να έχει υλοποιήσει περισσότερες διασυνδέσεις χρήστη ή να έχει αναλύσει περισσότερα συστήματα. Είναι επομένως σε ευνοϊκότερη θέση μπροστά σε προβλήματα που έχει ήδη αντιμετωπίσει.

- Εμπειρία σε παρόμοιες εφαρμογές. Παρόμοιες εφαρμογές (π.χ. διασυνδέσεις χρήστη για άτομα με ειδικές ανάγκες) παρουσιάζουν παρόμοια προβλήματα.
- Εμπειρία στο ενοποιημένο περιβάλλον ανάπτυξης (IDE). Ο μηχανικός μπορεί να έχει ήδη χρησιμοποιήσει σε προηγούμενα έργα το ίδιο περιβάλλον ανάπτυξης.

Τα μέλη της ομάδας όμως μπορεί να διαφέρουν μεταξύ τους και σε πολλά σημεία σχετικά με την προσωπικότητά τους. Παρατίθεται στη συνέχεια ένας δεύτερος κατάλογος σχετικών σημείων:

- Ικανότητα επικοινωνίας. Πρόκειται για πολύ σημαντική ιδιότητα, αφού οι μηχανικοί καλούνται να συνεργασθούν μεταξύ τους αλλά και με άτομα εξωτερικά προς την ομάδα ανάπτυξης (προσωπικό του πελάτη του έργου, συμβούλους, χρήστες, εξωτερικούς προμηθευτές). Η ικανότητα επικοινωνίας αφορά την ταχύτητα και την ποιότητα της επικοινωνίας με άλλα άτομα.
- Ικανότητα διαμοίρασης ευθύνης με άλλα μέλη της ομάδας. Πρόκειται για ιδιότητα πολύ σημαντική, αφού όπως συζητήθηκε τα μέλη της ομάδας πρέπει να έχουν κοινό στόχο. Η ικανότητα διαμοίρασης της ευθύνης αφορά την ανάληψη και διεκπεραίωση ενός δίκαιου μεριδίου της ευθύνης στο πλαίσιο της ομαδικής δραστηριότητας
- Ικανότητα διαχείρισης. Είναι η ικανότητα για προγραμματισμό, παρακολούθηση και έλεγχο εργασιών, λήψη αποφάσεων, προετοιμασία αναφορών κλπ.
- Έφεση για καινοτομία. Πρόκειται για μία ικανότητα που αναδεικνύεται σε έργα που η ομάδα αντιμετωπίζει για πρώτη φορά ή για έργα που έχουν έναν εγγενή καινοτομικό χαρακτήρα.

Είναι προφανές ότι τόσο η επιλογή των μελών μίας ομάδας από το σύνολο των διαθέσιμων στελεχών του οργανισμού, όσο και η ανάθεση ρόλου ή ρόλων σε αυτά εξαρτάται από την αξιολόγηση όλων αυτών των ικανοτήτων των υποψηφίων. Για παράδειγμα η ικανότητα επικοινωνίας είναι σημαντική για ρόλους όπως του μηχανικού προδιαγραφών που θα πρέπει να έρθει σε επαφή με χρήστες. Η ικανότητα

διαχείρισης έχει προφανή σχέση με την επιλογή ενός ατόμου για τη θέση του τεχνικού διαχειριστή.

*Προτιμήσεις σχετικά με τον τρόπο εργασίας των μελών της ομάδας*

Θα ολοκληρώσουμε τη συζήτηση για την οργάνωση ομάδων εξετάζοντας τους διαφορετικούς τρόπους με τους οποίους κάθε μέλος μίας ομάδας προτιμά να εργάζεται. Κάποιοι από τους μηχανικούς που έχουμε στη διάθεσή μας μπορεί να προτιμούν να εργάζονται μόνοι τους μπροστά στην οθόνη του υπολογιστή τους, χωρίς να βασίζονται ιδιαίτερα στην επικοινωνία με άλλα μέλη της ομάδας τους. Αντίθετα, κάποιοι άλλοι μπορεί να βασίζονται ακριβώς στην απόκτηση πληροφοριών από άλλα μέλη και στη μετέπειτα επεξεργασία τους. Άλλοι πάλι μπορεί να βασίζονται στη χρήση της διαίσθησής τους για να παίρνουν αποφάσεις, χωρίς να ολοκληρώνουν την επεξεργασία όλων των στοιχείων που έχουν στη διάθεσή τους. Επίσης μπορεί να έχουμε συμπεριφορές που κινούνται μεταξύ αυτών των περιπτώσεων ή και να συνδυάζουν ορισμένες εξ αυτών.

Όταν ο μηχανικός λογισμικού προτιμά να εξωτερικεύει τις σκέψεις του θεωρείται ότι ανήκει στην κατηγορία των *εξωστρεφών* (Jung). Αν προτιμά να ακούει πρώτα τους άλλους και με βάση τις πληροφορίες που συλλέγει να καταλήγει στις τελικές αποφάσεις, επιλογές και ενέργειές του ανήκει στους *εσωστρεφείς*. Η κατηγορία των *δισαισθητικών* είναι αυτοί οι οποίοι βασίζονται στη διαίσθησή τους ενώ οι *ορθολογιστές* αποφασίζουν αφού εξετάσουν διεξοδικά τα στοιχεία που έχουν στη διάθεσή τους. Με βάση αυτή την ανάλυση μπορούμε να ορίσουμε τέσσερις κατηγορίες εργαζομένων, τους *εσωστρεφείς δισαισθητικούς*, τους *εξωστρεφείς δισαισθητικούς*, τους *εσωστρεφείς ορθολογιστές* και τους *εξωστρεφείς ορθολογιστές*. Η κατάταξη ενός εργαζόμενου σε μία από τις τέσσερις κατηγορίες μπορεί να γίνει δισαισθητικά ή και με την εφαρμογή συγκεκριμένων ερωτηματολογίων – τεστ προσωπικότητας.

Ορισμένες δραστηριότητες της ανάπτυξης ταιριάζουν περισσότερο με κάποιους από αυτούς τους τύπους παρά με άλλους. Για παράδειγμα ο διαχειριστής έργου συχνά πρέπει να είναι περισσότερο δισαισθητικός παρά ορθολογιστής. Ο λόγος είναι ότι

πολλές φορές απαιτείται η ταχεία λήψη αποφάσεων επειδή δεν υπάρχει χρόνος για πλήρη ανάλυση των διαθέσιμων πληροφοριών. Συχνά μάλιστα δεν υπάρχουν και όλες οι πληροφορίες και μετρήσεις που θα χρειαζόμασταν για να έχουμε μία πλήρη εικόνα του προβλήματος.

Αντίθετα για τις δραστηριότητες των προδιαγραφών και της ανάλυσης ίσως θα προτιμούσαμε εξωστρεφείς ορθολογιστές, οι οποίοι θα έπαιρναν υπόψη τους όλα τα απαραίτητα στοιχεία στη διάθεσή τους για να αποφύγουν τυχόν λάθη και ταυτόχρονα θα εξωτερίκευαν την όλη διαδικασία για να μοιραστούν με τους πελάτες, χρήστες και τους άλλους μηχανικούς τις σκέψεις και απόψεις τους.

Τέλος, θα πρέπει να σημειώσουμε ότι οι τύποι εργασίας μπορούν να μας βοηθήσουν και στη γενικότερη σύνθεση μίας ομάδας. Έχει παρατηρηθεί στην τεχνολογία λογισμικού ότι οι ομάδες που αποδίδουν περισσότερο είναι αυτές που αποτελούνται από συνδυασμούς όλων των τύπων. Ο λόγος είναι ότι αφενός μεν συνολικά όλοι οι τύποι μπορούν να καλύψουν όλους τους ρόλους, αφετέρου τύποι με διαφορετικές προσωπικότητες συνεργάζονται καλύτερα. Για παράδειγμα, πειράματα που έγιναν από έλληνες ερευνητές σε ελληνικό ίδρυμα έδειξαν ότι προγραμματιστές με διαφορετικές προσωπικότητες ήταν πιο αποδοτικοί, όταν εργάστηκαν σε ζεύγη (*pair programming*) για την ανάπτυξη λογισμικού.

#### **Άσκηση Αυτοαξιολόγησης 2.4**

Να περιγράψετε συνοπτικά τους στοχους της οργάνωσης ομάδας.

#### **Δραστηριότητα 2.2**

A) Να αναθέσετε ρόλους σε μία ομάδα ανάπτυξης λογισμικού που αποτελείται από δέκα άτομα. Υπόδειξη: Θα πρέπει να προσπαθήσετε να καλύψετε όλους τους ρόλους, δίνοντας περισσότερο βάρος στις κύριες φάσεις.

B) Πως θα αλλάξει η ανάθεση ρόλων αν το λογισμικό (1) θα έχει πολλούς και μάλλον άπειρους χρήστες, (2) θα είναι τμήμα ενός κρίσιμου για ανθρώπινες ζωές

συστήματος, (3) θα εκτελεσθεί στο πλαίσιο συμφωνίας με έναν ιδιαίτερα αυστηρό σε προδιαγραφές και γραφειοκρατικό φορέα (π.χ. Υπουργείο Αμύνης);

### **Δραστηριότητα 2.3**

Στις τρεις περιπτώσεις της απάντησης στην ερώτηση (B) της δραστηριότητας 2.2 να υποδείξετε στοιχεία των μελών των ομάδων που πιστεύετε ότι θα μπορούσαν να οδηγήσουν στην ανάθεση νέων ρόλων. Π.χ. με ποιο κριτήριο θα επιλέγατε κάποιον για μηχανικό προδιαγραφών.

### **Δραστηριότητα 2.4**

Χρησιμοποιώντας μία μηχανή αναζήτησης να προσπαθήσετε να εντοπίσετε στο Διαδίκτυο ερωτηματολόγια τα οποία ανιχνεύουν την προσωπικότητα και κατατάσσουν τον εξεταζόμενο σε μία από τις τέσσερις κατηγορίες που συζητήθηκαν στην τελευταία παράγραφο.

## **2.4 Σύνοψη – Βιβλιογραφία**

---

Στο κεφάλαιο αυτό είδαμε δύο βασικές οργανωτικές δομές: την οργάνωση των διαφορετικών έργων ενός οργανισμού και την οργάνωση των ομάδων που μπορεί να συμμετέχουν στα έργα.

Γνωρίσαμε τρεις οργανωτικές δομές για τα έργα, από τις οποίες η οργάνωση λειτουργιών φαίνεται να είναι η πιο απλή, αλλά ίσως και η πιο προβληματική. Η οργάνωση έργου είναι μία καλή επιλογή, ενώ η οργάνωση μητρώου φαίνεται να είναι εκείνη που μπορεί να επιτύχει το βέλτιστο αποτέλεσμα, εφόσον εφαρμοσθεί σωστά. Σε γενικές γραμμές, ένας συνδυασμός της οργάνωσης έργου και της οργάνωσης πίνακα (δηλ. η δημιουργία μικρών σχετικά ομάδων έργου μέσα σε μία μεγάλη οργάνωση πίνακα) φαίνεται να είναι η προσφορότερη λύση για οργανισμούς που αναπτύσσουν τόσο μεγάλα όσο και μικρά έργα.

Η οργάνωση ομάδας μας θέτει μπροστά σε μία σειρά από προκλήσεις και οι αποφάσεις μας μπορούν να επηρεάσουν καθοριστικά την επιτυχία της κάθε ομάδας και επομένως την πορεία του έργου. Γνωρίσαμε τα κριτήρια για επιλογή του προσωπικού και ανάθεση ρόλων σε αυτό με βάση τις γνώσεις, τις ικανότητες, την εμπειρία αλλά και την προσωπικότητα και την προτίμηση του τρόπου εργασίας

## Βιβλιογραφία

---

- [1] ‘Software Management’, Donald J. Reifer, IEEE Computer Society Press, 2002. Πρόκειται για Τόμο με συλλογή δημοσιευμένων άρθρων σχετικά με όλα τα σημαντικά θέματα της διαχείρισης έργων λογισμικού
- [2] Software Project Management, Joel Henry, Addison-Wesley. Βιβλίο για τη διαχείριση έργων που εστιάζει στην πρακτική εφαρμογή
- [3] Software Engineering, Maciaszek, Liang, Addison Wesley. Βιβλίο γενικά για την Τεχνολογία Λογισμικού, με έμφαση στη χρήση εργαλείων
- [4] Software Engineering, 7<sup>th</sup> ed, Ian Sommerville, Addison-Wesley. Κλασικό σύγγραμμα για την Τεχνολογία Λογισμικού
- [5] Software Engineering, Theory and Practice, S. L. Pfleeger, Prentice Hall, Βιβλίο γενικά για την Τεχνολογία Λογισμικού, αφιερώνει ένα κεφάλαιο στα θέματα του παρόντος Τόμου. Εκδίδεται και σε ελληνική μετάφραση από τον Κλειδάριθμο



## ΚΕΦΑΛΑΙΟ 3: ΔΙΑΧΕΙΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΟΥ ΕΡΓΑΣΙΩΝ

### Συγγραφέας: Πάνος Φιτσιλής

#### **Σκοπός:**

Σκοπός του Κεφαλαίου 3 είναι να παρουσιάσουμε τη διαχείριση του αντικειμένου εργασιών. Ο βασικός στόχος της διαδικασίας του ορισμού του αντικειμένου των εργασιών είναι να καταγράψει τις απαιτήσεις των χρηστών από το υπό ανάπτυξη σύστημα και στη συνέχεια να προσδιορίσει επακριβώς τις ικανές και αναγκαίες εργασίες που πρέπει να εκτελεσθούν για την επιτυχή περάτωση ενός έργου. Αποτελεί τη βάση για την κατανόηση των στόχων του έργου από όλους τους συμμετέχοντες. Επιπλέον, στο κεφάλαιο αυτό παρουσιάζεται η δομή ανάλυσης έργου, καθώς και οι βασικές έννοιες διαχείρισης αλλαγών.

#### **Προσδοκώμενα Αποτελέσματα:**

Στο κεφάλαιο αυτό :

- Θα κατανοήσετε τις έννοιες της διαχείρισης αντικειμένου εργασιών (scope management) και της διαχείρισης απαιτήσεων έργου (requirements management).
- Θα κατανοήσετε τον τρόπο δημιουργίας της δομής ανάλυσης εργασιών έργου (Work Breakdown Structure, WBS).
- Θα κατανοήσετε βασικές αρχές διαχείρισης αλλαγών (change management), καθώς και τον τρόπο διαχείρισης των παραδοτέων ενός έργου.

#### **Έννοιες – Κλειδιά:**

Διαχείριση αντικειμένου εργασιών (scope management).

Διαχείριση απαιτήσεων (requirements management).

Δομή ανάλυσης εργασιών (Work Breakdown Structure, WBS).

Παραδοτέο (deliverable).

Διαχείριση σχηματισμών (configuration management).

Ενοποιημένη προσέγγιση (unified process).

Αποδέσμευση (release).

Έλεγχος αλλαγών (change control).

### Ενότητα 3.1 Εισαγωγή.

---

Η διαχείριση αντικειμένου εργασιών είναι η διατήρηση της λεπτής ισορροπίας μεταξύ των αναγκών του πελάτη και της ρεαλιστικής δυνατότητας να αναπτύξουμε και να παραδώσουμε ένα πληροφοριακό σύστημα.

*Rational Software Corporation*

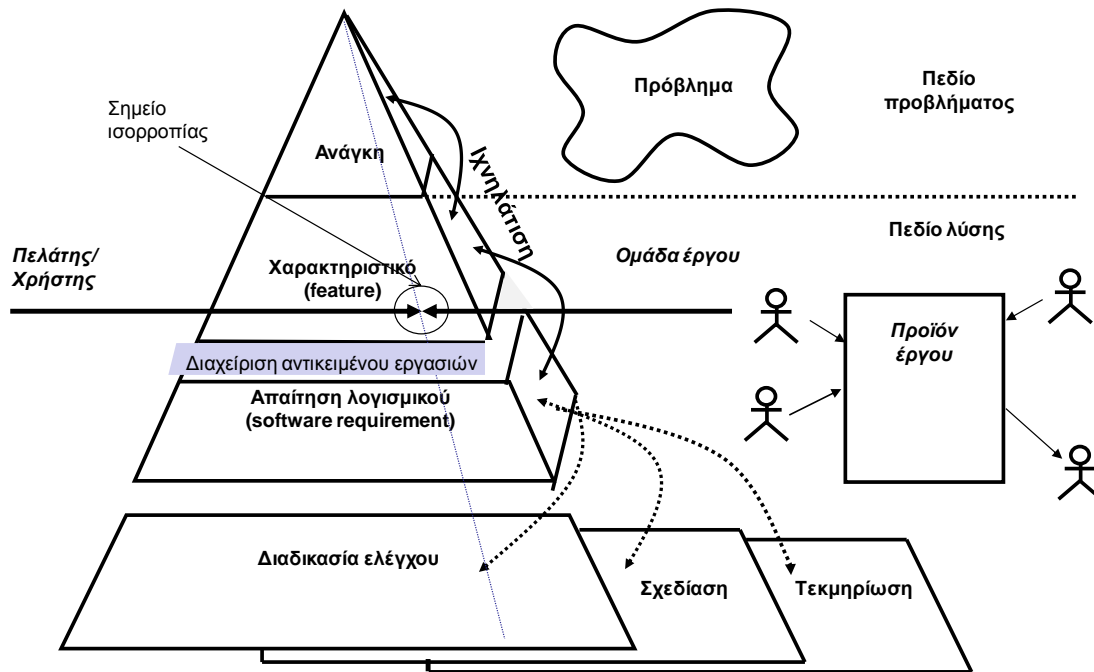
Ο βασικός στόχος της διαχείρισης έργων λογισμικού είναι η ανάπτυξη ποιοτικού λογισμικού, στον προκαθορισμένο χρόνο και με το προϋπολογισθέν κόστος, το οποίο ταυτόχρονα να ικανοποιεί τις απαιτήσεις του πελάτη.

Τις περισσότερες φορές η επιτυχία ενός έργου εξαρτάται από την επιτυχή κατανόηση και καταγραφή των αναγκών του πελάτη. Η λανθασμένη κατανόηση των αναγκών οδηγεί σε σφάλματα, τα οποία σύμφωνα με σχετικές μελέτες (Standish, 1994) είναι ιδιαίτερα δύσκολο να διορθωθούν. Επιπλέον, είναι ιδιαίτερα σημαντικό και δύσκολο να προσδιορίσουμε με ακρίβεια τα όρια του συστήματος που καλούμαστε να αναπτύξουμε, αφού πολλές φορές τα χαρακτηριστικά του λογισμικού που καλούμαστε να αναπτύξουμε δεν προσδιορίζονται με σαφήνεια.

Η διαχείριση του αντικειμένου εργασιών, σε συνδυασμό με τη διαχείριση απαιτήσεων, ξεκινά από την κατανόηση του πεδίου προβλήματος και τη μετάφραση των αναγκών του πελάτη σε χαρακτηριστικά του συστήματος (system features) και στη συνέχεια σε απαιτήσεις λογισμικού (software requirements). Συνεπώς, η καλή κατανόηση του πεδίου προβλήματος και των επιχειρηματικών και τεχνικών προβλημάτων του πελάτη αποτελεί ακρογωνιαίο λίθο για την επιτυχία ενός έργου. Επιπλέον, το μοντέλο των απαιτήσεων αποτελεί τη βάση για τη σχεδίαση, τον έλεγχο και την τεκμηρίωση του συστήματος.

Τέλος, η ιχνηλάτιση των απαιτήσεων επιτρέπει:

- Την αξιολόγηση των αλλαγών στις απαιτήσεις.
- Την αξιολόγηση των αποτελεσμάτων του ελέγχου.
- Την επαλήθευση της ικανοποίησης των απαιτήσεων.
- Τη διαχείριση των αλλαγών.



Σχήμα 3.5.: Η συνολική εικόνα της διαχείρισης του αντικειμένου εργασιών.

### Ενότητα 3.2 Ο Ορισμός του Αντικειμένου Εργασιών ενός Έργου.

Ο βασικός στόχος της διαδικασίας του **ορισμού του αντικειμένου των εργασιών** ενός έργου (PMI, 2004) είναι να καταγράψει τις απαιτήσεις των χρηστών από το υπό ανάπτυξη σύστημα και στη συνέχεια να προσδιορίσει επακριβώς τις ικανές και αναγκαίες εργασίες που πρέπει να εκτελεστούν για την επιτυχή περάτωση ενός έργου. Η διαδικασία αυτή αποτελεί τη βάση για την κατανόηση των στόχων του έργου απ' όλους τους συμμετέχοντες.

Μπορούμε να πούμε ότι το αντικείμενο των εργασιών του έργου προκύπτει από την ισορροπία τριών παραγόντων: της απαιτούμενης λειτουργικότητας, των διαθέσιμων πόρων και του χρόνου μέσα στον οποίο πρέπει να εκτελεσθεί το έργο. Έτσι, αν η προσπάθεια (effort) που απαιτείται για την υλοποίηση των απαιτήσεων του έργου είναι ίση με τη διαθέσιμη προσπάθεια ( $\text{πόροι} \times \text{χρόνος}$ ), τότε λέμε ότι η υλοποίηση του έργου είναι εφικτή, ενώ σε αντίθετη περίπτωση θα πρέπει να μειώσουμε το εύρος του αντικειμένου των εργασιών, δηλαδή να ικανοποιήσουμε λιγότερες απαιτήσεις (requirements) και να προσφέρουμε στον πελάτη ένα σύστημα με λιγότερα χαρακτηριστικά (features). Εναλλακτικά, μπορούμε να προσθέσουμε πόρους στο έργο ώστε να εξασφαλίσουμε μεγαλύτερη διαθέσιμη προσπάθεια, αλλά σύμφωνα με τον Brooks (Brooks, 1975) αυτό είναι επικίνδυνο και δεν έχει πάντα θετικό αποτέλεσμα.



**Σχήμα 3.6.:** Το Αντικείμενο Εργασιών ενός Έργου.

Στο Σχήμα 3.6, το σκιασμένο τετράγωνο αντιπροσωπεύει το αντικείμενο εργασιών ενός έργου, το οποίο περιορίζεται από τους διαθέσιμους πόρους και τα ορόσημα του έργου. Όπως αναφέραμε και πριν, ο Brooks στο πολύ γνωστό του βιβλίο “The Mythical Man Month” (Brooks, 1975), απέδειξε ότι αν προσθέσουμε εργαζόμενους σε ένα ήδη καθυστερημένο έργο μπορεί να καθυστερήσει ακόμη πιο πολύ, αφού οι νέοι εργαζόμενοι στο έργο χρειάζονται εκπαίδευση και υποστήριξη, γεγονός που καθυστερεί τους ήδη εργαζόμενους.

Η καταγραφή του αντικειμένου των εργασιών ενός έργου γίνεται στο έγγραφο του ορισμού του αντικειμένου εργασιών (project scope statement) (PMI, 2004), το οποίο περιέχει τις πληροφορίες σχετικά με τους στόχους του έργου (project objectives), τον προσδιορισμό των χαρακτηριστικών του λογισμικού ή των υπηρεσιών που αναπτύσσονται στο έργο, τον ορισμό των βασικών απαιτήσεων του έργου, τα παραδοτέα, τα κριτήρια αποδοχής των παραδοτέων του έργου, τους περιορισμούς (constraints), καθώς και τις παραδοχές (assumptions). Εναλλακτικά, αντί για την ανάπτυξη του εγγράφου του ορισμού του αντικειμένου εργασιών, στην ενοποιημένη προσέγγιση προτείνεται η ανάπτυξη του οράματος του συστήματος (vision document). Το έγγραφο του οράματος παρουσιάζει σε υψηλό επίπεδο, τόσο το πρόβλημα, όσο και τη λύση του.

Συνεπώς, ο βασικός τρόπος για τη διαχείριση του αντικειμένου των εργασιών ενός έργου είναι η καταγραφή των απαιτήσεων. Προφανώς, δεν αναφερόμαστε στη λεπτομερή καταγραφή όλων των απαιτήσεων, αλλά στη δημιουργία υψηλού επιπέδου απαιτήσεων που θα μας επιτρέψουν να δημιουργήσουμε μια γραμμή αναφοράς (baseline).

### Ενότητα 3.3 Διαχείριση Απαιτήσεων Λογισμικού.

*Απαίτηση* είναι μια ικανότητα που θα πρέπει να διαθέτει το λογισμικό με σκοπό ο χρήστης να λύνει ένα πρόβλημα ή να πετυχαίνει ένα στόχο του ή να ικανοποιείται ένα πρότυπο ή ένας όρος του συμβολαίου, κ.λπ. (Dorfman, 1990).

Αντίστοιχα, *διαχείριση απαιτήσεων* είναι ο συστηματικός τρόπος εύρεσης, οργάνωσης και τεκμηρίωσης των απαιτήσεων του συστήματος, καθώς και ο έλεγχος των αλλαγών που συμβαίνουν σε όλη τη διάρκεια του έργου (Leffingwell, 2003).

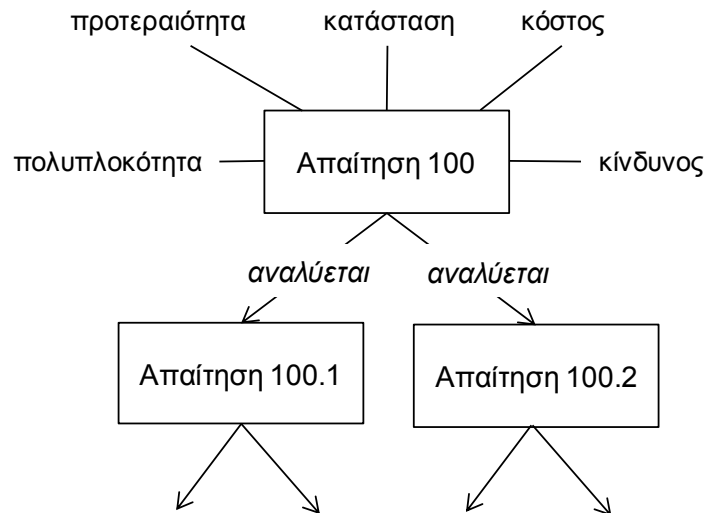
Ο λόγος που θα πρέπει να διαχειριζόμαστε τις απαιτήσεις είναι ότι ακόμη και για ένα απλό σύστημα λογισμικού υπάρχουν δεκάδες, αν όχι εκατοντάδες απαιτήσεις (Jalotte, 2002). Συνεπώς, θα πρέπει να βρούμε ποιες είναι οι απαιτήσεις σε συνεργασία με τον πελάτη του συστήματος, θα πρέπει να τις οργανώσουμε, καθώς και να τις τεκμηριώσουμε.

Σκοπός μας είναι να μπορούμε να απαντήσουμε σε ερωτήσεις του ακόλουθου όπως:

- Ποιο μέλος της ομάδας έργου είναι υπεύθυνο για την απαίτηση #347; Ποιο μέλος της ομάδας έχει το δικαίωμα να την τροποποιήσει ή να τη διαγράψει;
- Αν αλλάξουμε την απαίτηση #210, ποια άλλη απαίτηση επηρεάζεται;
- Πως μπορούμε να είμαστε βέβαιοι ότι η απαίτηση #142 έχει υλοποιηθεί; Ποιες είναι οι περιπτώσεις ελέγχου (test cases) που σχετίζονται με την απαίτηση #142;

Για τον ορισμό των απαιτήσεων αναπτύσσουμε το μοντέλο των περιπτώσεων χρήσης (use case model), το οποίο περιγράφεται με αναλυτικό τρόπο στον τόμο Β της Θ.Ε. ΠΛΗ24 του Προγράμματος Σπουδών «Πληροφορική» του ΕΑΠ.

Επιπλέον, οι απαιτήσεις του συστήματος δημιουργούν ιεραρχίες, καθώς οι απαιτήσεις υψηλού επιπέδου αναλύονται σε άλλες χαμηλότερου επιπέδου, οι οποίες είναι πιο λεπτομερείς. Για τον καλύτερο χειρισμό των απαιτήσεων ορίζουμε μια σειρά από πεδία (requirements attributes), τα οποία περιγράφουν τα χαρακτηριστικά και τις ιδιότητες των απαιτήσεων. Τέτοια πεδία είναι το κόστος υλοποίησης, η πολυπλοκότητα της απαίτησης, οι κίνδυνοι που υφίστανται από την υλοποίηση της απαίτησης, καθώς και η προτεραιότητα. Η προτεραιότητα αποτελεί βασικό χαρακτηριστικό για τη διαχείριση του αντικειμένου των εργασιών ενός έργου, αφού η υλοποίηση απαιτήσεων χαμηλής προτεραιότητας μπορεί να αναβληθεί για επόμενη έκδοση του συστήματος, σε περίπτωση έλλειψης χρόνου ή πόρων. Η ιεραρχία των απαιτήσεων παρουσιάζεται γραφικά στο Σχήμα 3.7.



Σχήμα 3.7.: Ιεραρχία απαιτήσεων.

Η χρήση των πεδίων για τον ορισμό του αντικειμένου εργασιών ενός έργου παρουσιάζεται στο Σχήμα 3.8.

Χαρακτηριστικά - Υψηλού επιπέδου απαιτήσεις	Προτεραιότητα	Δυσκολία	Κίνδυνος	Πολυπλοκότητα	Συνολική αξιολόγηση
1: Αποθήκευση κριτηρίων αναζήτησης	Μέτρια	Χαμηλή	Χαμηλή	Υψηλή	
2: Ικανότητα αποθήκευσης δεδομένων σε RIF μορφή	Μέτρια	Χαμηλή	Χαμηλή	Υψηλή	
3: Δυνατότητα αναιρέσης	Μέτρια	Υψηλή	Μέτρια	Μέτρια	
4: Υποστήριξη περισσότερων μορφών ημερομηνίας.	Μέτρια	Μέτρια	Χαμηλή	Μέτρια	
5: Δυνατότητα μετάφρασης σε συγκεκριμένο σημείο του εγγράφου	Μέτρια	Μέτρια	Μέτρια	Υψηλή	
6: Δυνατότητα παρουσίασης πεδίων	Μέτρια	Μέτρια	Μέτρια	Υψηλή	
7: Δυνατότητα δημιουργίας προτύπων	Μέτρια	Μέτρια	Μέτρια	Υψηλή	
8: Δημιουργία εγγράφου με πάτημα ενός πλήκτρου	Μέτρια	Υψηλή	Υψηλή	Μέτρια	
9: Δυνατότητα λειτουργίας σε νέο λειτουργικό σύστημα	Υψηλή	Μέτρια	Υψηλή	Υψηλή	

Σχήμα 3.8.: Πίνακας αξιολόγησης απαιτήσεων.

### Δραστηριότητα 3.1

Η διαχείριση απαιτήσεων στα έργα πληροφορικής γίνεται με τη χρήση CASE εργαλείων. Ερευνήστε στο internet με σκοπό να εντοπίσετε CASE εργαλεία για τη διαχείριση απαιτήσεων. Ποια είναι τα βασικά τους χαρακτηριστικά.

## Ενότητα 3.4 Δομή Ανάλυσης Εργασιών.

Η δομή ανάλυσης εργασιών (Work Breakdown Structure, WBS) υποδιαιρεί το αντικείμενο ενός έργου σε πακέτα εργασιών με σκοπό τον πιο εύκολο χειρισμό τους

(PMI, 2001). Η σωστή ανάλυση ενός έργου, σε συνδυασμό με τις κατάλληλες διαδικασίες, αποτελεί κρίσιμο παράγοντα επιτυχίας του έργου. Αν και η δομή WBS είναι ιδιαίτερα γνωστή και ευρέως χρησιμοποιούμενη, η σχετική βιβλιογραφία είναι ιδιαίτερα περιορισμένη, αφού η δομή WBS εξαρτάται σε μεγάλο βαθμό από το είδος του έργου, την οργανωσιακή κουλτούρα του οργανισμού, κ.λπ. (Greene, 2005; Humphrey, 1989).

Στόχος μας είναι να αναλύσουμε τις εργασίες ενός έργου με τέτοιο τρόπο ώστε να είναι:

- Διαχειρίσιμες, από ένα μέλος της ομάδας του έργου ή από ένα μικρό σύνολο εργαζομένων στο έργο.
- Ανεξάρτητες ή με τις μικρότερες δυνατές εξαρτήσεις από άλλες εργασίες.
- Ομαδοποιημένες, ώστε να μπορούμε να διαπιστώνουμε την πρόοδο του έργου.
- Μετρήσιμες, ώστε να μπορούμε να μετρούμε την πρόοδο του έργου.

Εκτός από την ανάλυση ενός έργου σε επιμέρους δραστηριότητες που θα πρέπει να εκτελεστούν, η δομή WBS είναι ιδιαίτερα χρήσιμη και στη γενικότερη διαχείριση του έργου, αφού παρέχει πληροφορίες για:

- το αντικείμενο των εργασιών του έργου,
- τις εργασίες που πρέπει να γίνουν έτσι ώστε να αναπτύξουμε ένα παραδοτέο,
- τις αναθέσεις εργασιών σε εργαζομένους,
- την ιεραρχία των παραδοτέων,
- την κοστολόγηση των παραδοτέων και την χρονοδρομολόγηση των εργασιών και
- γενικότερα μας διευκολύνει στην παρακολούθηση του έργου.

Η δομή WBS είναι ιεραρχική και ο αριθμός των επιπέδων κυμαίνεται από τρία έως έξι ανάλογα με την πολυπλοκότητα του έργου. Όπως έχουμε ήδη αναφέρει η δομή WBS που θα χρησιμοποιήσουμε για το έργο εξαρτάται από το είδος του έργου, τον οργανισμό που εκτελεί το έργο, κ.ά. Μια αρκετά γνωστή και διαδεδομένη προσέγγιση είναι αυτή που παρουσιάζεται στο Σχήμα 3.5.

	<i>Επίπεδο</i>	<i>Περιγραφή</i>
<i>Διοικητικά Επίπεδα</i>	1	Πρόγραμμα (programme)
	2	Έργο (project)
	3	Δράση (task)
<i>Τεχνικά Επίπεδα</i>	4	Υποδράση(subtask)
	5	Πακέτο εργασίας (work package)
	6	Δραστηριότητα(activity)

**Σχήμα 3.9.:** Τυπική δομή WBS.

Τα διοικητικά επίπεδα της δομής WBS συνήθως προσδιορίζονται από τον πελάτη ως μέρος της διακήρυξης και χρησιμεύουν για τη δημιουργία αναφορών προόδου, ενώ τα τεχνικά επίπεδα προσδιορίζονται από την ομάδα του έργου.

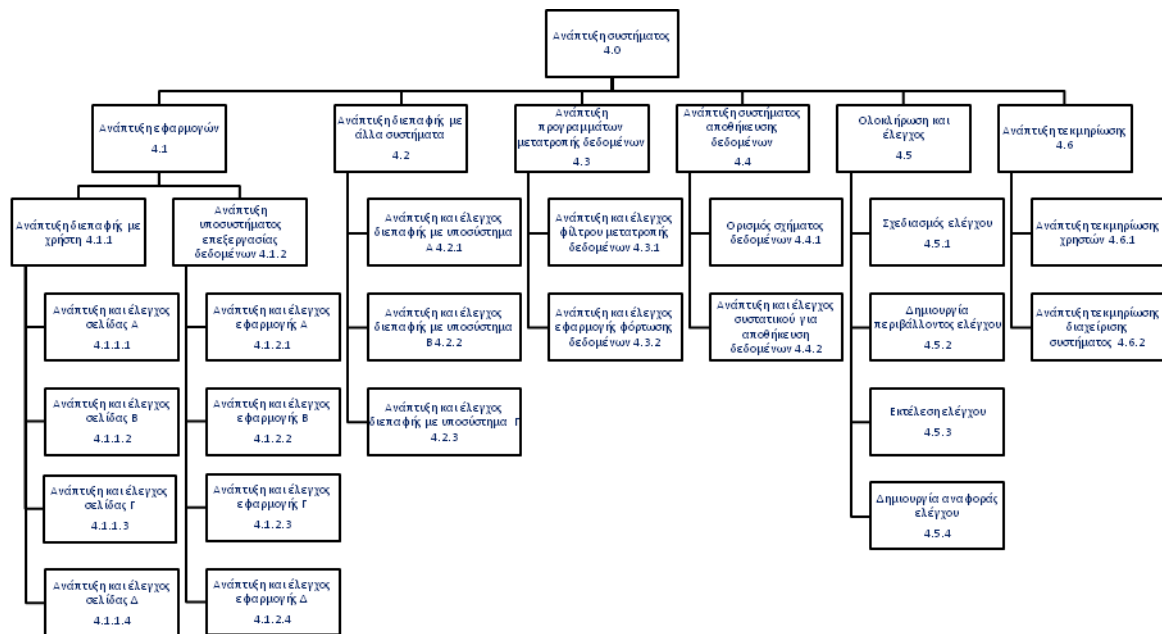
Μερικές γενικές παρατηρήσεις είναι οι ακόλουθες:

- Τα πρώτα τρία επίπεδα προσδιορίζουν συνολική προσπάθεια και όχι την προσπάθεια ενός ατόμου ή μιας ομάδας.
- Κάθε επίπεδο είναι το άθροισμα των δραστηριοτήτων των παρακάτω επιπέδων.
- Κάθε δραστηριότητα ανήκει μόνο σε ένα πακέτο εργασίας.
- Το επίπεδο στο οποίο διαχειριζόμαστε το έργο είναι αυτό του πακέτου εργασίας.
- Ο διαχειριστής του έργου συνήθως εργάζεται με τα τρία πρώτα επίπεδα WBS με σκοπό τη δημιουργία αναφορών προόδου (progress reports) για τα στοιχεία της δομής WBS των τριών πρώτων επιπέδων.

### **Παράδειγμα 3.1**

Το παράδειγμα του Σχήματος 3.6. παρουσιάζει τη δομή WBS για ένα απλό έργο ανάπτυξης λογισμικού. Στο παράδειγμα παρουσιάζεται μόνο η δομή WBS για την ανάπτυξη του λογισμικού του συστήματος και όχι για το έργο συνολικά.





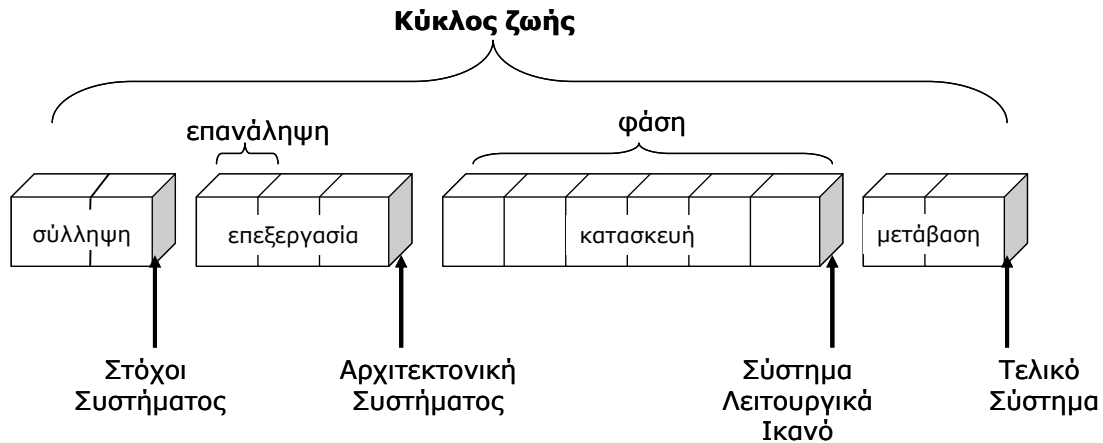
Σχήμα 3.10.: Παράδειγμα WBS για την ανάπτυξη συστήματος λογισμικού.

### Παράδειγμα 3.2

Θα παρουσιάσουμε τη δομή WBS για ένα πιο σύνθετο έργο, το οποίο ακολουθεί την ενοποιημένη προσέγγιση (unified process). Το παράδειγμα αυτό μπορεί να χρησιμοποιηθεί ως πρότυπο για τη δημιουργία της δομής WBS για όλα τα έργα που ακολουθούν την ενοποιημένη προσέγγιση. Η συνοπτική παρουσίαση των τεσσάρων φάσεων της δόθηκε στο Κεφάλαιο 1. Εκεί αναφέραμε τις φάσεις της σύλληψης (inception), της λεπτομερούς επεξεργασίας (elaboration), της κατασκευής (construction) και της μετάβασης (transition).

Σε κάθε φάση μπορεί να εκτελεστεί ένας αριθμός επαναλήψεων, όπου σε κάθε επανάληψη μπορεί να εκτελείται ένας αριθμός διαδικασιών. Κάθε διαδικασία περιγράφεται με ένα σύνολο ροών εργασίας (workflows). Αφετηρία αποτελεί ο γενικός κύκλος ζωής ενός έργου, συνεχίζουμε με τις φάσεις, όπου σε κάθε φάση εκτελούμε επαναλήψεις διαδικασιών και οι διαδικασίες με τη σειρά τους περιγράφονται με ροές εργασιών.

Στο Σχήμα 3.11 παρουσιάζονται με γραφικό τρόπο οι βασικές έννοιες του κύκλου ζωής και των επαναλήψεων που εκτελούνται (Royce, 1988).



**Σχήμα 3.11.:** Βασικές έννοιες του κύκλου ζωής στην ενοποιημένη προσέγγιση.

Μια ροή εργασίας είναι μια λογική ομαδοποίηση ενός συνόλου δραστηριοτήτων που λαμβάνουν χώρα κατά την ανάπτυξη ενός συστήματος και περιγράφει το ρόλο καθενός που συμμετέχει στις δραστηριότητες ενός έργου ανάπτυξης λογισμικού, τις δραστηριότητες αυτές καθαυτές, καθώς και τα παραγόμενα (artifacts) των δραστηριοτήτων. Τα παραγόμενα στην περίπτωση μιας διαδικασίας ανάπτυξης λογισμικού μπορεί να είναι παραδοτέα (deliverables), έγγραφα, κώδικας, πρότυπα ή οτιδήποτε άλλο μπορεί να παράγει μια δραστηριότητα (Γερογιάννης, 2006).

Για την οργάνωση της δομής WBS θα πρέπει να ακολουθήσουμε τους παρακάτω κανόνες:

- Τα στοιχεία του πρώτου επιπέδου της δομής WBS είναι οι ροές εργασιών. Ενδεικτικά αναφέρονται οι ροές εργασιών της διαχείρισης έργου, του ορισμού του περιβάλλοντος του έργου, της καταγραφής των απαιτήσεων, της ανάλυσης και σχεδίασης, της υλοποίησης, του ελέγχου, κ.ά.
- Τα στοιχεία του δεύτερου επιπέδου της δομής WBS είναι οι φάσεις της ενοποιημένης προσέγγισης, δηλαδή οι φάσεις της σύλληψης, της λεπτομερούς επεξεργασίας, της κατασκευής και της μετάβασης σε λειτουργία. Τα στοιχεία αυτού του επιπέδου επιτρέπουν στο πλάνο του έργου να εξελίσσεται και να προσαρμόζεται ανάλογα με τις απαιτήσεις που προκύπτουν.
- Τα στοιχεία του τρίτου επιπέδου της δομής WBS είναι οι δραστηριότητες που παράγουν τα παραδοτέα του έργου. Στο επίπεδο αυτό μπορούμε να υπολογίσουμε το κόστος παραγωγής καθενός εκ των παραδοτέων και για κάθε φάση του έργου. Εναλλακτικά, μπορούμε να αναλύσουμε τις δραστηριότητες με ακόμη μεγαλύτερη λεπτομέρεια εισάγοντας περισσότερα επίπεδα στη δομή WBS.

Μια ενδεικτική δομή WBS για ένα έργο που ακολουθεί την ενοποιημένη προσέγγιση παρουσιάζεται στον παρακάτω πίνακα:

A	Διαχείριση έργου
AA	Φάση της σύλληψης
AAA	Ορισμός σκοπού συστήματος
AAB	Προδιαγραφές για τη φάση της λεπτομερούς επεξεργασίας
AAΓ	Δημιουργία δομής WBS για τη φάση της λεπτομερούς επεξεργασίας
AAΔ	Σχεδιασμός ανάπτυξης λογισμικού
AAE	Αξιολόγηση αποτελέσματος φάσης σύλληψης
AB	Φάση της λεπτομερούς επεξεργασίας
ABA	Προδιαγραφές για τη φάση της κατασκευής
ABB	Δημιουργία δομής WBS για τη φάση της κατασκευής
ABΓ	Αξιολόγηση αποτελέσματος φάσης λεπτομερούς επεξεργασίας
AΓ	Φάση της κατασκευής
AΓA	Σχεδίαση μετάβασης συστήματος σε λειτουργία
AΓB	Δημιουργία δομής WBS για τη φάση της μετάβασης
AΓΓ	Αξιολόγηση αποτελέσματος φάσης κατασκευής
AΔ	Φάση της μετάβασης
AΔA	Σχεδιασμός για επόμενη επανάληψη
AΔB	Αξιολόγηση αποτελέσματος φάσης μετάβασης
B	Περιβάλλον έργου
BA	Προδιαγραφές για το περιβάλλον του έργου στη φάση της σύλληψης
BB	Περιβάλλον του έργου για τη φάση της λεπτομερούς επεξεργασίας
BBA	Εγκατάσταση και διαχείριση περιβάλλοντος ανάπτυξης συστήματος
BBB	Ολοκλήρωση περιβάλλοντος ανάπτυξης συστήματος και ειδικών εργαλείων
BBΓ	Δημιουργία βάσης δεδομένων για διαχείριση αλλαγών
BΓ	Συντήρηση περιβάλλοντος έργου για τη φάση της κατασκευής
BΓA	Εγκατάσταση και διαχείριση περιβάλλοντος ανάπτυξης συστήματος
BΓB	Συντήρηση βάσης δεδομένων για διαχείριση αλλαγών
BΔ	Συντήρηση περιβάλλοντος έργου για τη φάση της μετάβασης
BΔA	Εγκατάσταση και διαχείριση περιβάλλοντος ανάπτυξης συστήματος

	BΔB	Συντήρηση βάσης δεδομένων για διαχείριση αλλαγών
	BΔΓ	Συντήρηση περιβάλλοντος πακεταρίσματος λογισμικού (packaging)
Γ		Διαχείριση απαιτήσεων
	ΓΑ	Ανάπτυξη απαιτήσεων για τη φάση της σύλληψης
	ΓΑΑ	Όραμα (vision) συστήματος
	ΓΑΒ	Μοντελοποίηση με περιπτώσεις χρήσης
	ΓΒ	Ανάπτυξη απαιτήσεων για τη φάση της λεπτομερούς επεξεργασίας
	ΓΒΑ	Δημιουργία γραμμής αναφοράς (baseline) για το όραμα του συστήματος
	ΓΒΒ	Δημιουργία γραμμής αναφοράς για τις περιπτώσεις χρήσης του συστήματος
	ΓΓ	Συντήρηση απαιτήσεων κατά τη φάση της κατασκευής
	ΓΔ	Συντήρηση απαιτήσεων κατά τη φάση της μετάβασης
Δ		Σχεδιασμός και ανάλυση
	ΔΑ	Πρωτοτυποποίηση αρχιτεκτονικής κατά τη φάση της σύλληψης
	ΔΒ	Αρχιτεκτονική συστήματος κατά τη φάση της λεπτομερούς επεξεργασίας
	ΔΒΑ	Σχεδίαση αρχιτεκτονικής συστήματος
	ΔΒΒ	Παρουσίαση αρχιτεκτονικής συστήματος
	ΔΒΓ	Τεκμηρίωση αρχιτεκτονικής συστήματος
	ΔΓ	Φάση της κατασκευής – σχεδίαση συστήματος
	ΔΓΑ	Συντήρηση μοντέλου αρχιτεκτονικής συστήματος
	ΔΓΒ	Σχεδίαση συστατικών (components)
	ΔΔ	Φάση της μετάβασης – συντήρηση μοντέλου σχεδίασης συστήματος
Ε		Υλοποίηση
	ΕΑ	Φάση της σύλληψης – πρωτοτυποποίηση
	ΕΒ	Φάση της λεπτομερούς επεξεργασίας – υλοποίηση συστατικών
	ΕΒΑ	Ολοκλήρωση των «κρίσιμων» συστατικών
	ΕΓ	Φάση της κατασκευής
	ΕΓΑ	Αρχική έκδοση του συστήματος και έλεγχος
	ΕΓΒ	Έκδοση alpha του συστήματος και έλεγχος
	ΕΓΓ	Έκδοση beta του συστήματος και έλεγχος
	ΕΔ	Φάση της μετάβασης – συντήρηση συστατικών
Z		Έλεγχος
	ΖΑ	Φάση της σύλληψης – σχεδιασμός του ελέγχου

εκδόσεων	ZB	Φάση της λεπτομερούς επεξεργασίας
	ZBA	Μοντέλο ελέγχων
	ZBB	Σενάριο ελέγχου αρχιτεκτονικής
	ZBΓ	Παρουσίαση του μοντέλου ελέγχων και περιγραφή των
	ZΓ	Φάση της κατασκευής
	ZΓΑ	Έλεγχος αρχικής έκδοσης του συστήματος
	ZΓΒ	Έλεγχος alpha έκδοσης του συστήματος
	ZΓΓ	Έλεγχος beta έκδοσης του συστήματος
	ZΔ	Φάση της μετάβασης
	ZΔΑ	Αξιολόγηση τελικής έκδοσης συστήματος

Η παραπάνω δομή WBS είναι ενδεικτική και δείχνει πως από μια σύνθετη μεθοδολογία, όπως η ενοποιημένη προσέγγιση, μπορούμε να δημιουργήσουμε ένα πλάνο για την εκτέλεση του έργου. Σε κάθε περίπτωση, η παραπάνω δομή προσαρμόζεται στις ανάγκες του κάθε έργου, αφού λάβουμε υπόψη τις παρακάτω παραμέτρους:

- *Μέγεθος*: Ανάλογα με το μέγεθος του έργου είναι πιθανό να χρειασθεί να προσθέσουμε περισσότερα επίπεδα ανάλυσης.
- *Οργανωτική δομή*: Έργα στα οποία εμπλέκονται υπεργολάβοι ή εκτελούνται σε περισσότερες από μια γεωγραφικές τοποθεσίες εισάγουν περιορισμούς που οδηγούν σε διαφορετική κατάτμηση του έργου.
- *Είδος έργου*: Σε ένα έργο επιχειρηματικού ανασχεδιασμού (business re-engineering) δίνεται έμφαση στην καταγραφή και ανάλυση απαιτήσεων και όχι στη σχεδίαση και υλοποίηση του συστήματος. Σε αυτή την περίπτωση, έμφαση δίνεται στο τμήμα της δομής WBS που αναφέρεται στις απαιτήσεις, σε αντίθεση με ένα έργο ανάπτυξης νέου συστήματος, όπου η έμφαση δίνεται στο τμήμα της σχεδίασης και υλοποίησης. Επιπλέον, έργα που αναφέρονται στην κατασκευή προϊόντος, όπου υπάρχουν πολλοί και διαφορετικοί χρήστες, διαφέρουν σημαντικά από έργα που προορίζονται για χρήση μέσα σε ένα οργανισμό και ιδιαίτερα στη φάση της μετάβασης σε λειτουργία (deployment).
- *Προηγούμενη εμπειρία*. Η προηγούμενη εμπειρία του οργανισμού που εκτελεί το έργο αποτελεί σημαντικό παράγοντα στην κατασκευή της δομής WBS.

Αν και η δομή WBS είναι εξαιρετικά απλή στην κατανόησή της, η δημιουργία της παρουσιάζει δυσκολίες ιδιαίτερα όταν το μέγεθος του έργου είναι μεγάλο. Μερικά από τα προβλήματα που μπορεί να παρουσιαστούν είναι τα εξής:

- Η διάσπαση των εργασιών σε πολύ μικρά κομμάτια δημιουργεί πολυπλοκότητα και αυξάνει το κόστος της διαχείρισης ενός έργου. Συνήθως, ένα πακέτο εργασίας έχει διάρκεια εργασίας μερικών εβδομάδων και περιλαμβάνει προσπάθεια μερικών εκατοντάδων ανθρωπο-ωρών. Στην περίπτωση μεγάλων έργων, η διάσπαση σε πακέτα εργασίας τέτοιου μεγέθους, δημιουργεί μεγάλο αριθμό πακέτων εργασίας, γεγονός που δημιουργεί δυσκολίες στις αναθέσεις εργασιών σε ομάδες εργαζομένων.
- Η δομή WBS διαφέρει από έργο σε έργο και συνεπώς δεν είναι εύκολο να συγκρίνουμε διαφορετικά έργα.

Για να αξιολογήσουμε την πληρότητα της δομής WBS θα πρέπει να πληρούνται τα παρακάτω έξι κριτήρια:

- Να μπορούμε να μετρήσουμε το ποσοστό ολοκλήρωσης της δραστηριότητας.
- Η έναρξη και η λήξη της δραστηριότητας να είναι καλά ορισμένη.
- Η κάθε δραστηριότητα να σχετίζεται με ένα παραδοτέο.
- Η διάρκεια και ο προϋπολογισμός της δραστηριότητας να μπορεί να υπολογισθεί εύκολα.
- Η διάρκεια της δραστηριότητας να είναι αποδεκτή.
- Οι δραστηριότητες να είναι ανεξάρτητες.

### **Δραστηριότητα 3.2**

Δημιουργήστε τη δομή WBS για ένα έργο της επιλογής σας.

α) Σχολιάστε τις επιλογές σας.

β) Αξιολογήστε τις δραστηριότητες σύμφωνα με τα κριτήρια πληρότητας που αναφέρθηκαν παραπάνω.

## **Ενότητα 3.5 Διαχείριση Παραδοτέων.**

Όπως αναφέραμε και στην προηγούμενη ενότητα κάθε δραστηριότητα η οποία ορίζεται στη δομή WBS θα πρέπει να συνδέεται με ένα παραδοτέο (deliverable). Ορίζουμε ως παραδοτέο οποιοδήποτε μοναδικό και επαληθεύσιμο προϊόν ή αποτέλεσμα που πρέπει να παραχθεί προκειμένου να ολοκληρωθεί μία δραστηριότητα, μία φάση ή

ένα έργο. Ο ορισμός των παραδοτέων συνδέεται στενά με τη διαχείριση του αντικειμένου των εργασιών σε ένα έργο, μια και η λίστα των παραδοτέων είναι αυτή που προσδιορίζει με ακρίβεια το αποτέλεσμα του έργου.

Για κάθε παραδοτέο του έργου θα πρέπει να προσδιορίσουμε με ακρίβεια τις παρακάτω πληροφορίες:

- Το όνομα της δραστηριότητας η οποία θα παράγει το παραδοτέο.
- Το πακέτο εργασίας στο οποίο ανήκει η δραστηριότητα.
- Το όνομα του παραδοτέου.
- Τον τίτλο του παραδοτέου.
- Τον τύπο του παραδοτέου. Ο τύπος προσδιορίζει τη μορφή του παραδοτέου που μπορεί να είναι έγγραφο, λογισμικό, υλικό, δεδομένα ή οτιδήποτε άλλο.
- Το αναμενόμενο μέγεθος του παραδοτέου, ώστε να μπορέσουμε να εκτιμήσουμε την προσπάθεια που απαιτείται για την αξιολόγηση των παραδοτέων.
- Την αναμενόμενη ημερομηνία παράδοσης του παραδοτέου.
- Τη διαδικασία αποδοχής του παραδοτέου. Η διαδικασία αποδοχής (acceptance procedure) είναι σύνθετη διαδικασία και καθορίζει τη συνολική προσπάθεια που απαιτείται στο έργο, τόσο από τον ανάδοχο του έργου, όσο και από τον πελάτη. Συνεπώς, είναι ιδιαίτερα χρήσιμο να γνωρίζουμε τις απαιτήσεις της διαδικασίας αποδοχής, τόσο σε χρόνο (elapsed time), όσο και σε συνολική προσπάθεια (effort).
- Το είδος του παραδοτέου. Το είδος του παραδοτέου μπορεί να είναι τελικό ή ενδιάμεσο. Τελικά είναι τα παραδοτέα που αποτελούν τμήμα του τελικού προϊόντος του έργου, ενώ ενδιάμεσα χαρακτηρίζονται τα παραδοτέα που δεν αποτελούν μέρος του τελικού προϊόντος, αλλά είναι απαραίτητα για την παραγωγή του τελικού προϊόντος. Για παράδειγμα, τελικό παραδοτέο είναι ο κώδικας του συστήματος ή η τεκμηρίωση του συστήματος, ενώ ενδιάμεσο παραδοτέο είναι το έγγραφο που περιέχει την ανάλυση του συστήματος.
- Την περιοδικότητα του παραδοτέου. Έχει σχέση με το εάν θα παραχθούν περισσότερες από μια εκδόσεις του παραδοτέου. Για παράδειγμα, οι αναφορές προόδου είναι παραδοτέα που παράγονται σε μηνιαία βάση.

Για την οργάνωση των παραδοτέων δημιουργούμε μια βάση δεδομένων στην οποία αποθηκεύονται όλες οι παραπάνω πληροφορίες, ενώ ταυτόχρονα όλα τα έγγραφα υπόκεινται σε διαχείριση σχηματισμών (configuration management). Το σύνολο της

παραπάνω πληροφορίας έχει επικρατήσει να ονομάζεται πίνακας ιχνηλάτισης παραδοτέων (Deliverable Tracking Matrix). Ένα παράδειγμα αυτού του πίνακα παρουσιάζεται στο Σχήμα 3.8. Βέβαια, σήμερα σε πολλές περιπτώσεις η διαχείριση των παραδοτέων γίνεται με τη χρήση συστημάτων διαχείρισης έργων.

WP	Δραστηριότητα	Όνομα παραδοτέου	Τίτλος παραδοτέου	Τύπος	Προβλεπόμενο μέγεθος	Περιοδικότητα	Πραγματικό μέγεθος	Ημερομηνία υποβολής για έλεγχο	Κύκλος ελέγχου	Ημερομηνία αποδοχής
WP0	WP0.1	XYZ-PMP-VNNNN	Πλάνο Διαχείρισης Έργου	DOC	150	3	450	7/2/2006	7-5-3	28/2/2006
WP0	WP0.2	XYZ-SCHEDULE-VNNNN	Χρονοδιάγραμμα Έργου	DOC	1 file	17	17	Μηνιαίο. Πρώτη Δευτέρα κάθε μήνα	7-5-3	Επόμενη σύσκεψη
WP0	WP0.3	XYZ-PMP-DTMVNNNN	Πίνακας Ιχνηλάτισης Παραδοτέων	DOC	This file	17	17	Μηνιαίο. Πρώτη Δευτέρα κάθε μήνα	7-5-3	Επόμενη σύσκεψη
WP0	WP0.4	XYZ-MOM-SCNN-VNNNN	Πρακτικά σύσκεψης διοικητικής επιτροπής έργου	DOC	10	17	170	Μηνιαίο και δύο ημέρες μετά την σύσκεψη	7-5-3	Επόμενη σύσκεψη
WP0	WP0.5	XYZ-MOM-PGNN-VNNNN	Πρακτικά σύσκεψης τεχνικής επιτροπής έργου	DOC	10	35	350	Μηνιαίο και δύο ημέρες μετά την σύσκεψη	7-5-3	Επόμενη σύσκεψη
WP0	WP0.6	XYZ-MPR-AMVY-VNNNN	Μηνιαία αναφορά προόδου	DOC	10	17	170	Μηνιαίο. Πρώτη Δευτέρα κάθε μήνα	7-5-3	Επόμενη σύσκεψη
WP0	WP0.7	XYZ-MOM-QANN-VNNNN	Πρακτικά σύσκεψης QA επιτροπής έργου	DOC	10	20	200	Κατά περίπτωση	7-5-3	Επόμενη σύσκεψη
WP0	WP0.8	XYZ-QRAN-VNNNN	Αναφορά έλεγχου ποιότητας	DOC	20	1	20	8/11/2006	7-5-3	29/11/2006
WP0	WP0.9	XYZ-PFR-VNNNN	Τέλικη αναφορά έργου	DOC	50	1	50	31/7/2003	7-5-3	21/8/2003
WP0	WP0.10	XYZ2-RMP-V0100-01	Πλάνο Διαχείρισης κινδύνου	DOC	20	2	40	16/4/2006	7-5-3	
WP0	WP0.11	XYZ-CMR-NN-VNNNN	Πλάνο Διαχείρισης διαμορφώσεων	DATA				Κατά περίπτωση	7-5-3	Κατά περίπτωση
WPI	WPI.1	XYZ-SWC-ESP	Πρωτότυπο συστήματος	SW				29/4/2006	Δεν απαιτείται	

**Σχήμα 3.12.:** Παράδειγμα πίνακα ιχνηλάτισης παραδοτέων.

Ιδιαίτερα σημαντικό ρόλο έχει η ονοματολογία των παραδοτέων που χρησιμοποιείται στο έργο, ώστε το κάθε παραδοτέο να προσδιορίζεται με μοναδικό τρόπο. Η ονοματολογία των αρχείων σε ένα έργο βοηθάει στην καλύτερη οργάνωση των παραδοτέων του έργου και στην αποτελεσματικότερη επικοινωνία μεταξύ του πελάτη και της ομάδας του έργου. Ειδικά σήμερα που όλα τα έγγραφα συντάσσονται ηλεκτρονικά η ανάγκη αυτή είναι ιδιαίτερα μεγάλη. Για παράδειγμα, σε ένα μεγάλο έργο μπορεί να χρησιμοποιηθεί το παρακάτω σχήμα ονοματολογίας.

<ΟΝΟΜΑ ΕΡΓΟΥ>-<ΕΙΔΟΣ ΕΓΓΡΑΦΟΥ>-<ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΕΓΓΡΑΦΟΥ>-  
V<ΕΚΔΟΣΗ.ΑΝΑΘΕΩΡΗΣΗ>

Το «είδος του εγγράφου» είναι μια συντομογραφία που μας επιτρέπει να καταλάβουμε αμέσως το είδος του εγγράφου. Για παράδειγμα, θα μπορούσε να είναι ένα από τα παρακάτω:

ΠΡΣ	Για τα πρακτικά σύσκεψης
ΠΔΕ	Για το Πλάνο Διαχείρισης Έργου
ΜΑΠ	Για τη Μηνιαία Αναφορά Προόδου
ΑΝΑ	Για τα έγγραφα ανάλυσης του συστήματος
ΣΧΕ	Για τα έγγραφα σχεδίασης του συστήματος



Στην περίπτωση που ένα έγγραφο είναι περιοδικό ή έχει πολλά τμήματα που αποτελούν ξεχωριστά έγγραφα, το «είδος του εγγράφου» δεν επαρκεί από μόνο του για να προσδιορίσει επακριβώς το έγγραφο. Στην περίπτωση αυτή χρησιμοποιούμε έναν επιπλέον προσδιορισμό. Για παράδειγμα, το είδος του εγγράφου «ΜΑΠ» του έργου είναι ένα έγγραφο που παράγεται κάθε μήνα. Στην περίπτωση αυτή ο προσδιορισμός του εγγράφου θα ήταν ο μήνας και το έτος στον οποίο αναφέρεται η αναφορά προόδου. Αντίστοιχα, ένα έγγραφο τύπου «ΑΝΑ» ή «ΣΧΕ» μπορεί να συνδυαστεί, για παράδειγμα, με τους παρακάτω προσδιορισμούς:

ΑΠΑ	Απαιτήσεις έργου
ΠΧ	Περιπτώσεις χρήσης
ΔΔΡ	Διαγράμματα δραστηριοτήτων
ΔΚΛ	Διαγράμματα κλάσεων

Το όνομα του εγγράφου κλείνει με την έκδοση (version) του εγγράφου.

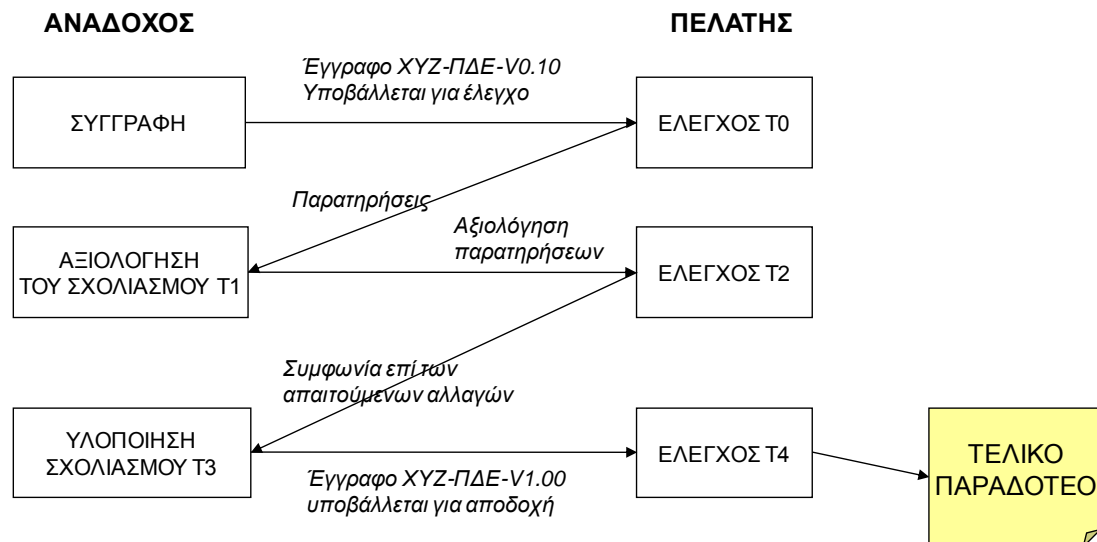
### **Παράδειγμα 3.3**

XYZ-ΜΑΠ-0207-V0.10	Είναι η έκδοση 0.10 (πρόχειρη) της μηνιαίας αναφοράς προόδου για το μήνα Φεβρουάριο του 2007 και για το έργο XYZ.
XYZ-ΠΔΕ-V0.10	Είναι η έκδοση 0.10 (πρόχειρη) του πλάνου διαχείρισης έργου του έργου XYZ.
XYZ-ΑΝΑ-ΠΧ-V1.10	Είναι η έκδοση 1.10 των περιπτώσεων χρήσης του έργου XYZ. Συνήθως, κάνουμε την παραδοχή ότι η έκδοση 1.00 είναι η πρώτη αποδεκτή από τον πελάτη του έργου έκδοση του εγγράφου. Συνεπώς, η έκδοση του εγγράφου 1.10 είναι μια επόμενη έκδοση η οποία περιέχει αλλαγές και τροποποιήσεις σε σχέση με την 1.00 έκδοση του συστήματος.

Στο Σχήμα 3.13 παρουσιάζεται ο κύκλος ζωής μιας τεχνικής αναφοράς. Ο ανάδοχος του έργου γράφει την τεχνική αναφορά και την υποβάλλει στον πελάτη προς τελική έγκριση. Η τεχνική αναφορά υποβάλλεται με αριθμό έκδοσης V0.10. Ο πελάτης ελέγχει την έκθεση και μέσα σε χρόνο T0 δίνει τα σχόλιά του στον ανάδοχο. Ο χρόνος T0 προς

έλεγχο των παραδοτέων ποικίλει και εξαρτάται από το μέγεθος του παραδοτέου, το είδος του παραδοτέου, κ.λπ.

Στη συνέχεια, ο ανάδοχος αξιολογεί τις παρατηρήσεις του πελάτη και κάνει τις αντιπροτάσεις του μέσα σε χρονικό διάστημα T1. Ο πελάτης μαζί με τον ανάδοχο συζητάνε τις παρατηρήσεις και τις απαιτούμενες αλλαγές και ο ανάδοχος προχωρά στην υλοποίηση των σχολίων. Το έγγραφο υποβάλλεται στον πελάτη για τελική έγκριση.



Σχήμα 3.13.: Διαδικασία αποδοχής παραδοτέου.

### Άσκηση αυτοαξιολόγησης 3.1

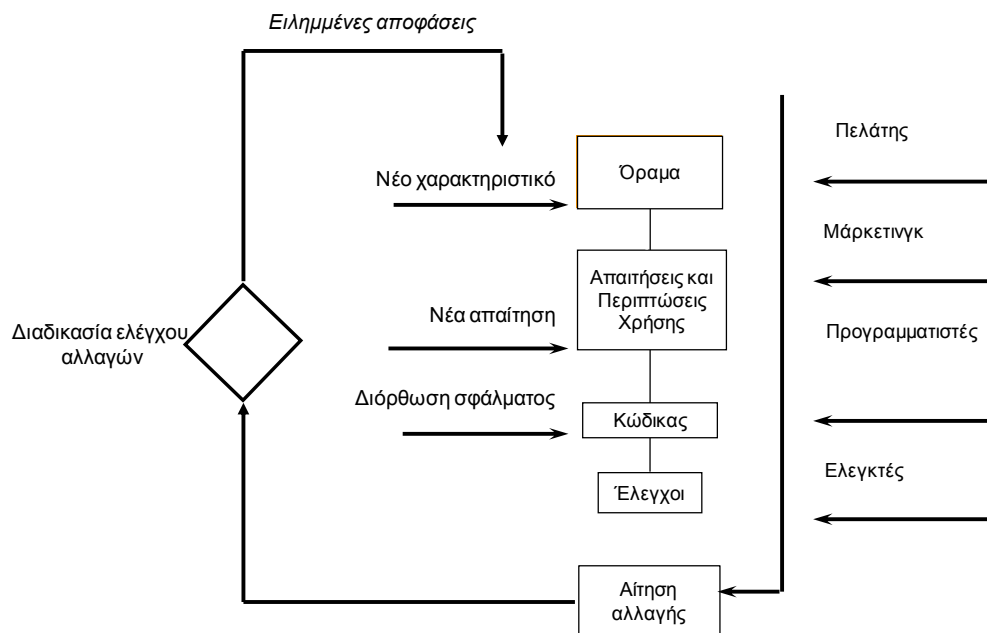
Συμπληρώστε στον παρακάτω πίνακα τις περιγραφές των παραδοτέων.

XYZ-ΜΑΠ-0207-V1.00	
XYZ-ΑΝΑ-ΔΚΛ-V1.00	
XYZ-ΣΧΕ-ΔΚΛ-V1.00	

## Ενότητα 3.6 Έλεγχος Αλλαγών.

Είναι κανόνας ότι κατά τη διάρκεια ενός έργου θα συμβούν αλλαγές. Οι αλλαγές συμβαίνουν γιατί το περιβάλλον του έργου άλλαξε, οι συνθήκες της αγοράς άλλαξαν, οι ανάγκες έχουν αλλάξει, υπήρξαν τεχνικές δυσκολίες, παρουσιάστηκαν καινούρια προϊόντα στην αγορά που κάνουν τη δουλειά με διαφορετικό τρόπο ή για πολλούς άλλους λόγους. Όσο πιο μεγάλη διάρκεια έχει ένα έργο, τόσο περισσότερες αλλαγές είναι

δυνατό να συμβούν. Όσο πιο ασαφές είναι το έργο στο ξεκίνημά του, τόσο πιο επιρρεπές στην αλλαγή είναι. Οι αλλαγές μπορεί να προέρχονται από τους πελάτες ή χρήστες του έργου, το περιβάλλον του έργου (μάρκετινγκ), την ίδια την ομάδα του έργου ή από σφάλματα που έχουν εντοπισθεί στο ίδιο το υπό ανάπτυξη σύστημα. Αντίστοιχα, οι αλλαγές μπορεί να επηρεάζουν τα χαρακτηριστικά του συστήματος, τις απαιτήσεις, τον κώδικα, κ.λπ. Ο μόνος τρόπος για την επιτυχή διαχείριση των αλλαγών είναι η ανάπτυξη ενός ολοκληρωμένου συστήματος ελέγχου αλλαγών. Αναφέρουμε τον όρο ολοκληρωμένο σύστημα αλλαγών γιατί μια αλλαγή στο φυσικό αντικείμενο ενός έργου επιφέρει αλλαγές στον προϋπολογισμό του έργου ή στη διάρκειά του. Συνεπώς, ο μόνος τρόπος για να διαχειρισθούμε τις αλλαγές σε ένα έργο είναι λαμβάνοντας υπόψη μας όλες τις παραμέτρους – διαστάσεις ενός έργου. Το Σχήμα 3.14 παρουσιάζει με γραφικό τρόπο τη διαδικασία ελέγχου αλλαγών.



**Σχήμα 3.14.:** Η διαδικασία ελέγχου αλλαγών.

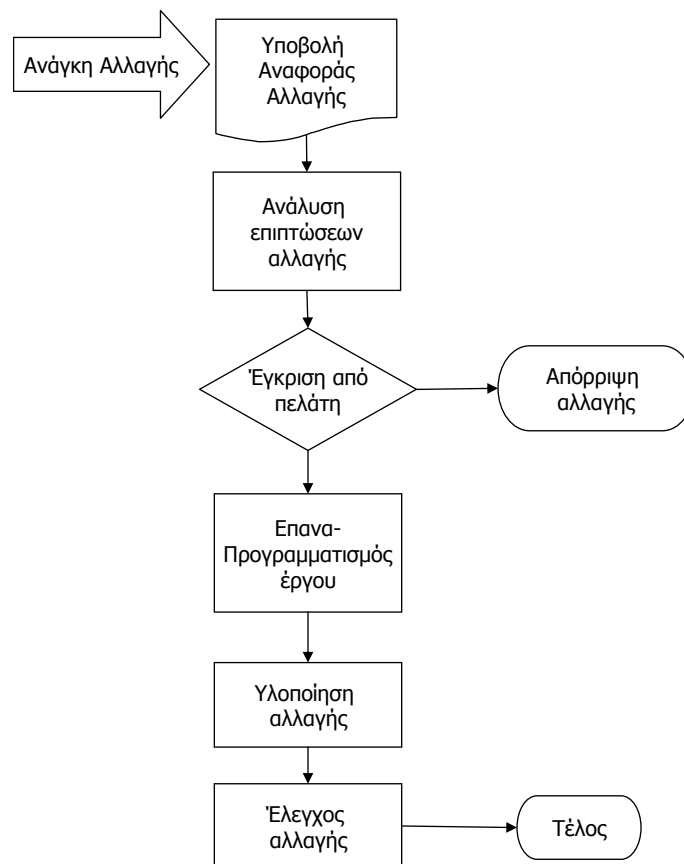
Βασικό σημείο για το χαρακτηρισμό ενός γεγονότος ως αλλαγή είναι οι συμβατικές υποχρεώσεις του αναδόχου του έργου, όπως αυτές περιγράφονται στη σύμβαση του έργου. Η σύμβαση του έργου οφείλει να περιγράψει με σαφή και ξεκάθαρο τρόπο το αντικείμενο του έργου. Κάθε απόκλιση από τις συμβατικές υποχρεώσεις μπορεί να θεωρηθεί ως αλλαγή. Συνεπώς, αλλαγές μπορεί να προκληθούν, όχι μόνο από τυχαίες μεταβολές των επιχειρηματικών διαδικασιών, των στόχων ή των προτεραιοτήτων του πελάτη, αλλά και από τη συμμετοχή και εκτέλεση δραστηριοτήτων που δεν περιγράφονται στη σύμβαση έργου, από την τροποποίηση του αντικειμένου εργασιών

του έργου με πρόσθεση ή αφαίρεση νέων χαρακτηριστικών ή από την πραγμάτωση κινδύνων του έργου.

Στο Σχήμα 3.15 παρουσιάζεται μια απλή διαδικασία διαχείρισης αλλαγών κατά την οποία οι αλλαγές καταγράφονται αρχικά σε μια αίτηση αλλαγής (change request form). Στη συνέχεια γίνεται μια ανάλυση των επιπτώσεων της αλλαγής στο έργο. Η αιτούμενη αλλαγή εκτιμάται ως προς την αναγκαιότητα, τη σημασία, την προτεραιότητα και την τυχόν επίπτωση που μπορεί να έχει στο κόστος, στο χρονοδιάγραμμα, στην ποιότητα και στα παραδοτέα του έργου.

Τα αποτελέσματα αυτής της ανάλυσης υποβάλλονται προς έγκριση στην ομάδα διοίκησης του έργου. Στην περίπτωση που η προτεινόμενη αλλαγή επηρεάζει το κόστος ή/και τις ημερομηνίες ολοκλήρωσης των ορόσημων η/και οποιοδήποτε όρο της σύμβασης του έργου, αρμόδιος να αποφασίσει είναι ο πελάτης του έργου. Στην απόφαση συμφωνείται η έκδοση του συστήματος στην οποία θα ενσωματωθεί η αλλαγή.

Μετά τη γραπτή έγκριση του πελάτη προγραμματίζεται η υλοποίηση της αλλαγής με την κατάλληλη τροποποίηση του πλάνου του έργου.



**Σχήμα 3.15.:** Η διαδικασία έγκρισης αλλαγών.

Ο ολοκληρωμένος έλεγχος αλλαγών απαιτεί τη δημιουργία ενός συστήματος παρακολούθησης των αλλαγών. Στο σύστημα αυτό κάθε αλλαγή πρέπει να καταγράφεται με αναλυτικό τρόπο με τη χρήση αναφορών αλλαγής. Μια αίτηση αλλαγής, στη γενική περίπτωση, περιέχει:

- Περιγραφή της προτεινόμενης αλλαγής.
- Κατηγοριοποίηση της αλλαγής σύμφωνα με το σύστημα που χρησιμοποιείται στο έργο. Κριτήρια κατηγοριοποίησης μπορεί να είναι η σημαντικότητα, το μέγεθος, το πόσο επείγουσα είναι, κ.λπ.
- Ανάλυση των επιπτώσεων της αλλαγής σε οικονομικό, χρονικό, ποιοτικό και τεχνικό επίπεδο.
- Κίνδυνοι που σχετίζονται με την αλλαγή.
- Τεκμηρίωση που απαιτείται για την αλλαγή.
- Κριτήρια αποδοχής της αλλαγής.
- Εγκρίσεις για την υλοποίηση της αλλαγής από τον πελάτη και από τη διοίκηση του έργου.

Στη γενική περίπτωση, η κατηγοριοποίηση των αλλαγών μπορεί να γίνει χρησιμοποιώντας τις παρακάτω κατηγορίες:

Τύπου 0	Σοβαρό σφάλμα του συστήματος. Το σύστημα δεν μπορεί να εκτελέσει μια από τις βασικές του λειτουργίες. Αυτού του είδους οι αλλαγές προκύπτουν κατά τη διάρκεια του ελέγχου του συστήματος και υλοποιούνται πριν την παράδοση του συστήματος.
Τύπου 1	Σφάλμα του συστήματος. Πιο συγκεκριμένα, σφάλμα στη λειτουργικότητα του συστήματος, η οποία δεν μας εμποδίζει στη βασική του λειτουργία ή υπάρχει εναλλακτικός τρόπος λειτουργίας.
Τύπου 2	Βελτίωση του συστήματος. Αλλαγή που προκύπτει από αίτηση για βελτίωση της χρηστικότητας, απόδοσης, κ.λπ. του συστήματος.
Τύπου 3	Αλλαγή απαιτήσεων. Νέες απαιτήσεις χρηστών ή τροποποίηση υπάρχουσας λειτουργικότητας.
Τύπου 4	Άλλου είδους αλλαγές. Πρόκειται για αλλαγές που δεν ανήκουν σε καμία από τις παραπάνω κατηγορίες.

### **Παράδειγμα 3.4**

Ο παρακάτω πίνακας παρουσιάζει παραδείγματα αλλαγών για το έργο «ανάπτυξη συστήματος ελέγχου εναερίου κυκλοφορίας».

<b>Είδος αλλαγής</b>	<b>«Ανάπτυξη συστήματος ελέγχου εναερίου κυκλοφορίας»</b>
Τύπου 0	Απώλεια δεδομένων ή κατάσταση «αδιεξόδου» (deadlock).
Τύπου 1	Χρόνος απόκρισης μεγαλύτερος από 0,2 sec.
Τύπου 2	Βελτίωση των μηνυμάτων που εμφανίζονται στις οθόνες των χειριστών.
Τύπου 3	Αύξηση της δυνατότητας δρομολόγησης πτήσεων από 1200 σε 2400.
Τύπου 4	Αναβάθμιση λογισμικού συστήματος διαχείρισης βάσης δεδομένων.

### **Άσκηση αυτοαξιολόγησης 3.2**

Χαρακτηρίστε την κάθε αλλαγή ανάλογα με τον τύπο της για το έργο της «ανάπτυξης επιχειρηματικής εφαρμογής».

<b>Είδος αλλαγής</b>	<b>«Ανάπτυξη επιχειρηματικής εφαρμογής»</b>
	Τα δεδομένα παρουσιάζονται ομαδοποιημένα ανά τύπο. Πατώντας το σύμβολο (+) δίπλα στο όνομα της ομάδας το σύστημα παρουσιάζει όλα τα ομαδοποιημένα δεδομένα αναλυτικά. Η αντίστροφη λειτουργία, δηλαδή η σύμπτυξη των δεδομένων της ομάδας δεν λειτουργεί κανονικά.
	Απώλεια δεδομένων.
	Μεταφορά της εφαρμογής στη νέα έκδοση του φυλλομετρητή.
	Χρήση χρωματικού κώδικα για την παρουσίαση «νέων» και «παλαιών δεδομένων».
	Βελτίωση του σχήματος αποθήκευσης δεδομένων με σκοπό τη γρήγορη εύρεση του ιστορικού πελατών.

Μια βασική διαδικασία για την ολοκληρωμένη διαχείριση αλλαγών είναι η διαχείριση σχηματισμών (configuration management). Η διαχείριση σχηματισμών (Humphrey,

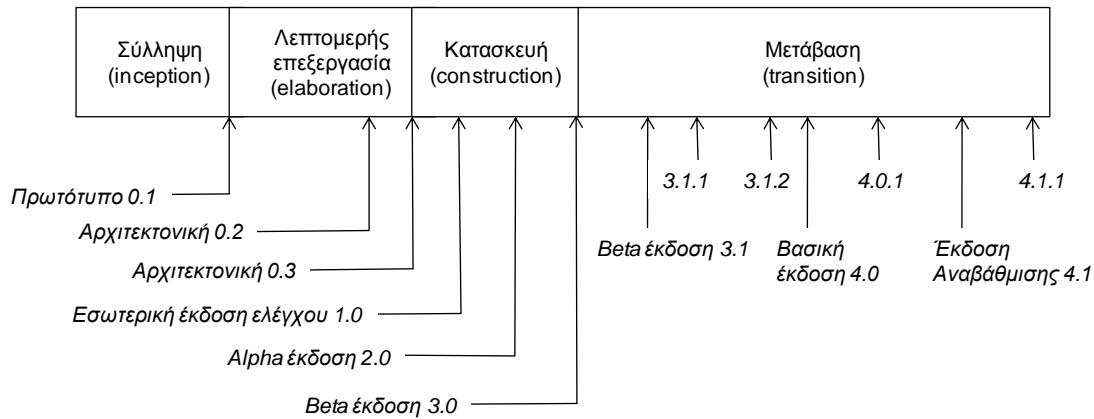
1989) είναι μια διαδικασία καταγραφής και τεκμηρίωσης των αλλαγών του συστήματος με τη χρήση γραμμών αναφοράς (baselines), η οποία :

1. Παρακολουθεί και καταγράφει τα λειτουργικά και φυσικά χαρακτηριστικά ενός αντικειμένου και ενός συστήματος, καθώς αυτό εξελίσσεται μέσα στο χρόνο.
2. Ελέγχει και καταγράφει τις αλλαγές αυτών των χαρακτηριστικών μέσα στο χρόνο.
3. Μπορεί να παρουσιάσει ανά πάσα στιγμή και με λεπτομέρεια την ιστορία εξέλιξης του συστήματος, καθώς και την παρούσα κατάσταση.
4. Ελέγχει ότι όλα τα επιμέρους στοιχεία ενός συστήματος ακολουθούν τους κανόνες.

Τα παραδοτέα τα οποία υπόκεινται σε διαχείριση σχηματισμών είναι συνήθως τα έγγραφα του έργου, ο κώδικας του συστήματος, τα σχέδια και γενικότερα όλα τα στοιχεία του έργου που μπορούμε να διαχειρισθούμε με ηλεκτρονικό τρόπο. Η διαχείριση σχηματισμών αποκτά ιδιαίτερη βαρύτητα σε έργα με αυξημένη πολυπλοκότητα, σε έργα με πολλά υποσυστήματα, καθώς και σε έργα που περιλαμβάνουν συστήματα με μεγάλη διάρκεια ζωής.

Η συνολική κατάσταση του συστήματος που υπόκειται σε διαχείριση σχηματισμών συνήθως προσδιορίζεται από έναν αριθμό της μορφής N.X.Y, όπου το N προσδιορίζει τη βασική αποδέσμευση (major release) ή έκδοση (version), το X την υπο-αποδέσμευση (minor release), ενώ το Y την ενδιάμεση αποδέσμευση (interim release). Μια βασική αποδέσμευση προσδιορίζει μια νέα έκδοση του συστήματος, μια υπο-αποδέσμευση προσδιορίζει μια έκδοση του συστήματος με νέα βελτιωμένα χαρακτηριστικά, ενώ μια ενδιάμεση αποδέσμευση μια έκδοση του συστήματος με διορθωμένη λειτουργικότητα.

Στο Σχήμα 3.16 παρουσιάζεται η ιστορία των αποδεσμεύσεων για ένα έργο ανάπτυξης λογισμικού το οποίο ακολουθεί την ενοποιημένη προσέγγιση.



**Σχήμα 3.16.:** Παράδειγμα αποδεσμεύσεων έργου ανάπτυξης προϊόντος λογισμικού.

Το Κεφάλαιο 5 του τόμου Β της Θ.Ε. ΠΛΗ24 του Προγράμματος Σπουδών «Πληροφορική» του ΕΑΠ παρουσιάζει σε μεγαλύτερη ανάλυση τα θέματα της διαχείρισης σχηματισμών.

## ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

Στο κεφάλαιο αυτό παρουσιάσαμε τις βασικές έννοιες της διαχείρισης αντικειμένου εργασιών ενός έργου σε συνδυασμό με τη διαχείριση απαιτήσεων. Στη συνέχεια, παρουσιάσαμε τη δομή ανάλυσης εργασιών, η οποία είναι ιδιαίτερα χρήσιμη για τη διαχείριση των έργων, αφού παρέχει πληροφορίες για το αντικείμενο των εργασιών ενός έργου, τις εργασίες που πρέπει να γίνουν, έτσι ώστε να αναπτύξουμε ένα παραδοτέο, τις αναθέσεις εργασιών σε εργαζομένους, την ιεραρχία των παραδοτέων, την κοστολόγηση των παραδοτέων και τη χρονοδρομολόγηση των εργασιών, ενώ γενικότερα μας διευκολύνει στην παρακολούθηση του έργου. Τέλος, παρουσιάσαμε τη διαχείριση των παραδοτέων ενός έργου σε συνδυασμό με τη διαχείριση αλλαγών.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Brooks, F., (1975). The Mythical Man Month: Essays on Software Engineering. Addison-Wesley.

Dorfman, M., και Thayer P., (1990). Standards, Guidelines, and Examples of System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press.

Greene, J., Stellman, A., (2005). Applied Software Project Management, O'Reilly.

Humphrey, W., (1989). Managing the Software Process. Addison-Wesley.



Jalotte, P., (2002). *Software Project Management in Practice*. Addison-Wesley.

Leffingwell, D., Widrig, D., (2003). *Managing Software Requirements: A Use Case Approach, Second Edition*. Addison Wesley.

PMI Institute, (2001). *Practice Standard for Work Breakdown Structures*. Project Management Institute.

PMI Institute, (2004). *A Guide to the Project Management Body of Knowledge*. PMI Standard Committee.

Royce, W., (1998). *Software Project Management: A Unified Framework*. Addison-Wesley.

Γερογιάννης, Β., Κακαρόντζας, Γ., Καμέας, Α., Σταμέλος, Γ., Φιτσιλής, Π., (2006). *Αντικειμενοστρεφής Ανάπτυξη Λογισμικού με τη UML*. Εκδόσεις Κλειδάριθμος.

## **ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΕΡΓΩΝ**

**Συγγραφέας Πάνος Φιτσιλής**

### **Σκοπός:**

Σκοπός του Κεφαλαίου 4 είναι να παρουσιάσουμε κάποιες από τις βασικές αρχές σχεδιασμού έργων. Παρουσιάζουμε το βασικό παραδοτέο της σχεδίασης ενός έργου, το πλάνο διαχείρισης έργων, καθώς και βασικές έννοιες για τη διαχείριση αλλαγών. Συνεχίζουμε με το χρονοπρογραμματισμό έργων αναλύοντας τις μεθόδους CPM/PERT, οι οποίες αποτελούν τις γνωστότερες και πλέον διαδεδομένες μεθόδους χρονοπρογραμματισμού έργων. Παρουσιάζουμε αναλυτικά παραδείγματα για την κατασκευή του δικτύου ενός έργου, τον υπολογισμό της κρίσιμης διαδρομής, τον υπολογισμό της πιθανότητας ολοκλήρωσης έργου σε μια δεδομένη χρονική στιγμή και τη συντόμευση της διάρκειας έργου.

Επιπλέον, στο κεφάλαιο παρουσιάζονται οι βασικές έννοιες διαχείρισης πόρων και παρουσιάζεται αναλυτικά ένα παράδειγμα εξισορρόπησης πόρων.

Το κεφάλαιο κλείνει με μια σύντομη εισαγωγή στη διαχείριση έργων με τη μέθοδο της κρίσιμης αλυσίδας.

### **Προσδοκώμενα Αποτελέσματα:**

Στο κεφάλαιο αυτό:

- Θα μάθετε τις βασικές έννοιες του σχεδιασμού έργων.
- Θα κατανοήσετε τη δομή του πλάνου διαχείρισης ενός έργου.
- Θα μάθετε τις βασικές έννοιες του χρονοπρογραμματισμού του έργου και της δικτυακής ανάλυσης έργων.
- Θα μάθετε να εφαρμόζετε τις μεθόδους CPM/PERT.
- Θα μάθετε για τη συντόμευση των δραστηριοτήτων ενός έργου.
- Θα μάθετε για τη διαχείριση πόρων και την εξισορρόπηση αυτών.
- Θα μάθετε για τη διαχείριση έργων με τη μέθοδο της κρίσιμης αλυσίδας, για τα αποθέματα χρόνου και τους απομονωτές.

### **Έννοιες κλειδιά:**

Πλάνο διαχείρισης έργου (Project management Plan, PMP).

Χρονοπρόγραμμα έργου (project schedule).

Μέθοδος CPM/PERT.

Διάγραμμα Gantt.

Κρίσιμη διαδρομή (critical path).  
Κρίσιμη αλυσίδα (critical chain).  
Διαχείριση πόρων (resource management).  
Εξισορρόπηση πόρων (resource leveling).

## ΕΝΟΤΗΤΑ 4.1 Εισαγωγή

---

*Σε κάθε εργασία σκεφτείτε τι προηγείται και τι ακολουθεί· στη συνέχεια εκτελέστε την.*

*Επίκτητος, 50-138 μ.Χ., Έλληνας στωικός φιλόσοφος.*

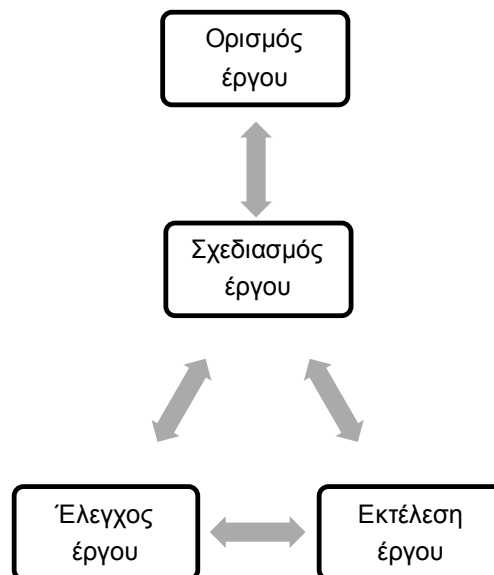
Ακόμη και τα πιο μικρά έργα απαιτούν σχεδιασμό και έλεγχο για την επιτυχή ολοκλήρωσή τους. Ο σχεδιασμός ενός έργου ως λειτουργία έχει ως στόχο τον ορισμό των προτύπων, των διαδικασιών και των μεθόδων που είναι απαραίτητα ώστε να επιτύχουμε τους στρατηγικούς στόχους του οργανισμού σε σχέση με το έργο. Επομένως, μια από τις βασικές αρμοδιότητες του διοικητή έργου είναι ο σχεδιασμός (planning), η ολοκλήρωση (integration) και η εκτέλεση (execution) των πλάνων του έργου.

Αν προσπαθούσαμε να απαντήσουμε στην ερώτηση, ποιες είναι οι βασικές συνιστώσες του σχεδιασμού ενός έργου, θα λέγαμε ότι ο σχεδιασμός ενός έργου περιλαμβάνει τον ορισμό:

- Των τελικών παραδοτέων (των προϊόντων, του λογισμικού, των συστημάτων) που θα παραχθούν από το έργο.
- Των δραστηριοτήτων που απαιτούνται για τη δημιουργία των παραδοτέων του έργου.
- Των δραστηριοτήτων που απαιτούνται για τον ποιοτικό έλεγχο των παραδοτέων του έργου.
- Των πόρων που απαιτούνται για την εκτέλεση των δραστηριοτήτων του έργου.
- Των εξαρτήσεων μεταξύ των δραστηριοτήτων (εσωτερικές, προερχόμενες από τις ανάγκες του έργου ή εξωτερικές, προερχόμενες από το περιβάλλον του έργου).

Το βασικό παραδοτέο του σχεδιασμού ενός έργου είναι το πλάνο διαχείρισης έργου (project management plan), ο ρόλος και τα περιεχόμενα του οποίου ορίζονται στην επόμενη παράγραφο. Το πλάνο διαχείρισης έργου θα πρέπει να παρουσιάζει μια συστηματική προσέγγιση για τη διαχείριση των δραστηριοτήτων του έργου, η οποία ταυτόχρονα θα είναι αρκετά εύκαμπτη για να διευκολύνει την αντιμετώπιση και ενσωμάτωση των αλλαγών στο έργο.

Είναι ιδιαίτερα σημαντικό να κατανοήσουμε ότι ο σχεδιασμός ενός έργου είναι μια επαναληπτική διαδικασία (βλ. Σχήμα 4.1.), η οποία ξεκινά από τη φάση της προκήρυξης του έργου και ολοκληρώνεται με τη λήξη του έργου. Η επαναληπτικότητα εξασφαλίζει ότι η διοίκηση του έργου έχει τη δυνατότητα να αλλάξει τον τρόπο εκτέλεσης του έργου, ώστε να ανταποκριθεί σε αλλαγές που προέρχονται, είτε από αλλαγές στις απαιτήσεις, είτε από το περιβάλλον του έργου ή απλά για να βελτιώσει την αποδοτικότητα στο έργο.



Σχήμα 4.17.: Η επαναληπτικότητα στο σχεδιασμό ενός έργου.

## ΕΝΟΤΗΤΑ 4.2 Ανάπτυξη Πλάνου Διαχείρισης Έργου.

Το πλάνο διαχείρισης έργου αποτελεί το πιο σημαντικό παραδοτέο που απαιτείται για τη σχεδίαση ενός έργου. Το πλάνο διαχείρισης έργου χρησιμοποιείται για να τεκμηριώσει τις παραδοχές του έργου, να τεκμηριώσει τις αποφάσεις που έχουν ληφθεί σε σχέση με τις εναλλακτικές λύσεις του έργου, να καθοδηγήσει την εκτέλεση του έργου, να τεκμηριώσει τις αποφάσεις που έχουν ληφθεί σε σχέση με τις εναλλακτικές λύσεις, να τεκμηριώσει τις πρακτικές εργασίας που θα ακολουθηθούν, να καθορίσει τα πρότυπα και τον τρόπο που αυτά θα χρησιμοποιηθούν, και γενικότερα να ορίσει:

- το πως θα γίνεται η επικοινωνία διευκολύνοντάς την,
- τον τρόπο διοίκησης του έργου,
- τον τρόπο επίλυσης διαφορών,
- τον τρόπο ελέγχου και αποδοχής των παραδοτέων, κ.ά.

Αποτελεί επομένως τη «βίβλο» ενός έργου, αφού ορίζει, καταγράφει, και επεξηγεί το πώς θα διοικήσουμε το έργο (Bechtold, 1999). Είναι ένα σύνθετο έγγραφο το οποίο περιέχει πληροφορίες για:

1. Το σκοπό του έργου. Ο σκοπός του έργου ορίζεται σε σχέση με τις ανάγκες του πελάτη, ενώ ταυτόχρονα δίνονται χαρακτηριστικά παραδείγματα για τη λειτουργικότητα του συστήματος, ώστε να γίνουν κατανοητές οι βασικές προτεραιότητες του συστήματος.
2. Τους στόχους του έργου. Οι στόχοι του έργου θα πρέπει να δίνονται με σαφήνεια, ενώ ταυτόχρονα να ορίζονται οι μετρικές με τις οποίες θα γίνει η αξιολόγηση της εκπλήρωσής τους. Οι στόχοι του έργου θα πρέπει να είναι εκφρασμένοι με τέτοιο τρόπο, ώστε να είναι πλήρως κατανοητοί από όλους όσους συμμετέχουν στο έργο.
3. Τα πρότυπα που θα χρησιμοποιηθούν στο έργο. Σε κάθε έργο είναι απαραίτητο να ορίσουμε έναν αριθμό προτύπων που θα χρησιμοποιηθούν κατά τη διάρκειά του. Η απαίτηση για τη χρήση ενός προτύπου προκύπτει, είτε από απαίτηση του πελάτη, είτε από την ανάγκη να εκτελέσουμε μια δραστηριότητα με συστηματικό και σωστό τρόπο. Επιπλέον, η υιοθέτηση των προτύπων θα πρέπει να είναι τέτοια ώστε να τυποποιούνται, τόσο οι διαδικασίες του έργου, όσο και τα παραδοτέα του έργου.
4. Τον κύκλο ζωής που θα υιοθετηθεί στο έργο. Η επιλογή του μοντέλου κύκλου ζωής του έργου είναι απόφαση ιδιαίτερα σημαντική για το έργο, αφού καθορίζει σε μεγάλο βαθμό τον τρόπο εργασίας, τα παραδοτέα (ιδιαίτερα τα ενδιάμεσα), τους απαιτούμενους πόρους, κ.ά.
5. Όλες τις βασικές διαδικασίες που θα χρησιμοποιηθούν κατά τη διάρκεια του έργου. Οι διαδικασίες που είναι απαραίτητες σε κάθε έργο σχετίζονται με έναν αριθμό παραγόντων, όπως το είδος του έργου, ο κύκλος ζωής του έργου, η τεχνογνωσία του οργανισμού που εκτελεί το έργο, κ.λπ. Για κάθε διαδικασία που θα χρησιμοποιήσουμε θα πρέπει να ορίσουμε τα δεδομένα εισόδου, την

απαιτούμενη εργασία ή επεξεργασία, καθώς και τα παραγόμενα αποτελέσματα.

Ο λεπτομερής ορισμός των διαδικασιών που θα χρησιμοποιηθούν στο έργο τις περισσότερες φορές προϋπάρχει, αφού αυτές έχουν ορισθεί κατά τη διάρκεια ανάπτυξης του ποιοτικού συστήματος (quality system) του οργανισμού. Για το λόγο αυτό, στο πλάνο διαχείρισης του έργου χρειάζεται να ορίσουμε το πώς οι διαδικασίες αυτές διαφοροποιούνται στο συγκεκριμένο έργο ή να ορίσουμε διαδικασίες που θα εφαρμοσθούν αποκλειστικά στο συγκεκριμένο έργο. Παραδείγματα τέτοιων διαδικασιών είναι η διαδικασία εκτίμησης κόστους λογισμικού (software cost estimation), η διαδικασία διαχείρισης σχηματισμών (configuration management), η διαδικασία διαχείρισης απαιτήσεων (requirements management), κ.ά.

6. Τα *εργαλεία* που θα χρησιμοποιηθούν, καθώς και τον τρόπο χρήσης τους. Ο ορισμός των εργαλείων που θα χρησιμοποιηθούν στο έργο αποτελεί βασική ανάγκη ιδιαίτερα για τα έργα πληροφορικής. Ο ορισμός των εργαλείων καθορίζει τον τρόπο εργασίας, την απαραίτητη τεχνογνωσία στο έργο, κ.ά. Ο ορισμός θα πρέπει να είναι λεπτομερής και να περιλαμβάνει τη συγκεκριμένη έκδοση που θα χρησιμοποιηθεί, το περιβάλλον εργασίας, τον αριθμό των αδειών, κ.λπ.
7. Την *αναλυτική λίστα με τα παραδοτέα του έργου*. Ο σαφής προσδιορισμός όλων των παραδοτέων, ενδιάμεσων και τελικών, αποτελεί αναγκαία προϋπόθεση για την αποτελεσματική διοίκηση του έργου. Ο ορισμός των παραδοτέων θα πρέπει να γίνεται με λεπτομέρεια ορίζοντας το είδος του παραδοτέου, το μέγεθός τους, την πολυπλοκότητα, τον τρόπο και τις απαιτήσεις ελέγχου.
8. Τη *δομή ανάλυσης εργασιών του έργου (WBS)*. Η ανάπτυξη της δομής WBS σε επαρκή λεπτομέρεια εξασφαλίζει ότι όλες οι απαραίτητες εργασίες έχουν καταγραφεί.
9. Τις *ανάγκες σε πόρους διαφόρων κατηγοριών*. Το πλάνο διαχείρισης έργου θα πρέπει να παρουσιάζει με σαφήνεια όλες τις διαφορετικές ειδικότητες προσωπικού που θα απαιτηθεί να εργασθούν στο έργο, καθώς και όλους τους διαφορετικούς ρόλους που θα απαιτηθούν. Επιπλέον, θα πρέπει να γίνει καταγραφή των αναγκών σε πόρους ανά δραστηριότητα.

10. Τον αναλυτικό προϋπολογισμό του έργου. Το πλάνο διαχείρισης έργου θα πρέπει να περιλαμβάνει τον αναλυτικό προϋπολογισμό του έργου, ενώ θα πρέπει να ορίζονται οι διαδικασίες για την παρακολούθηση των αλλαγών, αλλά και της προόδου του έργου.
11. Το χρονοδιάγραμμα, μαζί με τα βασικά ορόσημα (milestones) του έργου.
12. Τους κινδύνους του έργου, καθώς και τον τρόπο απόκρισης σε αυτούς. Ιδιαίτερα σε μεγάλου μεγέθους έργα η διαχείριση κινδύνου θα πρέπει να γίνεται με συστηματικό και αναλυτικό τρόπο.

Σύμφωνα με το PMBOK (PMI, 2004) το πλάνο διαχείρισης έργου είναι ένα σύνθετο έγγραφο, το οποίο περιέχει τουλάχιστον ένα κεφάλαιο για κάθε μια από τις γνωστικές περιοχές που απαρτίζουν το PMBOK. Έτσι, υπάρχουν κεφάλαια για το:

1. Πλάνο αντικειμένου εργασιών.
2. Πλάνο διαχείρισης χρονοδιαγράμματος.
3. Πλάνο διαχείρισης κόστους.
4. Πλάνο διαχείρισης ποιότητας.
5. Πλάνο διαχείρισης ανθρωπίνου δυναμικού.
6. Πλάνο επικοινωνίας.
7. Πλάνο διαχείρισης κινδύνου και απόκρισης στον κίνδυνο.
8. Πλάνο προμηθειών.
9. Πλάνο διαχείρισης αλλαγών.

Αν και ο βασικός στόχος του πλάνου διαχείρισης έργου είναι να τεκμηριώσει τον τρόπο με τον οποίο ένα έργο θα εκτελεσθεί, η χρησιμότητά του δεν εξαντλείται εκεί. Εξίσου σημαντικός στόχος είναι η καταγραφή του τρόπου παρακολούθησης του έργου και η διαχείριση των αλλαγών.

#### 4.2.1 Πληροφοριακά συστήματα διαχείρισης έργων

Η σύγχρονη τάση στη διαχείριση έργων απαιτεί την αυτοματοποιημένη διαχείρισή τους με τη χρήση πληροφοριακών συστημάτων (Project Management Information System, PMIS). Επιπλέον, το μέγεθος ορισμένων έργων και η πολυπλοκότητά τους επιβάλλει τη χρήση τέτοιων εργαλείων. Σήμερα στην αγορά κυκλοφορούν εκατοντάδες εργαλεία καλύπτοντας μεγάλο φάσμα αναγκών.

Για το λόγο αυτό η επιλογή του κατάλληλου εργαλείου αποτελεί μια επίπονη και δύσκολη διαδικασία. Η διαδικασία της επιλογής πρέπει να βασίζεται στον ορισμό των

κριτηρίων και των αναγκών βάση των οποίων θα γίνει η αξιολόγηση των εργαλείων. Η βασική λειτουργικότητα που υλοποιούν τα εργαλεία διαχείρισης έργων είναι η ακόλουθη:

- Υποστήριξη WBS.
- Υποστήριξη ανάλυσης έργου σε πολλά επίπεδα.
- Χρονοπρογραμματισμός των έργων με υποστήριξη των μεθόδων PERT, CPM και GANTT.
- Διαχείριση πόρων με υποστήριξη πρόβλεψης αναγκών, ανάθεση εργασιών, εξομάλυνση πόρων, παρακολούθηση κατανάλωσης πόρων, κ.λπ.
- Οικονομική διαχείριση με δημιουργία προϋπολογισμού, λογιστική παρακολούθηση του έργου, ανάλυση πιστοποιημένης αξίας, κ.λπ.

Η σύγχρονη τάση στην αγορά των εργαλείων διαχείρισης έργων είναι η ενοποίηση των συστημάτων αυτών με τα συστήματα διαχείρισης επιχειρησιακών πόρων ( Enterprise Resource Planning, ERP). Με τη χρήση των συστημάτων ERP επιτυγχάνεται η ενοποίηση των δεδομένων του έργου με τα εταιρικά δεδομένα, με αποτέλεσμα την καλύτερη διαχείριση των πόρων σε εταιρικό επίπεδο.

Επιπλέον, η χρήση συστημάτων PMIS σε συνδυασμό με συστήματα ERP επιτρέπει:

- Την έναρξη της διαχείρισης του έργου από τη στιγμή που υποβάλλεται η προσφορά για το έργο.
- Τον καλύτερο έλεγχο του συμβολαίου και των σχέσεων με τον πελάτη, καθώς και την καλύτερη λογιστική παρακολούθηση του έργου. Η διαχείριση συμβολαίου αφορά στη διαχείριση των παραγγελιών που γίνονται στα πλαίσια ενός συμβολαίου / έργου. Στην περίπτωση μεγάλων έργων με σύμβαση με ποσοστό επί του απολογιστικού κόστους ή με τιμή μονάδος η διαχείριση συμβολαίου αποτελεί ιδιαίτερα σημαντική και πολύπλοκη διαδικασία καθώς μπορεί σε μηνιαία βάση να υπάρχουν εκατοντάδες παραγγελίες / τιμολόγια.
- Την καλύτερη τεκμηρίωση του έργου με ολοκληρωμένη διαχείριση εγγράφων. Η ολοκληρωμένη διαχείριση εγγράφων (document management) επιτρέπει το συστηματικό έλεγχο όλων των εγγράφων ενός έργου με την αποθήκευσή τους με ομοιόμορφο τρόπο, την ταξινόμησή τους και την εύκολη εύρεσή τους. Επιπλέον, με την ολοκληρωμένη διαχείριση εγγράφων είναι δυνατόν να ελέγξουμε την πρόσβαση στα έγγραφα, τόσο μέσα στην ομάδα του έργου, όσο και έξω από αυτήν.



- Την υποστήριξη με αυτοματοποιημένο τρόπο της ροής των εργασιών. Η διαχείριση ροής εργασιών (workflow management) αποτελεί βασικό εργαλείο για την αποτελεσματική και γρήγορη διεκπεραίωση καλά ορισμένων και τεκμηριωμένων διαδικασιών που απαιτούν συνεργασία πολλών διαφορετικών ατόμων. Η έγκριση παραγγελιών, η ανάθεση εργασιών και ο απολογισμός ενεργειών είναι τυπικά παραδείγματα τυποποιημένων διαδικασιών στα πλαίσια ενός έργου που μπορούν να επιταχυνθούν και να ελεγχθούν καλύτερα με τη χρήση ενός αυτοματοποιημένου πληροφοριακού συστήματος.
- Την ολοκληρωμένη διαχείριση των προμηθειών στα πλαίσια του οργανισμού.
- Τον καλύτερο απολογιστικό έλεγχο του έργου.

#### **Δραστηριότητα 4.1**

Ποια κατά τη γνώμη σας είναι τα πλεονεκτήματα και τα μειονεκτήματα που προκύπτουν από τη χρήση των συστημάτων PMIS.

---

### **ΕΝΟΤΗΤΑ 4.3 Χρονοπρογραμματισμός Έργου.**

---

Το χρονοπρόγραμμα (schedule) του έργου αποτελεί ένα από τα βασικά εργαλεία για την παρακολούθηση, αλλά και τον έλεγχο ενός έργου. Είναι τόσο σημαντικό ώστε η ύπαρξή του αποτελεί πολλές φορές ζητούμενο από τους πελάτες ενός έργου.

Η βασική προσέγγιση για τη δημιουργία χρονοδιαγράμματος είναι η ανάπτυξη ενός δικτύου δραστηριοτήτων και σχέσεων μεταξύ τους, οι οποίες αναπαριστούν γραφικά την ακολουθία με την οποία θα πρέπει να εκτελεστούν οι δραστηριότητες ενός έργου. Επιπλέον, στο χρονοδιάγραμμα παρουσιάζονται οι πόροι που απαιτούνται για την ολοκλήρωση της κάθε δραστηριότητας, το κόστος αυτών, καθώς και οι υπάρχοντες περιορισμοί. Στόχος είναι η ανάπτυξη του χρονοδιαγράμματος να δώσει απάντηση σε μια σειρά από βασικά ερωτήματα που σχετίζονται με τη διαχείριση του κάθε έργου, όπως είναι τα ακόλουθα:

- Ποια είναι η ημερομηνία ολοκλήρωσης του έργου;
- Ποιες είναι οι κρίσιμες δραστηριότητες του έργου, δηλαδή οι δραστηριότητες που αν καθυστερήσουν καθυστερεί συνολικά όλο το έργο;

- Ποιες είναι οι δραστηριότητες που μπορούν να καθυστερήσουν χωρίς να καθυστερήσει συνολικά το έργο και πόσο είναι δυνατό αυτές να καθυστερήσουν;
- Πότε πρέπει να ξεκινήσει και πότε να τελειώσει η κάθε δραστηριότητα;
- Είναι επιθυμητό / αποδοτικό να επιταχύνουμε την εκτέλεση κάποιων δραστηριοτήτων, ώστε να συντομεύσουμε τη συνολική διάρκεια του έργου;
- Ποιο είναι το κόστος που έχει το έργο μια δεδομένη χρονική στιγμή και πως αυτό συγκρίνεται με το προϋπολογισθέν κόστος;
- Ποια είναι η πιθανότητα ολοκλήρωσης ενός έργου μια δεδομένη χρονική στιγμή;

Για την ανάπτυξη του χρονοδιαγράμματος έχουν αναπτυχθεί διάφορες τεχνικές. Οι πιο γνωστές από αυτές είναι η μέθοδος PERT (Program Evaluation and Review Technique) και η μέθοδος της κρίσιμης διαδρομής (Critical Path Method, CPM). Η μέθοδος PERT αναπτύχθηκε από τους Booz, Allen και Hamilton το 1958 με σκοπό την καλύτερη διαχείριση των περίπου 11000 υπεργολάβων που συμμετείχαν στο έργο της ανάπτυξης του πυραύλου Polaris. Αντίστοιχα, η μέθοδος CPM είναι αποτέλεσμα της κοινής προσπάθειας των M. Walker της εταιρείας DuPont και του J. Kelly της εταιρείας Remington Rand, που είχαν ως βασικό στόχο το χρονοπρογραμματισμό της συντήρησης σε χημικά εργοστάσια. Η βασική διαφορά των δύο μεθόδων είναι ότι στη μέθοδο CPM η χρονική διάρκεια των δραστηριοτήτων είναι προκαθορισμένη, ενώ στη μέθοδο PERT η χρονική διάρκεια μιας δραστηριότητας αναπαρίσταται ως μια τυχαία μεταβλητή, η οποία σχετίζεται με τρεις εκτιμήσεις διάρκειας: Την αισιόδοξη, την απαισιόδοξη και την πιο πιθανή.

Στις επόμενες παραγράφους θα περιγράψουμε αναλυτικά αυτές τις μεθόδους.

### **4.3.1 Το Διάγραμμα Gantt.**

Το πλέον χρησιμοποιούμενο εργαλείο για το χρονοπρογραμματισμό ενός έργου είναι το διάγραμμα Gantt, το οποίο αναπτύχθηκε από το H. Grantt.

Το διάγραμμα Gantt αποτελεί ένα γραμμικό διάγραμμα που παρουσιάζει με απλό και εύληπτο τρόπο, την πορεία υλοποίησης ενός έργου. Πιο συγκεκριμένα, ένα διάγραμμα Gantt για ένα έργο παρουσιάζει τη λίστα των δραστηριοτήτων του έργου, τη διάρκειά τους, την ημερομηνία έναρξης, κ.ά. Χρησιμοποιώντας ένα διάγραμμα Gantt είναι δυνατό να χρονοπρογραμματίσουμε το έργο χρησιμοποιώντας, είτε τη λογική της νωρίτερης

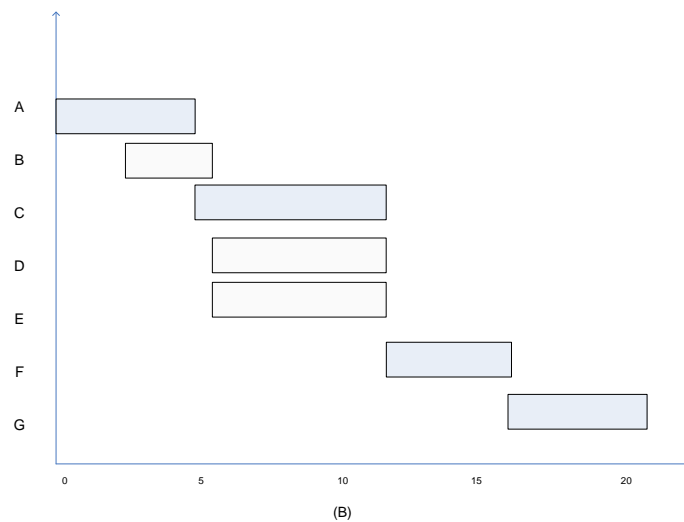
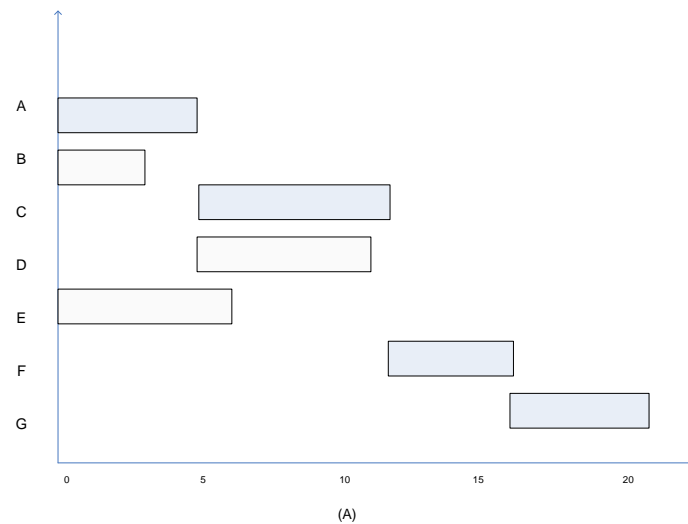
έναρξης (early-start), είτε τη λογική της αργότερης έναρξης (late-start). Όταν χρησιμοποιούμε τη λογική της νωρίτερης έναρξης, κάθε δραστηριότητα ξεκινά όσο το δυνατό νωρίτερα, ενώ χρησιμοποιώντας τη λογική της αργότερης έναρξης, καθυστερούμε την έναρξη της κάθε δραστηριότητας με την απαίτηση να μην επιμηκύνεται η συνολική διάρκεια του έργου.

#### **Παράδειγμα 4.5**

Ο επόμενος πίνακας παρουσιάζει τις δραστηριότητες ενός έργου, τις σχέσεις μεταξύ τους, καθώς και τη διάρκειά τους σε εβδομάδες.

<b>Δραστηριότητα</b>	<b>Προηγούμενη Δραστηριότητα</b>	<b>Διάρκεια (εβδομάδες)</b>
A	-	5
B	-	3
C	A	8
D	A, B	7
E	-	7
F	C, E, D	4
G	F	5

Εάν εφαρμόσουμε τη λογική της νωρίτερης έναρξης οδηγούμαστε στην κατασκευή ενός χρονοδιαγράμματος που μας δίνει τη συντομότερη διάρκεια του έργου (βλ. Σχήμα 4.18.(α)). Στη συνέχεια και αφού κατασκευάσουμε το συντομότερο χρονοδιάγραμμα μπορούμε να προχωρήσουμε στην κατασκευή του χρονοδιαγράμματος όπου οι δραστηριότητες αρχίζουν το αργότερο δυνατό, μετατοπίζοντας τις δραστηριότητες του έργου, αλλά με την προϋπόθεση ότι η συνολική διάρκεια του έργου δεν υπερβαίνει τις 22 εβδομάδες (βλ. Σχήμα 4.18.(β)).



**Σχήμα 4.18.:** Διάγραμμα Gantt για νωρίτερη και αργότερη έναρξη δραστηριοτήτων

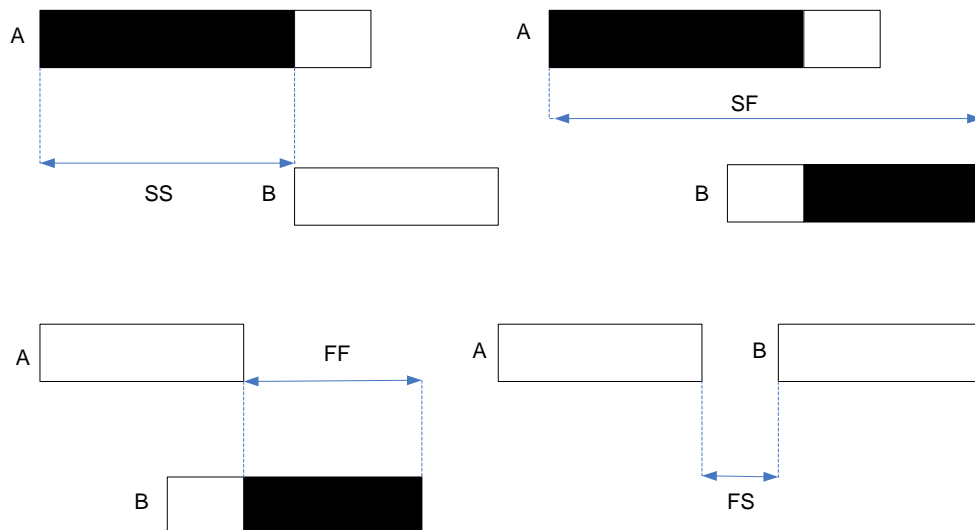
Η διαφορά στη έναρξη (ή στη λήξη) των δραστηριοτήτων μεταξύ των δύο χρονοδιαγραμμάτων απεικονίζει το χρονικό περιθώριο μιας δραστηριότητας. Στο παραπάνω παράδειγμα η δραστηριότητα B μπορεί να τελειώσει το νωρίτερο την τρίτη ημέρα και το αργότερο την έκτη ημέρα. Συνεπώς έχει χρονικό περιθώριο τριών ημερών, που σημαίνει ότι μια τριήμερη καθυστέρηση της δραστηριότητας δε θα προκαλέσει καθυστέρηση στο έργο συνολικά.

Δραστηριότητες που δεν έχουν χρονικό περιθώριο καθυστέρησης ονομάζονται κρίσιμες δραστηριότητες. Αντίστοιχα, καθυστερημένη εκτέλεση μιας τέτοιας δραστηριότητας θα προκαλέσει καθυστέρηση στη λήξη όλου του έργου. Η ακολουθία των δραστηριοτήτων που είναι κρίσιμες και οδηγούν από την αρχή στο τέλος του έργου ορίζουν το κρίσιμο μονοπάτι του έργου που λογικά είναι και το μεγαλύτερο σε διάρκεια μονοπάτι.

Στο παραπάνω παράδειγμα κάναμε την παραδοχή ότι για να ξεκινήσει μια δραστηριότητα θα πρέπει πρώτα να τελειώσει η προηγούμενη. Έτσι, για να ξεκινήσει η δραστηριότητα C θα πρέπει να λήξει η δραστηριότητα A. Ο περιορισμός αυτός ονομάζεται “Λήξη-για-Έναρξη” (Finish-to-Start, FS). Ο περιορισμός αυτός είναι ο πιο συνηθισμένος που συναντάμε σε έργα (π.χ. πρέπει να τελειώσει ο έλεγχος για να αρχίσει η εγκατάσταση του λογισμικού). Αντίστοιχα, μπορούμε να ορίσουμε τους περιορισμούς “Έναρξη-για-Έναρξη” (Start-to-Start, SS), “Λήξη-για-Λήξη” (Finish-to-Finish, FF) και “Έναρξη-για-Λήξη” (Start-to-Finish, SF) ως εξής:

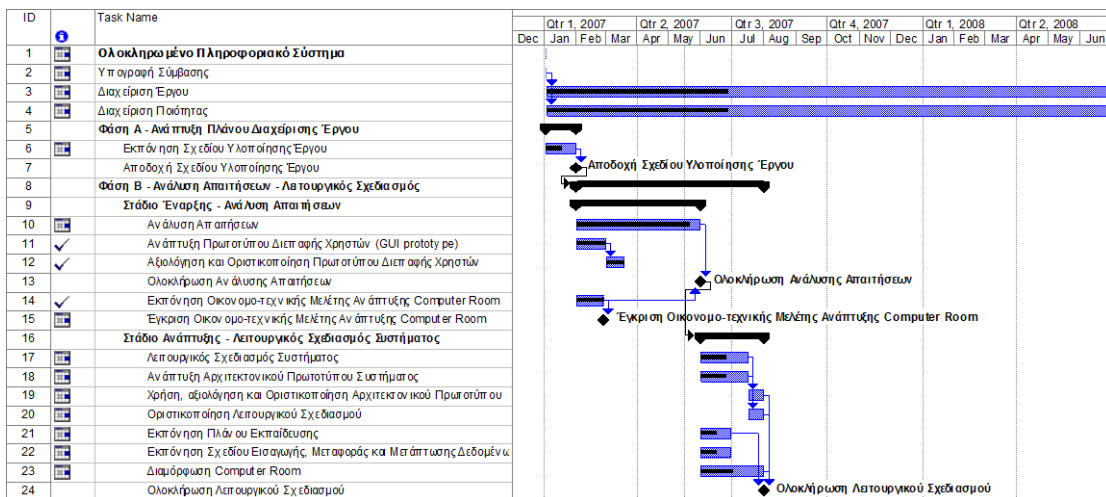
- $FS_{AB}$  Η δραστηριότητα B δεν μπορεί να ξεκινήσει πριν το πέρας FS χρονικών μονάδων μετά τη λήξη της δραστηριότητας A.
- $SS_{AB}$  Η δραστηριότητα B δεν μπορεί να ξεκινήσει πριν την έναρξη της A και το πέρας SS χρονικών μονάδων.
- $FF_{AB}$  Η δραστηριότητα B δεν μπορεί να ολοκληρωθεί πριν την ολοκλήρωση της A και αφού παρέλθουν FF χρονικές μονάδες.
- $SF_{AB}$  Πρέπει να μεσολαβούν τουλάχιστον SF χρονικές μονάδες μεταξύ της έναρξης της A και της λήξης της B.

Το Σχήμα 4.19. παρουσιάζει τους περιορισμούς προτεραιοτήτων μεταξύ των δραστηριοτήτων.



**Σχήμα 4.19.:** Περιορισμοί προτεραιοτήτων.

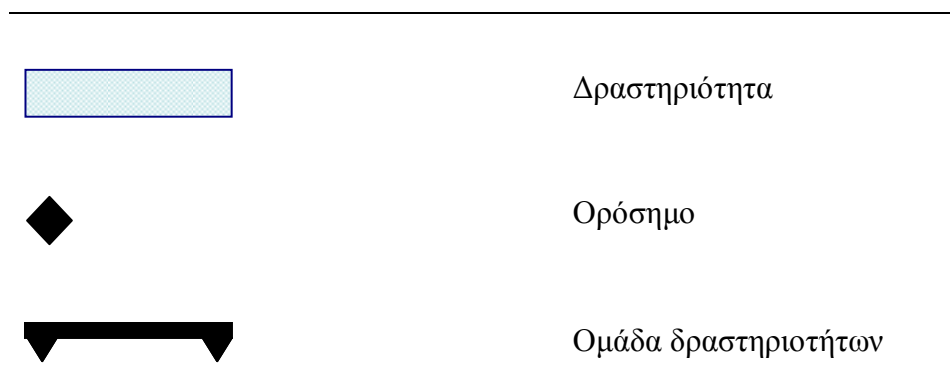
Στο Σχήμα 4.4. παρουσιάζεται ένα παράδειγμα ενός διαγράμματος Gantt, το οποίο έχει δημιουργηθεί με τη βοήθεια του εργαλείου MS-Project, που αποτελεί κατά πολλούς το πιο δημοφιλές εργαλείο στη διαχείριση έργων.



Σχήμα 4.20.: Παράδειγμα διαγράμματος Gantt.

Στο Σχήμα 4.4., στο αριστερό τμήμα παρουσιάζεται η λίστα των δραστηριοτήτων για δύο φάσεις ενός έργου ανάπτυξης ενός πληροφοριακού συστήματος, ενώ στο δεξί μέρος παρουσιάζεται με γραφικό τρόπο η κάθε δραστηριότητα όπως αυτή εξελίσσεται στο χρόνο. Επίσης, παρουσιάζονται οι τυχόν περιορισμοί μεταξύ των δραστηριοτήτων, οι οποίοι στη συγκεκριμένη περίπτωση είναι τύπου “Finish-to-Start”.

Ο παρακάτω πίνακας επεξηγεί τα σύμβολα που χρησιμοποιούνται στο διάγραμμα του Σχήματος 4.4.:



### Δραστηριότητα 4.2

Αναπτύξτε το διάγραμμα Gantt για την εκπόνηση μιας πτυχιακής εργασίας. Η πτυχιακή εργασία στο ΕΑΠ ακολουθεί την ίδια δομή με μια θεματική ενότητα. Για την ανάπτυξη του διαγράμματος μπορείτε να χρησιμοποιήσετε ένα από τα πολλά εργαλεία που είναι διαθέσιμα στο διαδίκτυο.

### 4.3.2 Ανάλυση Κρίσιμου Μονοπατιού.

Για να εφαρμόσουμε τις μεθόδους PERT/CPM θα πρέπει να κατανοήσουμε πολύ καλά τις ανάγκες του έργου, αλλά και τις απαιτήσεις του. Πιο συγκεκριμένα, για να ξεκινήσουμε να κατασκευάζουμε το δίκτυο θα πρέπει να είμαστε σε θέση να απαντήσουμε σε τέσσερις βασικές ερωτήσεις:

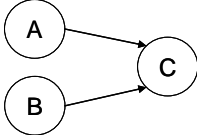
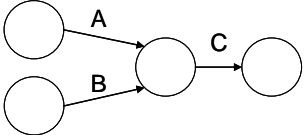
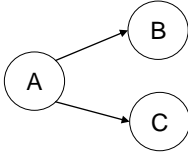
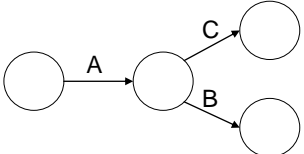
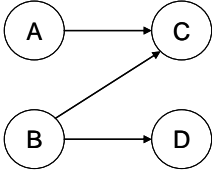
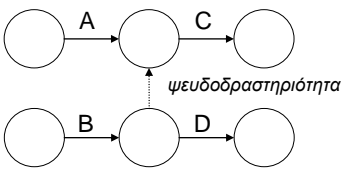
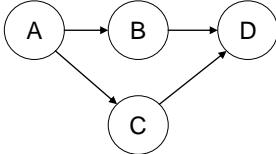
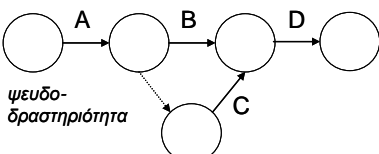
1. Ποιες είναι οι δραστηριότητες του έργου;
2. Ποια είναι η σειρά εκτέλεσης των δραστηριοτήτων και ποιοι είναι οι περιορισμοί που υπάρχουν;
3. Ποιες δραστηριότητες μπορούν να εκτελεστούν παράλληλα;
4. Ποια είναι η χρονική διάρκεια της κάθε δραστηριότητας;

Στις μεθόδους PERT/CPM το έργο αναπαρίσταται ως ένα δίκτυο. Οι δύο πιο δημοφιλείς προσεγγίσεις αναπαράστασης είναι οι εξής:

- A. Οι δραστηριότητες να τοποθετούνται πάνω σε τόξα του δικτύου (Activity on Arrow, AOA). Η ουρά του τόξου αναπαριστά την αρχή και η κεφαλή του το τέλος της δραστηριότητας του χρονοδιαγράμματος. Οι δραστηριότητες του χρονοδιαγράμματος συνδέονται σε σημεία που καλούνται κόμβοι (και που συνήθως σχεδιάζονται ως μικροί κύκλοι), ώστε να επιδειχθεί η ακολουθία με την οποία αναμένεται να εκτελεστούν οι δραστηριότητες του χρονοδιαγράμματος.
- B. Οι δραστηριότητες να τοποθετούνται πάνω σε κόμβους του δικτύου (Activity on Node, AON).

Στον πίνακα του Σχήματος 4.5. συνοψίζονται οι διαφορετικές περιπτώσεις που εμφανίζονται για κάθε διαφορετική αναπαράσταση.

Activity on Node (AON)	Ερμηνεία	Activity on Arrow (AOA)
	<p>Η δραστηριότητα A πρέπει να ολοκληρωθεί για να ξεκινήσει η δραστηριότητα B και στη συνέχεια η</p>	

Activity on Node (AON)	Ερμηνεία	Activity on Arrow (AOA)
	<p>δραστηριότητα C.</p> <p>Οι δραστηριότητες A και B πρέπει να ολοκληρωθούν για να ξεκινήσει η δραστηριότητα C.</p>	
	<p>Οι δραστηριότητες B και C μπορούν να ξεκινήσουν μόνο όταν ολοκληρωθεί η A.</p>	
	<p>Η δραστηριότητα C θα ξεκινήσει όταν ολοκληρωθούν οι δραστηριότητες A και B.</p> <p>Στην αναπαράσταση AOA είναι απαραίτητη η εισαγωγή μιας ψευδοδραστηριότητας (dummy activity) με σκοπό να απεικονισθεί αυτή η εξάρτηση.</p>	
	<p>Οι δραστηριότητες B και C δεν μπορούν να ξεκινήσουν μέχρι να ολοκληρωθεί η A. Αντίστοιχα, η D δεν μπορεί να ξεκινήσει μέχρι</p>	



Activity on Node (AON)	Ερμηνεία	Activity on Arrow (AOA)
	να ολοκληρωθούν οι δραστηριότητες B και C. Και στο παράδειγμα αυτό είναι αναγκαία η εισαγωγή ψευδο-δραστηριότητας στην αναπαράσταση AOA.	

Σχήμα 4.21.: Συμβολισμοί αναπαράστασεων AOA και AON.

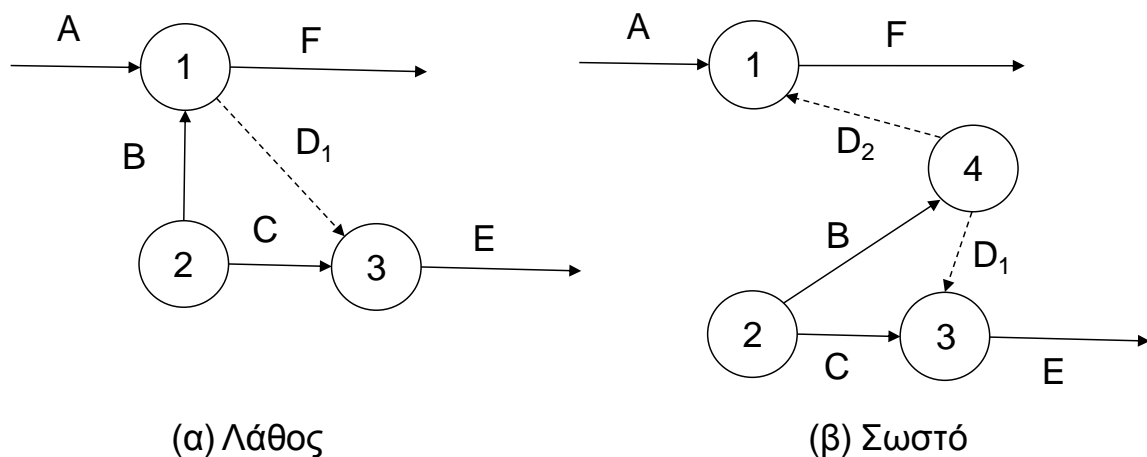
#### Παράδειγμα 4.6

Σχεδιάστε το δίκτυο AOA που ικανοποιεί του ακόλουθους περιορισμούς:

1. Η δραστηριότητα E είναι επόμενη των B και C.
2. Η δραστηριότητα F είναι επόμενη των A και B.

#### Απάντηση

Στο Σχήμα 4.6. παρουσιάζονται δύο δίκτυα. Ένα λανθασμένο και ένα σωστό. Στο λανθασμένο διάγραμμα οι προηγούμενες δραστηριότητες της E δεν είναι οι σωστές. Σύμφωνα με το διάγραμμα (α) η δραστηριότητα E έχει ως προηγούμενες τις A, B και C και όχι τη B και C όπως θα έπρεπε.



Σχήμα 4.22.: Παράδειγμα δικτύου AOA.

#### Άσκηση Αυτοαξιολόγησης 4.1

Δεδομένου του παρακάτω πίνακα δραστηριοτήτων κατασκευάστε το δίκτυο AOA και στη συνέχεια το δίκτυο AON.

<b>Δραστηριότητα</b>	<b>Προηγούμενη Δραστηριότητα</b>
A	-
B	-
C	A
D	A, B
E	C
F	C
G	D, E
H	F, G

Από τη στιγμή που κατασκευάσουμε το δίκτυο AOA, μπορούμε να αναπαραστήσουμε το δίκτυο AOA με μορφή πίνακα, όπου οι γραμμές αντιστοιχούν στα γεγονότα (κόμβους) έναρξης, ενώ οι στήλες στα γεγονότα λήξης. Σε περίπτωση που υπάρχει δραστηριότητα που συνδέει τα δύο γεγονότα, στο αντίστοιχο κελί του πίνακα αποθηκεύουμε την τιμή «Αληθές» ή τη διάρκεια της δραστηριότητας.

Λόγω του σχήματος αρίθμησης που χρησιμοποιείται έχουμε για κάθε δραστηριότητα  $a(i,j)$ ,  $i > j$ . Το γεγονός αυτό σημαίνει ότι ο πίνακας έχει τιμή πάνω από τη διαγώνιο. Τα κελιά κάτω από τη διαγώνιο χρησιμοποιούνται για την αποθήκευση πληροφοριών σχετικά με τους απαιτούμενους πόρους, τον προϋπολογισμό της δραστηριότητας, κ.λπ.

Σε μεγάλα έργα η δημιουργία της αρίθμησης των γεγονότων δεν είναι προφανής. Για την αρίθμηση των κόμβων σε ένα δίκτυο AOA μπορούμε να εφαρμόσουμε τον παρακάτω αλγόριθμο.

Ας υποθέσουμε ότι το δίκτυο περιγράφεται σε έναν πίνακα γειτονικότητας  $A$ , όπου  $a_{i,j} = 1$ , εάν ο κόμβος  $i$  είναι άμεσα προηγούμενος του  $j$ , ενώ σε κάθε άλλη περίπτωση

$a_{i,j} = 0$ . Στον πίνακα αυτό οι κόμβοι είναι διατεταγμένοι σε γραμμές και στήλες με τυχαίο τρόπο. Έστω ότι  $v(j)$  συμβολίζει το νούμερο ενός καινούργιου κόμβου  $j$  και  $d_j^{(in)}$  συμβολίζει τον αριθμό των εισερχομένων τόξων στον κόμβο  $j$ . Αρχικά υπολογίζουμε το  $d_j^{(in)}$  για όλους τους κόμβους, αθροίζοντας τον αριθμό των μη μηδενικών στοιχείων της στήλης  $j$  του πίνακα  $A$ . Ο κόμβος  $k$  για τον οποίο  $d_k^{(in)} = 0$  είναι αρχικός κόμβος και γι' αυτό το λόγο θέτουμε  $v(k) = 1$ . Στη συνέχεια, υπολογίζουμε ξανά τις τιμές των στοιχείων που βρίσκονται στη γραμμή  $k$ . Ο αλγόριθμος παρουσιάζεται παρακάτω:

Βήμα 0:

Υπολογίζουμε  $d_j^{(in)} = \sum_{i=1}^n a_{ij}, j = 1, 2, \dots, n$

Θέτουμε  $N = \{1, 2, \dots, n\}$

Θέτουμε  $m = 1$

Βήμα 1: Υπολογισμός των αρχικών κόμβων.

Βρίσκουμε  $k \in N$  με  $d_k^{(in)} = 0$ . Εάν δεν υπάρχει τέτοιο  $k$ , τότε το δίκτυο περιέχει κύκλους οπότε ο αλγόριθμος σταματά.

Θέτουμε  $v(k) = m$

Θέτουμε  $m = m + 1$

Θέτουμε  $N = N - \{k\}$

Εάν  $N = \emptyset$ , τότε ο αλγόριθμος σταματά έχοντας αριθμήσει με σωστό τρόπο όλους τους κόμβους.

Βήμα 2: Υπολογίζουμε ξανά τον αριθμό των εισερχομένων τόξων.

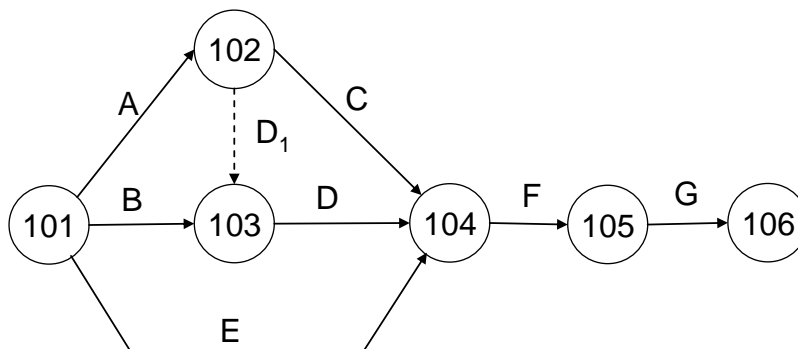
Θέτουμε  $d_j^{(in)} = d_j^{(in)} - a_{kj}$ , για όλα τα  $j \in N$

Επιστρέφουμε στο βήμα 1

Εάν δεν είναι δυνατό να αριθμήσουμε όλους τους κόμβους, τότε υπάρχει λογικό λάθος στον ορισμό των προηγούμενων δραστηριοτήτων και πιθανόν να υπάρχει κυκλική εξάρτηση.

### Παράδειγμα 4.7

Έστω το έργο του Παραδείγματος 4.1. Ξεκινάμε από το Σχήμα 4.23, όπου παρουσιάζεται το δίκτυο ΑΟΑ, στο οποίο η αρίθμηση των κόμβων έχει γίνει με τυχαίο τρόπο. Στόχος μας είναι να εφαρμόσουμε τον αλγόριθμο που παρουσιάστηκε στο Παράδειγμα 4.2, με σκοπό να αριθμήσουμε τους κόμβους του δικτύου.



**Σχήμα 4.23.:** Παράδειγμα δικτύου ΑΟΑ με τυχαία αρίθμηση.

Η εκτέλεση του βήματος 0 αφορά στην κατασκευή του πίνακα γειτονικότητας και στον υπολογισμό του  $d_j^{(in)}$ .

		Τελικό γεγονός					
		101	102	103	104	105	106
Αρχικό γεγονός	101		1	1	1		
	102			1	1		
	103				1		
	104					1	
	105						1
	106						
$d^{(in)}$		0	1	2	3	1	1

Είναι  $N = \{101, 102, 103, 104, 105, 106\}$  και  $m = 1$ .

Αναλύοντας τον παραπάνω πίνακα διαπιστώνουμε ότι ο κόμβος 101 έχει  $d^{(in)} = 0$  και γι' αυτό το λόγο έχουμε:

$$v(101) = 1$$

$$N = \{102, 103, 104, 105, 106\} \text{ και}$$

$$m = 2.$$

Εφαρμόζουμε το βήμα 2 του αλγορίθμου ξανα-υπολογίζοντας τα  $d_j^{(in)}$ , οπότε έχουμε

	102	103	104	105	106
$d^{(in)}$	0	1	2	1	1

Στον παραπάνω πίνακα ο κόμβος 102 έχει  $d^{(in)} = 0$  και γι' αυτό το λόγο έχουμε:

$$v(102) = 2$$

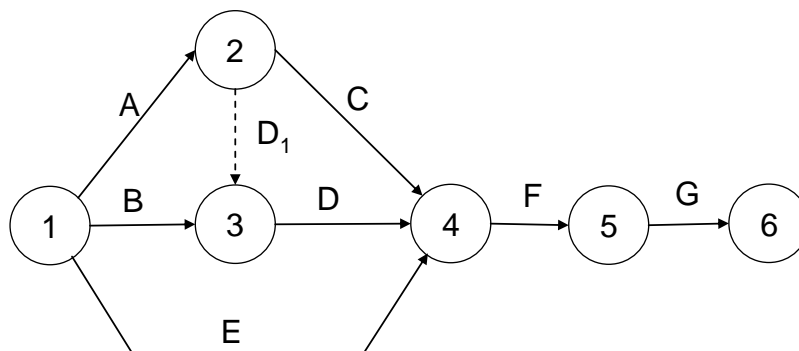
$$N = \{103, 104, 105, 106\} \text{ και}$$

$$m = 3.$$

Εφαρμόζουμε ξανά το βήμα 2 του αλγορίθμου υπολογίζοντας τα  $d_j^{(in)}$ , οπότε έχουμε:

	103	104	105	106
$d^{(in)}$	0	1	1	1

Επαναλαμβάνουμε διαδοχικά τα βήματα του αλγορίθμου μέχρι  $N = \emptyset$ . Το δίκτυο που προκύπτει έχει την αρίθμηση που παρουσιάζεται στο Σχήμα 4.8.



**Σχήμα 4.24.:** Παράδειγμα δικτύου AOA με κανονική αρίθμηση.

Έχοντας κατασκευάσει το δίκτυο είναι εύκολο να υπολογίσουμε όλα τα δυνατά μονοπάτια που ξεκινούν από τους αρχικούς κόμβους και καταλήγουν στους τελικούς κόμβους. Επιπλέον, εάν γνωρίζουμε τη διάρκεια των δραστηριοτήτων (τόξων) μπορούμε να υπολογίσουμε τη διάρκεια του έργου, η οποία ισούται με τη διάρκεια του μακρύτερου μονοπατιού.

Για το δίκτυο που παρουσιάζεται στο Σχήμα 4.24. έχουμε:

<b>Ακολουθία γεγονότων</b>	<b>Δραστηριότητες Διαδρομής</b>	<b>Διάρκεια Διαδρομής</b>
1-2-4-5-6	A, C, E, G	22
1-2-3-4-5-6	A, D <sub>1</sub> , D, F, G	21
1-3-4-5-6	B, D, F, G	19
1-4-5-6	E, F, G	16

Στον παραπάνω πίνακα, η διαδρομή με τη μεγαλύτερη διάρκεια είναι αυτή της ακολουθίας γεγονότων 1-2-4-5-6, η οποία έχει συνολική διάρκεια 22 χρονικών μονάδων. Όλες οι δραστηριότητες αυτής της διαδρομής θεωρούνται κρίσιμες, γεγονός που σημαίνει ότι εάν η εκτέλεσή τους καθυστερήσει, θα καθυστερήσει όλο το έργο. Αντίστοιχα, η διαδρομή E-F-G έχει διάρκεια 16 χρονικών μονάδων που σημαίνει ότι οι δραστηριότητες E, F και G μπορούν να έχουν μια συνολική καθυστέρηση 6 χρονικών μονάδων.

Στο σημείο αυτό μπορούμε να ορίσουμε τις έννοιες του ελεύθερου περιθωρίου (free float, free slack) και του συνολικού περιθωρίου (total float, total slack) .

Ελεύθερο περιθώριο είναι το χρονικό διάστημα κατά το οποίο μία δραστηριότητα του χρονοπρογράμματος μπορεί να καθυστερήσει χωρίς να καθυστερήσει η νωρίτερη έναρξη καμίας από τις αμέσως ακόλουθες δραστηριότητες του χρονοπρογράμματος.

Αντίστοιχα, συνολικό περιθώριο είναι το χρονικό διάστημα κατά το οποίο μία δραστηριότητα του χρονοπρογράμματος μπορεί να καθυστερήσει πέρα από την ημερομηνία νωρίτερης έναρξης, χωρίς να καθυστερήσει η ημερομηνία λήξης του έργου ή να παραβιασθεί ένας περιορισμός του χρονοπρογράμματος.

### **4.3.3 Υπολογισμός Κρίσιμης Διαδρομής.**

Ο υπολογισμός της κρίσιμης διαδρομής μπορεί να γίνει χρησιμοποιώντας, είτε την αναπαράσταση AOA, είτε την αναπαράσταση AON.

Θα επιλέξουμε να χρησιμοποιήσουμε την αναπαράσταση AON στην οποία οι δραστηριότητες του έργου αντιστοιχούν στους κόμβους του δικτύου. Στην αναπαράσταση του έργου με χρήση δικτύου AON ισχύουν οι παρακάτω κανόνες:

1. Όλοι οι κόμβοι με εξαίρεση τους τελικούς έχουν τουλάχιστον έναν επόμενο κόμβο (successor).
2. Όλοι οι κόμβοι εκτός από τον αρχικό έχουν τουλάχιστον έναν προηγούμενο κόμβο (predecessor).
3. Υπάρχει τουλάχιστον ένας αρχικός και τουλάχιστον ένας τελικός κόμβος.
4. Κάθε ακμή θα πρέπει να συνδέει υποχρεωτικά δύο κόμβους.
5. Το δίκτυο που προκύπτει από τις δραστηριότητες του έργου, καθώς και από τις σχέσεις μεταξύ τους είναι ακυκλικό.

Για τον υπολογισμό της κρίσιμης διαδρομής χρειάζεται να ορίσουμε τις ακόλουθες έννοιες:

1. **Νωρίτερη έναρξη (Early Start, ES):** Η νωρίτερη ημερομηνία κατά την οποία μπορεί να ξεκινήσει μια δραστηριότητα, με την προϋπόθεση ότι όλες οι προηγούμενες δραστηριότητες έχουν ολοκληρωθεί κανονικά.
2. **Νωρίτερη λήξη (Early Finish, EF):** Η νωρίτερη ημερομηνία κατά την οποία μπορεί να ολοκληρωθεί μια δραστηριότητα, με την προϋπόθεση ότι όλες οι προηγούμενες δραστηριότητες έχουν ολοκληρωθεί κανονικά.
3. **Αργότερη έναρξη (Late Start, LS):** Η αργότερη ημερομηνία κατά την οποία μπορεί να ξεκινήσει μια δραστηριότητα, έτσι ώστε το έργο να ολοκληρωθεί την προγραμματισμένη ημερομηνία.
4. **Αργότερη λήξη (Late Finish, LF):** Η αργότερη ημερομηνία κατά την οποία μπορεί να λήξει μια δραστηριότητα, έτσι ώστε το έργο να ολοκληρωθεί την προγραμματισμένη ημερομηνία.
5. **Χρονικό περιθώριο (slack, SL):** Το πλεόνασμα χρόνου στον προγραμματισμό μιας διαδικασίας.

Για τον υπολογισμό της κρίσιμης διαδρομής διατρέχουμε το δίκτυο από την αρχή του έργου προς τη λήξη του έργου (forward pass) υπολογίζοντας τους χρόνους της νωρίτερης έναρξης και της νωρίτερης λήξης. Στη συνέχεια, διατρέχουμε το δίκτυο αντίστροφα, από τις τελικές δραστηριότητες προς τις αρχικές (backward pass) και υπολογίζουμε τους χρόνους της αργότερης έναρξης και της αργότερης λήξης. Οι δραστηριότητες που έχουν μηδενικό χρονικό περιθώριο ανήκουν στην κρίσιμη διαδρομή.

Για τον υπολογισμό των χρόνων χρησιμοποιούμε τον παρακάτω αλγόριθμο:

Υπολογισμός των ES, EF (εργαζόμαστε από τις αρχικές προς τις τελικές δραστηριότητες):

$ES(k)=0$ , όπου  $k$  είναι αρχικές δραστηριότητες

$ES(k) = \max \{EF(j) : \forall \text{ δραστηριότητα } j \text{ που είναι προηγούμενη της δραστηριότητας } k\}$

$EF(k) = ES(k) + L(k)$ , όπου  $L(k)$  είναι η διάρκεια της δραστηριότητας  $k$

Υπολογισμός των LS, LF (εργαζόμαστε από τις τελικές προς τις αρχικές δραστηριότητες):

$LF(k) = EF(k)$ , όπου  $k$  είναι η τελική δραστηριότητα

$LF(k) = \min\{LS(j) : \forall \text{ δραστηριότητα } j \text{ που είναι επόμενη της δραστηριότητας } k\}$

$LS(k) = LF(k) - L(k)$

Υπολογισμός συνολικού χρονικού περιθωρίου και ελεύθερου χρονικού περιθωρίου:

Το συνολικό χρονικό περιθώριο μιας δραστηριότητας είναι:

$SL(k) = LS(k) - ES(k)$  ή  $SL(k) = LF(k) - EF(k)$

Το ελεύθερο χρονικό περιθώριο μιας δραστηριότητας είναι:

$FSL(k) = \min \{ ES(j) \} - EF(k)$ ,

όπου  $\min \{ ES(j) : \forall \text{ δραστηριότητα } j \text{ που είναι επόμενη της δραστηριότητας } k\}$

#### **Παράδειγμα 4.8**

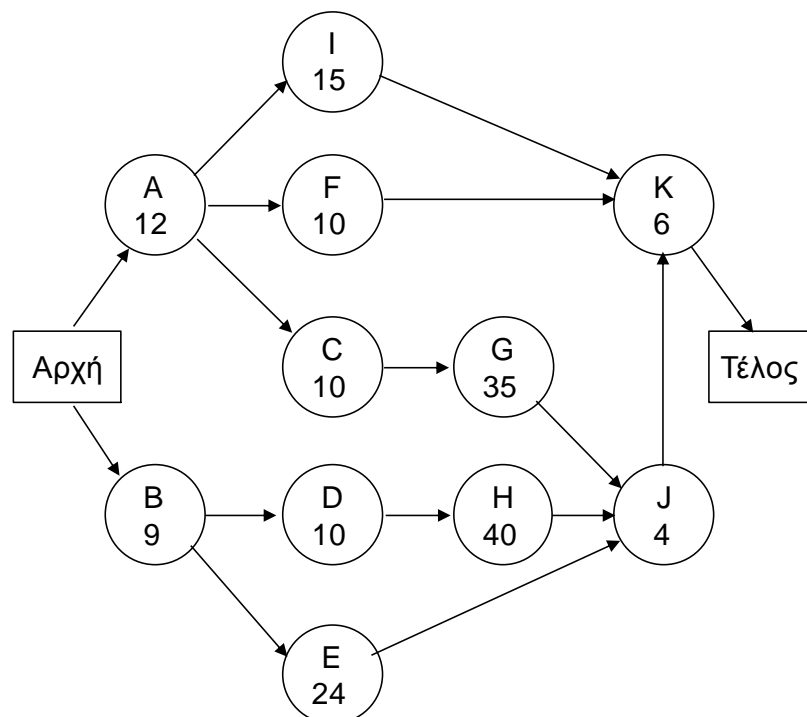
Στον παρακάτω πίνακα παρουσιάζονται οι δραστηριότητες ενός έργου ανάπτυξης ενός πληροφοριακού συστήματος, η διάρκειά τους και οι εξαρτήσεις τους.

<b>Δραστηριότητα</b>	<b>Περιγραφή</b>	<b>Προηγούμενη</b>	<b>Διάρκεια</b>
----------------------	------------------	--------------------	-----------------



Δραστηριότητα (εβδομάδες)				
A	Δημιουργία ομάδας έργου	ομάδας	-	12
B	Καταγραφή απαιτήσεων		-	9
C	Ανάλυση απαιτήσεων		A	10
D	Σχεδίαση αρχιτεκτονικής συστήματος		B	10
E	Προμήθεια εξοπλισμού		B	24
F	Λεπτομερής σχεδίαση		A	10
G	Κατασκευή συστήματος	του	C	35
H	Έλεγχος του συστήματος		D	40
I	Ολοκλήρωση του συστήματος		A	15
J	Εγκατάσταση εξοπλισμού		E,G,H	4
K	Εκπαίδευση προσωπικού		F,I,J	6

Στο Σχήμα 4.25. παρουσιάζεται το δίκτυο AON του παραπάνω έργου.



**Σχήμα 4.25:** Δίκτυο AON έργου Παραδείγματος 4.4 αναφορικά με την ανάπτυξη ενός πληροφοριακού συστήματος.

Σύμφωνα με τον ορισμό του έργου οι δραστηριότητες A και B είναι αρχικές και για το λόγο αυτό  $ES(A) = 0$  και  $ES(B) = 0$ . Η διάρκεια της δραστηριότητας A είναι 12 εβδομάδες και συνεπώς:

$$EF(A) = ES(A) + L(A) = 0 + 12 = 12$$

Με τον ίδιο τρόπο υπολογίζουμε ότι  $EF(B) = 9$ .

Συνεχίζουμε με τη δραστηριότητα C. Η δραστηριότητα C έχει ως μοναδική προηγούμενη δραστηριότητα τη δραστηριότητα A. Συνεπώς,  $ES(C) = EF(A) = 12$  και  $EF(C) = 12 + 10 = 22$ .

Με τον ίδιο τρόπο υπολογίζουμε τους χρόνους ES και EF (βλέπε πίνακα που ακολουθεί) για τις δραστηριότητες D, E, F, I, H και G. Ο υπολογισμός των ES και EF των δραστηριοτήτων J και K διαφοροποιείται, αφού τόσο η δραστηριότητα J, όσο και η δραστηριότητα K έχουν περισσότερες από μια προηγούμενες δραστηριότητες. Για το λόγο αυτό:

$$ES(J) = \max \{ EF(G), EF(H), EF(E) \} = \max \{ 57, 59, 33 \} = 59$$

$$EF(J) = 59 + 4 = 63$$

και

$$ES(K) = \max \{ EF(I), EF(F), EF(J) \} = \max \{ 27, 22, 63 \} = 63$$

$$EF(K) = 63 + 6 = 69$$

Η λογική ερμηνεία της επιλογής του μεγαλύτερου χρόνου EF είναι ότι για να ξεκινήσει μια δραστηριότητα πρέπει να ολοκληρωθούν όλες οι προηγούμενες δραστηριότητες. Επιπλέον, αυτό που καθορίζει το πότε θα ξεκινήσουμε μια δραστηριότητα είναι η ολοκλήρωση της απαιτητικότερης σε χρόνο προηγούμενης δραστηριότητας.

Δραστηριότητα	Προηγούμενη Δραστηριότητα	Διάρκεια (εβδομάδες)	ES	EF
A	-	12	0	12
B	-	9	0	9
C	A	10	12	22
D	B	10	9	19

E	B	24	9	33
F	A	10	12	22
G	C	35	22	57
H	D	40	19	59
I	A	15	12	27
J	E,G,H	4	59	63
K	F,I,J	6	63	69

Αφού έχουμε υπολογίσει τους χρόνους ES και EF μπορούμε να ξεκινήσουμε τον υπολογισμό των χρόνων LS και LF. Για τον υπολογισμό των χρόνων LS και LF ξεκινάμε να εργαζόμαστε από το τέλος του δικτύου (έργου) προς την αρχή.

Στο παράδειγμά μας η δραστηριότητα K είναι τελική και συνεπώς:

$$LF(K) = EF(K) = 69$$

$$LS(K) = LF(K) - L(K) = 69 - 6 = 63$$

Συνεχίζουμε με τον κόμβο J εφαρμόζοντας το σχετικό βήμα του αλγορίθμου:

$$LF(J) = \min\{LS(K)\} = \min\{63\} \quad (\text{υπάρχει μόνο μια επόμενη})$$

$$LS(J) = LF(J) - L(J) = 63 - 4 = 59$$

Με τον ίδιο τρόπο υπολογίζουμε τους χρόνους LS και LF για όλες τις δραστηριότητες του δικτύου (βλέπε πίνακα που ακολουθεί). Στην περίπτωση που μια δραστηριότητα έχει δύο επόμενες, όπως για παράδειγμα η δραστηριότητα B, ο υπολογισμός των LS και LF γίνεται ως ακολούθως:

$$LF(B) = \min\{LS(D), LS(E)\} = \min\{9, 35\} = 9$$

$$LS(B) = LF(B) - L(B) = 9 - 9 = 0$$

Δραστηριότητα	Προηγούμενη Δραστηριότητα	Διάρκεια (εβδομάδες)	LS	LF	SL
A	-	12	2	14	2
B	-	9	0	9	0
C	A	10	14	24	2
D	B	10	9	19	0
E	B	24	35	59	26
F	A	10	53	63	41

G	C	35	24	59	2
H	D	40	19	59	0
I	A	15	48	63	36
J	E,G,H	4	59	63	0
K	F,I,J	6	63	69	0

Στην τελευταία στήλη του παραπάνω πίνακα παρουσιάζεται το χρονικό περιθώριο της κάθε δραστηριότητας, το οποίο υπολογίζεται από τη διαφορά μεταξύ LS και ES ή LF και EF. Για παράδειγμα:

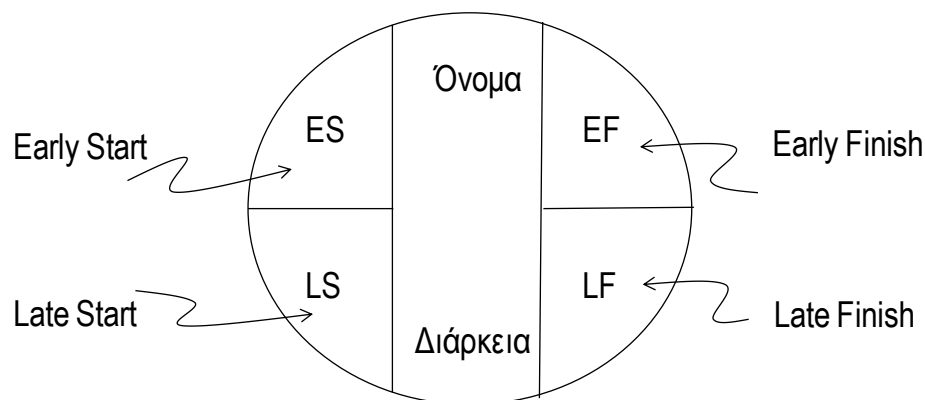
$$SL(F) = LS(F) - ES(F) = 53 - 12 = 41 \text{ ή}$$

$$SL(F) = LF(F) - EF(F) = 63 - 22 = 41$$

Οι δραστηριότητες με μηδενικό χρονικό περιθώριο δεν μπορούν να καθυστερήσουν, γιατί στην περίπτωση αυτή θα καθυστερήσει όλο το έργο. Στο παράδειγμά μας οι δραστηριότητες με μηδενικό χρονικό περιθώριο ή αλλιώς η κρίσιμη διαδρομή είναι:

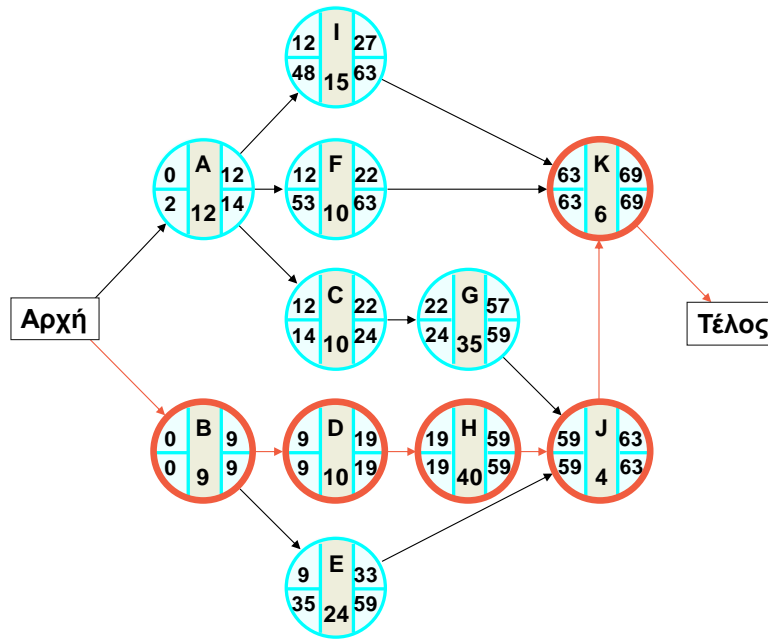
B-D-H-J-K

Εναλλακτικά μπορούμε να χρησιμοποιήσουμε γραφική απεικόνιση για να υπολογίσουμε την κρίσιμη διαδρομή, όπως αυτή παρουσιάζεται στο Σχήμα 4.26.



**Σχήμα 4.26.:** Γραφική αναπαράσταση δραστηριοτήτων.

Χρησιμοποιώντας τη γραφική αναπαράσταση η επίλυση του δικτύου μπορεί να γίνει με απλό και εύκολο τρόπο. Η χρησιμότητα όμως της μεθόδου περιορίζεται σε έργα με μικρό αριθμό δραστηριοτήτων.



Σχήμα 4.27.: Γραφική επίλυση δικτύου AON για έργο ανάπτυξης ενός πληροφοριακού συστήματος.

#### Άσκηση Αυτοαξιολόγησης 4.2

Να υπολογισθεί η κρίσιμη διαδρομή για το παρακάτω έργο:

Δραστηριότητα	Προηγούμενη Δραστηριότητα	Διάρκεια
A		2
B	A	3
C	A	2
D	B,C	4
E	D	4

#### 4.3.4 Η Αβεβαιότητα στο Χρόνο Ολοκλήρωσης των Δραστηριοτήτων και η Μέθοδος PERT.

Στη μέθοδο CPM κάνουμε την παραδοχή ότι ο χρόνος – διάρκεια των δραστηριοτήτων ενός έργου είναι γνωστός από την αρχή του έργου και καθορισμένος. Η

παραδοχή αυτή είναι πολύ περιοριστική, αφού μόνο σε λίγα έργα μπορούμε να είμαστε σίγουροι από την αρχή του έργου για τη διάρκεια μιας δραστηριότητας. Για το λόγο αυτό η μέθοδος CPM δε δίνει ακριβή αποτελέσματα όταν στο έργο υπάρχει αβεβαιότητα σχετική με τη διάρκεια των δραστηριοτήτων. Στα έργα πληροφορικής είναι πολύ συνηθισμένο φαινόμενο να μη γνωρίζουμε με ακρίβεια το χρόνο που θα απαιτηθεί, αφού δεν έχουν καταγραφεί με λεπτομέρεια όλες οι απαιτήσεις του συστήματος.

Με σκοπό να βελτιώσουμε τις εκτιμήσεις μας σχετικά με τα έργα, εισάγουμε για κάθε δραστηριότητα τρεις διαφορετικούς χρόνους ολοκλήρωσης. Οι χρόνοι αυτοί είναι:

- Ο αισιόδοξος χρόνος ολοκλήρωσης –  $a$ .
- Ο πιο πιθανός χρόνος ολοκλήρωσης -  $m$ .
- Ο απαισιόδοξος χρόνος ολοκλήρωσης -  $b$ .

Με βάση τους τρεις αυτούς χρόνους ολοκλήρωσης μπορούμε να υπολογίσουμε τον αναμενόμενο χρόνο ολοκλήρωσης (TE) της δραστηριότητας, καθώς και την τυπική απόκλιση ( $\sigma$ ). Οι υπολογισμοί αυτοί γίνονται κάνοντας την παραδοχή ότι ο χρόνος ολοκλήρωσης της δραστηριότητας ακολουθεί την κατανομή beta. Πιο συγκεκριμένα:

$$TE = \frac{a + 4m + b}{6}$$

και

$$\sigma = \sqrt{\frac{b - a}{6}}$$

#### **Παράδειγμα 4.9**

Έστω το έργο του Παραδείγματος 4.1. Στον παρακάτω πίνακα παρουσιάζονται οι αναμενόμενοι χρόνοι και οι τυπικές αποκλίσεις για κάθε δραστηριότητα του έργου αυτού.

Για τη δραστηριότητα A έχουμε ότι:

$$TE_A = \frac{a + 4m + b}{6} = \frac{2 + 4 \times 5 + 8}{6} = 5$$

$$\sigma = \sqrt{\frac{b - a}{6}} = \sqrt{\frac{8 - 2}{6}} = \sqrt{1} = 1$$

Με τον ίδιο τρόπο υπολογίζουμε τους χρόνους για τις υπόλοιπες δραστηριότητες.

---

Δραστηριότητα	Αισιόδοξος Χρόνος	Πιθανός Χρόνος	Απαισιόδοξος Χρόνος	Αναμενόμενος Χρόνος (TE)	Τυπική Απόκλιση
---------------	-------------------	----------------	---------------------	--------------------------	-----------------

---

	(α)	(m)	(b)		(σ)
A	2	5	8	5	1
B	1	3	5	3	0,66
C	7	8	9	8	0,33
D	4	7	10	7	1
E	6	7	8	7	0,33
F	2	4	6	4	0,66
G	4	5	6	5	0,33

Αν θεωρήσουμε ότι ο χρόνος κάθε δραστηριότητας είναι μια τυχαία μεταβλητή, τότε μπορούμε να υπολογίσουμε τον αναμενόμενο χρόνο ολοκλήρωσης του έργου ως το άθροισμα των αναμενόμενων χρόνων ολοκλήρωσης των δραστηριοτήτων του έργου που είναι στην κρίσιμη διαδρομή. Δηλαδή:

$$TE_{\text{έργου}} = TE_1 + TE_2 + \dots + TE_n$$

Στη συνέχεια, εφαρμόζουμε το κεντρικό οριακό θεώρημα (central limit theorem) για να υπολογίσουμε την πιθανότητα ολοκλήρωσης του έργου για μια δεδομένη χρονική στιγμή με τον ακόλουθο τύπο:

$$Z = \frac{\tau - TE_{\text{έργου}}}{\sqrt{\sigma_{\text{έργου}}^2}}$$

Στον τύπο αυτό,  $\tau$  είναι ο χρόνος ολοκλήρωσης του έργου, την πιθανότητα του οποίου θέλουμε να υπολογίσουμε,  $TE_{\text{έργου}}$  είναι ο αναμενόμενος χρόνος ολοκλήρωσης του έργου,  $\sigma_{\text{έργου}}^2$  το άθροισμα των αποκλίσεων των κρίσιμων δραστηριοτήτων και  $Z$  ο αριθμός των τυπικών αποκλίσεων μιας τυπικής κανονικής κατανομής. Έχοντας υπολογίσει το  $Z$  μπορούμε να υπολογίσουμε την πιθανότητα ολοκλήρωσης του έργου σε χρόνο μικρότερο από  $\tau$  με τη χρήση του πίνακα αθροιστικής πιθανότητας της κανονικής κατανομής (βλ.

**Παράρτημα).**

#### **Παράδειγμα 4.10**

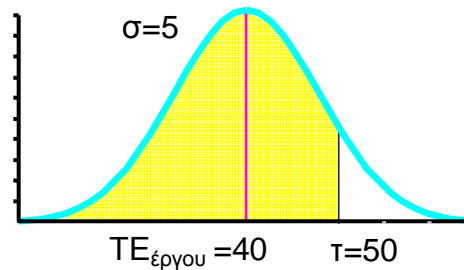
Ποια είναι η αναμενόμενη πιθανότητα ολοκλήρωσης ενός έργου διάρκειας 40 εβδομάδων με τυπική απόκλιση 5 εβδομάδες; Ποια είναι η πιθανότητα να τελειώσει το έργο σε 50 εβδομάδες ή λιγότερο.

#### **Απάντηση**

Εφαρμόζουμε τον τύπο:

$$Z = \frac{\tau - TE_{\text{έργου}}}{\sqrt{\sigma_{\text{έργου}}^2}} = \frac{50 - 40}{5} = 2,0$$

Ανατρέχοντας στον πίνακα αθροιστικής πιθανότητας της κανονικής κατανομής που παρουσιάζεται στο Παράρτημα βρίσκουμε ότι όταν  $Z=2,0$ , τότε  $P(X \leq 2,0) = 0,97725$



**Σχήμα 4.28.:** Κατανομή πιθανότητας για το χρόνο ολοκλήρωσης του έργου.

Στο Σχήμα 4.12. παρουσιάζεται με γραφικό τρόπο ο υπολογισμός της πιθανότητας ολοκλήρωσης του έργου σε λιγότερο από 50 εβδομάδες. Η σκιασμένη περιοχή κάτω από την καμπύλη και αριστερά του σημείου  $\tau=50$  δίνει αυτή ακριβώς την πιθανότητα. Επειδή κάνουμε την παραδοχή ότι οι χρόνοι ολοκλήρωσης του έργου ακολουθούν την κανονική κατανομή είναι εύκολο να υπολογίσουμε το εμβαδόν για  $\tau < 50$  ανατρέχοντας στο σχετικό πίνακα του παραρτήματος, και συνεπώς η πιθανότητα να τελειώσει το έργο σε λιγότερο από 50 εβδομάδες είναι 97,725 %.

### Άσκηση Αυτοαξιολόγησης 4.3

Δίνεται το παρακάτω έργο:

Δραστηριότητα	Προηγούμενη Δραστηριότητα	Αισιόδοξος Χρόνος(a)	Πιθανός Χρόνος	Απαισιόδοξος Χρόνος(b)
A		1	4	7
B	-	2	2	2
C	A	2	5	8
D	A	3	4	5
E	C, B	4	6	8
F	C, B	0	0	6



G

D, E

3

6

9

Κατασκευάστε το δίκτυο AON και στη συνέχεια υπολογίστε:

1. Τους αναμενόμενους χρόνους ολοκλήρωσης TE, την απόκλιση  $\sigma^2$  και το χρονικό περιθώριο των δραστηριοτήτων (SL).
2. Την κρίσιμη διαδρομή και τον αναμενόμενο χρόνο ολοκλήρωσης του έργου  $TE_{\text{έργου}}$ .
3. Την πιθανότητα ολοκλήρωσης του έργου σε 23 ημέρες.
4. Τη διάρκεια του έργου που έχει πιθανότητα ολοκλήρωσης 99%.

Συνοψίζοντας, τα βήματα που θα πρέπει να εφαρμόζει κανείς για τη μελέτη ενός έργου με τη μέθοδο PERT είναι τα ακόλουθα:

- 1) Για κάθε δραστηριότητα k, αξιολογούμε την κατανομή πιθανότητας για το χρόνο εκτέλεσης των δραστηριοτήτων ή θεωρούμε ότι ακολουθεί την κατανομή beta και κάνουμε τρεις εκτιμήσεις (απαισιόδοξη, πιο πιθανή, αισιόδοξη). Οι εκτιμήσεις αυτές γίνονται από το διοικητή του έργου ή από τα πιο έμπειρα μέλη της ομάδας του έργου.
- 2) Υπολογίζουμε τον αναμενόμενο χρόνο εκτέλεσης (TE) της κάθε δραστηριότητας, καθώς και την απόκλιση αυτής ( $\sigma^2$ ).
- 3) Χρησιμοποιούμε τη μέθοδο CPM για να υπολογίσουμε την κρίσιμη διαδρομή.
- 4) Στη συνέχεια μπορούμε να υπολογίσουμε την αναμενόμενη διάρκεια του έργου το  $TE_{\text{έργου}}$ , καθώς και την πιθανότητα ολοκλήρωσης του έργου για μια δεδομένη διάρκεια.

#### 4.3.5 Συντόμευση Διάρκειας Έργου.

Στις προηγούμενες παραγράφους μελετήσαμε τις μεθόδους CPM και PERT με τις οποίες κατασκευάζουμε το χρονοδιάγραμμα του έργου και προσδιορίζουμε την κρίσιμη διαδρομή, τη συνολική διάρκεια του έργου, καθώς και την πιθανότητα ολοκλήρωσης του έργου σε μια δεδομένη χρονική στιγμή. Όμως, σε πολλές περιπτώσεις ο χρόνος που απαιτείται για την ολοκλήρωση του έργου δεν ικανοποιεί τις απαιτήσεις και συνεπώς υπάρχει ανάγκη για τη συντόμευση της διάρκειας του έργου. Στην περίπτωση αυτή θα

πρέπει να είμαστε σε θέση να καθορίσουμε ποιες δραστηριότητες θα πρέπει να συντομευθούν, ώστε να έχουμε τη βέλτιστη συντόμευση του έργου συνολικά (Υψηλάντης, 2003; Kerzner, 2003).

Για τη συντόμευση του έργου χρησιμοποιούμε τον παρακάτω αλγόριθμο:

- 1) Κατασκευάζουμε το δίκτυο CPM/PERT.
- 2) Υπολογίζουμε την κρίσιμη διαδρομή.
- 3) Για τις δραστηριότητες της κρίσιμης διαδρομής βρίσκουμε αυτή με το μικρότερο κόστος συντόμευσης.
- 4) Συντομεύουμε το έργο κατά μια χρονική μονάδα.
- 5) Υπολογίζουμε το νέο κόστος για το έργο και την κρίσιμη διαδρομή.
- 6) Ελέγχουμε εάν ικανοποιούνται οι χρονικοί περιορισμοί του έργου. Εάν ικανοποιούνται ο αλγόριθμος τερματίζει, αλλιώς πηγαίνουμε στο βήμα 3.

#### **Παράδειγμα 4.11**

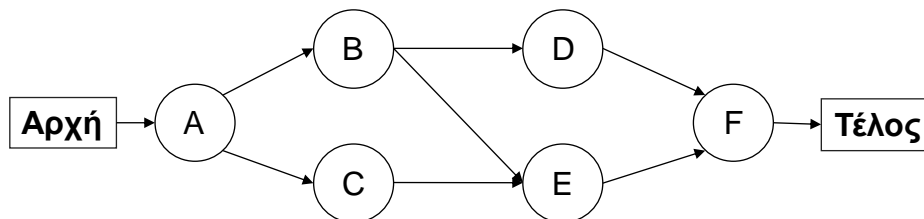
Στον παρακάτω πίνακα, η στήλη με τίτλο «κανονική διάρκεια» μας δίνει τη συνολική διάρκεια της δραστηριότητας, ενώ η στήλη με τίτλο «διάρκεια συντόμευσης» παρουσιάζει πόσες εβδομάδες μπορεί να συντομευθεί μια δραστηριότητα. Εάν και πόσο μπορεί να συντομευθεί μια δραστηριότητα εξαρτάται από το είδος της συγκεκριμένης δραστηριότητας και τους διαθέσιμους πόρους στο έργο. Προφανώς δεν είναι δυνατόν να συντομεύσουμε όλες τις δραστηριότητες. Στο παράδειγμα μας, η «διάρκεια συντόμευσης» είναι δεδομένο εισόδου, καθώς δεν μας απασχολεί ο υπολογισμός της. Αντίστοιχα, η στήλη με τίτλο «κανονικό κόστος» παρουσιάζει το συνολικό κόστος της δραστηριότητας, ενώ η στήλη με τίτλο «κόστος συντόμευσης» παρουσιάζει το κόστος συντόμευσης μιας δραστηριότητας ανά εβδομάδα.

<b>Δραστηριότητα</b>	<b>Προηγούμενη Δραστηριότητα</b>	<b>Κανονική Διάρκεια</b>	<b>Διάρκεια Συντόμευσης</b>	<b>Κανονικό Κόστος</b>	<b>Κόστος Συντόμευσης</b>
<b>α</b>	<b>α</b>	<b>α</b>	<b>ς</b>	<b>ς</b>	<b>ς</b>

Δραστηριότητα α	Προηγούμενη Δραστηριότητα α	Κανονική Διάρκεια α	Διάρκεια Συντόμευση ς	Κανονικό Κόστος	Κόστος Συντόμευσης ς
A	-	4	2	10000	2000
B	A	6	1	30000	12500
C	A	2	1	8000	1500
D	B	2	1	12000	6000
E	B, C	7	3	40000	6000
F	D, E	6	3	20000	3000

Έστω ότι θέλουμε να εκτελέσουμε το πιο πάνω έργο σε χρονικό διάστημα 15 εβδομάδων, δηλαδή συντομότερα κατά 8 εβδομάδες σε σχέση με την αρχική του διάρκεια. Για το λόγο αυτό εφαρμόζουμε τον αλγόριθμο της υποενότητας 4.3.5:

- 1) Κατασκευάζουμε το δίκτυο δραστηριοτήτων και υπολογίζουμε την κρίσιμη διαδρομή:

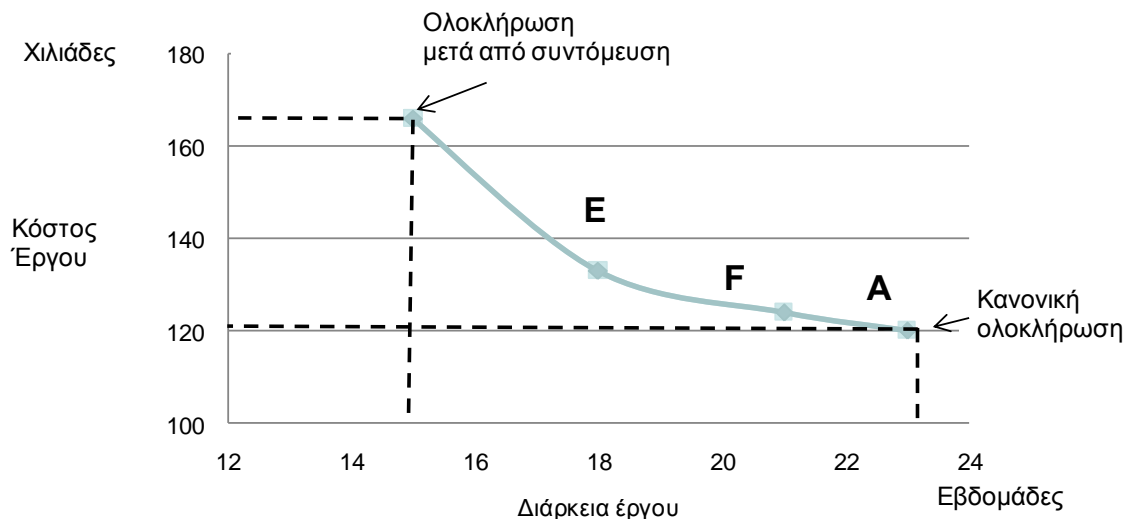


- 2) Η κρίσιμη διαδρομή είναι A-B-E-F και έχει διάρκεια 23 εβδομάδων (δεν παρουσιάζεται αναλυτικά ο υπολογισμός).
- 3) Εξετάζουμε τις δραστηριότητες της κρίσιμης διαδρομής και βλέπουμε ότι από τις δραστηριότητες A, B, E και F αυτή που έχει το ελάχιστο κόστος συντόμευσης είναι η δραστηριότητα A, η οποία μπορεί να συντομευθεί κατά δύο εβδομάδες με συνολικό κόστος 4000 €.
- 4) Το συνολικό κόστος του έργου είναι τώρα 124000 € από 120000 € που ήταν αρχικά. Η κρίσιμη διαδρομή δεν αλλάζει γιατί η δραστηριότητα A είναι η μόνη αρχική δραστηριότητα.
- 5) Η διάρκεια του έργου είναι 21 εβδομάδες και όχι 15 εβδομάδες που θέλουμε. Συνεπώς, συνεχίζουμε να μειώνουμε τη διάρκεια του έργου (πηγαίνουμε πάλι στο βήμα 3).

Συνεχίζοντας βρίσκουμε ότι η επόμενη δραστηριότητα της κρίσιμης διαδρομής με το ελάχιστο κόστος συντόμευσης είναι η δραστηριότητα F, η οποία μπορεί να συντομευθεί για τρεις εβδομάδες με συνολικό κόστος 9000 €. Η διάρκεια του έργου μειώνεται στις 18 εβδομάδες, ενώ το συνολικό κόστος αυξάνεται στις 133000 €. Η κρίσιμη διαδρομή παραμένει η ίδια γιατί η δραστηριότητα F είναι η μοναδική τελική δραστηριότητα.

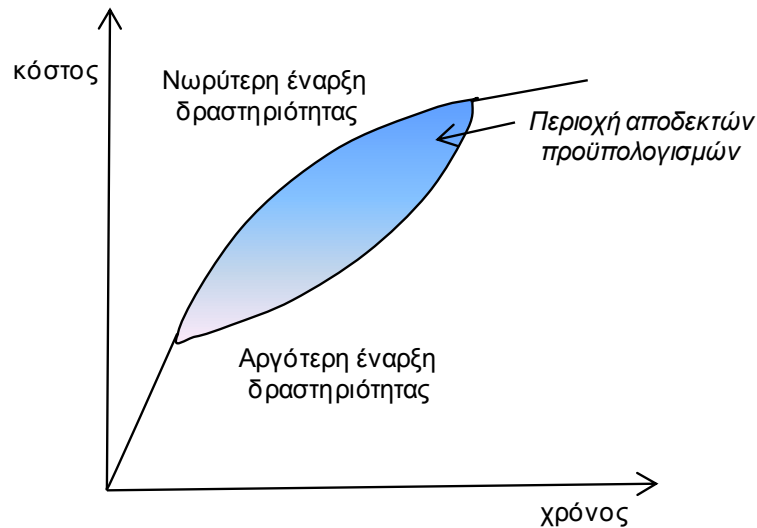
Η επόμενη δραστηριότητα που συντομεύουμε είναι η δραστηριότητα E, η οποία μπορεί να συντομευθεί για τρεις εβδομάδες με κόστος 6000 € ανά εβδομάδα. Μειώνουμε τη διάρκεια της δραστηριότητας E κατά τρεις εβδομάδες, ώστε η συνολική διάρκεια του έργου να μειωθεί στις 15 εβδομάδες που είναι το ζητούμενο. Αντίστοιχα, το συνολικό κόστος αυξάνεται από τις 133000 € στις 151000 €.

Στο Σχήμα 4.29. παρουσιάζεται η σχέση μεταξύ της μείωσης της διάρκειας και της αύξησης του κόστους του έργου.



**Σχήμα 4.29.:** Κόστος συντόμευσης σε σχέση με τη διάρκεια.

Ο γενικότερος στόχος της διαδικασίας συντόμευσης των δραστηριοτήτων είναι η βέλτιστη χρήση των πόρων, δηλαδή να μεγιστοποιήσουμε την χρήση των πόρων μας και ειδικά των ανθρώπινων, ενώ ταυτόχρονα να ελαχιστοποιήσουμε το κόστος. Η βέλτιστη λύση για τη διάρκεια του έργου σε σχέση με το κόστος βρίσκεται στη σκιασμένη περιοχή του Σχήματος 4.14., η οποία ορίζεται από τους νωρίτερους και από τους αργότερους χρόνους έναρξης των δραστηριοτήτων.



Σχήμα 4.30.: Περιοχή αποδεκτών προϋπολογισμών.

#### 4.3.6 Σχολιασμός των Μεθόδων CPM και PERT.

Αν και οι μέθοδοι CPM και PERT είναι ευρέως διαδεδομένες και ιδιαίτερα χρήσιμες υπάρχουν σημαντικά προβλήματα στην εφαρμογή των μεθόδων για τους ακόλουθους λόγους:

- 1) Υπάρχει αντικειμενική δυσκολία στην εκτίμηση του χρόνου των δραστηριοτήτων. Στα μεγάλα και σύνθετα έργα υπάρχει αντικειμενική δυσκολία στο να ορίσουμε με ακρίβεια τις δραστηριότητες του έργου, αλλά και τη διάρκειά τους.
- 2) Η χρήση της κατανομής beta για τη μοντελοποίηση της διάρκειας των δραστηριοτήτων δεν είναι πάντα η καταλληλότερη επιλογή. Ο λόγος είναι ότι η απόκλιση της διάρκειας του έργου προκύπτει ως άθροισμα των αποκλίσεων της διάρκειας των δραστηριοτήτων, γεγονός που μπορεί να οδηγήσει σε συστηματικό λάθος. Επιπλέον, σε πολλές περιπτώσεις είναι ιδιαίτερα δύσκολο να παράγουμε τρεις εκτιμήσεις για τη διάρκεια των δραστηριοτήτων και η τυχαία εκτίμηση της αισιόδοξης και απαισιόδοξης διάρκειας δε βελτιώνει τη γενικότερη πρόβλεψη.
- 3) Δίνεται μεγάλη έμφαση στις δραστηριότητες της κρίσιμης διαδρομής. Δίνοντας έμφαση στην κρίσιμη διαδρομή παραβλέπουμε τη διαθεσιμότητα των πόρων, η έλλειψη των οποίων οδηγεί το έργο σε καθυστέρηση.

### ΕΝΟΤΗΤΑ 4.4 Διαχείριση Πόρων.

Στις προηγούμενες παραγράφους μελετήσαμε τις μεθόδους CPM και PERT με τις οποίες κατασκευάζουμε το χρονοδιάγραμμα του έργου και προσδιορίζουμε την κρίσιμη

διαδρομή. Η μελέτη αυτή βασίσθηκε στην παραδοχή ότι μόλις τελειώνει μια δραστηριότητα ξεκινάει αμέσως η επόμενη, δηλαδή όλοι οι περιορισμοί είναι “Λήξη-για-Έναρξη” (Finish-to-Start, FS). Αυτό σημαίνει ότι υπάρχουν αρκετοί πόροι για την εκτέλεση όλων των δραστηριοτήτων του έργου. Προφανώς, αυτό δεν ισχύει στα περισσότερα έργα. Η διαχείριση πόρων (resource management) είναι η διαδικασία της απόκτησης πόρων, της ανάθεσης πόρων σε δραστηριότητες και της παρακολούθησης εκτέλεσης των δραστηριοτήτων σε σχέση με την απόδοση των πόρων του έργου.

Η διαδικασία της διαχείρισης πόρων σε ένα έργο είναι εξαιρετικά πολύπλοκη ιδιαίτερα σε οργανισμούς που εκτελούν ταυτόχρονα περισσότερα από ένα έργα. Οι λόγοι που συμβάλουν στην πολυπλοκότητα είναι οι ακόλουθοι:

- Η επιλογή των πόρων θα πρέπει να γίνεται έχοντας ως στόχο, όχι μόνο τις ανάγκες του έργου, αλλά και τη βέλτιστη χρήση ενός πόρου (π.χ. ένας υπολογιστής απαραίτητος για το έργο μας, μια ημέρα την εβδομάδα δε χρησιμοποιείται με βέλτιστο τρόπο).
- Το κόστος των πόρων που κοστολογούνται ανά μονάδα χρήσης μειώνεται με την ποσότητα.
- Υπάρχει ένα σημείο πέρα από το οποίο η ανάθεση περισσότερων πόρων σε μια δραστηριότητα δε μειώνει τη διάρκειά της. Για παράδειγμα, μπορεί η ανάπτυξη μιας εφαρμογής να απαιτεί 15 ανθρωπομήνες από έναν προγραμματιστή, 7,5 ανθρωπομήνες από δύο προγραμματιστές, αλλά σε καμία περίπτωση δεν μπορούμε να πούμε ότι 15 προγραμματιστές θα τελειώσουν το έργο σε ένα μήνα.
- Κάποιοι πόροι είναι διακριτοί (π.χ. δεν είναι δυνατό να εργάζονται 3,2 προγραμματιστές για την εκτέλεση ενός έργου).
- Οι πόροι ανήκουν στον οργανισμό που εκτελεί το έργο. Συνεπώς, θα πρέπει να λαμβάνουμε υπόψη, όχι μόνο τη βέλτιστη χρήση για το έργο, αλλά και για τον οργανισμό στο οποίο ανήκει ο πόρος.

Ανάλογα με τη διαθεσιμότητα των πόρων μπορούμε να ταξινομήσουμε τους πόρους σε:

1. Ανανεώσιμους (renewable), όπως για παράδειγμα το προσωπικό.
2. Μη ανανεώσιμους. Είναι οι πόροι που εξαντλούνται με τη χρήση τους. Όταν εξαντληθεί ένας τέτοιος πόρος αναπληρώνεται με νέα παραγγελία. Τυπικό παράδειγμα μη ανανεώσιμου πόρου είναι τα υλικά κάθε είδους.

3. Περιορισμένους ανά χρονική περίοδο (constrained), καθώς και συνολικά για όλο το έργο (double constrained). Τα διαθέσιμα χρήματα για το έργο αποτελούν το καλύτερο παράδειγμα περιορισμένου πόρου.

#### 4.4.1 Εξισορρόπηση Πόρων.

Για να κατανοήσουμε τη διαδικασία της εξισορρόπησης πόρων σε ένα έργο θα πρέπει να κατανοήσουμε τη σχέση μεταξύ εργασίας, πόρων και διάρκειας. Για κάθε δραστηριότητα έχουμε την ποσότητα εργασίας που απαιτείται και συνήθως εκφράζεται σε ανθρωποημέρες ή ανθρωπομήνες, την ποσότητα των πόρων που θα απασχοληθεί στη συγκεκριμένη δραστηριότητα και τη χρονική διάρκεια της δραστηριότητας, η οποία εξαρτάται από την ποσότητα των πόρων που απασχολείται σε αυτή. Η διάρκεια της δραστηριότητας προκύπτει από την ποσότητα εργασίας διά την ποσότητα των πόρων που θα χρησιμοποιηθούν. Ένα τέταρτο μέγεθος που εξαρτάται από τις παραπάνω παραμέτρους είναι το κόστος της δραστηριότητας, το οποίο είναι ιδιαίτερα σημαντικό για τη διοίκηση του έργου.

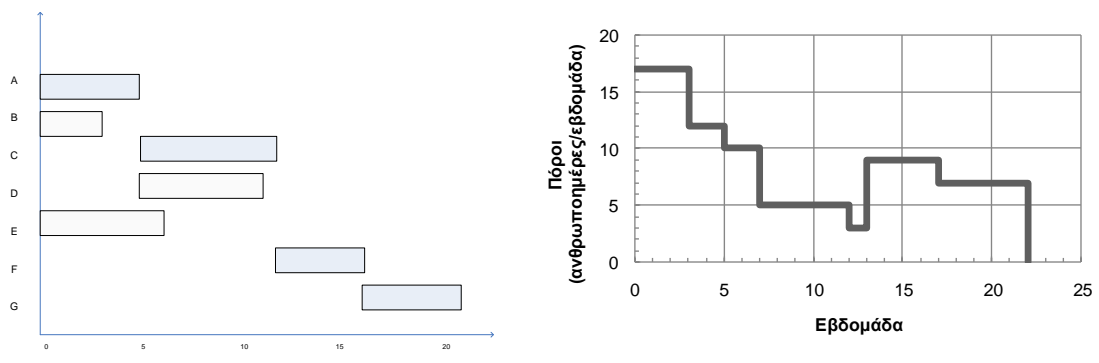
##### **Παράδειγμα 4.12**

Για να παρουσιάσουμε τη σχέση μεταξύ χρονοδιαγράμματος και απαιτήσεων σε πόρους θα χρησιμοποιήσουμε το έργο του Παραδείγματος 4.1. Στον ακόλουθο πίνακα παρουσιάζονται οι απαιτήσεις σε πόρους για κάθε δραστηριότητα του έργου αυτού.

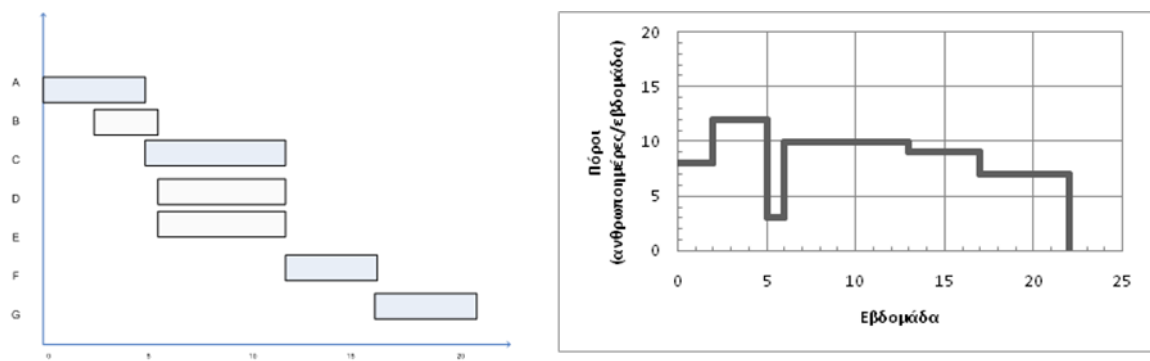
<b>Δραστηριότητα</b>	<b>Προηγούμενη Δραστηριότητα</b>	<b>Διάρκεια (εβδομάδες)</b>	<b>Απαιτούμενοι Πόροι ανά Εβδομάδα</b>	<b>Συνολικοί Απαιτούμενοι Πόροι</b>	<b>SL</b>	<b>FS</b>
A	-	5	8	40	0	0
B	-	3	4	12	3	2
C	A	8	3	24	0	0
D	A,B	7	2	14	1	1
E	-	7	5	35	6	6
F	C, E, D	4	9	36	0	0
G	F	5	7	35	0	0

Στον παραπάνω πίνακα οι πόροι καταναλώνονται με σταθερό ρυθμό, δηλαδή για κάθε εβδομάδα που διαρκεί η δραστηριότητα A καταναλώνουμε την ίδια ποσότητα πόρων. Σε περίπτωση που οι ανάγκες σε πόρους είναι μεταβαλλόμενες θα πρέπει να ορίσουμε τις ανάγκες ανά χρονική μονάδα (ημέρα ή εβδομάδα).

Στο Σχήμα 4.31. παρουσιάζεται το διάγραμμα Gantt, καθώς και το διάγραμμα απαιτήσεων πόρων για χρονοδιάγραμμα με νωρίτερη έναρξη δραστηριοτήτων. Αντίστοιχα, στο Σχήμα 4.32. παρουσιάζεται το ίδιο έργο, αλλά με τις δραστηριότητες να ξεκινούν την αργότερη δυνατή ημερομηνία.



**Σχήμα 4.31.:** Διάγραμμα Gantt και διάγραμμα απαιτήσεων πόρων για χρονοδιάγραμμα με νωρίτερη έναρξη δραστηριοτήτων.



**Σχήμα 4.32.:** Διάγραμμα Gantt και διάγραμμα απαιτήσεων πόρων για χρονοδιάγραμμα με αργότερη έναρξη δραστηριοτήτων.

Το πρώτο διάγραμμα Gantt (βλ. Σχήμα 4.31.), όπου οι δραστηριότητες ξεκινούν την νωρίτερη ημερομηνία έχει μεγαλύτερες απαιτήσεις σε πόρους στην αρχή του έργου, αφού



τις τρεις πρώτες εβδομάδες του έργου απαιτούνται 17 ανθρωποημέρες ανά εβδομάδα. Αυτό σημαίνει ότι απαιτούνται 3,4 εργαζόμενοι ανά ημέρα, απαίτηση που μπορεί να καλυφθεί από τρεις εργαζόμενους πλήρους απασχόλησης και από υπερωρίες ή από κάποιον εργαζόμενο μερικής απασχόλησης. Μια δεύτερη παρατήρηση είναι ότι οι απαιτήσεις σε πόρους μεταβάλλονται σημαντικά κατά τη διάρκεια του έργου, από το μέγιστο των 17 ανθρωποημερών ανά εβδομάδα, στο ελάχιστο των 3 ανθρωποημερών ανά εβδομάδα. Αυτή η σημαντική διαφοροποίηση μπορεί να προκαλέσει δυσκολίες στην ανάθεση των πόρων και τελικά αύξηση στο κόστος του έργου.

Αντίστοιχα, στο δεύτερο διάγραμμα Gantt (βλ. Σχήμα 4.32.) οι απαιτήσεις σε πόρους δεν είναι τόσο συγκεντρωμένες, αλλά παρουσιάζουν μια πιο ομοιόμορφη εικόνα. Επίσης, η διαφορά μεταξύ μέγιστου και ελάχιστου αριθμού απαιτούμενων πόρων είναι μικρότερη.

---

Επομένως, στόχος της εξισορρόπησης πόρων είναι να ανακατανεύσουμε το χρονικό περιθώριο των δραστηριοτήτων, ώστε να μειώσουμε τις αυξομειώσεις στις απαιτήσεις πόρων. Όπως αναφέραμε και πριν, η αυξομείωση στις απαιτήσεις πόρων οδηγεί στην αύξηση κόστους. Όταν αναφερόμαστε σε ανθρώπινους πόρους σημαίνει πρόσληψη νέου προσωπικού, απόλυση υπάρχοντος προσωπικού, προσωπικό σε διαθεσιμότητα, ανάθεση εργασιών σε εξωτερικούς συνεργάτες, κ.λπ. Αντίστοιχα, αυξομείωση στα υλικά προκαλούν αυξημένες προμήθειες, υψηλή διαθεσιμότητα εξοπλισμού που δε χρησιμοποιείται πλήρως, κ.λπ.

Η εξισορρόπηση πόρων μπορεί να γίνει με πολλούς διαφορετικούς τρόπους ανάλογα με τις παραμέτρους του προβλήματος. Ένας γενικός αλγόριθμος που μπορεί να χρησιμοποιηθεί είναι ο ακόλουθος:

1. Υπολογίζουμε το μέσο όρο των ανθρωποημερών που απαιτούνται ανά χρονική μονάδα και ανά δραστηριότητα.
2. Δημιουργούμε το χρονοδιάγραμμα με τις δραστηριότητες να ξεκινούν την νωρίτερη δυνατή ημερομηνία.
3. Χρησιμοποιούμε το χρονοδιάγραμμα και επιλέγουμε τις μη κρίσιμες δραστηριότητες:
  - (α) Καθυστερούμε τις δραστηριότητες που έχουν το μεγαλύτερο χρονικό περιθώριο.
  - (β) Για κάθε αλλαγή ελέγχουμε τις απαιτήσεις πόρων του νέου χρονοδιαγράμματος.
4. Επιλέγουμε το χρονοδιάγραμμα που ελαχιστοποιεί τις αυξομειώσεις στις απαιτήσεις πόρων.

### **Παράδειγμα 4.13**

Συνεχίζοντας με το έργο του Παραδείγματος 4.7 παρατηρούμε ότι η δραστηριότητα E έχει το μεγαλύτερο ελεύθερο χρονικό περιθώριο (έξι εβδομάδων). Αν καθυστερήσουμε την έναρξη της δραστηριότητας E για 3 εβδομάδες, ώστε να ξεκινήσει μετά τη λήξη της δραστηριότητας B, έχουμε ως αποτέλεσμα τη μείωση των απαιτήσεων σε πόρους τις τρεις πρώτες εβδομάδες.

Οι νέες απαιτήσεις σε πόρους παρουσιάζονται στον παρακάτω πίνακα:

<b>Εβδομάδα</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
Αρχικές απαιτήσεις	17	17	17	13	13	10	10	5	5	5	5
Νέες απαιτήσεις	12	12	12	13	13	10	10	10	10	10	5
<b>Εβδομάδα</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>
Αρχικές απαιτήσεις	5	3	9	9	9	9	7	7	7	7	7
Νέες απαιτήσεις	5	3	9	9	9	9	7	7	7	7	7

Παρατηρούμε ότι η μέγιστη απαίτηση σε πόρους μετά τη μετακίνηση του E είναι 12 αντί για 17 που ήταν αρχικά, ενώ η ελάχιστη παραμένει 3. Έτσι, η διαφορά της μέγιστης από την ελάχιστη απαίτηση σε πόρους είναι τώρα  $12-3=9$ , αντί για  $17-3=14$  που ήταν.

Επειδή η μέγιστη απαίτηση παραμένει τις εβδομάδες 4 και 5 μπορούμε να μετακινήσουμε ξανά τη δραστηριότητα E, ώστε να ξεκινά μετά τη λήξη της A. Οι νέες απαιτήσεις σε πόρους παρουσιάζονται στον παρακάτω πίνακα:

<b>Εβδομάδα</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
Νέες απαιτήσεις	12	12	12	8	8	10	10	10	10	10	10
<b>Εβδομάδα</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>
Νέες απαιτήσεις	10	3	9	9	9	9	7	7	7	7	7

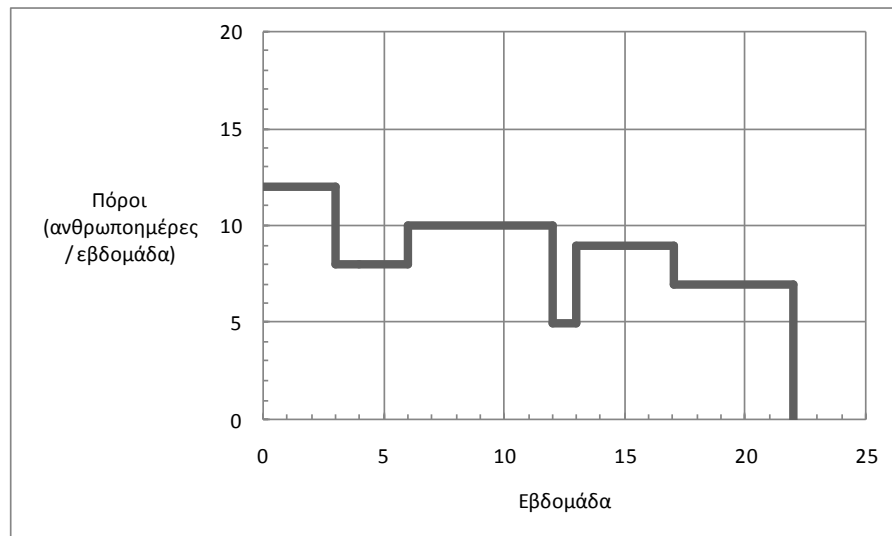
Η διαφορά της μέγιστης από την ελάχιστη απαίτηση σε πόρους είναι τώρα  $12-3=9$ . Η επόμενη δραστηριότητα που θα μπορούσε να μετακινηθεί είναι η B που έχει χρονικό περιθώριο 2 εβδομάδων. Όμως, η μετακίνηση της δραστηριότητας B για μια ή δύο εβδομάδες αυξάνει τις απαιτήσεις σε πόρους από 8 σε 12, γεγονός που δεν αποτελεί βελτίωση.

Για το λόγο αυτό αφήνουμε τη δραστηριότητα B και μετακινούμε την επόμενη δραστηριότητα με χρονικό περιθώριο, δηλαδή τη δραστηριότητα D. Αν καθυστερήσουμε την έναρξη της δραστηριότητας D κατά μία εβδομάδα, τότε προκύπτει ο παρακάτω πίνακας απαιτήσεων.

<b>Εβδομάδα</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
Νέες απαιτήσεις	12	12	12	8	8	8	10	10	10	10	10
<b>Εβδομάδα</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>
Νέες απαιτήσεις	10	5	9	9	9	9	7	7	7	7	7

Η διαφορά της μέγιστης από την ελάχιστη απαίτηση σε πόρους είναι τώρα  $12-5=7$ , που είναι και η βέλτιστη για το δεδομένο έργο.

Το Σχήμα 4.33. παρουσιάζει με γραφικό τρόπο τις απαιτήσεις σε πόρους μετά τη διαδικασία της εξισορρόπησης.



**Σχήμα 4.33.:** Απαιτήσεις πόρων μετά από εξισορρόπηση.

Στην περίπτωση που οι πόροι για την έγκαιρη περάτωση του έργου δεν επαρκούν, τότε θα πρέπει να μεταθέσουμε το τέλος του έργου και να εκτελέσουμε το έργο με μικρότερο αριθμό πόρων. Αυτό βέβαια ισχύει στην περίπτωση που μια δραστηριότητα μπορεί να εκτελεστεί με μικρότερο αριθμό πόρων. Άλλοι τρόποι για την αντιμετώπιση της έλλειψης πόρων είναι η κατάτμηση των δραστηριοτήτων σε άλλες μικρότερες, η τροποποίηση των περιορισμών του δικτύου δραστηριοτήτων ή η αναζήτηση εναλλακτικών πόρων.

Η αναζήτηση λύσεων για την ικανοποίηση των περιορισμών στο χρονοπρογραμματισμό πόρων (constraint resource scheduling) είναι ιδιαίτερα σύνθετη επειδή τα προβλήματα που προκύπτουν σε πραγματικά έργα έχουν μεγάλο αριθμό παραμέτρων και από τη φύση τους δεν είναι γραμμικά. Για την επίλυση τέτοιων προβλημάτων χρησιμοποιούνται συνήθως δύο στρατηγικές:

- Ευριστικοί αλγόριθμοι.
- Μοντέλα βελτιστοποίησης.

Στην περίπτωση χρήσης ευριστικών (heuristics) αλγόριθμων ξεκινάμε με τη κατασκευή του δικτύου CPM/PERT και στη συνέχεια αναλύουμε τις ανάγκες σε πόρους

ανά περίοδο και ανά είδος πόρου. Εάν σε μια περίοδο του έργου δεν επαρκούν οι πόροι, ο ευριστικός αλγόριθμος εξετάζει τις δραστηριότητες που εκτελούνται σε αυτή την περίοδο του έργου και αναθέτει πόρους με τη χρήση κανόνων προτεραιοτήτων, λαμβάνοντας πάντα υπόψη τις τεχνικές απαιτήσεις των δραστηριοτήτων. Μερικοί κανόνες που χρησιμοποιούνται ευρέως είναι:

- **Το συντομότερο δυνατό** (as-soon-as-possible). Οι δραστηριότητες χρονοδρομολογούνται το συντομότερο δυνατό. Είναι ο πλέον γνωστός και χρήσιμος κανόνας.
- **Το αργότερο δυνατό** (as-late-as-possible). Οι δραστηριότητες εκτελούνται το αργότερο δυνατό με την προϋπόθεση ότι δεν καθυστερεί το έργο συνολικά. Ο κανόνας αυτός είναι χρήσιμος όταν θέλουμε να καθυστερήσουμε τα έξοδα του έργου.
- **Η συντομότερη δραστηριότητα πρώτα** (shortest-task-first). Οι δραστηριότητες κατατάσσονται ανάλογα με τη διάρκειά τους και στη συνέχεια εκτελούμε αυτές με τη συντομότερη διάρκεια. Ο κανόνας αυτός είναι χρήσιμος όταν θέλουμε να μεγιστοποιήσουμε τον αριθμό των ολοκληρωμένων δραστηριοτήτων σε μια δεδομένη χρονική περίοδο.
- **Η δραστηριότητα με τις υψηλότερες απαιτήσεις σε πόρους** (most-resources-first). Οι δραστηριότητες κατατάσσονται ανάλογα με τις απαιτήσεις τους σε πόρους. Η λογική υπαγορεύει ότι οι δραστηριότητες που απαιτούν πολλούς πόρους είναι συνήθως και οι πιο σημαντικές.

Η παραπάνω λίστα είναι ενδεικτική και παρουσιάζει μόνο μερικούς από τους διαθέσιμους στη βιβλιογραφία ευριστικούς αλγόριθμους.

### **Δραστηριότητα 4.3**

Είσατε διοικητής έργου και έχετε να αντιμετωπίσετε τις παρακάτω καταστάσεις. Ποια λύση προτείνεται για κάθε μια από τις παρακάτω περιπτώσεις;

α) Τον επόμενο μήνα χρειάζεστε ένα διαχειριστή βάσης δεδομένων 4 ημέρες την εβδομάδα, ενώ είναι διαθέσιμος μόνο 2 ημέρες.

β) Το τμήμα ανθρώπινων πόρων σας υποσχέθηκε ότι τον επόμενο μήνα θα είναι διαθέσιμος ένας ακόμη προγραμματιστής να εργασθεί στο έργο, ενώ θα έπρεπε να είναι διαθέσιμος άμεσα.

## ΕΝΟΤΗΤΑ 4.5 Διαχείριση Έργων με τη Μέθοδο της Κρίσιμης Αλυσίδας.

Όπως αναφέραμε και στις προηγούμενες παραγράφους η κρίσιμη διαδρομή είναι η μεγαλύτερης διάρκειας διαδρομή του έργου. Η μέθοδος της κρίσιμης αλυσίδας επεκτείνει την έννοια της κρίσιμης διαδρομής και ορίζει ότι θα πρέπει να εξετάζουμε, όχι μόνο τις εξαρτήσεις μεταξύ των δραστηριοτήτων, αλλά και τους περιορισμούς των πόρων του έργου. Η διαχείριση έργων με τη μέθοδο της κρίσιμης αλυσίδας (Critical Chain Project Management, CCPM) αποτελεί το συνδυασμό της θεωρίας των περιορισμών (theory of constraints) με τη διαχείριση έργων (Goldratt, 1997; Leach, 2000). Η μέθοδος CCPM δίνει έμφαση στις δραστηριότητες της κρίσιμης αλυσίδας μια και αυτές είναι που περιορίζονται από τους διαθέσιμους πόρους. Οι πόροι αυτοί ονομάζονται κρίσιμοι πόροι (critical resources / bottlenecks).

Τα βασικά βήματα για να δημιουργήσουμε ένα χρονοδιάγραμμα με τη μέθοδο της κρίσιμης αλυσίδας για ένα έργο είναι:

1. Δημιουργούμε το χρονοδιάγραμμα με την αργότερη έναρξη δραστηριοτήτων. Για τον υπολογισμό της διάρκειας των δραστηριοτήτων χρησιμοποιούμε το χρόνο που έχει 50% πιθανότητα ολοκλήρωσης. Αυτό σημαίνει ότι αφαιρούμε τα χρονικά περιθώρια των δραστηριοτήτων.
2. Εντοπίζουμε τις αλληλοσυγκρουόμενες απαιτήσεις σε πόρους. Ξεκινάμε να εργαζόμαστε με περιορισμούς που αφορούν δραστηριότητες κοντά στη λήξη του έργου ή από αυτές τις δραστηριότητες που έχουν την μεγαλύτερη ανάγκη για περισσότερους πόρους.
3. Ικανοποιούμε τους περιορισμούς που προκύπτουν από τις αλληλοσυγκρουόμενες απαιτήσεις σε πόρους μετακινώντας τις δραστηριότητες σε νωρίτερες ημερομηνίες έναρξης. Εργαζόμαστε πάντα από το τέλος του έργου προς την αρχή.
4. Επαναλαμβάνουμε τη διαδικασία για όλους τους πόρους του έργου.
5. Η κρίσιμη αλυσίδα είναι η μεγαλύτερη αλυσίδα εξαρτόμενων γεγονότων.

Η έννοια του χρονικού αποθέματος (contingency), που αναφέραμε στο βήμα 1 ορίζεται ως εξής:

Έστω,

μια δραστηριότητα  $j$ ,

$t_j^1$ , ο χρόνος ολοκλήρωσης δραστηριότητας με 50% πιθανότητα ολοκλήρωσης

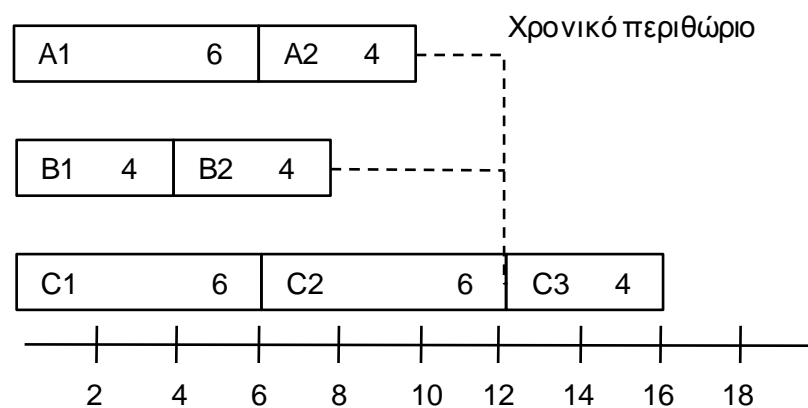
$t_j^2$ , ο χρόνος ολοκλήρωσης δραστηριότητας με 90% πιθανότητα ολοκλήρωσης ορίζουμε ως χρονικό απόθεμα τη διαφορά του  $t_j^1$  και  $t_j^2$ ,

$$c_j = t_j^1 - t_j^2$$

Για παράδειγμα, έστω μια δραστηριότητα που στο 50% των περιπτώσεων που εκτελείται ολοκληρώνεται σε 10 ώρες ή λιγότερο. Η ίδια δραστηριότητα στο 90% των περιπτώσεων ολοκληρώνεται σε 15 ώρες ή λιγότερο. Η δραστηριότητα αυτή θα πρέπει να έχει χρονικό απόθεμα 5 ωρών (15-10).

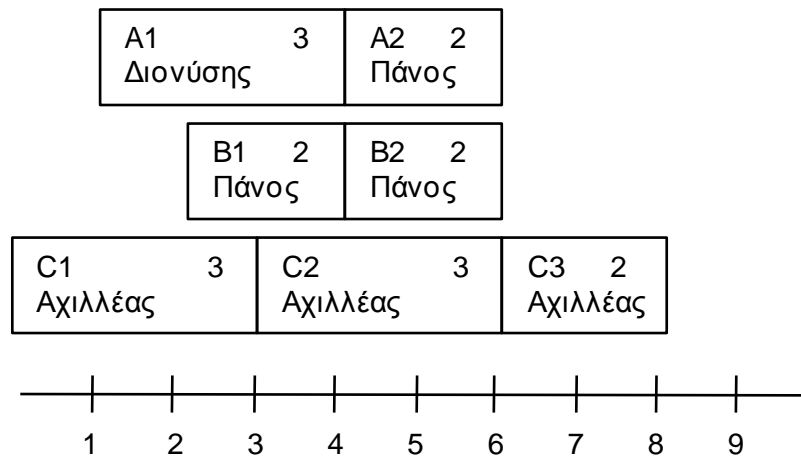
#### **Παράδειγμα 4.14**

Στο Σχήμα 4.34. παρουσιάζεται το χρονοδιάγραμμα ενός έργου, όπου οι δραστηριότητες ξεκινούν την νωρίτερη δυνατή ημερομηνία. Η διάρκεια του έργου είναι 16 εβδομάδες, ενώ το κρίσιμο μονοπάτι είναι το C1-C2-C3.



**Σχήμα 4.34.:** Χρονοδιάγραμμα με νωρίτερη έναρξη δραστηριοτήτων.

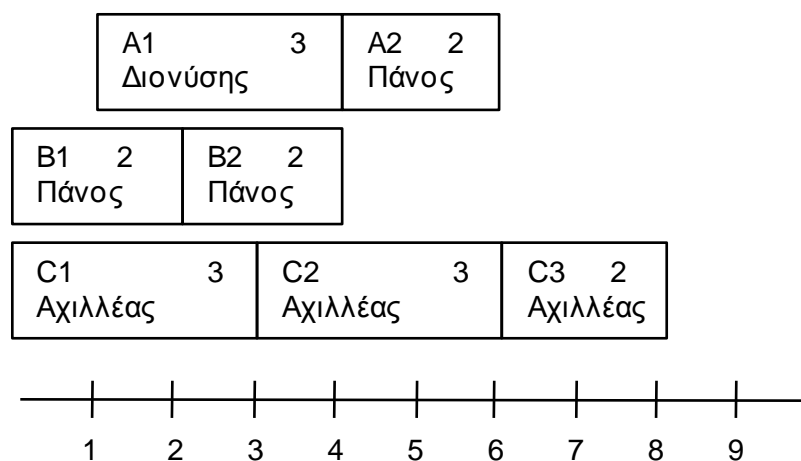
Στο Σχήμα 4.35. παρουσιάζεται το χρονοδιάγραμμα του έργου. όπου οι δραστηριότητες ξεκινούν την αργότερη δυνατή ημερομηνία. Επίσης, η διάρκεια των δραστηριοτήτων έχει αντικατασταθεί από τη διάρκεια με 50% πιθανότητα ολοκλήρωσης, που σημαίνει ότι έχουν αφαιρεθεί τα χρονικά αποθέματα. Για το λόγο αυτό η συνολική διάρκεια του έργου μειώνεται από 16 εβδομάδες σε 8 εβδομάδες. Επιπλέον, θεωρούμε ότι υπάρχουν τρεις εργαζόμενοι στο έργο (Πάνος, Διονύσης, Αχιλλέας), των οποίων οι αναθέσεις παρουσιάζονται επίσης στο Σχήμα 4.35. Σύμφωνα με την ανάθεση αυτή, ο εργαζόμενος «Πάνος» θα πρέπει να εργασθεί ταυτόχρονα στη δραστηριότητα A2 και B2, γεγονός που αποτελεί απαίτηση που δεν μπορεί να ικανοποιηθεί.



**Σχήμα 4.35.:** Χρονοδιάγραμμα με αργότερη έναρξη δραστηριοτήτων.

Για την αφαίρεση των αλληλοσυγκρουόμενων απαιτήσεων σε πόρους ξεκινούμε από τις δραστηριότητες της κρίσιμης διαδρομής C1-C2-C3, όπου παρατηρούμε ότι δεν υπάρχει περιορισμός. Συνεχίζουμε με τις δραστηριότητες που έχουν το μικρότερο χρονικό περιθώριο. Παρατηρούμε, στο Σχήμα 4.34., ότι η δραστηριότητα με το μικρότερο χρονικό περιθώριο είναι η A2. Για το λόγο αυτό αναθέτουμε στον εργαζόμενο «Πάνο» τη δραστηριότητα A2 και όχι τη B2. Το γεγονός, ότι η ανάθεση της δραστηριότητας B2 στον ίδιο εργαζόμενο δεν είναι εφικτή μας οδηγεί στο να μετακινήσουμε τη δραστηριότητα B2, ώστε να ξεκινήσει μια νωρίτερη ημερομηνία.

Το νέο χρονοδιάγραμμα παρουσιάζεται στο Σχήμα 4.36.



**Σχήμα 4.36.:** Χρονοδιάγραμμα μετά από αφαίρεση αλληλοσυγκρουόμενων απαιτήσεων σε πόρους,

Μετά την αφαίρεση των αλληλοσυγκρουόμενων απαιτήσεων, η μέθοδος CCPM εισάγει στο χρονοδιάγραμμα απομονωτές (buffers). Οι απομονωτές είναι χρονικές



διάρκειες που τοποθετούνται στο τέλος μιας ομάδας δραστηριοτήτων με σκοπό την εξασφάλιση και την προστασία του έργου από καθυστερήσεις. Ο υπολογισμός των απομονωτών γίνεται με βάση τα χρονικά αποθέματα των δραστηριοτήτων που απαρτίζουν την ομάδα (Wysocki, 2003). Τρεις είναι οι βασικές κατηγορίες απομονωτών:

- Απομονωτές έργου (Project Buffers, PB). Είναι χρονικοί απομονωτές που τοποθετούνται στο τέλος του έργου με σκοπό τη συνολική προστασία του χρονοδιαγράμματος. Ο απομονωτής έργου υπολογίζεται από την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των αποθεμάτων των δραστηριοτήτων που απαρτίζουν την κρίσιμη αλυσίδα.
- Απομονωτές τροφοδοσίας (Feeding Buffers, FB). Είναι χρονικοί απομονωτές που τοποθετούνται στο τέλος μιας ακολουθίας δραστηριοτήτων και οδηγούν σε δραστηριότητα της κρίσιμης αλυσίδας. Υπολογίζονται με τον ίδιο τρόπο όπως και οι απομονωτές έργου.
- Απομονωτές πόρων (Resource Buffers, RF). Ένας απομονωτής αυτού του τύπου σηματοδοτεί στην κρίσιμη αλυσίδα τη χρονική στιγμή που θα απαιτηθεί ένας πόρος.

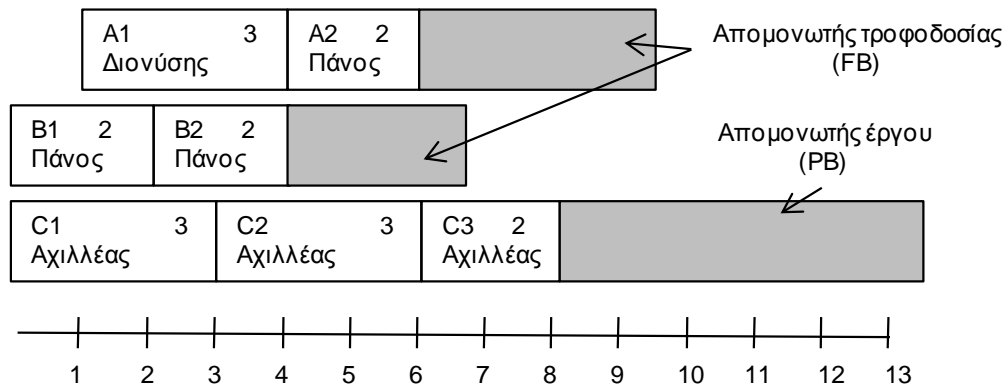
#### **Παράδειγμα 4.15**

Συνεχίζοντας το Παράδειγμα 4.14 εισάγουμε μετά την κρίσιμη αλυσίδα C1-C2-C3 έναν απομονωτή έργου (βλ. Σχήμα 4.21.). Η διάρκεια του απομονωτή υπολογίζεται από την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των διαφορών μεταξύ της αρχικής διάρκειας και της μειωμένης διάρκειας των δραστηριοτήτων.

Για τις δραστηριότητες C1 και C2 η αρχική διάρκεια ήταν 6, ενώ η μειωμένη ήταν 3. Αντίστοιχα, για τη δραστηριότητα C3 η αρχική διάρκεια ήταν 4, ενώ η μειωμένη ήταν 2. Συνεπώς:

$$PB_{C1-C2-C3} = \sqrt{3^2 + 3^2 + 2^2} = \sqrt{9 + 9 + 4} = \sqrt{22} = 4,7 \text{ ημέρες}$$

Αντίστοιχα,  $FB_{A1-A2}=3,6$  ημέρες και  $FB_{B1-B2}=2,8$  ημέρες



**Σχήμα 4.37.:** Απομονωτές.

Αξίζει να σημειωθεί ότι το αρχικό χρονοδιάγραμμα (αυτό του Σχήματος 4.18.) έχει διάρκεια 16 ημερών, ενώ αυτό που προέκυψε από τη μέθοδο CCPM έχει διάρκεια 12,7 ημερών, γεγονός που αποτελεί σημαντική βελτίωση του χρονοδιαγράμματος.

Η μέθοδος της κρίσιμης αλυσίδας είναι ιδιαίτερα χρήσιμη στη καθημερινή διαχείριση του έργου, αφού ο διοικητής έργου δεν ασχολείται με την παρακολούθηση της ημερομηνία έναρξης και λήξης της κάθε δραστηριότητας, αλλά με τη κρίσιμη αλυσίδα και τη διαχείριση απομονωτών. Έτσι, καθυστέρηση μιας δραστηριότητας ή μιας αλυσίδας δραστηριοτήτων σημαίνει είσοδο στη χρονική διάρκεια του απομονωτή. Στην περίπτωση αυτή, εάν η διεύθυνση της αλυσίδας των δραστηριοτήτων είναι:

- Μικρότερη του ενός τρίτου της χρονικής διάρκειας του απομονωτή τότε ο διοικητής του έργου δεν χρειάζεται να λάβει καμία διορθωτική ενέργεια
- Μικρότερη των δύο τρίτων της χρονικής διάρκειας του απομονωτή τότε ο διοικητής του έργου, χρειάζεται να διερευνήσει τις αιτίες του προβλήματος και τέλος εάν η διεύθυνση είναι
- Μεγαλύτερη των δύο τρίτων θα πρέπει να κάνει διορθωτικές κινήσεις στη διαχείριση του έργου.

#### Απομονωτής

1 <sup>ο</sup> τρίτο	2 <sup>ο</sup> τρίτο	3 <sup>ο</sup> τρίτο
Καμία ενέργεια	Διερεύνηση αιτίας καθυστέρησης	Διορθωτικές ενέργειες

**Σχήμα 4.38.:** Διεύθυνση δραστηριοτήτων στη χρονική διάρκεια απομονωτή.

## ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

---

Στο κεφάλαιο αυτό ασχοληθήκαμε με βασικά θέματα διαχείρισης έργων. Παρουσιάσαμε το πλάνο διαχείρισης έργου, το οποίο αποτελεί το πιο σημαντικό παραδοτέο που απαιτείται για τη σχεδίαση ενός έργου. Το πλάνο διαχείρισης έργου χρησιμοποιείται για να τεκμηριώσει τις παραδοχές του έργου, να τεκμηριώσει τις αποφάσεις που έχουν ληφθεί σε σχέση με τις εναλλακτικές λύσεις του έργου, να καθοδηγήσει την εκτέλεση του έργου, να τεκμηριώσει τις αποφάσεις που έχουν ληφθεί σε σχέση με τις εναλλακτικές λύσεις, να τεκμηριώσει τις πρακτικές εργασίας που θα ακολουθηθούν, να καθορίσει τα πρότυπα που θα χρησιμοποιηθούν, αλλά και τον τρόπο που αυτά θα εφαρμοσθούν, κ.α. Αποτελεί τη «βίβλο» ενός έργου, αφού ορίζει, καταγράφει, επεξηγεί το πώς θα διοικήσουμε το έργο.

Στη συνέχεια παρουσιάσαμε βασικές μεθόδους διαχείρισης χρόνου. Πιο συγκεκριμένα, παρουσιάστηκαν οι μέθοδοι PERT/CPM, η ανάλυση του κρίσιμου μονοπατιού, και ο τρόπος υπολογισμού της κρίσιμης διαδρομής. Μετά τον υπολογισμό της κρίσιμης διαδρομής παρουσιάστηκε μια μέθοδος της συντόμευσης της διάρκειας έργου.

Το επόμενο θέμα που παρουσιάστηκε είναι οι βασικές έννοιες της διαχείρισης πόρων του έργου, καθώς και μια απλουστευμένη μέθοδο εξισορρόπησης πόρων.

Το κεφάλαιο ολοκληρώνεται με μια σύντομη εισαγωγή στη μέθοδο της κρίσιμης αλυσίδας.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

Bechtold, P., (1999). Essentials of Software Project Management. Management Concepts.

PMI Institute, (2004). A guide to the Project Management Body of Knowledge, PMI Standard Committee.

Hughes, B., Cotterell, M., (1999). Software Project Management. McGraw Hill.

Meredith, J., Mantel, S., Mantel, Jr. S., (2005). Project Management: A Managerial Approach, 5th edition. John Wiley & Sons.

Kerzner, H., (2003). Project Management: A Systems Approach to Planning, Scheduling and Controlling, 8th Edition. John Wiley & Sons.

Υψηλάντης, Π., (2003). Χρονικός προγραμματισμός. ΤΕΙ Λάρισας.

Leach, L., (2000). Critical Chain Project Management. Boston, Artech House.

Goldratt, E., (1997). Critical Chain. Great Barrington, Mass.: The North River Press.

Wysocki, R., McGary, R., (2003). Effective Project Management, 3rd Edition. John Wiley & Sons.

## ΚΕΦΑΛΑΙΟ 5. ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ

**Συγγραφέας: Ι. ΣΤΑΜΕΛΟΣ**

Στο κεφάλαιο αυτό θα μάθετε:

- τον ορισμό της εκτίμησης κόστους
- τη χρησιμότητα της εκτίμησης κόστους
- τα χαρακτηριστικά των έργων λογισμικού που επηρεάζουν το κόστος
- τα εγγενή προβλήματα στην εκτίμηση κόστους
- ποιες είναι οι τρεις κυριότερες κατηγορίες τεχνικών εκτίμησης κόστους
- ποιες είναι οι τεχνικές που ανήκουν σε κάθε κατηγορία
- θα ενημερωθείτε για παραδείγματα εργαλείων που μπορούν να υποστηρίξουν την εφαρμογή αυτών των τεχνικών
- να συγκρίνετε τεχνικές μεταξύ τους με βάση τα πλεονεκτήματα και μειονεκτήματα της κάθε μίας

Θα αποκτήσετε δεξιότητες σχετικά με

- την εφαρμογή τεχνικών εκτίμησης κόστους
- τη χρήση εργαλείων εκτίμησης κόστους

### ΠΕΡΙΕΧΟΜΕΝΑ

#### 5.1 Εισαγωγή στην εκτίμηση κόστους

##### 5.1.1 Ορισμός της εκτίμησης κόστους

##### 5.1.2 Χρησιμότητα της εκτίμησης κόστους

##### 5.1.2.1. Συνιστώσες του κόστους

##### 5.1.3 Παράγοντες κόστους φόρτου ανάπτυξης

##### 5.1.4 Τεχνικές εκτίμησης κόστους

##### 5.1.5 Προβλήματα της εκτίμησης κόστους

#### 5.2 Εκτίμηση κόστους με παραμετρικά μοντέλα

##### 5.2.1 Μοντέλο COCOMO81

- 5.2.1.1 Γενικά – Βασικό μοντέλο COCOMO
- 5.2.1.2 Ενδιάμεσο μοντέλο COCOMO81
- 5.2.2 Μοντέλο COCOMO -2.0
- 5.2.3 Μοντέλα Παλινδρόμησης
- 5.3. Εκτίμηση με βάση τις αναλογίες
- 5.4 Συμπεράσματα – Βιβλιογραφία
- 5.5 Απαντήσεις Ασκήσεων Αυτοαξιολόγησης – Δραστηριοτήτων

## 5.1 Εισαγωγή στην εκτίμηση κόστους

---

### 5.1.1 Ορισμός της εκτίμησης κόστους

Η Εκτίμηση Κόστους Λογισμικού (για συντομία *εκτίμηση κόστους*) είναι μία επιστημονική περιοχή που άρχισε να αναπτύσσεται την εποχή που το λογισμικό άρχισε να διαδίδεται ραγδαία, βρίσκοντας ευρεία εφαρμογή σε κλάδους όπως εμπορικές εφαρμογές, εφαρμογές διαχείρισης δεδομένων, παράλληλα με τις επιστημονικές εφαρμογές από τις οποίες και ξεκίνησε.

Με τον όρο *εκτίμηση κόστους* εννοούμε το σύνολο των δραστηριοτήτων που αποσκοπούν στον προσδιορισμό του απαιτούμενου κατασκευαστικού κόστους για την ολοκλήρωση ενός έργου λογισμικού. Πρόκειται για μία αναγκαία διοικητική ενέργεια που αποτελεί την αρχική δραστηριότητα για τον χρονοπρογραμματισμό και τη στελέχωση του έργου (φάσεις *Ανάπτυξη Χρονοδιαγράμματος και Ανάπτυξη Προϋπολογισμού* βλ. παρ. 1.1, σχ.2). Θα δούμε στη συνέχεια ότι η κυριότερη συνιστώσα κόστους είναι η ανθρώπινη προσπάθεια που απαιτείται για την εκτέλεση του έργου. Επομένως η εκτίμηση κόστους σε μεγάλο βαθμό ανάγεται σε *εκτίμηση φόρτου (effort estimation)*.

Στις φάσεις που προαναφέραμε αλλά και προηγούμενα (π.χ. στον Προσδιορισμό των Κριτηρίων Επιτυχίας, στη φάση σύλληψης στη διεργασία RUP, κλπ) μας απασχολούν ερωτήματα όπως πόσο μεγάλο θα είναι το έργο, πόσο θα κοστίζει, πόσο θα διαρκέσει η ανάπτυξή του, από ποιους παράγοντες θα επηρεασθούν όλα αυτά και σε ποιο βαθμό. Η εκτίμηση κόστους είναι η δραστηριότητα που βοηθά στο να δώσουμε απαντήσεις, έστω και μερικά, στα παραπάνω ερωτήματα.

### 5.1.2 Χρησιμότητα της εκτίμησης κόστους

Οι εκτιμήσεις κόστους αποτελούν μία πηγή πολύτιμης πληροφορίας για τη διοίκηση ενός έργου λογισμικού. Πιο συγκεκριμένα είναι απαραίτητες για τις εξής λειτουργίες της διοίκησης:

- Κατανομή Πόρων: Ανάλογα με το προϋπολογιζόμενο κόστος θα ανατεθούν πόροι σε διάφορες δραστηριότητες του έργου (π.χ. ανθρώπινοι πόροι στη διαδικασία συλλογής απαιτήσεων, συγγραφή κώδικα, διασφάλιση ποιότητας). Επίσης θα επιλεγούν και άτομα με τα κατάλληλα προσόντα που θα στελεχώσουν τις διάφορες ομάδες του έργου (π.χ. εκτίμηση υψηλού κόστους για τη φάση της ανάλυσης θα οδηγήσει σε ανάγκη στελέχωσης του έργου με αναλυτές λογισμικού).
- Χρονοπρογραμματισμός Έργου: Η εκτίμηση του ανθρώπινου φόρτου, εκτός από τον αριθμό των εργαζόμενων και την ένταση της προσπάθειάς τους οδηγεί και σε εκτίμηση της χρονικής διάρκειας του έργου και στη συνέχεια της χρονικής διάρκειας κάθε φάσης και επιμέρους εργασίας. Επομένως τα αποτελέσματα της εκτίμησης μπορούν να αποτελέσουν τη βάση με την οποία θα τοποθετηθούν οι φάσεις του έργου στο σχετικό χρονοδιάγραμμα.
- Ικανοποίηση Χρονικών και Οικονομικών Περιορισμών: Η εκτίμηση του κόστους από την πλευρά του κατασκευαστή του δίνει τη δυνατότητα (α) να διεκδικήσει την ανάθεση έργων με ρεαλιστικό, αποδοτικό και ανταγωνιστικό τίμημα (β) να αιτιολογήσει τα οικονομικά αιτήματά του προς το χρηματοδότη οργανισμό και (γ) να καταρτίζει ρεαλιστικά χρονοδιαγράμματα σε σχέση με τους χρονικούς περιορισμούς που τίθενται (π.χ. να συμπίεσει σε κάποιο βαθμό τη διάρκεια εκτέλεσης του έργου αυξάνοντας τον απαιτούμενο φόρτο εργασίας). Ανάλογα, η εκτίμηση του κόστους από την πλευρά του χρηματοδότη / πελάτη του λογισμικού του δίνει τη δυνατότητα (α) να αξιολογεί τις προσφορές των κατασκευαστών, (β) να ορίζει ρεαλιστικά επίπεδα προϋπολογισμών για τα έργα για τα οποία πρόκειται να ζητήσει προσφορές από κατασκευαστές λογισμικού, και (γ) να καταρτίζει ρεαλιστικούς χρονικούς στόχους.

#### 5.1.2.1 Συνιστώσες του κόστους



Το κόστος εκτέλεσης ενός έργου λογισμικού εξαρτάται από μία σειρά παραγόντων. Το ποσοστό συμμετοχής τους στο συνολικό κόστος ποικίλλει και επηρεάζεται από τα ιδιαίτερα χαρακτηριστικά του έργου. Οι κυριότεροι παράγοντες είναι οι ακόλουθοι:

- Μισθοί και αποζημιώσεις για την εργασία των συμμετεχόντων στο έργο. Πρόκειται για το σημαντικότερο παράγοντα καθώς ο απαιτούμενος φόρτος για ένα έστω και μέτριο έργο είναι αρκετά μεγάλη, ενώ συχνά απαιτούνται άτομα με εξειδικευμένες γνώσεις των οποίων οι αμοιβές είναι συνήθως υψηλές
- Κόστος απόκτησης υλικού και λογισμικού (της λεγόμενης πλατφόρμας ανάπτυξης) πάνω στα οποία θα αναπτυχθεί και θα εγκατασταθεί το λογισμικό. Στο κόστος αυτό συμπεριλαμβάνεται η αγορά servers, δικτυακής υποδομής, λειτουργικών συστημάτων, αδειών χρήσης (π.χ. για εργαλεία CASE, βάσεις δεδομένων) κλπ. Ο παράγοντας αυτός αυξάνεται όταν έχουμε να αναπτύξουμε κατακευματισμένα συστήματα, συστήματα που θα εγκατασταθούν σε πολλά, ξεχωριστά γεωγραφικά σημεία κλπ.
- Έξοδα μετακινήσεων και εκπαίδευσης προσωπικού. Ο παράγοντας αυτός είναι αυξημένος όταν απαιτούνται μετακινήσεις επειδή οι ομάδες ανάπτυξης και οι ομάδες με τις οποίες αυτές θα συνεργασθούν (π.χ. ομάδες χρηστών) δεν βρίσκονται στο ίδιο σημείο. Έτσι θα απαιτηθούν μετακινήσεις για συναντήσεις εργασίας (για την πορεία του έργου, για τη διασφάλιση της ποιότητας), για συνεντεύξεις με τους χρήστες, παρουσιάσεις κλπ. Πρέπει να σημειωθεί ότι σήμερα σε μεγάλα έργα συχνά η ανάπτυξη γίνεται από ομάδες που βρίσκονται σε απομακρυσμένα σημεία (π.χ. η συλλογή απαιτήσεων γίνεται στην πόλη-έδρα του πελάτη, η ανάλυση γίνεται στην έδρα του κατασκευαστή, ενώ η συγγραφή του κώδικα γίνεται σε άλλη χώρα λόγω χαμηλού κόστους εργασίας).
- Πάγια έξοδα του κατασκευαστή, όπως ενοίκια, λειτουργικά έξοδα, τέλη επικοινωνιών κλπ
- Αναλώσιμα (χαρτί, τόներ, κλπ)

Από τους παράγοντες αυτούς ο πιο σημαντικός είναι ο ανθρώπινος φόρτος εργασίας, όχι μόνο γιατί ποσοτικά υπερβαίνει τους υπόλοιπους, αλλά και επειδή είναι σαφώς και ο δυσκολότερος να προβλεφθεί.

### 5.1.3 Παράγοντες κόστους φόρτου εργασίας

Υπάρχουν διάφοροι παράγοντες του έργου οι οποίοι συντελούν στην αυξομείωση του απαιτούμενου φόρτου εργασίας. Αναφέρονται εδώ συνοπτικά μερικοί από αυτούς, τους οποίους και θα ξανασυναντήσουμε πιο κάτω συζητώντας τις διάφορες τεχνικές εκτίμησης. Αυτοί είναι:

- Ο τύπος του λογισμικού που αναπτύσσεται (π.χ. λογισμικό πραγματικού χρόνου, λογισμικό διαχείρισης δεδομένων, λογισμικό συστήματος), το πεδίο εφαρμογής (π.χ. βιομηχανία, τραπεζική, ηλ.κή διακυβέρνηση).
- Το μέγεθος του λογισμικού είναι ίσως ο σημαντικότερος παράγοντας (όσο πιο μεγάλο το λογισμικό τόσο μεγαλύτερος ο απαιτούμενος φόρτος εργασίας). Διακρίνουμε το φυσικό μέγεθος (συνηθισμένη μονάδα μέτρησης του όγκου του κώδικα είναι ο αριθμός των εντολών ή των γραμμών κώδικα. βλ. κεφ. COCOMO) και το λειτουργικό μέγεθος που μετράται με τους Βαθμούς Λειτουργίας (Function Points) και τις παραλλαγές τους.
- Η ποιότητα και εμπειρία των διαφόρων ομάδων ανάπτυξης (αναλυτές, προγραμματιστές κλπ).
- Τα εργαλεία ανάπτυξης που χρησιμοποιούνται (Integrated Design Environments, test tools, κλπ )
- Η διεργασία ανάπτυξης που ακολουθείται. Πολύπλοκες διεργασίες (π.χ. το Σπειροειδές μοντέλο) με μεγάλες απαιτήσεις ελέγχου της ποιότητας και τεκμηρίωσης απαιτούν σημαντικά περισσότερο φόρτο. Αντίθετα ευέλικτες διεργασίες μειώνουν το κόστος.

### 5.1.4. Τεχνικές εκτίμησης κόστους

Διακρίνουμε τρεις μεγάλες κατηγορίες τεχνικών εκτίμησης κόστους:

1. Εκτίμηση από ειδικούς
2. Εκτίμηση με παραμετρικά μοντέλα
3. Εκτίμηση με βάση τις αναλογίες

Στις τρεις αυτές κατηγορίες τα τελευταία χρόνια έχει προστεθεί μία τέταρτη ομάδα τεχνικών εκτίμησης κόστους που βρίσκεται στο στάδιο της έρευνας ακόμη και δεν έχει διαδοθεί ιδιαίτερα στην πράξη. Πρόκειται για τις τεχνικές μηχανικής μάθησης (νευρωνικά δίκτυα, κανόνες συσχέτισης, δένδρα CART, δίκτυα Bayes). Στο κεφάλαιο αυτό θα επικεντρωθούμε στις μεθόδους των παραμετρικών μοντέλων και των αναλογιών.

Στην εκτίμηση από ειδικούς, οι εκτιμήσεις παράγονται με βάση αποκλειστικά την εμπειρία ενός ή περισσότερων 'ειδικών', δηλ. σχετικά έμπειρων στελεχών και τεχνολόγων λογισμικού. Δε χρησιμοποιούνται ιδιαίτερα καταγραμμένα στοιχεία για προγενέστερα, κοστολογημένα έργα λογισμικού (λέγονται *ιστορικά έργα λογισμικού*). Η τεχνική εστιάζεται στο πως θα συνδυασθούν οι εκτιμήσεις περισσότερων του ενός ειδικών. Μία (αρκετά παλιά) σχετική τεχνική είναι αυτή των Δελφών. Η επιτυχία της τεχνικής των ειδικών εξαρτάται πολύ από τη διαθεσιμότητα του ειδικού ή των ειδικών και από την αξιοποίηση της εμπειρίας των ειδικών και τη σωστή υποκειμενική αξιολόγηση του υπό εκτίμηση έργου. Είναι η πιο διαδομένη μέθοδος στην πράξη, αλλά συνήθως εφαρμόζεται με μη συστηματικό τρόπο και οδηγεί συχνά σε πολύ λανθασμένες εκτιμήσεις.

Στην εκτίμηση με παραμετρικά μοντέλα, χρησιμοποιούνται μεμονωμένες εξισώσεις ή ομάδες εξισώσεων, οι οποίες, με βάση τιμές που δίδονται σε επιλεγμένα χαρακτηριστικά (παραμέτρους) του προς εκτίμηση έργου, υπολογίζουν το κόστος του. Οι εξισώσεις αυτές βασίζονται σε ένα σύνολο από ιστορικά έργα τα οποία έχουν χρησιμοποιηθεί για να επιλεγεί το είδος της σχέσης του κόστους με τις παραμέτρους του (π.χ. γραμμική, εκθετική σχέση) και οι συντελεστές βαρύτητας με τους οποίους κάθε παράμετρος συμμετέχει στον υπολογισμό του κόστους. Διακρίνουμε δύο κατηγορίες, τα μοντέλα παλινδρόμησης (regression models) και τα άτυπα μοντέλα (ad hoc models).

Τα πρώτα προέρχονται από την εφαρμογή της στατιστικής μεθόδου της παλινδρόμησης στα δεδομένα των ιστορικών έργων. Τα άτυπα μοντέλα είναι μοντέλα κατασκευασμένα με μη αυτόματο τρόπο. Τα πιο γνωστά μοντέλα αυτού του τύπου είναι τα μοντέλα της οικογένειας COCOMO. Κρίσιμα σημεία της μεθόδους είναι η ακρίβεια εκτίμησης των ίδιων των ιστορικών έργων από το μοντέλο και η συνάφεια του νέου

έργου με τα ιστορικά. Η τεχνική των παραμετρικών μοντέλων είναι αρκετά διαδομένη στην πράξη.

Η μέθοδος με αναλογίες στηρίζεται επίσης σε ένα σύνολο από ιστορικά έργα. Το προς εκτίμηση έργο συγκρίνεται με κάθε ένα ιστορικό έργο ξεχωριστά. Επιλέγονται ένα ή περισσότερα ιστορικά έργα ως “ανάλογα” με το προς εκτίμηση, και με βάση το καταγραμμένο κόστος τους γίνεται η εκτίμηση του κόστους του νέου έργου. Κρίσιμο σημείο της μεθόδου είναι η ύπαρξη έργου ή έργων που είναι παρόμοια με το προς εκτίμηση. Η μέθοδος των αναλογιών είναι η λιγότερο διαδομένη στην πράξη από τις τρεις βασικές τεχνικές.

Τέλος, η εκτίμηση με μοντέλα μηχανική μάθησης μοιάζει με αυτή των παραμετρικών μοντέλων, με τη διαφορά ότι την εκτίμηση του νέου έργου αναλαμβάνουν τώρα δίκτυα (νευρωνικά, Bayes), κανόνες ή δένδρα κατηγοριοποίησης που έχουν εκπαιδευθεί προηγουμένως πάνω στα ιστορικά έργα.

### **Δραστηριότητα 5.1**

Να αναφέρετε τις τρεις κυριότερες κατηγορίες τεχνικών εκτίμησης κόστους και να τις περιγράψετε σύντομα. Κατά τη γνώμη σας, γιατί και οι τρεις τεχνικές βασίζονται στην ανάλυση ιστορικών έργων;

#### **5.1.5. Προβλήματα της εκτίμησης κόστους**

Η απάντηση στη δραστηριότητα 5.1. σας δίνει ήδη μία ιδέα για την πολυπλοκότητα και την εγγενή δυσκολία της εκτίμησης κόστους. Γενικότερα, η εκτίμηση κόστους είναι ιδιαίτερα δύσκολη για μια σειρά από λόγους:

- Στην αρχική φάση του έργου δεν είναι γνωστές πολλές σημαντικές λεπτομέρειες, π.χ. η ακριβής σύνθεση της ομάδας έργου
- Πολλές παράμετροι του έργου είναι γνωστές σε μικρό βαθμό και υπάρχει μεγάλη αοριστία και ασάφεια (π.χ. οι απαιτήσεις από το σύστημα και το λογισμικό)

- Διάφορες παράμετροι είναι αρκετά πιθανό να τροποποιηθούν σε σημαντικό βαθμό κατά την εκτέλεση του έργου (π.χ. η ανάγκη μετάβασης σε μία νέα έκδοση της πλατφόρμας ανάπτυξης)
- Πολύ συχνά δεν υπάρχουν πληροφορίες οι οποίες θα μπορούσαν να υποστηρίξουν αποτελεσματικά την εκτίμηση κόστους, όπως π.χ. πληροφορίες κόστους για έργα που έχει αναπτύξει στο σχετικά κοντινό παρελθόν η ομάδα ανάπτυξης
- Δεν είναι επαρκώς γνωστή η δυναμική των σχέσεων που αναπτύσσονται κατά τη διάρκεια της εκτέλεσης του έργου, π.χ. δεν είναι γνωστή η παραγωγικότητα ενός προγραμματιστή ή μίας ομάδας προγραμματιστών κάτω από τις συγκεκριμένες συνθήκες που θα αναπτυχθούν στο έργο.
- Δεν είναι ιδιαίτερα διαδεδομένες τόσο οι πρωταρχικές τεχνικές όσο και οι προχωρημένες τεχνικές εκτίμησης κόστους που έχουν δει το φως τα τελευταία χρόνια, οι οποίες θα μπορούσαν να αυξήσουν την ακρίβεια των εκτιμήσεων

Για τους παραπάνω λόγους η εκτίμηση κόστους παραμένει μία δύσκολη δραστηριότητα παρόλη την αναμφισβήτητη χρησιμότητά της.

### **Άσκηση Αυτοαξιολόγησης 5.1**

Να αναφέρετε τρία χαρακτηριστικά έργων λογισμικού που επηρεάζουν σημαντικά το κόστος τους

## **5.2. Εκτίμηση κόστους με παραμετρικά μοντέλα**

---

Στο κεφάλαιο αυτό θα ασχοληθούμε με την περιγραφή, την κατασκευή και την ανάλυση των παραμετρικών μοντέλων, δηλ. των μοντέλων παλινδρόμησης και των ad-hoc μοντέλων. Θα δώσουμε περισσότερη έμφαση αρχικά στα ad-hoc μοντέλα καθώς υπάρχει περισσότερη εμπειρία από την πρακτική εφαρμογή τους σε σχέση με τα πρώτα. Από τα μοντέλα αυτά θα δούμε τα μοντέλα COCOMO81 και COCOMO-2. Το κοινό γνώρισμα όλων αυτών των μοντέλων είναι ότι χρησιμοποιούν μία ή περισσότερες εξισώσεις για την εκτίμηση του κόστους.

### 5.2.1. Μοντέλο COCOMO81

Το μοντέλο αυτό προτάθηκε από τον B. Boehm το 1981. Πρόκειται για το πρώτο διαδομένο μοντέλο εκτίμησης κόστους και για περίπου δεκαπέντε χρόνια χρησιμοποιήθηκε σχεδόν μονοπωλιακά από τους διοικητές λογισμικού ανά τον κόσμο. Ακόμη και σήμερα βρίσκουμε εφαρμογές του μοντέλου αυτού, αν και πλέον θεωρείται παρωχημένο. Παρόλα αυτά θα δούμε τις βασικές του λεπτομέρειες καθώς παρουσιάζει μεγάλο ενδιαφέρον από εκπαιδευτικής άποψης. Στη συνέχεια θα περιγράψουμε και τη νεότερη έκδοσή του, το COCOMO-2 (προτάθηκε το 2000), περισσότερο για να καταλάβουμε τις διαφορές των δύο μοντέλων.

#### 5.21.1. Γενικά – Βασικό μοντέλο COCOMO

*Τα αρχικά COCOMO προέρχονται από τις λέξεις COnstructive COst Model ('Κατασκευαστικό' Μοντέλο Κόστους, δηλ. μοντέλο που κατασκευάζει σταδιακά την εκτίμηση κόστους). Πράγματι το μοντέλο εκτιμά το κόστος με τελείως διαφανή τρόπο, αυξομοιώνοντας την αρχική ονομαστική εκτίμησή του. Το μοντέλο αποτελείται από τρία επιμέρους μοντέλα, το βασικό (basic), το ενδιάμεσο (intermediate) και το λεπτομερές (detailed) μοντέλο COCOMO. Στην παράγραφο αυτή θα αναπτύξουμε το βασικό μοντέλο και στην επόμενη μόνο το ενδιάμεσο. Θα αφήσουμε το λεπτομερές μοντέλο για τον αναγνώστη που ενδιαφέρεται να εμβαθύνει περισσότερο. Τα μοντέλα COCOMO περιγράφονται με λεπτομέρεια στην αναφορά [1] και στη διεύθυνση <http://sunset.usc.edu/COCOMOII/cocomo.html>*

Το βασικό στοιχείο του έργου για την εκτίμηση του κόστους είναι το μέγεθος του σε *KDSI (Kilo-Delivered Source Instructions)*. Πρόκειται για τον όγκο του πηγαίου κώδικα του έργου μετρημένο σε χιλιάδες εντολών κώδικα που παραδίνονται στον πελάτη κατά την ολοκλήρωση του έργου. Ο εκτιμητής κατά COCOMO81 θα πρέπει να εκτιμήσει ή να μετρήσει πρώτα το μέγεθος του κώδικα για να προχωρήσει στη συνέχεια της εκτίμησης.

Το δεύτερο βασικό στοιχείο είναι η κατηγορία στην οποία ανήκει το έργο. Σύμφωνα με το μοντέλο τα έργα χωρίζονται σε τρεις κατηγορίες, τα *οργανικά (organic)*, τα *ημι-*

αποσπασμένα (*semi-detached*) και τα ενσωματωμένα (*embedded*) έργα. Η κατηγορία του έργου ουσιαστικά προκαθορίζει τη δυσκολία ανάπτυξής του.

Τα οργανικά έργα αναπτύσσονται με κάπως χαλαρές απαιτήσεις, δεν υπάρχουν έντονοι περιορισμοί γενικά και επιτρέπονται αποκλίσεις από τις αρχικές προδιαγραφές. Γενικά είναι έργα που αναπτύσσονται με σχετικά χαλαρό χρονοδιάγραμμα και διοίκηση έργου. Παραδείγματα αυτού του είδους έργου μπορεί να είναι το σύστημα μηχανογράφησης ενός τμήματος πανεπιστημίου, μία εφαρμογή που παρουσιάζει μορφοποιημένα κάποια δεδομένα, μία εφαρμογή που βρίσκει στο διαδίκτυο πληροφορίες κλπ.

Στην άλλη άκρη βρίσκονται τα ενσωματωμένα έργα που αναπτύσσουν συστήματα με αυστηρές απαιτήσεις, σημαντικούς περιορισμούς (π.χ. χρονικούς) και αυστηρή διοικητική διαδικασία. Παραδείγματα μπορεί να είναι ένα σύστημα αποφυγής συγκρούσεων σε κάποιο όχημα, ένα σύστημα που εκτελεί μεταφορές κεφαλαίων, ένα σύστημα που παρακολουθεί κρίσιμες σωματικές λειτουργίες ενός ασθενούς σε μία εντατική μονάδα παρακολούθησης σε ένα νοσοκομείο.

Τέλος τα ημι-αποσπασμένα έργα βρίσκονται ανάμεσα στις δύο προηγούμενες κατηγορίες. Έχουν σημαντικές απαιτήσεις και περιορισμούς, όχι όμως στο βαθμό που έχουν τα ενσωματωμένα συστήματα.

Η κατάταξη του προς εκτίμηση έργου σε μία από τις τρεις κατηγορίες γίνεται με βάση μία σειρά χαρακτηριστικών γνωρισμάτων του έργου που συνοψίζονται στον πίνακα 1. Ο εκτιμητής επιλέγει υποκειμενικά (και συγκρίνοντας με άλλα έργα) τιμές για κάθε ένα χαρακτηριστικό γνώρισμα και καταλήγει στην κατηγορία στην οποία ανήκει το έργο ανάλογα με το πλήθος των χαρακτηριστικών του έργου που εμπίπτουν στη κάθε κατηγορία. Σε περίπτωση που τα χαρακτηριστικά του έργου μοιράζονται στις τρεις κατηγορίες συνίσταται η επιλογή της ημι-αποσπασμένης κατηγορίας.

Μετά την επιλογή της κατηγορίας έργου και την εκτίμηση του μεγέθους είναι πλέον δυνατή η εφαρμογή του απλούστερου μοντέλου, δηλ. του βασικού. Ο υπολογισμός του ονομαστικού φόρτου ανάπτυξης  $MM_{NOM}$  γίνεται σύμφωνα με τις εξισώσεις του Πίνακα 2.

Το αποτέλεσμα δίνεται σε ανθρωπομήνες (*man-months*). Για το βασικό μοντέλο COCOMO81 ο ονομαστικός φόρτος εργασίας (δηλ. ένας μέσος ή τυπικός φόρτος) είναι ίδιος με τον φόρτο ανάπτυξης  $MM_{DEV}$ . Η διαφορά των δύο θα φανεί στο ενδιάμεσο μοντέλο.

### **Άσκηση αυτοαξιολόγησης 5.2**

Δίνονται τα παρακάτω παραδείγματα έργων. Από εσάς ζητείται να τα κατατάξετε διαισθητικά στην πλησιέστερη κατηγορία σύμφωνα με τον πίνακα 1. Αιτιολογήστε την απάντησή σας.

- Η διπλωματική εργασία ενός φοιτητή
- Σύστημα συναλλαγών χρηματιστηρίου
- Σύστημα υπολογισμού βαθμών φοιτητών του ΕΑΠ
- Υποσύστημα λογιστικής ενός συστήματος ERP



**Πίνακας 1. Κατηγορίες έργων λογισμικού κατά COCOMO81**

Χαρακτηριστικά	Κατηγορία		
	Οργανική	Ημιαποσπασμένη	Ενσωματωμένη
Κατανόηση από όλους εμπλεκόμενους των αντικειμενικών σκοπών του προϊόντος	Λεπτομερής	Λεπτομερής	Γενική
Εμπειρία που να αποκτήθηκε σε σχετικά έργα λογισμικού	Εκτενής	Σημαντική	Μέτρια
Ανάγκη για συμμόρφωση του λογισμικού με προκαθορισμένες απαιτήσεις	Μέτρια	Σημαντική	Πλήρης
Ανάγκη για συμμόρφωση του λογισμικού με προϋπάρχουσα εξωτερική διασύνδεση	Μέτρια	Σημαντική	Πλήρης
Ταυτόχρονη ανάπτυξη νέου λογισμικού και υλικού	Μερική	Μέτρια	Πλήρης
Ανάγκη για μοντέρνες αρχιτεκτονικές λύσεις και νέους αλγόριθμους	Μικρή	Μερική	Σημαντική
Κίνητρο για γρήγορη αποπεράτωση	Χαμηλό	Μέτριο	Υψηλό
Μέγεθος προϊόντος	<50KDSI	<300KDSI	Όλα
Παραδείγματα	Μαζική επεξεργασία	Απλά συστήματα	Μεγάλα σύνθετα

	δεδομένων. Επιστημονικά μοντέλα. Εμπορικά μοντέλα. Μεταγλωττιστές. Συστήματα ελέγχου παραγωγής	συναλλαγών. Λογισμικό Συστήματος, Συστήματα Διαχείρισης Βάσεων Δεδομένων	συστήματα δοσοληψιών. Μεγάλα φιλόδοξα συστατικά λογισμικού συστήματος, Συστήματα Πραγματικού Χρόνου
--	--	---	--

**Πιν. 2** Εξισώσεις ονομαστικού φόρτου / φόρτου ανάπτυξης ανά κατηγορία έργου  
 ( $MM_{NOM} = MM_{DEV}$ )

Κατηγορία	Ονομαστικός φόρτος	Διάρκεια ανάπτυξης
Οργανική	$MM_{NOM} = 3.2(KDSI)^{1.05}$	$TDEV = 2.5(MM_{DEV})^{0.38}$
Ημιαποσπασμένη	$MM_{NOM} = 3.0(KDSI)^{1.12}$	$TDEV = 2.5(MM_{DEV})^{0.35}$
Ενσωματωμένη	$MM_{NOM} = 2.8(KDSI)^{1.20}$	$TDEV = 2.5(MM_{DEV})^{0.32}$

### 5.2.1.2. Ενδιάμεσο μοντέλο COCOMO81

Το ενδιάμεσο μοντέλο βασίζεται στις ίδιες αρχές όπως το βασικό, δηλ. μέγεθος μετρημένο σε KDSI και επιλογή της κατηγορίας έργου για τον υπολογισμό του ονομαστικού φόρτου, αλλά στη συνέχεια προχωρά στη ρύθμισή του με βάση έναν πολλαπλασιαστική φόρτου που εξαρτάται από συγκεκριμένα χαρακτηριστικά του έργου.

Ο πολλαπλασιαστική φόρτου με τη σειρά του υπολογίζεται δουλεύοντας με 15 παράγοντες κόστους που φαίνονται στον Πίνακα 3. Ο πίνακας δίνει μία συντομογραφία και μία σύντομη επεξήγηση για τον καθένα τους. Οι παράγοντες χωρίζονται σε 4 κατηγορίες (παράγοντες λογισμικού, υπολογιστή, προσωπικού και έργου). Ορισμένοι από αυτούς είναι μάλλον παρωχημένοι (π.χ. ο χρόνος εξυπηρέτησης χρήστη αναφέρεται στο χρόνο που διατίθεται σε time sharing υπολογιστές, κάτι που πλέον δε γίνεται συχνά), οι περισσότεροι όμως εξακολουθούν να είναι ισχυροί ακόμη και σήμερα.

Σε κάθε έναν παράγοντα ο εκτιμητής θα πρέπει, πάντα σύμφωνα με τα χαρακτηριστικά του έργου, να δώσει μία από τις κατηγορικές τιμές στις στήλες του πίνακα 4 (ονομαστική, υψηλή, κλπ). Ο πίνακας δίνει μία συνοπτική επεξήγηση, αλλά στην αναφορά [1] υπάρχουν αρκετά λεπτομερείς πίνακες που καθοδηγούν τον εκτιμητή στο να δώσει τιμές με υποκειμενικό αλλά συνεπή τρόπο.

Έχοντας επιλέξει τις κατηγορικές τιμές των 15 παραγόντων, χρησιμοποιώντας τον πίνακα 5, παίρνουμε αριθμητικές τιμές που ποσοτικοποιούν τη συμμετοχή του κάθε παράγοντα στο συνολικό κόστος. Το γινόμενο αυτών των 15 τιμών είναι ο πολλαπλασιαστική φόρτου  $q$ . Ο τελικός υπολογισμός του φόρτου γίνεται με βάση και πάλι με βάση τις τρεις εξισώσεις του πίνακα 2, οι οποίες δίνουν τον ονομαστικό φόρτο  $MM_{NOM}$ , σύμφωνα με τον τύπο  $MM_{NOM} = aKDSI^b$ , όπου  $a$ ,  $b$  εξαρτώνται από την κατηγορία έργου.

Ο υπολογισμένος ονομαστικός φόρτος αναπροσαρμόζεται τώρα με τον πολλαπλασιαστική φόρτου ώστε να μας δώσει τον τελικό φόρτο ανάπτυξης σύμφωνα με τον τύπο  $MM_{DEV} = q \cdot MM_{NOM}$ .

## Δραστηριότητα 5.2

Έστω ότι θέλουμε να εκτιμήσουμε το κόστος ανάπτυξης ενός συστήματος το οποίο θα εκτελεί μεταφορές κεφαλαίων μεταξύ π.χ. χρηματιστηριακών εταιρειών, τραπεζών κλπ. Τα κύρια χαρακτηριστικά του συστήματος είναι η υψηλή αξιοπιστία, υψηλές επιδόσεις ταχύτητας, αυστηρό χρονοδιάγραμμα ανάπτυξης και καλή διασύνδεση χρήστη (GUI). Με τη βοήθεια ενός έμπειρου σχεδιαστή / προγραμματιστή που έχει αναπτύξει στο παρελθόν ένα παρόμοιο σύστημα έχει εκτιμηθεί ότι το μέγεθος θα είναι περίπου 10 KDSI. Από εσάς ζητείται να εκτιμήσετε το κόστος του έργου αυτού σύμφωνα με το μοντέλο COCOMO81. Υπενθυμίζεται ότι τα βήματα της μεθόδου εφαρμογής του μοντέλου είναι τα ακόλουθα:

*Εκτίμηση μεγέθους (έχει ήδη γίνει, 10 KDSI)*

*Επιλογή κατηγορίας έργου*

*Υπολογισμός ονομαστικού φόρτου*

*Ανάθεση τιμών στους παράγοντες κόστους*

*Υπολογισμός Πολλαπλασιαστή Φόρτου και Φόρτου Ανάπτυξης*

*Υπολογισμός απαιτούμενης διάρκειας ανάπτυξης*

Πίνακας 3. Παράγοντες κόστους κατά COCOMO.

Συνομογραφία παράγοντα κόστους	Περιγραφή	Κατηγορία Παράγοντα
RELY	Απαιτούμενη αξιοπιστία λογισμικού	ΛΟΓΙΣΜΙΚΟ
DATA	Μέγεθος βάσης δεδομένων	
CPLX	Πολυπλοκότητα συστήματος	
TIME	Περιορισμοί στο χρόνο εκτέλεσης	ΥΠΟΛΟΓΙΣΤΗΣ
STOR	Περιορισμοί στην κύρια μνήμη	
VIRT	Διαθεσιμότητα πλατφόρμας ανάπτυξης	
TURN	Χρόνος εξυπηρέτησης χρήστη	
ACAP	Ικανότητα αναλυτών	ΠΡΟΣΩΠΙΚΟ
AEXP	Εμπειρία αναλυτών σε παρόμοιες εφαρμογές	
PCAP	Ικανότητα προγραμματιστών	
VEXP	Εμπειρία με την πλατφόρμα ανάπτυξης	
LEXP	Εμπειρία με τη γλώσσα προγραμματισμού	
MODP	Χρησιμοποίηση μοντέρνων διεργασιών ανάπτυξης	ΕΡΓΟ
TOOL	Χρησιμοποίηση εργαλείων ανάπτυξης	
SCED	Απαιτούμενο χρονοδιάγραμμα	

	ανάπτυξης	
--	-----------	--

**Πίν. 4. Κατάταξη παραγόντων κόστους κατά COCOMO81**

Παράγοντας κόστους	Χαμηλότερη	χαμηλή	ονομαστική	υψηλή	υψηλότερη	υψηλότερη
RELY	Αποτέλεσμα: μικρή ενόχληση	Μικρές, εύκολα 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%XΔX E	70%XΔXE	85%XΔXE	95%XΔ XE
STOR			<50%XΔM	70%XΔM	85%XΔM	95%XΔ M
VIRT		Σημαντική αλλαγή κάθε 12 μήνες.	Σημαντική αλλαγή κάθε 6 μήνες.	Σημαντική αλλαγή κάθε 2 μήνες.	Σημαντική αλλαγή κάθε 2 εβδομάδες.	
TURN		Διαλογική	ΜΧΚΕ<4 ώρες	4 έως 8 ώρες	>12 ώρες	
ACAP	15 ET	35 ET	55 ET	75 ET	90 ET	
AEXP	<4 μήνες	1 χρόνο	3 χρόνια	6 χρόνια	12 χρόνια	
PCAP	15 ET	35 ET	55 ET	75 ET	90 ET	
VEXP	<1 μήνα	4 μήνες	1 χρόνο	3 χρόνια		
LEXP	<1 μήνα	4 μήνες	1 χρόνο	3 χρόνια		
MODP	Όχι	Βασική χρήση	Μερική χρήση	Γενική χρήση	Αποκλειστική χρήση	

TOO L	Βασικά εργαλεία μικρο- υπολογιστή	Βασικά εργαλεία μινι- υπολογιστή	Βασικά εργαλεία μεσαίου και μεγάλου υπολογιστή	Ισχυρά εργαλεία προγραμματισμού και ελέγχου ορθότητας	Επιπρόσθε- τα εργαλεία για προδιαγραφές απαιτήσεων, σχεδίου κλπ.	
SCE D	75%	85%	100%	130%	160%	

Επεξήγηση συντομογραφιών:

- L=(DBbytes)/(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

**Πίνακας 5. Πολλαπλασιαστές προσπάθειας**

Κατάταξη						
Παράγον- τας κόστους	χαμηλότ- ερη	χαμηλ ή	ονομαστ ική	υψηλ ή	υψηλότ ερη	υψηλότ ατη
RELY	.75	.88	1.00	1.15	1.40	
DATA		.94	1.00	1.08	1.16	
CPLX	.70	.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1.66
STOR			1.00	1.06	1.21	1.56
VIRT		.87	1.00	1.15	1.30	
TURN		.87	1.00	1.07	1.15	
ACAP	1.46	1.19	1.00	.86	.71	
AEXP	1.29	1.13	1.00	.91	.82	

PCAP	1.42	1.17	1.00	.86	.70	
VEXP	1.21	1.10	1.00	.90		
LEXP	1.14	1.07	1.00	.95		
MODR	1.24	1.10	1.00	.91	.82	
TOOL	1.24	1.10	1.00	.91	.83	
SCED	1.23	1.08	1.00	1.04	1.10	

### Άσκηση Αυτοαξιολόγησης 5.2

Κατά την ανάπτυξη ενός έργου λογισμικού μεγέθους 10.000 εντολών, αναπτύχθηκε σειρά συστατικών κώδικα (modules) για τις ανάγκες της δοκιμής ολοκλήρωσης του συστήματος (integration testing), δηλ. οδηγοί (drivers) και στελέχη (stubs). Το συνολικό τους μέγεθος ήταν 500 εντολές. Επίσης αναπτύχθηκαν 1000 εντολές για δοκιμές ελέγχου αποδοχής από τους τελικούς χρήστες οι οποίες και αποτέλεσαν μέρος του τελικού κώδικα. Να εξηγήσετε πως οι δύο αυτές περιπτώσεις κώδικα θα επηρεάσουν την τελική εκτίμηση σύμφωνα με το μοντέλο COCOMO81.

### Δραστηριότητα 5.3

Να εκτιμήσετε το κόστος ενός πληροφοριακού συστήματος που θα επιλέξετε μόνοι σας από παραδείγματα άλλων θεματικών ενότητων. Να κάνετε παραδοχές και υποθέσεις όπου δεν έχετε επαρκείς πληροφορίες. Για το μέγεθος, να θεωρήσετε ότι το σύστημά σας θα αποτελείται από 50 KDSI.

### Δραστηριότητα 5.4

Να εκτιμήσετε το φόρτο μίας εργασίας σας προγραμματισμού από αυτήν ή άλλη θεματική ενότητα χρησιμοποιώντας το μοντέλο COCOMO81. Μπορείτε να μετρήσετε τον κώδικα απλά μετρώντας τις γραμμές του αρχείου που περιέχει τον πηγαίο κώδικα



(π.χ. το αρχείο **\*\*\*.c**). Προσέξτε μήπως ο αριθμός των σχολίων και των κενών γραμμών (δεν είναι ‘εντολές’) είναι πολύ μεγάλος. Στην περίπτωση αυτή θα πρέπει να αφαιρέσετε σχόλια και κενές γραμμές.

Να συγκρίνετε την εκτίμηση με την προσπάθεια που εσείς αφιερώσατε. Τι παρατηρείτε;

### 5.2.2. Μοντέλο COCOMO-2.0

Όπως προαναφέρθηκε, η κοστολόγηση είναι μια από τις δυσκολότερες δραστηριότητες στην παραγωγή και εκτίμηση του λογισμικού ([1]). Οι λόγοι είναι πολλοί: οι ελλιπείς πληροφορίες στις αρχικές φάσεις της ανάπτυξης, η δυσκολία στην εφαρμογή των διαθέσιμων εργαλείων και μεθόδων αλλά και τα μειονεκτήματά τους, η απουσία ιστορικών δεδομένων κόστους, η ελλιπής διάδοση της τεχνογνωσίας της κοστολόγησης, κλπ. Υπενθυμίζεται ότι οι τρεις βασικοί τρόποι κοστολόγησης είναι η κοστολόγηση με βάση την εμπειρία ενός ειδικού, τα παραμετρικά μοντέλα κοστολόγησης και η κοστολόγηση με βάση τις αναλογίες.

Είδαμε προηγουμένως ότι ένα από τα περισσότερο γνωστά εργαλεία στην Τεχνολογία Λογισμικού είναι το μοντέλο COCOMO81. Η επιτυχία του οφείλεται κυρίως στο ότι παράγει εκτιμήσεις με διάφανο τρόπο: κάθε επιλογή τιμής του χρήστη είναι ξεκάθαρο πως επηρεάζει το τελικό αποτέλεσμα. Έτσι το COCOMO81 αποτελεί εκτός των άλλων και ένα σημαντικό εργαλείο διδασκαλίας της κοστολόγησης. Παρ’ όλα αυτά το COCOMO81 έχει μερικά πολύ σημαντικά μειονεκτήματα:

1. Είναι βασισμένο σε έργα προγενέστερα του 1980. Αυτό σημαίνει ότι η παραγωγικότητα που έχει ενσωματωθεί στο μοντέλο δεν αντιστοιχεί στους σημερινούς ρυθμούς παραγωγικότητας. Επί πλέον είναι λογικό να υποθέσει κανείς ότι οι παράγοντες που επηρεάζουν το κόστος έχουν αλλάξει ή έχουν προστεθεί νέοι.
2. Η εκτίμηση στις αρχικές φάσεις γίνεται με βάση την εκτίμηση του μεγέθους του κώδικα εκφρασμένου σε εντολές της γλώσσας προγραμματισμού. Αν και υπάρχουν υποβοηθητικές μέθοδοι (π.χ. η *SSM – Software Sizing Method* του Bozoki [5]), αυτό

το είδος εκτίμησης εξακολουθεί να είναι μια δύσκολη εργασία που κινδυνεύει να οδηγήσει σε λάθος αποτελέσματα.

3. Η εκτίμηση γίνεται με βάση αρκετές επί μέρους υποκειμενικές εκτιμήσεις παραγόντων κόστους. Επί πλέον, αντιστοιχίζονται αριθμητικές τιμές (π.χ. 1, 1.10, 1.25 ...) σε ονομαστικές τιμές (NOM, HIGH, VERY HIGH ...), πάνω στις οποίες στη συνέχεια, εφαρμόζονται αυθαίρετα μαθηματικές πράξεις (π.χ. πολλαπλασιασμός).

Στην δεκαετία του 1990 αναπτύχθηκε το μοντέλο COCOMO-2.0 [2] που προσπαθεί να επιλύσει τα δυο πρώτα προβλήματα – το τρίτο πρόβλημα εξακολουθεί να ισχύει. Οι ερευνητές που σχεδίασαν το COCOMO-2.0, αναγνωρίζοντας το πρόβλημα της εκτίμησης του μεγέθους στις αρχικές φάσεις, διαχωρίζουν τον τρόπο εκτίμησης ανάλογα με τη φάση της ανάπτυξης του έργου. Οι τρεις φάσεις του έργου για τις οποίες διαφοροποιείται ο τρόπος εκτίμησης είναι:

- αρχική φάση (Application Composition)
- φάση σχεδίασης (Early Design)
- φάση υλοποίησης (Postarchitecture)

Στην αρχική φάση, συνήθως κατασκευάζεται ένα πρωτότυπο, με σκοπό να αντιμετωπισθούν κίνδυνοι σχετικά με την διαπροσωπεία χρήστη (user interface), την απόδοση του λογισμικού ή την ωριμότητα της τεχνολογίας που θα χρησιμοποιηθεί. Δεδομένης της δυσκολίας εκτίμησης του μεγέθους σε αυτή την φάση, η εκτίμησή του γίνεται με βάση ένα νέο σύστημα μέτρησης, τους λεγόμενους *αντικειμενικούς βαθμούς (object points)*. Στην φάση της σχεδίασης η εκτίμηση του μεγέθους γίνεται με βάση τους βαθμούς λειτουργίας και στην φάση της υλοποίησης με βάση την μέτρηση των γραμμών του πηγαίου κώδικα.

Ο υπολογισμός των αντικειμενικών βαθμών γίνεται πάνω σε στοιχεία των προδιαγραφών, όπως ο αριθμός των οθονών, των αναφορών και των στοιχείων γλωσσών 3ης γενιάς (ξεχωριστά σε πελάτη και σε εξυπηρετητή). Στοιχεία γλωσσών 3ης γενιάς μπορεί να είναι modules, functions, κλπ. που εξυπηρετούν πολύπλοκότερες ανάγκες σε σχέση με οθόνες και αναφορές. Όπως εύκολα αντιλαμβάνεται κανείς, οι αντικειμενικοί βαθμοί μοιάζουν με τους βαθμούς λειτουργίας (βλ. παρακάτω): υπάρχουν και εδώ system

types, που έχουν ένα βαθμό πολυπλοκότητας ανάλογα με κάποια χαρακτηριστικά τους. Η διαφορά είναι ότι εδώ δεν υπολογίζεται ένας αριθμός βαθμών λειτουργίας που αναπαριστά την ποσότητα λειτουργικότητας του συστήματος, αλλά υπολογίζεται απ' ευθείας ένας συντελεστής παραγωγικότητας. Ο λόγος είναι ότι αυτό που ενδιαφέρει ένα μοντέλο κόστους είναι πρωταρχικά η παραγωγικότητα και όχι ένα ενδιάμεσο χαρακτηριστικό του συστήματος, όπως θα ήταν η λειτουργικότητα.

Ο μαθηματικός τύπος για τον υπολογισμό του κόστους έχει παρόμοια μορφή με αυτή του κλασικού μοντέλου:

$$E = b S^c m(X)$$

Η αλλαγή εντοπίζεται κυρίως στο νέο τρόπο υπολογισμού του συντελεστή  $c$ . Τα χαρακτηριστικά του έργου που χρησιμοποιούνται είναι κατά σειρά:

1. Μέγεθος (Size)
2. Επαναχρησιμοποίηση κώδικα (Reuse)
3. Βαθμός αλλαγής των απαιτήσεων (Breakage)
4. Συντήρηση (Maintenance)
5. Συντελεστής  $c$  στον τύπο υπολογισμού (Scale)
6. Συντελεστές κόστους σχετικοί με το προϊόν (Product cost drivers)
7. Συντελεστές κόστους σχετικοί με την πλατφόρμα ανάπτυξης (Platform cost drivers)
8. Συντελεστές κόστους σχετικοί με το προσωπικό (Personnel cost drivers)
9. Συντελεστές κόστους σχετικοί με το έργο (Project cost drivers)

Η επαναχρησιμοποίηση και ο βαθμός αλλαγής των απαιτήσεων δεν έχουν έννοια στην αρχική φάση. Στη σχεδίαση, η επαναχρησιμοποίηση υπολογίζεται ως το ποσοστό κώδικα που επαναχρησιμοποιείται χωρίς μετατροπές (unmodified reuse) και με μετατροπές (modified reuse). Στην φάση της υλοποίησης υπολογίζεται ως αριθμός ισοδύναμων γραμμών κώδικα.

Η συντήρηση υπολογίζεται με βάση τους αντικειμενικούς βαθμούς, τον μέσο όρο των αλλαγών, και με βάση τον βαθμό επαναχρησιμοποίησης στις προχωρημένες φάσεις.

Ο συντελεστής  $c$  έχει σταθερή τιμή ίση με '1' στην αρχική φάση, ενώ στις άλλες δυο φάσεις μπορεί να πάρει τιμές στο διάστημα [1.02, 1.26]. Η τιμή του  $c$  εξαρτάται από παράγοντες, όπως ύπαρξη τεχνογνωσίας προερχόμενη από εμπειρία με παλαιότερα παρόμοια έργα (precedentedness), ευελιξία σε σχέση με τους στόχους (conformity, flexibility), επιτυχής ή όχι αντιμετώπιση των κινδύνων (risk resolution), αποδοτικότητα συνεργασίας της ομάδας ανάπτυξης και επίπεδο ωριμότητας κατά SEI (βλ. κεφ. για μοντέλα ωριμότητας). Η τιμή του συντελεστή  $c$  υπολογίζεται με τον εξής τρόπο: η τιμή 'πάρα πολύ υψηλό' (extra high) αντιστοιχεί στην τιμή 0, η τιμή 'πολύ υψηλό' (very high) στην τιμή 1, η τιμή 'υψηλό' (high) στην τιμή 2, η τιμή 'ονομαστικό' (nominal) στην τιμή 3, η τιμή 'χαμηλό' (low) στην τιμή 4, και η τιμή 'πολύ χαμηλό' (very low) στην τιμή 5.

Δεν θα παρουσιάσουμε εδώ με περισσότερη λεπτομέρεια την αντιστοιχία κατηγορικών και αριθμητικών τιμών, όπως κάναμε για το μοντέλο COCOMO81, για τρεις λόγους: (α) έχουμε ήδη κατανοήσει τον τρόπο με τον οποίο μπορεί κανείς να συνθέσει το κόστος, (β) δεν είναι γνωστές όλες οι λεπτομέρειες του COCOMO 2.0 και (γ) το μοντέλο COCOMO 2.0 μπορεί να εφαρμοσθεί μέσω μίας διαδικτυακής εφαρμογής (βλ. Σχόλιο Μελέτης στο τέλος της παραγράφου).

Όσον αφορά τους παράγοντες κόστους, παρατηρούμε ότι αυτοί δεν λαμβάνονται καθόλου υπ' όψιν στην αρχική φάση (αναγνωρίζοντας ότι δεν είναι ακόμη διαθέσιμες οι πληροφορίες που θα επιτρέψουν την αξιολόγησή τους). Στη φάση της σχεδίασης, όσον αφορά το προϊόν, λαμβάνεται υπ' όψιν η πολυπλοκότητα και η ανάγκη ή όχι για κατασκευή επαναχρησιμοποιήσιμου λογισμικού στα πλαίσια του έργου. Στην φάση της υλοποίησης, παρατηρούμε ότι έχουν προστεθεί στους κλασικούς παράγοντες του μοντέλου COCOMO οι ανάγκες της τεκμηρίωσης. Στους παράγοντες του προσωπικού έχει προστεθεί η συνέχεια του προσωπικού (personnel continuity), δηλ. το κατά πόσον παρέμεινε σταθερό το προσωπικό του έργου. Τέλος, στους παράγοντες έργου έχει προστεθεί η ανάπτυξη από ομάδες που εργάζονται σε διαφορετικές γεωγραφικές τοποθεσίες.

## Δραστηριότητα 5.5

Μελετώντας την περιγραφή των δύο μοντέλων να προσπαθήσετε να εντοπίσετε τις διαφορές και βελτιώσεις που παρουσιάζει το COCOMO 2.0 σε σχέση με το COCOMO81.

### Παράδειγμα εφαρμογής

Το μοντέλο COCOMO 2.0 θα εφαρμοσθεί πάνω σε ένα σύστημα ενοικίασης αυτοκινήτων, ώστε να φανούν τα βασικά βήματα που πρέπει να ακολουθηθούν. Το σύστημα αυτό επιτρέπει την επιλογή αυτοκινήτου, την επιλογή πακέτου πρόσθετων παροχών (ασφάλεια κλπ), παρουσιάζει στον χρήστη στοιχεία σχετικά με το κόστος κατανάλωσης καυσίμου ανάλογα με τον τύπο του αυτοκινήτου και τη διαδρομή, το ιστορικό του αυτοκινήτου (ενοικιάσεις, επισκευές), κλπ.

### *ΑΡΧΙΚΗ ΦΑΣΗ*

---

Για τον υπολογισμό των αντικειμενικών βαθμών ελέγχουμε:

1. τον αριθμό πινάκων που χρησιμοποιούνται από τις οθόνες και αναφορές (reports) του συστήματος, σε πελάτη και στον εξυπηρετητή ξεχωριστά.
2. το ποσοστό των οθονών, αναφορών και τμημάτων κώδικα που επαναχρησιμοποιούνται από άλλες εφαρμογές

Ας υποθέσουμε ότι η δεύτερη περίπτωση δεν συντρέχει, ενώ υπάρχουν τρεις οθόνες και μια αναφορά. Οι τιμές δίνονται με βάση υποθέσεις σχετικά με τον αριθμό των προβολών (views) που υπάρχουν σε κάθε πίνακα (βλ. πίνακες 6, 7 και 8):

Οθόνη επιλογής αυτοκινήτου: χρησιμοποιείται ένας πίνακας των διαθέσιμων αυτοκινήτων: αποδίδεται η τιμή απλό (επίπεδο πολυπλοκότητας από τον πίνακα 6), με συντελεστή πολυπλοκότητας = 1 (από τον πίνακα 8)

Οθόνη επιλογής πακέτου πρόσθετων παροχών: χρησιμοποιείται ένας πίνακας παροχών και πληροφοριών, ένας πίνακας με τιμολόγιο των παροχών, και ένας πίνακας με

προτάσεις συνδυασμών παροχών: αποδίδεται η τιμή μέτριας δυσκολίας (πίν. 6), συντ. πολ. = 2 (πίν. 8)

Οθόνη κόστους κατανάλωσης καυσίμου: χρησιμοποιείται ένας πίνακας καυσίμων, ένας πίνακας τύπων διαδρομών και ένας πίνακας προκαθορισμένων καταναλώσεων: αποδίδεται η τιμή μέτριας δυσκολίας (πίν 6), συντ. πολ. = 2 (πίν. 8)

Αναφορά για συνολικές ενοικιάσεις ανά μήνα και χρόνο, περιοδικές και έκτακτες συντηρήσεις: αποδίδεται η τιμή μέτριας δυσκολίας (πίν 7), συντ. πολ. = 5 (πίν. 8)

Αθροίζοντας τις τιμές αυτές προκύπτει ο αριθμός των αντικειμενικών βαθμών NOPS (new object points) =  $1 + 2 + 2 + 5 = 10$ , που εκφράζει το μέγεθος του λογισμικού ανεξάρτητα από τις γραμμές κώδικα από τις οποίες αυτό θα αποτελείται όταν θα υλοποιηθεί.

Υποθέτοντας ότι οι προγραμματιστές έχουν πολύ χαμηλή εμπειρία / ικανότητα μηχανικών και πολύ χαμηλή ωριμότητα και ικανότητα στη χρήση εργαλείων CASE (μία μάλλον αδόκιμη ομάδα ανάπτυξης!), από τον πίν. 9 υπολογίζουμε το ρυθμό παραγωγικότητας ίσο με 4 NOPS/MM, δηλ. με προσπάθεια που αντιστοιχεί σε έναν ανθρωπομήνα κατασκευάζεται λογισμικό που αντιστοιχεί σε 4 νέους αντικειμενικούς βαθμούς.

**Πίνακας 6. Επίπεδα πολυπλοκότητας αντικειμενικών βαθμών για οθόνες**

	Αριθμός και προέλευση των πινάκων δεδομένων		
Αριθμός των προβολών που περιέχονται	συνολικά <4 (<2 στον εξυπηρετητή, <2 στον πελάτη)	συνολικά <8 (2-3 στον εξυπηρετητή, 3-5 στον πελάτη)	συνολικά >8 (>3 στον εξυπηρετητή, >5 στον πελάτη)
<3	απλό	απλό	μέτριας δυσκολίας
3-7	απλό	μέτριας δυσκολίας	Δύσκολο
>8	μέτριας δυσκολίας	δύσκολο	Δύσκολο

**Πίνακας 7. Επίπεδα πολυπλοκότητας αντικειμενικών βαθμών για αναφορές**

Επίπεδα πολυπλοκότητας βαθμών εφαρμογής για αναφορές			
	Αριθμός και προέλευση των πινάκων δεδομένων		
Αριθμός των	συνολικά <4 (<2	συνολικά <8 (2-3	συνολικά >8 (>3

ενοτήτων που περιέχονται	στον εξυπηρετητή, <2 στον πελάτη)	στον εξυπηρετητή, 3-5 στον πελάτη)	στον εξυπηρετητή, >5 στον πελάτη)
0 ή 1	απλό	Απλό	μέτριας δυσκολίας
2 ή 3	απλό	μέτριας δυσκολίας	δύσκολο
>4	μέτριας δυσκολίας	Δύσκολο	Δύσκολο

**Πίνακας 8. Συντελεστές πολυπλοκότητας για αντικειμενικούς βαθμούς**

Τύπος Στοιχείου	Απλό	Μέτριας Δυσκολίας	Δύσκολο
Οθόνη	1	2	3
Αναφορά	2	5	8
Συστατικό 3GL	-	-	10

**Πίνακας 9. Επιλογή Συντελεστή Παραγωγικότητας  $c$**

Εμπειρία και Ικανότητα των μηχανικών	Πολύ χαμηλή	Χαμηλή	Μέτρια	Υψηλή	Πολύ υψηλή
Ωριμότητα και Ικανότητα στη χρήση εργαλείων CASE	Πολύ χαμηλή	Χαμηλή	Μέτρια	Υψηλή	Πολύ υψηλή
Συντελεστής Παραγωγικότητας	4	7	13	25	50

Ο τελικός υπολογισμός του κόστους με βάση τις πληροφορίες από την αρχική φάση έχει σαν αποτέλεσμα: φόρτος ανάπτυξης =  $\text{NOPS} / \text{Συντ. Παρ.} = 10/4 = \mathbf{2.5 \text{ MM}}$ .

### Σχόλιο Μελέτης

Στο διαδίκτυο, υπάρχει αυτή τη στιγμή διαθέσιμο ένα εργαλείο για την εφαρμογή του μοντέλου COCOMO 2.0, στη σελίδα:

[http://sunset.usc.edu/COCOMOII/expert\\_cocomo/expert\\_cocomo.html](http://sunset.usc.edu/COCOMOII/expert_cocomo/expert_cocomo.html)

## **Δραστηριότητα 5.6**

Να επιλέξετε ένα σύστημα (ή ένα υπο-σύστημά του αν είναι μεγάλο) το οποίο χρησιμοποιείτε συχνά (π.χ. κάποια εφαρμογή στο Διαδίκτυο, ή την εφαρμογή για το ηλεκτρονικό σας ταχυδρομείο). Να εφαρμόσετε το μοντέλο COCOMO-2.0 κάνοντας υποθέσεις για την ομάδα ανάπτυξης.

### **5.2.3 Μοντέλα Παλινδρόμησης**

Για την κατανόηση της εκτίμησης με μοντέλα παλινδρόμησης είναι χρήσιμο να θεωρήσουμε τον πίνακα 10, ο οποίος παρουσιάζει μία βάση δεδομένων κόστους, δηλ. ένα σύνολο από ιστορικά Έργα  $1, \dots, n$  με καταγραμμένα χαρακτηριστικά  $X_{n1}, X_{n1}, \dots, X_{nl}$  (μέγεθος, διάρκεια, τύπος λογισμικού, γλώσσα προγραμματισμού κλπ). Η σημαντικότερη πληροφορία είναι το κόστος  $E_n$  του έργου. Στην τελευταία γραμμή του πίνακα έχουμε τοποθετήσει το νέο έργο του οποίου τα χαρακτηριστικά θεωρούμε ότι είναι γνωστά ενώ το κόστος παραμένει άγνωστο και είναι ο στόχος της εκτίμησης. Ο ίδιος πίνακας θα μας χρησιμεύσει στην επόμενη παράγραφο για την κατανόηση της μεθόδου των αναλογιών.



**Πίν 10. Βάση δεδομένων κόστους για την εφαρμογή της μεθόδου της κοστολόγησης με παλινδρόμηση και με αναλογίες**

	<b>Κόστος</b>	<i>Χαρ/κό 1</i>	<i>Χαρ/κό 2</i>	...	<i>Χαρ/κό k</i>
<i>Έργο 1</i>	$E_1$	$X_{11}$	$X_{12}$	...	$X_{1k}$
<i>Έργο 2</i>	$E_2$	$X_{21}$	$X_{22}$	...	$X_{2k}$
<i>Έργο 3</i>	$E_2$	$X_{31}$	$X_{32}$	...	$X_{3k}$
....	...	...	...	...	...
<i>Έργο n</i>	$E_n$	$X_{n1}$	$X_{n2}$	...	$X_{nk}$
<i>Νέο Έργο</i>	Άγνωστο	$Y_1$	$Y_2$	...	$Y_k$

Η ανάλυση παλινδρόμησης είναι η στατιστική μεθοδολογία η οποία μας επιτρέπει την πρόβλεψη μίας ή περισσότερων *εξαρτημένων* μεταβλητών από ένα σύνολο *ανεξάρτητων* μεταβλητών. Η εξαρτημένη μεταβλητή είναι μία τυχαία μεταβλητή και μπορεί να παίρνει τυχαίες τιμές από ένα σύνολο τιμών. Οι τιμές της εξαρτώνται από τις τιμές των ανεξάρτητων μεταβλητών. Θα πρέπει να κατασκευασθεί ένα μοντέλο παλινδρόμησης για να εξηγήσει την επίδραση των ανεξάρτητων μεταβλητών στην εξαρτημένη και για να δώσει τη δυνατότητα να εκτιμηθεί η τιμή της εξαρτημένης μεταβλητής δεδομένων των τιμών των ανεξάρτητων.

Η γενική μορφή ενός μοντέλου παλινδρόμησης για την εκτίμηση κόστους με χρήση του Πίν.10 είναι η ακόλουθη (E είναι ο φόρτος του νέου έργου):

$$E = \beta_0 + \beta_1 Y_1 + \dots + \beta_k Y_k + \varepsilon$$

Οι συντελεστές  $\beta_0, \beta_1, \dots, \beta_k$  ονομάζονται συντελεστές παλινδρόμησης, ενώ  $\varepsilon$  είναι το σφάλμα εκτίμησης. Ο υπολογισμός τους γίνεται με χρήση της μεθόδου των ελαχίστων τετραγώνων και απαιτεί την εφαρμογή ενός στατιστικού εργαλείου.<sup>3</sup> Οι τιμές τόσο των ανεξάρτητων όσο και των εξαρτημένων μεταβλητών πρέπει να είναι κανονικά κατανοημένες. Συχνά γίνεται λογαριθμικός μετασχηματισμός (μετατροπή σε

<sup>3</sup> Ο αναγνώστης μπορεί εύκολα να βρει ένα τέτοιο εργαλείο στην εφαρμογή λογιστικών φύλλων *Excel* του πακέτου MICROSOFT OFFICE.

λογαριθμική κλίμακα) των μεταβλητών που αναπαριστούν το φόρτο και το μέγεθος του λογισμικού για να επιτευχθεί ο περιορισμός της κανονικής κατανομής. Μετά τη δημιουργία του μοντέλου παλινδρόμησης θα πρέπει να βεβαιωθούμε ότι η διασπορά των σφαλμάτων εκτίμησης για κάθε ένα από τα έργα Έργα 1, ..., n είναι τυχαία. Θα πρέπει επομένως να δημιουργήσουμε ένα διάγραμμα διασποράς και να δούμε τα σφάλματα να διαμοιράζονται στο επίπεδο χωρίς να παρουσιάζουν κάποιο σχήμα. Η πλήρης περιγραφή της κατασκευής ενός παλινδρομικού μοντέλου είναι εκτός των ορίων του παρόντος συγγράμματος και ο αναγνώστης παραπέμπεται στη βιβλιογραφία της στατιστικής.

Ένα παράδειγμα παλινδρομικού μοντέλου (έχει χρησιμοποιηθεί ανάλυση μεταβλητότητας) είναι το εξής:

$$lneffort = 3,3 + 0,78*lnsize - t,$$

όπου:

$lneffort$  = λογάριθμος του φόρτου σε ώρες εργασίας

$lnsize$  = λογάριθμος του μέγεθος σε βαθμούς λειτουργίας

$t$  = ένταση χρήσης εργαλείου  $T$  σύμφωνα με τον πίνακα 11.

**Πίν 11. Συντελεστές χρήσης εργαλείου T**

Ένταση Χρήσης	Συντελεστής
Μηδενική	0
Ελάχιστη	0,199
Μέτρια	0,241
Σημαντική	1,014
Πολύ συχνή	1,022

### Άσκηση Αυτοαξιολόγησης 5.4

Χρησιμοποιώντας το παράδειγμα του παλινδρομικού μοντέλου που σας δόθηκε, να υπολογίσετε το φόρτο τριών έργων με τα εξής χαρακτηριστικά:

Έργο Α: Μέγεθος = 50 Βαθμοί Λειτουργίας, μηδενική χρήση Τ

Έργο Β: Μέγεθος = 50 Βαθμοί Λειτουργίας, σημαντική χρήση Τ

Έργο Γ: Μέγεθος = 1000 Βαθμοί Λειτουργίας, μηδενική χρήση Τ

Τι παρατηρείτε; Επίσης πώς εξηγούνται οι τιμές του Πίν. 11; Γιατί πιστεύετε ότι μέχρι την τιμή χρήσης *μέτρια* το κέρδος παραγωγικότητας είναι χαμηλό;

### 5.3 Εκτίμηση με βάση τις αναλογίες

---

Η μέθοδος αυτή χρησιμοποιείται εναλλακτικά, όταν δεν είναι διαθέσιμο ένα αλγοριθμικό μοντέλο κόστους, υπάρχει μια ‘δεξαμενή’ από έργα των οποίων έχουν μετρηθεί και καταγραφεί το κόστος και τα χαρακτηριστικά τους [3]. Η μέθοδος αυτή, υπό ορισμένες συνθήκες (π.χ. μικρή βάση ιστορικών έργων) μπορεί να δώσει καλύτερες εκτιμήσεις σε σχέση με τα παραμετρικά αλγοριθμικά μοντέλα, επειδή βασίζεται στην πιθανή ομοιότητα μερικών έργων με άλλα. Σύμφωνα με τη μέθοδο, το προς εκτίμηση έργο  $E$  συγκρίνεται με τα διαθέσιμα ολοκληρωμένα έργα. Επιλέγονται ορισμένα εξ αυτών ως “ανάλογα” και με βάση το καταγραμμένο κόστος τους γίνεται η εκτίμηση του κόστους του  $E$ .

Η επιλογή των “anáλογων” έργων γίνεται με βάση τα χαρακτηριστικά του έργου και την έννοια της “απόστασης” (βλ. πίνακα 10). Μπορούν να χρησιμοποιηθούν διάφορες αποστάσεις. Η περισσότερο χρησιμοποιούμενη έννοια απόστασης είναι η ευκλείδεια απόσταση, η οποία δίνεται από τον τύπο:

$$d_{new,i} = \left\{ \sum_{j=1}^k (Y_j - X_{ij})^2 \right\}^{1/2}, \quad i = 1, 2, \dots, n$$

η οποία συνδυάζει τις ‘αποστάσεις’ των τιμών ενός χαρακτηριστικού του νέου έργου από τις τιμές του χαρακτηριστικού κάθε ιστορικού έργου. Για να μπορέσουμε να χρησιμοποιήσουμε μεγέθη μετρημένα σε διαφορετικές κλίμακες θα πρέπει πρώτα να τα κανονικοποιήσουμε (βλ. Παράδειγμα υπολογισμού απόστασης).

Άλλη μια απόσταση που μπορεί να χρησιμοποιηθεί είναι η γνωστή με τον όρο *feature dissimilarity* και εφαρμόζεται όταν τα χαρακτηριστικά μπορεί να έχουν και μη αριθμητικές τιμές (π.χ. ‘υψηλό’, ‘χαμηλό’). Σε αυτή την περίπτωση η απόσταση είναι ίση με 0 αν οι δυο τιμές είναι ίσες, και με 1 αν οι τιμές είναι διαφορετικές. Η απόσταση είναι ίση με τον όρο  $(Y_j - X_{ij})^2$ , αν οι τιμές των χαρακτηριστικών είναι αριθμητικές. Αυτές οι επί μέρους αποστάσεις κάθε χαρακτηριστικού μπορούν να συνδυασθούν με έναν άλλο τύπο απόστασης, ώστε να δώσουν μια συνολική απόσταση μεταξύ των δυο έργων. Μπορούμε για παράδειγμα να συνδυάσουμε την προσέγγιση αυτή με την ευκλείδεια απόσταση, όπως φαίνεται στο παρακάτω παράδειγμα.

Αφού ευρεθούν ανάλογα έργα, επιλέγονται μερικά από αυτά και το κόστος του νέου έργου υπολογίζεται με κάποια απλή στατιστική παράμετρο, π.χ. τον μέσο όρο ή το διάμεσο του κόστους των ανάλογων έργων. Η επιλογή των παραμέτρων της μεθόδου (τύπου απόστασης, αριθμού ανάλογων έργων, στατιστικής παραμέτρου κλπ) μπορεί να γίνει με βάση μία διεξοδική ανάλυση που πρέπει να προηγηθεί, κατά τη διάρκεια της οποίας πρέπει να δοκιμασθούν διάφοροι συνδυασμοί των παραμέτρων αυτών και να συγκριθούν τα αποτελέσματά τους. Για τον αριθμό των ανάλογων έργων ισχύει γενικά ο κανόνας ότι για μικρές ιστορικές βάσεις έργων (π.χ. 20 έργα) πρέπει να παίρνουμε λίγες αναλογίες (π.χ. 1, 2 ή 3).

### **Σχόλιο Μελέτης**

Για την εφαρμογή της παραμετροποίησης της μεθόδου και στη συνέχεια της ίδιας της μεθόδου των αναλογιών μπορείτε να χρησιμοποιήσετε το εργαλείο Angel που διατίθεται στη διεύθυνση:

[http://dec.bmth.ac.uk/dec\\_ind/decind22/web/Angel.html](http://dec.bmth.ac.uk/dec_ind/decind22/web/Angel.html)

Επίσης μπορείτε να κατεβάσετε το περισσότερο προχωρημένο εργαλείο BRACE-4 από την διεύθυνση

<http://sweng.csd.auth.gr>

### *Παράδειγμα εφαρμογής της μεθόδου των αναλογιών*

Δίνεται η παρακάτω ιστορική βάση έργων με το νέο έργο για το οποίο πρέπει να εκτιμηθεί το κόστος  $E'$ . Στόχος μας είναι να υπολογίσουμε την απόσταση (ανομοιότητα) των τριών ιστορικών έργων από το νέο έργο.

	Κόστος (Ανθρωπομήνες)	Μέγεθος (KDSI)	Μεγεθος (κανονικοποιημένο)	Τύπος
<i>Έργο 1</i>	50	120	1	Διαχείριση δεδομένων
<i>Έργο 2</i>	20	35	0.15	Πραγματικό χρόνου
<i>Έργο 3</i>	10	20	0	Διαχείριση δεδομένων
Νέο έργο	$E'$	70	0.5	Διαχείριση δεδομένων

Θα πρέπει πρώτα να κανονικοποιήσουμε την κλίμακα του μεγέθους στο διάστημα (0,1). Αυτό σημαίνει ότι η υψηλότερη τιμή θα πρέπει να αντιστοιχεί σε 1 και η χαμηλότερη στο 0. Θέτουμε επομένως  $20 = 0$  και  $120 = 1$ , οπότε προκύπτει η στήλη του κανονικοποιημένου μεγέθους ανάγοντας τη διαφορά της αρχικής τιμής από τη χαμηλότερη τιμή στο διάστημα που προκύπτει από τη διαφορά υψηλότερης και χαμηλότερης τιμής (π.χ. για την τιμή 35 έχουμε  $(35-20)/(120-20) = 0.15$ ). Τέλος το νέο έργο έχει μερική απόσταση (ως προς τον τύπο του έργου) ίση με 0 από τα έργα 1 και 3, και ίση με 1 από το έργο 2, επειδή το τελευταίο είναι διαφορετικού τύπου (πραγματικού χρόνου). Εφαρμόζοντας την ευκλείδεια απόσταση έχουμε:

$$D_{\text{new},1} = ((1-0.5)^2 + 0)^{1/2} = 0.5$$

$$D_{\text{new},2} = ((0.15-0.5)^2 + 1)^{1/2} = 1.06$$

$$D_{\text{new},3} = ((0-0.5)^2 + 0)^{1/2} = 0.5$$

Με βάση αυτές τις τιμές, το νέο έργο φαίνεται να προσομοιάζει στα έργα 1 και 3. Το κόστος του νέου έργου μπορεί τώρα να υπολογισθεί εύκολα χρησιμοποιώντας το μέσο όρο των Έργων 1 και 3. Επομένως  $E' = (50 + 10)/2 = 30$  ανθρωπομήνες.

□

Ένα από τα πλεονεκτήματα της κοστολόγησης με αναλογίες είναι ότι είναι πλησιέστερη στον ανθρώπινο τρόπο σκέψης. Η μέθοδος έχει τις ρίζες της σε ένα είδος τεχνικής της Τεχνητής Νοημοσύνης, της *συλλογιστικής με βάση περιπτώσεις (Case Based Reasoning)*. Πράγματι, ένας ειδικός 'κοστολόγος' στο λογισμικό προσπαθεί να σκεφθεί έργα που μοιάζουν με αυτό που θέλει να εκτιμήσει και στη συνέχεια βασίζει την πρόβλεψή του στο κόστος αυτών των έργων. Για παράδειγμα, εάν θέλει να προβλέψει το κόστος ενός συστήματος αυτόματης πλοήγησης δεν θα χρησιμοποιήσει το κόστος μιας εφαρμογής μηχανοργάνωσης. Η μέθοδος των αναλογιών βοηθά την αναγνώριση ομοιοτήτων μεταξύ έργων, ακόμη και όταν αυτές δεν είναι προφανείς, όπως στο παράδειγμα που μόλις αναφέραμε.

Ανεξάρτητα από τη μέθοδο κοστολόγησης που θα χρησιμοποιήσουμε, υπάρχει συνήθως ένα σημαντικό λάθος στις εκτιμήσεις μας. Η ακρίβεια των εκτιμήσεων μιας μεθόδου κοστολόγησης εξάγεται από την εφαρμογή της μεθόδου σε ένα σύνολο έργων γνωστού κόστους (πολλές φορές είναι τα ίδια τα ιστορικά έργα πάνω στα οποία ρυθμίστηκε) και βασίζεται σε ορισμένα κριτήρια που είναι διεθνώς αποδεκτά. Αυτά που χρησιμοποιούνται ευρύτερα είναι:

(α) το *Μέσο Σχετικό Λάθος Μεγέθους (Mean Magnitude Relative Error ή MMRE)*, που δίνεται από τον τύπο

$$MMRE = \frac{100}{n} \sum_{i=1}^n \frac{|E_i - \hat{E}_i|}{E_i},$$

όπου  $E_i$  το πραγματικό κόστος του έργου,  $\hat{E}_i$  η πρόβλεψη με βάση τη μέθοδο και  $n$  ο αριθμός των έργων γνωστού κόστους για τα οποία έγιναν προβλέψεις.

(β) η *Πρόβλεψη 25%* ή *Pred(25)*, δηλ. το ποσοστό των έργων επί του συνόλου των έργων γνωστού κόστους που εκτιμώνται από τη μέθοδο με λάθος μικρότερο από 25%.

Μια μέθοδος εκτίμησης κόστους θεωρείται καλή όταν επιτυγχάνει *MMRE* μικρότερο του 15%-25% και *Pred(25)* μεγαλύτερο του 75%.

Σε μια εκτεταμένη μελέτη ([3]) εφαρμόστηκε η μέθοδος των αναλογιών σε ένα σύνολο από ιστορικά δεδομένα κόστους και διαπιστώθηκε ότι αυτή επιτυγχάνει καλύτερα αποτελέσματα από αυτά της απλής παλινδρόμησης. Συγκεκριμένα με βάση τις αναλογίες επιτυγχάνεται *MMRE* μεταξύ 26% και 78 % (σε σχέση με 40% – 252 % της παλινδρόμησης) και *Pred(25)* μεταξύ 70% και 21 % (σε σχέση με 50% – 13 %). Η μελέτη επίσης διαπίστωσε ότι οι βέλτιστες προβλέψεις επιτυγχάνονται όταν λαμβάνονται υπ' όψιν δυο μόνον ανάλογα έργα. Τα αποτελέσματα δείχνουν ότι η απλή στατιστική ανάλυση δεν λύνει τα προβλήματα της κοστολόγησης, ενώ η μέθοδος των αναλογιών δείχνει ιδιαίτερα ενδιαφέροντα, αν και απαιτεί περισσότερη διερεύνηση.

Τα τυπικά βήματα για την προετοιμασία της εφαρμογής της μεθόδου των αναλογιών από έναν οργανισμό που παράγει ή αγοράζει λογισμικό είναι:

Επιλογή των χαρακτηριστικών με βάση τα οποία θα κρίνεται η αναλογία των έργων  
Συλλογή δεδομένων έργων για την ιστορικής βάσης κόστους – συνήθως γίνεται με ερωτηματολόγια, συνεντεύξεις, αναζήτηση μετρήσεων κλπ.

Υλοποίηση της βάσης

Δοκιμαστική εφαρμογή της μεθόδου και επιλογή των παραμέτρων που θα χρησιμοποιούνται (απόσταση, αριθμός ανάλογων έργων, τύπος εκτίμησης με βάση το κόστος των ανάλογων έργων)

Εκτίμηση κόστους έργων και καταγραφή της ακρίβειας της εκτίμησης

**Άσκηση Αυτοαξιολόγησης 5.5**

1. Ποια τα πλεονεκτήματα και ποια τα μειονεκτήματα της εκτίμησης με ειδικούς
2. Ποια τα πλεονεκτήματα και ποια τα μειονεκτήματα της εκτίμησης με παραμετρικά μοντέλα και με βάση τις αναλογίες

**Δραστηριότητα 5.6**

Απαντήστε στις παρακάτω ερωτήσεις πολλαπλών επιλογών

1. Το μοντέλο COCOMO81 βασίζει την εκτίμησή του:
  - A. Στο μέγεθος του έργου
  - B. Στο μέγεθος του έργου και τη χρονική του διάρκεια
  - Γ. Στο μέγεθος του έργου, στη χρονική του διάρκεια και σε ένα συνδυασμό παραγόντων
  - Δ. Σε συνδυασμό στοιχείων διαφορετικό από τους παραπάνω
2. Οι εκτιμήσεις του κόστους του λογισμικού χρειάζονται
  - A. Μόνο στην αρχή του έργου
  - B. Στην αρχή και κατά τη διάρκεια του έργου
  - Γ. Στην αρχή, κατά τη διάρκεια και στο τέλος του έργου
  - Δ. Κυρίως στην αρχή, αλλά και κατά τη διάρκεια και στο τέλος του έργου
3. Η παραγωγικότητα υπολογίζεται ως:
  - A. Φόρτος ανάπτυξης δια του χρόνου ανάπτυξης
  - B. Χρόνος ανάπτυξης δια του φόρτου ανάπτυξης
  - Γ. Μέγεθος δια του φόρτου ανάπτυξης
  - Δ. Φόρτος ανάπτυξης δια του μεγέθους



4. Το μοντέλο COCOMO81 υπολογίζει:
- A. Χρόνο ανάπτυξης και με βάση αυτόν το φόρτο ανάπτυξης
  - B. Φόρτος ανάπτυξης
  - Γ. Φόρτος ανάπτυξης και με βάση αυτήν το χρόνο ανάπτυξης
  - Δ. Τίποτα από τα παραπάνω
5. Ένα έργο μικρό, για το οποίο υπάρχει εμπειρία από άλλα παρόμοια έργα στο παρελθόν, με σχετικά χαλαρές απαιτήσεις ανήκει στην κατηγορία COCOMO81:
- A. Οργανική
  - B. Ενσωματωμένη
    - Γ. Ημι-αποσπασμένη
    - Δ. Σε καμία συγκεκριμένη κατηγορία
6. Η κατηγορία έργων COCOMO81 που απαιτεί το μεγαλύτερο φόρτο είναι η:
- A. Οργανική
  - B. Ενσωματωμένη
  - Γ. Ημι-αποσπασμένη
  - Δ. Δεν υπάρχει κατηγορία που να ξεχωρίζει ιδιαίτερα από τις άλλες
7. Στη μέθοδο των αναλογιών για ιστορικό σύνολο δεδομένων <20 έργων συνήθως προτιμάμε να έχουμε:
- A. Μία αναλογία
  - B. Μικρό αριθμό αναλογιών (1, 2 ή 3)
  - Γ. Πολλές αναλογίες (>3)
  - Δ. Τυχαίο αριθμό αναλογιών

## **Συμπεράσματα – Βιβλιογραφία**

---

Στο κεφάλαιο αυτό γνωρίσαμε έννοιες και τεχνικές της εκτίμησης κόστους. Ένα σημαντικό στοιχείο είναι ότι όλες οι τεχνικές βασίζονται σε ιστορικά έργα. Η γνώση των χαρακτηριστικών των ιστορικών έργων πάνω στα οποία ‘χτίστηκε’ μία τεχνική εκτίμησης κόστους είναι σημαντική γιατί επιτρέπει την αξιολόγηση της καταλληλότητας της τεχνικής για εκτίμηση ενός συγκριμένου έργου. Για παράδειγμα, σήμερα δεν χρησιμοποιείται το COCOMO81 επειδή είναι βασισμένο σε έργα προ του 1980, με παρωχημένη παραγωγικότητα σε σχέση με τα σημερινά. Επίσης δε θα χρησιμοποιούσαμε μία τεχνική που έχει ρυθμισθεί με βάση έργα πραγματικού χρόνου για να εκτιμήσουμε ένα έργο διαχείρισης εταιρικών δεδομένων (π.χ. μία εφαρμογή πελατολογίου) επειδή οι παράγοντες κόστους θα είχαν διαφορετική επίδραση στο κόστος και η παραγωγικότητα θα ήταν διαφορετική στις δύο περιπτώσεις.

Οι τεχνικές εκτίμησης κόστους μας προσφέρουν ένα χρήσιμο εργαλείο για τη διοίκηση έργου, το οποίο όμως πρέπει να χρησιμοποιούμε με προσοχή, μεθοδικότητα και με επίγνωση των ορίων του, δηλ. γνωρίζοντας ότι πάντα θα υπάρχει λάθος στην εκτίμησή μας, και μάλιστα το λάθος αυτό μπορεί να είναι σημαντικό.

[1] Software Engineering Economics, B. Boehm, Prentice Hall, 1981

[2] ‘Cost Models for Future Software Life Cycle Processes: COCOMO 2.0’,

Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, Ray Madachy, Richard Selby, Annals of Software Engineering, 1995. Υλικό για τα μοντέλα αυτής της οικογένειας μπορεί να βρεθεί στη σελίδα: <http://sunset.usc.edu/COCOMOII/cocomo.html>

[3] ‘Estimating software project effort using analogies’, J. Shepperd, C. Schofield, IEEE Transactions on Software Engineering (1997)

[4] ‘A systematic review of software cost estimation studies’, Shepperd, Jorgensen, Journal of Systems and Software, 2006

[5] Bozoki, G.J., 1991, Performance Simulation of SSM (Software Sizing Model), Proceedings, 13th Annual Conference, ISPA, 1991, Volume X, pp CM-14. Δείτε επίσης την παρακάτω διεύθυνση για μία σύντομη επεξήγηση της μεθόδου

<http://www.targetsoft-ware.com/expssm.htm>

[6] Software Engineering, Theory and Practice, S. L. Pfleeger, Prentice Hall, Βιβλίο γενικά για την Τεχνολογία Λογισμικού, αφιερώνει ένα κεφάλαιο στα θέματα του παρόντος Τόμου. Εκδίδεται και σε ελληνική μετάφραση από τον Κλειδάριθμο

## Απαντήσεις στις ασκήσεις αυτοαξιολόγησης

### Απάντηση Άσκησης Αυτοαξιολόγησης 1.1

	Σωστό	Λάθος
Η μελέτη και η κατασκευή κτιρίου / σπιτιού	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Η δημιουργία μιας κινηματογραφικής ταινίας	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Η ανάπτυξη μιας εφαρμογής μισθοδοσίας	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Η τεχνική υποστήριξη πελατών	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Η συντήρηση λογισμικού	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Τα βασικά κριτήρια για να απαντήσουμε στην ερώτηση αυτή είναι η μοναδικότητα του τελικού αποτελέσματος και το κατά πόσο είναι μια προσωρινή δραστηριότητα. Με βάση τα δύο αυτά κριτήρια α) η μελέτη και η κατασκευή κτιρίου / σπιτιού, β) η δημιουργία μιας κινηματογραφικής ταινίας και γ) η ανάπτυξη μιας εφαρμογής μισθοδοσίας είναι έργα, αφού πληρούν και τα δύο κριτήρια, ενώ η τεχνική υποστήριξη πελατών, καθώς και η συντήρηση λογισμικού δεν είναι, αφού δεν πληρούν το κριτήριο της προσωρινής δραστηριότητας.

Στο σημείο αυτό θα πρέπει να διακρίνουμε μια λεπτή διαφορά και να παρατηρήσουμε ότι, ενώ η συντήρηση λογισμικού θεωρείται υπηρεσία, αφού επαναλαμβάνεται με τον ίδιο τρόπο σε διαρκή βάση, πολλές φορές στην πράξη συναντούμε έργα συντήρησης λογισμικού. Ο λόγος είναι ότι συχνά οργανισμοί και επιχειρήσεις αποφασίζουν την ανάθεση των υπηρεσιών συντήρησης σε τρίτους (outsourcing) για ένα συγκεκριμένο χρονικό διάστημα, το οποίο τελικά λαμβάνει τη μορφή έργου συντήρησης για λόγους εύκολης διαχείρισης.

### Απάντηση Άσκησης Αυτοαξιολόγησης 1.2

Σύμφωνα με τη μελέτη των Kouzes και Posner (Kouzes, 1987) που βασίστηκε σε 900 ερωτηματολόγια που δόθηκαν σε εργαζόμενους εταιρειών τεχνολογίας και όπου υπήρχε η δυνατότητα πολλαπλών επιλογών το αποτέλεσμα ήταν το ακόλουθο:

Ικανότητα	Κατάταξη (ποσοστό)
-----------	-----------------------

Ηγετικές ικανότητες (έχει όραμα, εμπνέει τους συνεργάτες, κ.ά.)	68%
Ικανότητα δημιουργίας ομάδας (έχει ομαδικό πνεύμα, έχει κίνητρα, κ.ά.)	72%
Ικανότητα επικοινωνίας (ξέρει να πείθει, είναι καλός ακροατής, έχει ικανότητα έκφρασης)	84%
Οργανωτικές ικανότητες (γνωρίζει να θέτει στόχους, να δίνει προτεραιότητες, να αναλύει προβλήματα, κ.ά.)	75%
Χαρακτήρας (είναι ευέλικτος, δημιουργικός, επίμονος, κ.ά.)	59%
Γνώση της τεχνολογίας (γνωρίζει να αναπτύσσει πληροφοριακά συστήματα, γνωρίζει να διοικεί έργα, έχει εμπειρία, κ.ά.)	46%

Ίσως προκαλεί έκπληξη το γεγονός ότι η γνώση της τεχνολογίας παρουσιάζεται ως η τελευταία επιθυμητή ικανότητα. Το γεγονός αυτό σημαίνει για παράδειγμα ότι ο διαχειριστής έργων λογισμικού δεν χρειάζεται να είναι ειδικός (expert) στην ανάπτυξη πληροφοριακών συστημάτων.

## Απάντηση Άσκησης Αυτοαξιολόγησης 2.1

---

Παραδείγματα οργανισμών ανάπτυξης λογισμικού μπορεί να είναι τα ακόλουθα:

- Το τμήμα μηχανοργάνωσης μίας τράπεζας. Πρόκειται για οργανισμό που μπορεί να απασχολεί ακόμη και άνω των εκατό στελεχών, ανάλογα με το μέγεθος της τράπεζας. Αναπτύσσει και συντηρεί τραπεζικές εφαρμογές λογισμικού με αντικείμενο κυρίως τη διαχείριση δεδομένων (των καταθετών, των λογαριασμών τους, των χαρτοφυλακίων τους, των δανείων τους, κλπ)
- Το τμήμα μηχανοργάνωσης μίας νομαρχίας. Πρόκειται για οργανισμό που μπορεί να απασχολεί λίγες δεκάδες στελεχών, πάντα ανάλογα με το μέγεθος του νομού. Αναπτύσσει και συντηρεί εφαρμογές λογισμικού ηλεκτρονικής διακυβέρνησης με

αντικείμενο και πάλι κυρίως τη διαχείριση δεδομένων (των πολιτών της νομαρχίας, των έργων της, των υποδομών της, ενημέρωσης του κοινού κλπ)

- Μία εταιρεία ανάπτυξης εμπορικών εφαρμογών. Μπορεί να είναι πολύ μικρή (πέντε ατόμων), μέτριου μεγέθους (αρκετές δεκάδες προγραμματιστών) ή και μεγάλη (μερικές εκατοντάδες). Οι μικρού και μεσαίου μεγέθους εταιρείες αναπτύσσουν κάθετες εμπορικές εφαρμογές (δηλ. εφαρμογές που απευθύνονται σε συγκεκριμένες λειτουργίες, όπως π.χ. διαχείριση ΚΤΕΟ, διαχείριση της γραμματείας πανεπιστημιακού τμήματος), ενώ οι μεγάλες εταιρείες αναπτύσσουν οριζόντιου τύπου εφαρμογές (δηλ. εφαρμογές ευρείας αποδοχής, όπως λογιστική, διαχείριση αποθήκης, διαχείριση παραγγελιών κλπ). Και πάλι το λογισμικό που αναπτύσσεται είναι κυρίως για διαχείριση εταιρικών / οργανωσιακών δεδομένων.
- Εταιρεία ανάπτυξης τηλεπικοινωνιακού λογισμικού. Πρόκειται για οργανισμό που συνήθως είναι τμήμα μεγαλύτερης, πολυεθνικής εταιρείας που παράγει παράλληλα και τηλεπικοινωνιακό υλικό (π.χ. υλικό για ψηφιακά τηλεφωνικά κέντρα). Μπορεί να είναι μετρίου ή και μεγάλου μεγέθους και αναπτύσσει λογισμικό τύπου πραγματικού χρόνου (real time).

Αν δεν μπορέσατε να απαντήσετε σε αυτή την άσκηση μην απογοητεύεστε! Η άσκηση απαιτεί κάποια γνώση σχετικά με τον επιχειρηματικό κόσμο της πληροφορικής, δηλ. τις εταιρείες και οργανισμούς που παράγουν ή χρησιμοποιούν συστήματα πληροφορικής. Μπορείτε να παρακολουθείτε τις εξελίξεις στο χώρο αυτό από τα επιστημονικά περιοδικά της πληροφορικής, τον ημερήσιο και εβδομαδιαίο τύπο και από τα δελτία των επιστημονικών ενώσεων της πληροφορικής.

## **Απάντηση Άσκησης Αυτοαξιολόγησης 2.2**

---

Για την απάντηση βλ. παράγραφο 2.2.

A) Οργάνωση έργου, Οργάνωση Λειτουργιών, Οργάνωση Μητρώου

B) Μέγεθος του προς ανάπτυξη λογισμικού, Αριθμός των έργων προς υλοποίηση, Έκταση της ανάπτυξης, Περιβάλλον ανάπτυξης, Φυσικοί περιορισμοί, Οργανωτική κουλτούρα

### Απάντηση Άσκησης Αυτοαξιολόγησης 2.3

---

Η σταδιακή εισαγωγή της δομής μητρώου σε έναν οργανισμό θα μπορούσε να γίνει με τους εξής τρόπους:

- Με την εκπαίδευση των διαχειριστών του οργανισμού στη δομή μητρώου από ειδικούς συμβούλους
- Με την εφαρμογή της πιλοτικά σε μικρό αριθμό έργων, τα οποία επίσης θα έχουν σχετικά μικρό μέγεθος. Χρειάζεται προσοχή στο κόστος ενός παρόμοιου εγχειρήματος (βλ. μειονεκτήματα δομής μητρώου).
- Στην περίπτωση μίας μεγάλης επιχείρησης θα μπορούσε να γίνει με την εφαρμογή της στα έργα ενός μόνον τμήματος της επιχείρησης, π.χ. στα έργα της έρευνας αγοράς (marketing)
- Μέρος μόνον του προσωπικού του οργανισμού (το πιο ώριμο και δεκτικό σε νέες καταστάσεις) θα μπορούσε να εισάγεται σταδιακά στη δομή αυτή, ώστε το προσωπικό να απορροφήσει σταδιακά τα προβλήματα προσαρμογής

### Απάντηση Άσκησης Αυτοαξιολόγησης 2.4

---

Η οργάνωση ομάδας (βλ. παρ. 2.3) επιδιώκει τη δημιουργία ομάδων που η σύνθεσή τους θα διασφαλίζει:

- Την επιτυχή ολοκλήρωση της εργασίας για την οποία δημιουργήθηκε η ομάδα. Αυτό απαιτεί την κοινή κατανόηση από όλα τα μέλη των στόχων που θα τεθούν από τη διαχείριση.
- Την επίτευξη αποτελέσματος ανώτερου από αυτό που θα επιτύγχαναν τα μέλη της ομάδας, εργαζόμενα ξεχωριστά

- Την ικανοποίηση των προσδοκιών των μελών της ομάδας

### Απάντηση Άσκησης Αυτοαξιολόγησης 3.1

XYZ-ΜΑΠ-0207-V1.00	Η τελική έκδοση της μηνιαίας αναφοράς προόδου για το μήνα Φεβρουάριο του 2007 και για το έργο XYZ. Τελική είναι η έκδοση που έγινε αποδεκτή από τον πελάτη.
XYZ-ΑΝΑ-ΔΚΛ-V1.00	Η τελική έκδοση των διαγραμμάτων κλάσεων του συστήματος.
XYZ-ΣΧΕ-ΔΚΛ-V1.00	Τα διαγραμμάτων κλάσεων του συστήματος, αλλά για τη φάση του σχεδιασμού. Το παραδοτέο αυτό αποτελεί επέκταση του εγγράφου XYZ-ΑΝΑ-ΔΚΛ-V1.00.

### Απάντηση Άσκησης Αυτοαξιολόγησης 3.2

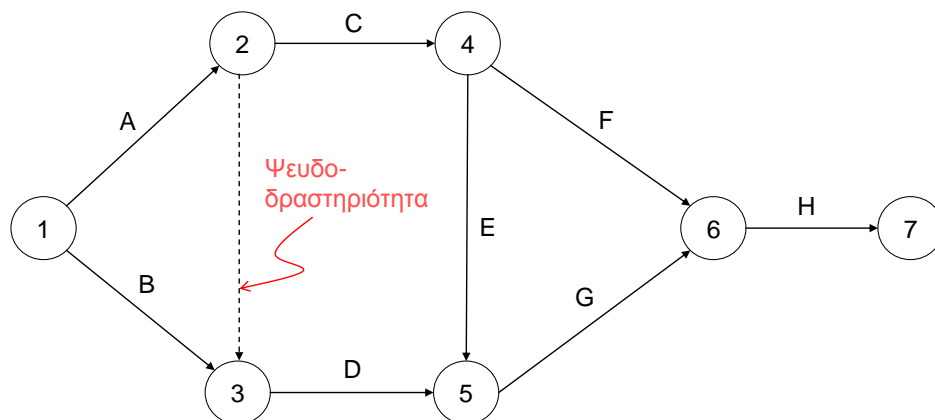
Είδος αλλαγής	«Ανάπτυξη επιχειρηματικής εφαρμογής»
Είναι τύπου 1, γιατί το σφάλμα αυτό μας επιτρέπει να συνεχίσουμε να χρησιμοποιούμε το σύστημα, έχοντας όμως μειωμένη λειτουργικότητα.	Τα δεδομένα παρουσιάζονται ομαδοποιημένα ανά τύπο. Πατώντας το σύμβολο (+) δίπλα στο όνομα της ομάδας το σύστημα παρουσιάζει όλα τα ομαδοποιημένα δεδομένα αναλυτικά. Η αντίστροφη λειτουργία, δηλαδή η σύμπτυξη των δεδομένων της ομάδας δεν λειτουργεί κανονικά.
Είναι τύπου 0, γιατί αποτελεί σοβαρό σφάλμα, το οποίο δεν μας επιτρέπει να συνεχίσουμε να χρησιμοποιούμε το σύστημα.	Απώλεια δεδομένων.
Είναι τύπου 3, γιατί αποτελεί νέα απαίτηση για το σύστημα	Μεταφορά της εφαρμογής στη νέα έκδοση του φυλλομετρητή.
Αποτελεί βελτίωση της λειτουργικότητας του συστήματος και	Χρήση χρωματικού κώδικα για την παρουσίαση «νέων» και «παλαιών



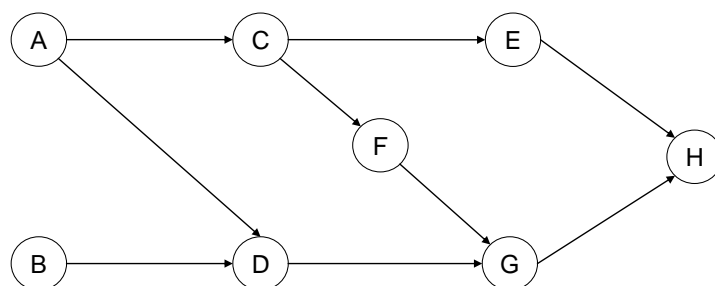
Είδος αλλαγής	«Ανάπτυξη επιχειρηματικής εφαρμογής»
για το λόγο αυτό η αλλαγή αυτή είναι τύπου 2.	δεδομένων».
Αποτελεί βελτίωση της λειτουργικότητας του συστήματος και για το λόγο αυτό η αλλαγή αυτή είναι τύπου 2.	Βελτίωση του σχήματος αποθήκευσης δεδομένων με σκοπό τη γρήγορη εύρεση του ιστορικού πελατών.

### Απάντηση Άσκησης Αυτοαξιολόγησης 4.1

Το δίκτυο AOA για το υπό εξέταση έργο είναι το ακόλουθο:



Αντίστοιχα, το δίκτυο AON είναι το ακόλουθο:



## Απάντηση Άσκησης Αυτοαξιολόγησης 4.2

Για την εύρεση της κρίσιμης διαδρομής θα πρέπει να υπολογίσουμε τους χρόνους ES, EF, LS και LF. Στη συνέχεια θα πρέπει να υπολογίσουμε το χρονικό περιθώριο της κάθε δραστηριότητας. Ο υπολογισμός των χρόνων δίνεται στον παρακάτω πίνακα:

Δραστηριότητα	Προηγούμενη Δραστηριότητα	Διάρκεια	ES	EF	LS	LF	SL
A	-	2	0	2	0	2	0
B	A	3	2	5	2	5	0
C	A	2	2	4	3	5	1
D	B,C	4	5	9	5	9	0
E	D	4	9	13	9	13	0

Η δραστηριότητα A είναι αρχική, συνεπώς  $ES(A) = 0$ . Η διάρκεια της δραστηριότητας A είναι 2 εβδομάδες και συνεπώς:

$$EF(A) = ES(A) + L(A) = 0 + 2 = 2.$$

Συνεχίζουμε με τη δραστηριότητα B. Η δραστηριότητα B έχει ως μοναδική προηγούμενη δραστηριότητα τη δραστηριότητα A. Συνεπώς,  $ES(B) = EF(A) = 2$  και  $EF(B) = 2 + 3 = 5$ .

Αντίστοιχα, η δραστηριότητα C έχει ως μοναδική προηγούμενη δραστηριότητα τη δραστηριότητα A. Συνεπώς,  $ES(C) = EF(A) = 2$  και  $EF(C) = 2 + 2 = 4$ .

Η δραστηριότητα D έχει δύο προηγούμενες δραστηριότητες, την B και C. Συνεπώς,

$$ES(D) = \max \{ EF(B), EF(C) \} = \max \{ 5, 4 \} = 5$$

$$EF(D) = ES(D) + L(D) = 5 + 4 = 9.$$

Συνεχίζουμε με τη δραστηριότητα E. Η δραστηριότητα E έχει ως μοναδική προηγούμενη δραστηριότητα τη δραστηριότητα D. Συνεπώς,  $ES(E) = EF(D) = 9$  και  $EF(E) = 9 + 4 = 13$ .

Αφού έχουμε υπολογίσει τους χρόνους ES και EF μπορούμε να ξεκινήσουμε τον υπολογισμό των χρόνων LS και LF. Για τον υπολογισμό των χρόνων LS και LF ξεκινάμε να εργαζόμαστε από το τέλος του δικτύου (έργου) προς την αρχή.

Στο παράδειγμά μας η δραστηριότητα E είναι τελική και συνεπώς:

$$LF(E) = EF(E) = 13$$

$$LS(E) = LF(E) - L(E) = 13 - 4 = 9.$$

Συνεχίζουμε με τον κόμβο D εφαρμόζοντας το σχετικό βήμα του αλγορίθμου:

$$LF(D) = \min\{LS(E)\} = 9 \text{ (υπάρχει μόνο μια επόμενη)}$$

$$LS(D) = LF(D) - L(D) = 9 - 4 = 5$$

Με τον ίδιο τρόπο υπολογίζουμε το LS, LF για τις δραστηριότητες B και C.

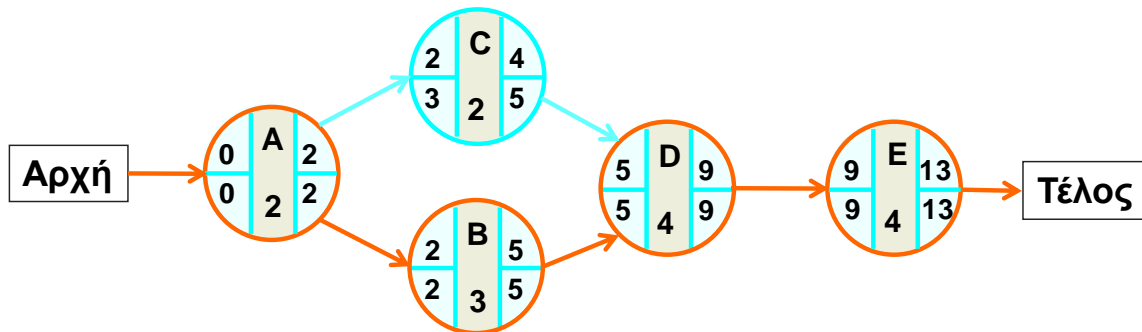
Η δραστηριότητα A έχει δύο επόμενες την B και C. Στην περίπτωση αυτή έχουμε:

$$LF(A) = \min\{LS(B), LS(C)\} = \min\{2, 3\} = 2$$

$$LS(A) = LF(A) - L(A) = 2 - 2 = 0$$

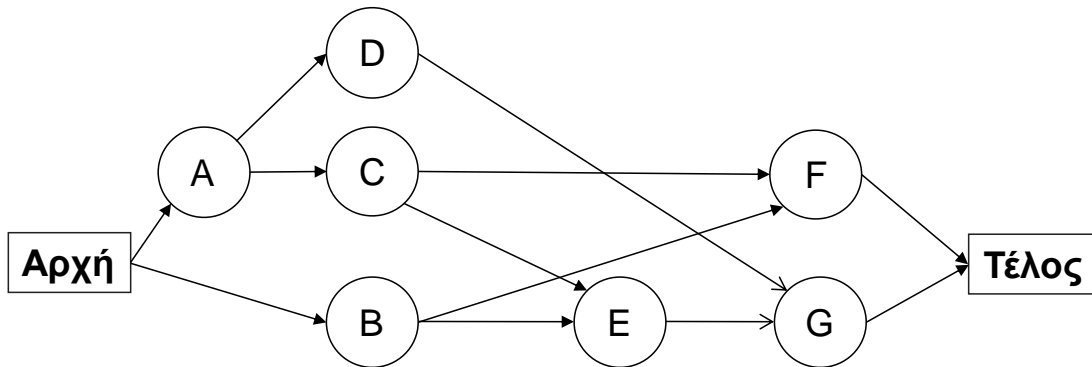
Υπολογίζουμε το SL, ως τη διαφορά LS-ES ή LF-EF για κάθε δραστηριότητα. Οι δραστηριότητες που ανήκουν στην κρίσιμη διαδρομή είναι αυτές που έχουν SL=0. Συνεπώς, η κρίσιμη διαδρομή είναι η A-B-D-E.

Η γραφική απεικόνιση του δικτύου παρουσιάζεται στο επόμενο σχήμα.



### Απάντηση Άσκησης Αυτοαξιολόγησης 4.3

Το ζητούμενο διάγραμμα AON είναι το ακόλουθο:



Υπολογίζουμε αρχικά το TE με τη χρήση του τύπου  $TE = \frac{a + 4m + b}{6}$  και στη

συνέχεια υπολογίζουμε τους χρόνους ES, EF, LS, LF και SL.

$a$	$m$	$b$	TE	ES	EF	LS	LF	SL	$\sigma^2$
-----	-----	-----	----	----	----	----	----	----	------------

A	-	1	4	7	4	0	4	0	4	0	1
B	-	2	2	2	2	0	2	7	9	7	0
C	A	2	5	8	5	4	9	4	9	0	1
D	A	3	4	5	4	4	8	11	15	7	0,11
E	C, B	4	6	8	6	9	15	9	15	0	0,44
F	C, B	0	0	6	1	9	10	20	21	11	1
G	D, E	3	6	9	6	15	21	15	21	0	1

Με βάση τον παραπάνω πίνακα βρίσκουμε ότι το κρίσιμο μονοπάτι είναι το A-C-E-G. Ο χρόνος ολοκλήρωσης του έργου είναι  $TE_{\text{έργου}}=21$  ημέρες.

Η πιθανότητα ολοκλήρωσης του έργου σε 23 ημέρες είναι:

$$Z = \frac{\tau - TE_{\text{έργου}}}{\sqrt{\sigma_{\text{έργου}}^2}} = \frac{23 - 21}{\sqrt{3,44}} = 1,078 \quad \text{το οποίο, σύμφωνα με τον πίνακα του}$$

παραρτήματος, αντιστοιχεί σε πιθανότητα 85,9%

Η πιθανότητα ολοκλήρωσης του έργου 99% αντιστοιχεί, σύμφωνα με τον πίνακα του παραρτήματος, σε  $Z=2,33$ . άρα:

$$2,33 = (\tau - 21)/1,855 \quad \text{και συνεπώς} \quad \tau = 25,32 \text{ ημέρες}$$

### Απάντηση Άσκησης Αυτοαξιολόγησης 5.1

Το κόστος επηρεάζεται σε μεγάλο βαθμό από τον τύπο του λογισμικού, το μέγεθος του λογισμικού, την ποιότητα (γνώση και εμπειρία) του προσωπικού ανάπτυξης, τα εργαλεία ανάπτυξης και τη διεργασία ανάπτυξης (βλ. παρ. 2.1.3).

### Απάντηση Άσκησης Αυτοαξιολόγησης 5.2

*Διπλωματική εργασία φοιτητή:* Οργανική κατηγορία. Πρόκειται για μία πολύ χαρακτηριστική περίπτωση έργου που ανήκει στην κατηγορία αυτή. Η εργασία έχει χαλαρές απαιτήσεις, οι οποίες είναι υπό συζήτηση ανάλογα με την πορεία της διπλωματικής και μπορούν να αλλάξουν μετά από συμφωνία με το επιβλέποντα καθηγητή. Η μη πλήρης ικανοποίηση των απαιτήσεων δε συνεπάγεται ποινές, απώλειες κλπ, αφού η διπλωματική εργασία συνήθως γίνεται για ερευνητικούς ή εκπαιδευτικούς σκοπούς. Επίσης η έκτασή της εργασίας είναι περιορισμένη σε σχέση με ένα πραγματικό έργο λογισμικού.

*Σύστημα συναλλαγών χρηματιστηρίου:* Ενσωματωμένη κατηγορία. Το σύστημα αυτό έχει αυστηρές απαιτήσεις ως προς την ακρίβεια των οικονομικών συναλλαγών (τυχόν λάθη θα αποτελούσαν πλήγμα στο κύρος του θεσμού του χρηματιστηρίου). Έχει αυξημένες απαιτήσεις ως προς την ταχύτητα εκτέλεσης των συναλλαγών αφού αυτές πρέπει να εκτελούνται σε πραγματικό χρόνο. Επίσης πρόκειται για ένα σύστημα μεγάλου μεγέθους (μεγάλος αριθμός χρηστών, εγκαταστάσεις του συστήματος σε πολλά διαφορετικά σημεία, πολλές και διαφορετικές λειτουργίες).

*Σύστημα υπολογισμού βαθμών φοιτητών του ΕΑΠ:* Ημι-αποσπασμένη κατηγορία. Το σύστημα έχει αυξημένες απαιτήσεις ακρίβειας για τους υπολογισμούς του, αφού τυχόν αστοχία του συστήματος (π.χ. έκδοση μίας λάθος βαθμολογίας σε μία ενότητα) θα έχει σαν αποτέλεσμα τη δυσφορία των φοιτητών. Ωστόσο το πρόβλημα μπορεί να αντιμετωπισθεί χειρωνακτικά χωρίς άλλες συνέπειες. Επίσης άλλες απαιτήσεις (π.χ. διασύνδεση χρήστη) είναι σχετικά χαλαρές. Το μέγεθος του συστήματος είναι μέτριο (σαφώς μικρότερο από το σύστημα του χρηματιστηρίου), ενώ δεν υπάρχουν απαιτήσεις πραγματικού χρόνου.

*Υποσύστημα λογιστικής ενός συστήματος ERP:* Ενσωματωμένη κατηγορία. Ίσως στην απάντησή σας να προτιμήσατε την Ημι-αποσπασμένη κατηγορία. Πρόκειται όμως για σύστημα το οποίο έχει χαρακτηριστικά παρόμοια με του συστήματος του χρηματιστηρίου, αν και δεν έχει απαιτήσεις πραγματικού χρόνου. Επειδή όμως προορίζεται για χρήση από πολλές επιχειρήσεις, θα πρέπει να ικανοποιεί σε μεγάλο βαθμό τις απαιτήσεις του, ώστε το ERP σύστημα στο οποίο ενσωματώνεται να έχει εμπορική επιτυχία.

### **Απάντηση Άσκησης Αυτοαξιολόγησης 5.3**

---

Η πρώτη περίπτωση κώδικα δεν θα επηρεάσει καθόλου την εκτίμηση σύμφωνα με το μοντέλο COCOMO81. Ο λόγος είναι ότι το μέγεθος του κώδικα προσμετράται σε KDSI, δηλ. χιλιάδες γραμμών παραδοτέου κώδικα (D = Delivered) (δείτε τον ορισμό στην παρ. 5.1.1).

Η δεύτερη περίπτωση κώδικα αντίθετα, εφόσον παραδίδεται στον πελάτη, θα πρέπει να προστεθεί στον υπόλοιπο κώδικα. Επομένως το μέγεθος είναι 11 KDSI. Να εξετάσετε μόνοι σας πως αυξάνεται ο φόρτος λόγω αυτής της κατά 10% αύξησης στο μέγεθος, εφαρμόζοντας και τις τρεις εξισώσεις του πίνακα 2. Παρατηρήσετε ότι η αύξηση του φόρτου σε σχέση με το μέγεθος δεν είναι γραμμική: ο φόρτος θα αυξηθεί περισσότερο από 10%.

### Απάντηση Άσκησης Αυτοαξιολόγησης 5.4

---

Αντικαθιστώντας τις τιμές για κάθε έργο στην εξίσωση του μοντέλου έχουμε:

Έργο Α: Μέγεθος = 50 Βαθμοί Λειτουργίας, μηδενική χρήση T, φόρτος = 573 ώρες εργασίας

Έργο Β: Μέγεθος = 50 Βαθμοί Λειτουργίας, σημαντική χρήση T, φόρτος = 208 ώρες εργασίας

Έργο Γ: Μέγεθος = 1000 Βαθμοί Λειτουργίας, μηδενική χρήση T, φόρτος = 5932 ώρες

Παρατηρούμε ότι:

(α) Στο συγκεκριμένο μοντέλο ο φόρτος αυξάνεται με μικρότερο ρυθμό σε σχέση με την αύξηση του μεγέθους (ο φόρτος των έργων Α και Γ δείχνει ότι εικοσαπλάσιο μέγεθος απαιτεί μόλις δεκαπλάσιο φόρτο). Συνήθως συμβαίνει το αντίθετο, αύξηση του μεγέθους συνοδεύεται από εντονότερη αύξηση του φόρτου (οικονομία κλίμακας)

(β) Η χρήση του εργαλείου T μειώνει σημαντικά το φόρτο (το έργο Β λόγω της σημαντικής χρήσης του T απαιτεί για το ίδιο μέγεθος λογισμικού υποδιπλάσιο φόρτο). Αυτό το γεγονός ίσως δε φαίνεται με την πρώτη ματιά από τον πίνακα 8 λόγω της χρήσης λογαρίθμων στο μοντέλο.

Τέλος μία εξήγηση για το μειωμένο όφελος από τη χρήση του T όταν η χρήση του δεν είναι σημαντική είναι το ότι για τη χρήση ενός εργαλείου απαιτείται κάποιος χρόνος και φόρτος για εξοικείωση, τυχόν μετατροπές, ενσωμάτωσή του στο έργο κλπ. Αυτό αναιρεί σε σημαντικό βαθμό το όποιο όφελος από περιορισμένη χρήση του εργαλείου, αλλά αποδίδει για υψηλές τιμές χρήσης του. Οι τιμές του πίνακα 11 δίνουν μία αναπαράσταση της λεγόμενης *καμπύλης εκμάθησης* (χαρακτηρίζεται από τις χαμηλές αρχικές τιμές αποδοτικότητας).

## Απάντηση Άσκησης Αυτοαξιολόγησης 5.5

---

Εκτίμηση με ειδικούς:

**Πλεονεκτήματα:** Καλές προβλέψεις ανάλογα με την εμπειρία των ειδικών και τον όγκο των πληροφοριών που έχουν στη διάθεσή τους

**Μειονεκτήματα:** Δεν είναι εύκολο να βρεθούν πάντα οι ειδικοί, οι ειδικοί μπορεί να μην είναι και τόσο 'ειδικοί', τελείως υποκειμενικές εκτιμήσεις, ελλειψη συστηματικής αντιμετώπισης της εκτίμησης, δυσκολία μεταφοράς των εμπειριών από επιτυχημένες ή αποτυχημένες εκτιμήσεις στο μέλλον.

Εκτίμηση με παραμετρικά μοντέλα και αναλογίες:

**Πλεονεκτήματα:** Συστηματική αντιμετώπιση της εκτίμησης, Αξιοποίηση στοιχείων ιστορικών έργων, Σχετικά αντικειμενικές εκτιμήσεις, Ευκολία μεταφοράς των εμπειριών από επιτυχημένες ή αποτυχημένες εκτιμήσεις στο μέλλον (π.χ. είναι δυνατό να εντοπισθεί το πρόβλημα που προκάλεσε μία λάθος εκτίμηση)

**Μειονεκτήματα:** Το νέο έργο πρέπει να μοιάζει σε αρκετά σημεία σε κάποια τουλάχιστον από τα ιστορικά έργα, Υποκειμενική κρίση για την επιλογή τιμών για παράγοντες κόστους, Ανάγκη χρήσης εργαλείων, Αδιαφάνεια στον τρόπο εκτίμησης (ισχύει για ορισμένα παραμετρικά μοντέλα όπως το COCOMO 2.0).

## Ενδεικτικές απαντήσεις στις δραστηριότητες

### Ενδεικτική Απάντηση Δραστηριότητας 1.1

---

Για να δώσουμε την απάντηση θα πρέπει να σκεφτούμε αναλυτικά διάφορους παράγοντες που επηρεάζουν τη διαχείριση ενός έργου. Ο παρακάτω πίνακας παρουσιάζει τον τρόπο με τον οποίο διαφοροποιούνται τα έργα ανάπτυξης λογισμικού από άλλα έργα.

	Έργο ανάπτυξης λογισμικού οικονομικής διαχείρισης	Έργο άλλου τύπου (π.χ. κτίσιμο γέφυρας)
Δομή έργου	Σχετίζεται άμεσα με επιχειρηματικές διαδικασίες και σίγουρα με άλλα	Είναι σχετικά ανεξάρτητο από επιχειρηματικές διαδικασίες. Πιθανόν να

	συστήματα λογισμικού της επιχείρησης (μισθοδοσία, διαχείριση πελατών, κ.λπ.).	συνδέεται με άλλα έργα κατασκευής οδών πρόσβασης.
Αντικείμενο του έργου	Στις περισσότερες περιπτώσεις τέτοιων έργων δεν είναι καλά ορισμένο και υπάρχει μια γενική ιδέα και ασαφείς απαιτήσεις. Είναι βέβαιο ότι θα υπάρχουν πολλές αλλαγές.	Καλά ορισμένο. Το σχέδιο του αναμενόμενου αποτελέσματος (π.χ. η γέφυρα) χαρακτηρίζεται από ικανοποιητικό επίπεδο λεπτομέρειας. Συνήθως υπάρχει προμελέτη και μελέτη εφαρμογής.
Διαχείριση αλλαγών	Στα περισσότερα έργα λογισμικού υπάρχει διαδικασία διαχείρισης αλλαγών, αλλά είναι δύσκολο να συσχετίσουμε τις αλλαγές με το τελικό προϊόν.	Είναι καλά ορισμένη.
Επένδρωση του έργου	Υπάρχει μεγάλος αριθμός διαφορετικών ειδικοτήτων που έχει ως αποτέλεσμα το προσωπικό του έργου να είναι μερικής απασχόλησης.	Οι εργαζόμενοι στο έργο είναι πλήρους απασχόλησης. Χρησιμοποιούνται εκτενώς υπεργολάβοι.
Διαχείριση κινδύνου	Οι κίνδυνοι δεν προσδιορίζονται εύκολα και συνήθως έχουν μεγάλο αντίκτυπο μια και συνήθως εντοπίζονται αργά.	Οι κίνδυνοι προσδιορίζονται σχετικά εύκολα και η διαχείρισή τους είναι τυποποιημένη. Υπάρχει σχετική νομοθεσία.
Διαχείριση ποιότητας	Τεκμηριωμένη αλλά δύσκολη στην εφαρμογή της.	Είναι πιο εύκολη η εφαρμογή τυποποιημένων ελέγχων.



## Ενδεικτική Απάντηση Δραστηριότητας 1.2

---

Τα παραδείγματα αποτυχημένων έργων που μπορεί να βρει κανείς με μια σύντομη έρευνα στο διαδίκτυο είναι πολλά και ποικίλα.

Για παράδειγμα στο δικτυακό τόπο <http://www.spectrum.ieee.org/sep05/1685> και από το άρθρο με τίτλο “Why software fails” του Robert Charette (Charette, 2005), μπορεί να διαβάσει για τις πλέον παταγώδεις αποτυχίες έργων ανάπτυξης λογισμικού (Hall of Shame) που απέφεραν ζημιές εκατομμυρίων ευρώ.

Αντίστοιχα, πολύ γνωστή είναι η αναφορά “CHAOS Report” από το Standish Group (Standish Group, 1995) το οποίο αναλύει τις αποτυχίες έργων λογισμικού στην Ηνωμένες Πολιτείες Αμερικής. Μια έκδοση αυτής της αναφοράς μπορεί να βρει κανείς στο δικτυακό τόπο (<http://www.standishgroup.com/public.php>).

## Ενδεικτική Απάντηση Δραστηριότητας 2.1

---

Αν δεν μπορείτε να απαντήσετε στην άσκηση αυτή, πρέπει να επαναλάβετε τη μελέτη των παραγράφων 2.2.1, 2.2.2 και 2.2.3 που περιγράφουν πλεονεκτήματα και μειονεκτήματα των τριών οργανωτικών δομών που εξετάζει το κεφάλαιο αυτό.

- Ανεξαρτησία στη λήψη απόφασης: όσο λιγότεροι εμπλέκονται στη λήψη μίας απόφασης, τόσο ταχύτερη και σαφέστερη θα είναι αυτή (πρέπει όμως να διασφαλισθεί ότι ο διαχειριστής ή οι διαχειριστές έχουν αποδεδειγμένα τις γνώσεις, την εμπειρία και την ικανότητα να πάρουν τη σωστή απόφαση)
- Ταχύτητα στη λήψη απόφασης: όσο πιο γρήγορα ληφθεί μία ορθή απόφαση τόσο μικρότερη καθυστέρηση θα έχουμε στην εκτέλεση του έργου. Επίσης τόσο λιγότεροι πόροι θα παραμένουν ανενεργοί. Σε ορισμένες περιπτώσεις η λήψη άμεσης απόφασης είναι απαραίτητη για τη συνέχιση του έργου (π.χ. στην περίπτωση εμφάνισης ενός κρίσιμου κινδύνου, βλ. επόμενο κεφάλαιο για την αντιμετώπιση του ρίσκου)
- Ελαχιστοποίηση της ανάγκης επικοινωνίας και επομένως του αριθμού των διασυνδέσεων μεταξύ των συμμετεχόντων σε ένα έργο ή σε μία ομάδα έργων. Με

τον τρόπο αυτό επιταχύνεται η ανάπτυξη του έργου, αυξάνεται ο διαθέσιμος χρόνος για δημιουργική εργασία στο έργο και αποφεύγονται λάθη που μπορεί να οφείλονται σε εσφαλμένη επικοινωνία.

- Σαφής ανάθεση ευθυνών. Είναι σημαντικό να υπάρχει ένας και μοναδικός υπεύθυνος για την ορθή εκτέλεση κάθε λειτουργίας και λήψη απόφασης. Σε έναν αποδοτικό, μη γραφειοκρατικό οργανισμό, αυτό βοηθάει στον εντοπισμό του αιτίου σε περίπτωση πλημμελούς εκτέλεσης του έργου, με στόχο όχι την τιμωρία του υπευθύνου, αλλά τη βελτίωση της όλης διεργασίας και την αποφυγή στο μέλλον παρομοίων λαθών.
- Ανάθεση της σωστής εργασίας στο σωστό άνθρωπο. Αυτό σημαίνει π.χ. ότι ένας καλός αναλυτής θα πρέπει να ασχολείται με την ανάλυση ενός συστήματος, αντίθετα δεν πρέπει να 'υποαπασχολείται' σαν προγραμματιστής. Αντίστοιχα, δεν πρέπει να αναθέτουμε κρίσιμες εργασίες κάποιου είδους σε άτομα που δε διαθέτουν τη σωστή εμπειρία και γνώση. Σημαντικό θέμα είναι η εκπαίδευση του προσωπικού, ώστε να επιτευχθούν τα ανωτέρω.
- Επιδίωξη της δημιουργίας προσωπικών κινήτρων για τους συμμετέχοντες στα έργα. Θα πρέπει οι επιτυχίες να συσχετίζονται με τα άτομα που είχαν συνεισφορά σε αυτές, ενώ καλό είναι το προσωπικό των έργων να αυτο-αναγνωρίζεται σε αυτά (δηλ. θα πρέπει ο X να είναι υπερήφανος για τη συμμετοχή του στο έργο Y).
- Ενημέρωση της ανώτερης διοίκησης σχετικά με την πορεία των έργων. Είναι απαραίτητη για το συντονισμό των έργων και την επέμβαση σε περίπτωση προβλημάτων που δε μπορούν να αντιμετωπισθούν σε χαμηλότερο επίπεδο.
- Βέλτιστη χρήση των πόρων του οργανισμού. Δεν είναι παραγωγικό ένας προγραμματιστής ή ένα ακριβό υπολογιστικό σύστημα να μην απασχολείται πλήρως σε κάποια δραστηριότητα. Αυτό σημαίνει ότι συχνά χρειάζεται μετακίνηση του προσωπικού του οργανισμού από έργο σε έργο.

- Είναι σημαντική η ανάπτυξη προτύπων μέσα από την εμπειρία που αποκτάται από έργα που εκτελεί ο οργανισμός και που θα εφαρμόσει στα μελλοντικά έργα. Η χρήση προτύπων βοηθά στην αποφυγή λαθών και στην επανάληψη καλών πρακτικών.
- Ο οργανισμός πρέπει να είναι σε θέση να ενημερώνει το προσωπικό του σχετικά με τις εξελίξεις σε νέες τεχνολογίες.
- Ο οργανισμός πρέπει να διαθέτει τόσο εξειδικευμένο προσωπικό όσο και προσωπικό με γενικές γνώσεις. Πρέπει να είναι δυνατή η αντιμετώπιση εξειδικευμένων προβλημάτων, αλλά πρέπει να υπάρχει και σημαντική ευελιξία, ώστε να είναι δυνατή η κάλυψη κενών στις ομάδες των έργων όταν αυτά δημιουργούνται.
- Άμεσος στόχος της διοίκησης του οργανισμού πρέπει να είναι η εκτέλεση των έργων μέσα στο προϋπολογισμένο κόστος, στο προδιαγραφμένο χρονοδιάγραμμα στα όρια ποιότητας που έχουν τεθεί.
- Ο οργανισμός πρέπει να δίνει προσοχή στις σχέσεις μεταξύ υπαλλήλων και προϊσταμένων. Η σχέση ενός υπαλλήλου με το μοναδικό προϊστάμενό του μπορεί να δημιουργήσει τριβές. Η ύπαρξη πολλών προϊσταμένων μπορεί να δημιουργήσει σύγχυση.
- Σε περιπτώσεις όπου ανακύπτουν διαφωνίες από διαφορετικές εκτιμήσεις μπορεί να είναι απαραίτητη η ύπαρξη συναίνεσης. Αυτό σημαίνει ότι απαιτείται αμοιβαία εμπιστοσύνη, ενώ πρέπει να αποφεύγεται η απόκρυψη στοιχείων και πληροφοριών για τα έργα μεταξύ των υπευθύνων.
- Η καλή προδιαγραφή του λογισμικού είναι απαραίτητη, όχι μόνον για την υλοποίησή του, αλλά και για τη διαχείριση της ανάπτυξής του. Ο λόγος είναι ότι χωρίς καλές προδιαγραφές δεν είναι δυνατή η παρακολούθηση της πορείας ανάπτυξης, η επίλυση διαφωνιών και η επίγνωση της κατάστασης των έργων.

- Πολλές φορές η ανάπτυξη λογισμικού θα πρέπει να ανατεθεί από ένα διαχειριστή σε εξωτερική ομάδα. Στην περίπτωση αυτή ο διαχειριστής θα πρέπει να φροντίσει να έχει στη διάθεση του τα απαραίτητα εργαλεία για την παρακολούθηση της πορείας του έργου (καλές προδιαγραφές, περιοδικοί έλεγχοι, καλές περιπτώσεις δοκιμών και δοκιμαστικά δεδομένα).
- Η διαχείριση των έργων έχει και αυτή το κόστος της, το οποίο είναι σημαντικό και πρέπει να λαμβάνεται υπόψη κατά το σχεδιασμό των έργων και των δομών τους.

## Ενδεικτική Απάντηση Δραστηριότητας 2.2

---

A) Μία πιθανή ανάθεση ρόλων θα μπορούσε να είναι η εξής

- 1 Τεχνικός Διαχειριστής
- 2 Μηχανικοί Προδιαγραφών και Αναλυτές
- 2 Σχεδιαστές
- 3 Προγραμματιστές (εκ των οποίων ο ένας και Συντηρητής)
- 1 Δοκιμαστής και Μηχανικός Ποιότητας
- 1 Εκπαιδευτής και Μηχανικός Τεκμηρίωσης

Είναι πιθανό να δώσατε διαφορετική απάντηση, η οποία μπορεί να είναι σωστή αρκεί να προσπαθεί να καλύψει όλους τους ρόλους σύμφωνα με την υπόδειξη και να μην περιέχει προφανείς ασυμμετρίες (π.χ. να έχει δύο τεχνικούς υπευθύνους και έναν προγραμματιστή!). Προσέξτε όμως πως στις δύο τελευταίες απαντήσεις προσπαθούμε να μοιράσουμε τις αυξημένες απαιτήσεις από το έργο σε όσο το δυνατόν περισσότερα μέλη της ομάδας, αποφεύγοντας να τις επικεντρώσουμε σέ ένα και μόνον άτομο (θυμηθείτε ότι τα μέλη της ομάδας πρέπει να μοιράζονται τους κοινούς στόχους του έργου).

B)

1) Θα πρέπει να ενισχυθεί ο ρόλος του εκπαιδευτή. Θα μπορούσαμε να αναθέσουμε την Τεκμηρίωση σε έναν προγραμματιστή και να έχουμε έναν εκπαιδευτή πλήρους απασχόλησης. Εναλλακτικά θα μπορούσαμε να αναθέσουμε το ρόλο του εκπαιδευτή και σε έναν ή δύο προγραμματιστές και σε έναν ή δύο αναλυτές.

2) Θα πρέπει να ενισχυθεί σημαντικά ο ρόλος του δοκιμαστή και του μηχανικού ποιότητας. Η νέα σύνθεση θα μπορούσε να είναι:

- 1 Τεχνικός Διαχειριστής
- 1 Μηχανικός Προδιαγραφών
- 1 Αναλυτής και Μηχανικός Ποιότητας
- 2 Σχεδιαστές και Δοκιμαστές
- 2 Προγραμματιστές (εκ των οποίων ο ένας και Συντηρητής)
- 1 Δοκιμαστής
- 1 Μηχανικός Ποιότητας
- 1 Εκπαιδευτής και Μηχανικός Τεκμηρίωσης

3) Θα πρέπει να ενισχυθεί ο ρόλος του μηχανικού προδιαγραφών, του μηχανικού ποιότητας και του μηχανικού τεκμηρίωσης. Η νέα σύνθεση θα μπορούσε να είναι:

- 1 Τεχνικός Διαχειριστής
- 1 Μηχανικός Προδιαγραφών
- 1 Αναλυτής και Μηχανικός Προδιαγραφών
- 1 Αναλυτής και Σχεδιαστής
- 1 Σχεδιαστής
- 3 Προγραμματιστές (εκ των οποίων ο ένας και Συντηρητής, και ο ένας και Μηχανικός Τεκμηρίωσης)
- 1 Δοκιμαστής και Μηχανικός Ποιότητας
- 1 Εκπαιδευτής και Μηχανικός Τεκμηρίωσης

### **Ενδεικτική Απάντηση Δραστηριότητας 2.3**

---

Για να απαντήσετε σωστά στη δραστηριότητα αυτή θα πρέπει να μελετήσετε τους δύο καταλόγους των ικανοτήτων των μελών μίας ομάδας στην παράγραφο σχετικά με τους ρόλους των μελών της ομάδας και να σκεφθείτε τους ρόλους στους οποίους η κάθε μία αποκτά ιδιαίτερη σημασία. Η απάντηση που ακολουθεί είναι ενδεικτική,

καθώς περισσότερο λεπτομερής περιγραφή του έργου θα μπορούσε να τροποποιήσει τα κριτήρια επιλογής που προτείνουμε εδώ.

- 1) Για το ρόλο του εκπαιδευτή είναι πολύ σημαντική η ικανότητα επικοινωνίας με άλλα άτομα, αφού ο εκπαιδευτής θα πρέπει να επικοινωνήσει σε μία σειρά μαθημάτων με τους χρήστες του συστήματος. Στις συναντήσεις αυτές θα πρέπει να τους μεταδώσει αποτελεσματικά τις γνώσεις και δεξιότητες που αυτοί χρειάζονται. Δευτερευόντως, εμπειρία σε παρόμοιες εφαρμογές θα ήταν ιδιαίτερα επιθυμητή.
- 2) Η εμπειρία σε παρόμοιες εφαρμογές πιθανώς να είναι ένα πολύ καλό κριτήριο επιλογής για το μηχανικό ποιότητας. Εμπειρία σε κρίσιμες εφαρμογές θα μπορούσε να μεταφερθεί στο έργο και να βοηθήσει στον εντοπισμό προβλημάτων και λαθών στη διεργασία ανάπτυξης (βλ. και το κεφάλαιο σχετικά με την ποιότητα του παρόντος Τόμου). Εμπειρία σε συγκεκριμένες λειτουργίες ή δραστηριότητες θα βοηθήσει μόνον τμηματικά, καθώς ο μηχανικός ποιότητας θα πρέπει να διασφαλίσει την ποιότητα όλης της διεργασίας ανάπτυξης. Ικανότητες επικοινωνίας και διαχείρισης θα μπορούσαν να είναι επίσης επιθυμητές λόγω της ανάγκης για επικοινωνία με όλη την ομάδα. Για το δοκιμαστή θα επιθυμούσαμε και πάλι εμπειρία σε παρόμοιες εφαρμογές, αλλά και εμπειρία σε δοκιμές σε προηγούμενα έργα.
- 3) Για το ρόλο του μηχανικού προδιαγραφών ιδιαίτερη σημασία έχει η ικανότητα επικοινωνίας με άτομα του πελάτη και χρήστες. Εμπειρία σε συγγραφή προδιαγραφών και παρόμοιες εφαρμογές είναι επίσης ιδιαίτερα επιθυμητές. Για το μηχανικό τεκμηρίωσης απαιτείται τουλάχιστον καλή ικανότητα γραπτής επικοινωνίας.

## **Ενδεικτική Απάντηση Δραστηριότητας 2.4**

---

Χρησιμοποιώντας μία μηχανή όπως το Google και λέξεις-κλειδί όπως personality test, temperament test μπορείτε να εντοπίσετε πολλές σχετικές ιστοσελίδες. Ενδεικτικές διευθύνσεις στο Διαδίκτυο είναι οι ακόλουθες:

<http://www.keirsey.com/>

<http://www.humanmetrics.com/cgi-win/JTypes2.asp>

<http://haleonline.com/>

Μπορείτε να εκτελέσετε και εσείς το τεστ για τον εαυτό σας και να διαπιστώσετε σε ποια κατηγορία ανήκετε!

### Ενδεικτική Απάντηση Δραστηριότητας 3.1

---

Υπάρχει μεγάλος αριθμός εργαλείων για τη διαχείριση απαιτήσεων. Βέβαια, η λίστα μεταβάλλεται διαρκώς και για το λόγο αυτό αναφέρουμε ενδεικτικά κάποια από αυτά:

- Rational RequisitePro της εταιρεία IBM (<http://www-306.ibm.com/software/awdtools/reqpro>)
- Analyst Pro Version 5.0 της εταιρείας Goda Software (<http://www.analysttool.com>)
- CaliberRM 2005 της εταιρείας Borland (<http://www.borland.com/us/products/caliber/>)
- Focal Point της εταιρείας Telelogic AB (<http://www.telelogic.com/>)

Η βασική λειτουργικότητα που προσφέρει ένα σύστημα διαχείρισης απαιτήσεων είναι:

- Ολοκληρωμένη διαχείριση του κύκλου ζωής των απαιτήσεων.
- Ιχνηλάτιση απαιτήσεων κατά τη διάρκεια του κύκλου ζωής ανάπτυξης λογισμικού.
- Ομαδοποίηση απαιτήσεων σε πολλαπλά επίπεδα.
- Αυτόματη συσχέτιση παρόμοιων, ομοειδών απαιτήσεων.
- Αυτόματη παραγωγή τεκμηρίωσης.
- Ολοκλήρωση, με άλλα εργαλεία CASE.
- κ.α.

### Ενδεικτική Απάντηση Δραστηριότητας 3.2

---

Έστω ένα σύστημα ηλεκτρονικής υποβολής προτάσεων, με τίτλο ΠΡΟΤΑΣΗ, για ερευνητικά έργα χρηματοδοτούμενα από την Γενική Γραμματεία Έρευνας και Τεχνολογίας (ΓΓΕΤ). Το έργο περιλαμβάνει:

- την ανάπτυξη, εγκατάσταση και συντήρηση του συστήματος πληροφορικής
- την παραμετροποίηση του συστήματος για τη χρήση του σε διαφορετικά προγράμματα ενίσχυσης ερευνητικών δραστηριοτήτων της ΓΓΕΤ και

- τη λειτουργία του συστήματος για δύο έτη από την ανάδοχο εταιρεία

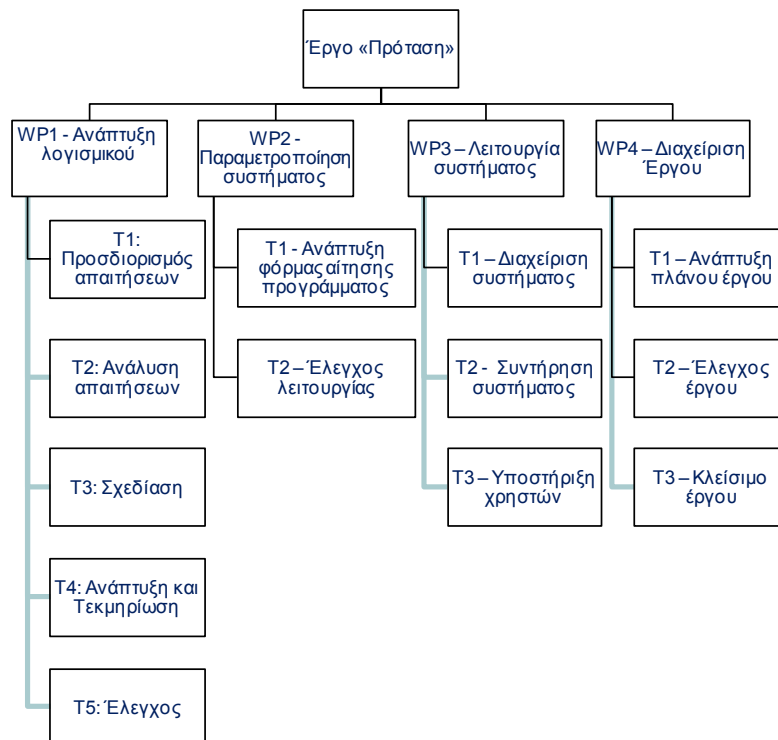
Σε ένα ανομοιογενές έργο όπως το έργο ΠΡΟΤΑΣΗ, όπου συνυπάρχει η ανάπτυξη λογισμικού, μαζί με την παραμετροποίηση και τη λειτουργία του συστήματος είναι λογικό να προσπαθήσουμε να διαχωρίσουμε αυτές τις ανομοιογενείς δραστηριότητες. Για το λόγο αυτό προτείνουμε τη δημιουργία τριών πακέτων εργασίας (work packages). Έτσι έχουμε:

- WP1 – Ανάπτυξη συστήματος
- WP2 – Παραμετροποίηση συστήματος για κάθε πρόγραμμα
- WP3 – Λειτουργία συστήματος

Επιπλέον, προσθέτουμε ένα τέταρτο πακέτο εργασίας σχετικό με τη διαχείριση του έργου.

- WP4 – Διαχείριση έργου

Στο παρακάτω σχήμα, παρουσιάζουμε την οργάνωση σε πακέτα εργασίας καθώς και ένα ενδεικτικό αριθμό δραστηριοτήτων ανά πακέτο εργασίας.



## Ενδεικτική Απάντηση Δραστηριότητας 4.1

Τα πλεονεκτήματα που δίνει η χρήση των συστημάτων PMIS είναι τα εξής:



- Αναλυτικός ορισμός των έργων.
- Χρονοπρογραμματισμός των έργων με αυτόματη ανάθεση και εξομάλυνση.
- Καλύτερος προγραμματισμός και έλεγχος πόρων.
- Καλύτερη ανάλυση και έλεγχος κόστους έργου.
- Ολοκλήρωση των δραστηριοτήτων του έργου με αυτές του οργανισμού φορέα.
- Καλύτερη διαχείριση των εγγράφων του έργου.

Αντίστοιχα, τα μειονεκτήματα από τη χρήση συστημάτων PMIS είναι τα ακόλουθα:

- Αυξημένο κόστος από την προμήθεια και εγκατάσταση του PMIS.
- Αναγκαιότητα εκπαίδευσης προσωπικού στη χρήση του εργαλείου.
- Αυξημένος χρόνος για την εισαγωγή και συντήρηση των δεδομένων.

## Ενδεικτική Απάντηση Δραστηριότητας 4.2

Σύμφωνα με τον κανονισμό σπουδών του Ελληνικού Ανοικτού Πανεπιστημίου κάθε πτυχιακή εργασία ολοκληρώνεται σε τέσσερις φάσεις. Κάθε επόμενη φάση βασίζεται στο καταγεγραμμένο υλικό της προηγούμενης, το οποίο επεκτείνεται και αναλύεται κατά τρόπο ανάλογο με την εξέλιξη της εργασίας. Οι φάσεις αυτές είναι η σύλληψη, η επεξεργασία, η ανάπτυξη και η μετάβαση, ενώ μετά τη λήξη τους ακολουθεί η προφορική παρουσίαση της πτυχιακής εργασίας σε ανοιχτή ημερίδα.

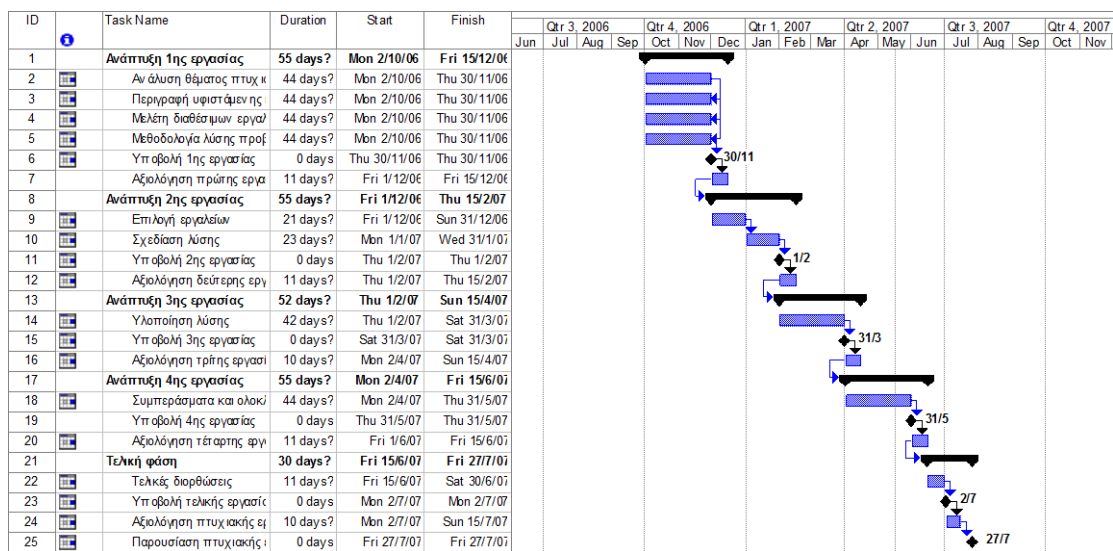
Βάσει των ανωτέρω, το χρονοδιάγραμμα μιας πτυχιακής εργασίας για το ακαδημαϊκό έτος 2006-2007 είναι το ακόλουθο:

Δραστηριότητα	Έναρξη	Λήξη
Ανάλυση θέματος πτυχιακής εργασίας	1/10/2006	30/11/2006
Περιγραφή υφιστάμενης κατάστασης	1/10/2006	30/11/2006
Μελέτη διαθέσιμων εργαλείων	1/10/2006	30/11/2006
Μεθοδολογία λύσης προβλήματος	1/10/2006	30/11/2006
Αξιολόγηση πρώτης εργασίας	1/12/2006	15/12/2006
Επιλογή εργαλείων	1/12/2006	31/12/2006

Σχεδίαση λύσης	1/1/2007	31/1/2007
Αξιολόγηση δεύτερης εργασίας	1/2/2007	15/2/2007
Υλοποίηση λύσης	1/2/2007	31/3/2007
Αξιολόγηση τρίτης εργασίας	1/4/2007	15/4/2007
Συμπεράσματα και ολοκλήρωση κειμένου	1/4/2007	31/5/2007
Αξιολόγηση τέταρτης εργασίας	1/6/2007	15/6/2007
Τελικές διορθώσεις	15/6/2007	30/6/2007
Αξιολόγηση πτυχιακής εργασίας	1/7/2007	15/7/2007
Παρουσίαση πτυχιακής εργασίας σε ημερίδα	27/7/2007	27/7/2007

Για τη δημιουργία του διαγράμματος Gantt κάνουμε τις ακόλουθες παραδοχές:

- Η παράδοση της κάθε εργασίας αποτελεί ορόσημο του έργου.
- Ο καθηγητής παραδίδει σχόλια για κάθε εργασία, ενώ ο φοιτητής συνεχίζει την εργασία του εκτός από την περίπτωση της τελικής υποβολής.



### Ενδεικτική Απάντηση Δραστηριότητας 4.3

α) Αρχικά, εξετάζουμε το ενδεχόμενο να αναθέσουμε τη δραστηριότητα σε άλλο μέλος της ομάδας έργου, είτε καθ' ολοκληρία, είτε στο τμήμα που δεν μπορεί να εκτελεσθεί. Εναλλακτικά, εξετάζουμε το ενδεχόμενο της παράτασης της δραστηριότητας

για ένα ακόμη μήνα, ώστε να μπορέσει να ολοκληρωθεί. Μια τρίτη λύση είναι να αναθέσουμε τη δραστηριότητα αυτή σε υπεργολάβους. Εάν καμία από τις λύσεις αυτές δεν είναι εφικτή, τότε ενημερώνουμε τη διοίκηση του οργανισμού για το πρόβλημα.

β) Εφόσον ο προγραμματιστής μπορεί να ενταχθεί στην ομάδα του έργου από τον επόμενο μήνα μπορούμε να προγραμματίσουμε υπερωρίες του προσωπικού ή επιμήκυνση του χρονοδιαγράμματος.

### Ενδεικτική Απάντηση Δραστηριότητας 5.1

---

Όλες οι τεχνικές βασίζονται σε ιστορικά έργα για να μπορέσουν να εκφράσουν ποσοτικά την επίδραση που έχουν οι διάφοροι παράγοντες στο κόστος και για να μπορέσουν να συλλάβουν την παραγωγικότητα που τα ιστορικά έργα απεικονίζουν. Αυτό οφείλεται στο ότι δεν είναι δυνατό με μία απλή επισκόπηση των χαρακτηριστικών ενός έργου να προβλέψουμε το κόστος του: πρέπει να ανατρέξουμε σε άλλες παρόμοιες περιπτώσεις έργων (όσο το δυνατόν περισσότερες) για να επαναχρησιμοποιήσουμε τη γνώση που έχει αποκτηθεί στο παρελθόν.

### Ενδεικτική Απάντηση Δραστηριότητας 5.2

---

#### *Επιλογή κατηγορίας έργου*

Από τον πίνακα 1 φαίνεται ότι η κατηγορία του έργου είναι η ενσωματωμένη. Το σύστημα εξάλλου παρουσιάζει πολλά κοινά χαρακτηριστικά με αυτό του συστήματος του χρηματιστηρίου της Δραστηριότητας 5.1.

#### *Υπολογισμός ονομαστικού φόρτου*

Με βάση τον τύπο  $MM_{\text{NOM}} = aKDSI^b$  και τις τιμές των  $a$  και  $b$  από τον πίνακα 2 για την ενσωματωμένη κατηγορία υπολογίζουμε ότι  $MM_{\text{NOM}} = 2.8 (10)^{1.20} = 44$  ανθρωπομήνες.

#### *Ανάθεση τιμών στους παράγοντες κόστους*

Μελετώντας τις υψηλού επιπέδου προδιαγραφές του συστήματος που έχουμε στη διάθεσή μας και κάνοντας υποθέσεις, εκτιμήσεις και παραδοχές όπου δεν έχουμε

συγκεκριμένα στοιχεία καταλήγουμε στις τιμές του πίνακα που ακολουθεί. Αν έχετε επιλέξει άλλες τιμές από αυτές του πίνακα δεν πρόκειται για λάθος, αλλά για διαφορετική εκτίμηση ή υπόθεση που έχετε κάνει, δεδομένου ότι δεν έχουμε στη διάθεσή μας λεπτομερή περιγραφή του συστήματος, επομένως μην απογοητεύεστε σε καμία περίπτωση! Επιπλέον υπάρχει σημαντικός βαθμός υποκειμενικότητας στην επιλογή αυτών των τιμών. Προσέξτε ότι όταν δεν έχουμε κάποια πληροφορία στη διάθεσή μας για κάποιον παράγοντα προτιμούμε την ονομαστική τιμή του, ώστε να μην επηρεάσουμε καθόλου την εκτίμηση.

Πίνακας τιμών παραγόντων κόστους για το σύστημα μεταφοράς κεφαλαίων

Συντομογραφία παράγοντα κόστους	Κατηγορική τιμή	Αριθμητική τιμή	Σύντομη αιτιολόγηση
RELY	Υψηλή	1.15	Φόβος για σημαντικές οικονομικές απώλειες σε περίπτωση αστοχίας του συστήματος
DATA	Χαμηλή	0.94	Μικρός όγκος δεδομένων συστήματος
CPLX	Υψηλότατη	1.30	Μεγάλη πολυπλοκότητα λόγω των πολλών ελέγχων που απαιτεί η μεταφορά κεφαλαίου
TIME	Υψηλή	1.11	Μεταφορές κεφαλαίων σε πραγματικό χρόνο
STOR	Υψηλή	1.06	Ανάγκη για καταγραφή όλων των κινήσεων για λογους ασφαλείας
VIRT	Ονομαστική	1.00	Δεν υπάρχουν πληροφορίες
TURN	Ονομαστική	1.00	Δεν υπάρχουν πληροφορίες
ACAP	Υψηλή	0.86	Υπόθεση: θα έχουμε στη διάθεσή μας αναλυτές υψηλού επιπέδου
AEXP	Ονομαστική	1.00	Δεν έγινε κάποια υπόθεση
PCAP	Υψηλή	0.86	Υπόθεση: θα έχουμε στη διάθεσή μας προγραμματιστές υψηλού επιπέδου
VEXP	Χαμηλή	1.10	Υπόθεση: το προσωπικό μας έχει λίγη εμπειρία με την πλατφόρμα υλικού/ λογισμικού που θα χρησιμοποιηθεί
LEXP	Ονομαστική	1.00	Δεν έγινε κάποια υπόθεση
MODP	Υψηλή	0.91	Θα ακολουθήσουμε μοντέρνα μεθοδολογία ανάπτυξης (π.χ. την

			RUP)
TOOL	Χαμηλή	1.10	Υπόθεση: θα κάνουμε μικρή χρήση εργαλείων CASE πέραν των βασικών (μεταγλωττιστής)
SCED	Ονομαστική	1.00	Δεν θα υπάρξουν πιέσεις για αυξομειώσεις της χρονικής διάρκειας του έργου

#### *Υπολογισμός Πολλαπλασιαστή Φόρτου και Φόρτου Ανάπτυξης*

Πολλαπλασιάζοντας τις 15 τιμές του πίνακα 6 παίρνουμε το αποτέλεσμα  $q = 1.35$ . Αυτό σημαίνει ότι ο ονομαστικός φόρτος θα αυξηθεί κατά 35% λόγω των ιδιαίτερα αυξημένων απαιτήσεων και των άλλων χαρακτηριστικών του έργου. Επομένως  $MM_{DEV} = q \cdot MM_{NOM} = 1.35 \cdot 44 = 59$  ανθρωπομήνες

#### *Υπολογισμός απαιτούμενης διάρκειας ανάπτυξης*

Σύμφωνα με τις εξισώσεις του πίνακα 2 για ενσωματωμένα έργα, έχουμε ότι  $T_{DEV} = 2.5 \cdot MM_{DEV}^{0.32} = 9$  μήνες.

### **Ενδεικτική Απάντηση Δραστηριότητας 5.4**

---

Το αποτέλεσμα που θα πάρετε με το μοντέλο COCOMO81 λογικά θα είναι αρκετά υψηλότερο από το φόρτο που εσείς αφιερώσατε. Παρά το ότι μπορεί να σας χρειάστηκε αρκετός χρόνος για να εκπονήσετε την εργασία σας, ή χαμηλή παραγωγικότητα των έργων του μοντέλου και το γεγονός ότι το μοντέλο υπολογίζει το κόστος για την ανάπτυξη επαγγελματικών προϊόντων (λεπτομερής τεκμηρίωση, καλά δοκιμασμένες λειτουργίες) θα πρέπει τελικά να οδηγήσει σε υπερεκτίμηση της προσπάθειάς σας.

### **Ενδεικτική Απάντηση Δραστηριότητας 5.5**

---

Το μοντέλο COCOMO 2.0 παρουσιάζει σημαντική πρόοδο ως προς την εκτίμηση του μεγέθους, καθώς εξαλείφει την ανάγκη για εκτίμηση του αριθμού των γραμμών κώδικα στις αρχικές φάσεις του έργου. Επίσης το μοντέλο COCOMO 2.0, εμφανίζει βελτιώσεις σε σχέση με το αρχικό μοντέλο COCOMO και ως προς τους παράγοντες κόστους:

λαμβάνονται υπ' όψιν σημαντικοί παράγοντες της μοντέρνας βιομηχανίας ανάπτυξης λογισμικού, όπως η επαναχρησιμοποίηση κώδικα, η συντήρηση, η αλλαγή των απαιτήσεων, καθώς επίσης και συγκεκριμένοι νέοι παράγοντες που επηρεάζουν σημαντικό το κόστος (π.χ. συνέχεια προσωπικού).

## Ενδεικτική Απάντηση Δραστηριότητας 5.6

1 Γ (Προσέξτε ότι στο μοντέλο COCOMO81 υπάρχει παράγοντας που αφορά το χρονοδιάγραμμα του έργου)

2 Δ (Στην αρχή χρησιμεύουν για το χρονοπρογραμματισμό του έργου, στη συνέχεια στον έλεγχο της πορείας του)

3 Δ (Παραγωγικότητα σημαίνει όγκος παραγόμενου προϊόντος ή εργασίας προς την προσπάθεια που απαιτήθηκε)

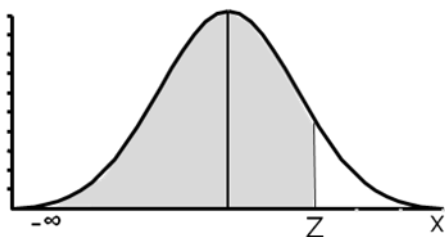
4 Γ (Δείτε τις εξισώσεις του Πίνακα 2)

5 Α (Δείτε τον Πίνακα 1)

6 Β (Προσέξτε τους εκθέτες στις εξισώσεις του Πίνακα 2)

7 Β (Δείτε προς το τέλος της παραγράφου που περιγράφει τη μέθοδο των αναλογιών τη συζήτηση για τις πιθανές επιλογές στην εφαρμογή της μεθόδου)

## Παράρτημα Α - Αθροιστική κανονική κατανομής πιθανότητας (από $-\infty$ ως Z)



Εάν  $Z=1,51$  πηγαίνουμε στην όγδοη στήλη και βρίσκουμε ότι η πιθανότητα είναι 93,45 %.

P(x<Z <sub>a</sub> )	Z <sub>a</sub>	P(x<Z <sub>a</sub> )	Z <sub>a</sub>	P(x<Z <sub>a</sub> )	Z <sub>a</sub>	P(x<Z <sub>a</sub> )	Z <sub>a</sub>	P(x<Z <sub>a</sub> )	Z <sub>a</sub>	P(x<Z <sub>a</sub> )	Z <sub>a</sub>
50,00%	0	69,15%	0,5	84,13%	1	93,32%	1,5	97,72%	2	99,38%	2,5
50,40%	0,01	69,50%	0,51	84,38%	1,01	93,45%	1,51	97,78%	2,01	99,40%	2,51
50,80%	0,02	69,85%	0,52	84,61%	1,02	93,57%	1,52	97,83%	2,02	99,41%	2,52
51,20%	0,03	70,19%	0,53	84,85%	1,03	93,70%	1,53	97,88%	2,03	99,43%	2,53

$P(x < Z_a)$	$Z_a$	$P(x < Z_a)$	$Z_a$	$P(x < Z_a)$	$Z_a$	$P(x < Z_a)$	$Z_a$	$P(x < Z_a)$	$Z_a$	$P(x < Z_a)$	$Z_a$
51,60%	0,04	70,54%	0,54	85,08%	1,04	93,82%	1,54	97,93%	2,04	99,45%	2,54
51,99%	0,05	70,88%	0,55	85,31%	1,05	93,94%	1,55	97,98%	2,05	99,46%	2,55
52,39%	0,06	71,23%	0,56	85,54%	1,06	94,06%	1,56	98,03%	2,06	99,48%	2,56
52,79%	0,07	71,57%	0,57	85,77%	1,07	94,18%	1,57	98,08%	2,07	99,49%	2,57
53,19%	0,08	71,90%	0,58	85,99%	1,08	94,29%	1,58	98,12%	2,08	99,51%	2,58
53,59%	0,09	72,24%	0,59	86,21%	1,09	94,41%	1,59	98,17%	2,09	99,52%	2,59
53,98%	0,1	72,57%	0,6	86,43%	1,1	94,52%	1,6	98,21%	2,1	99,53%	2,6
54,38%	0,11	72,91%	0,61	86,65%	1,11	94,63%	1,61	98,26%	2,11	99,55%	2,61
54,78%	0,12	73,24%	0,62	86,86%	1,12	94,74%	1,62	98,30%	2,12	99,56%	2,62
55,17%	0,13	73,57%	0,63	87,08%	1,13	94,84%	1,63	98,34%	2,13	99,57%	2,63
55,57%	0,14	73,89%	0,64	87,29%	1,14	94,95%	1,64	98,38%	2,14	99,59%	2,64
55,96%	0,15	74,22%	0,65	87,49%	1,15	95,05%	1,65	98,42%	2,15	99,60%	2,65
56,36%	0,16	74,54%	0,66	87,70%	1,16	95,15%	1,66	98,46%	2,16	99,61%	2,66
56,75%	0,17	74,86%	0,67	87,90%	1,17	95,25%	1,67	98,50%	2,17	99,62%	2,67
57,14%	0,18	75,17%	0,68	88,10%	1,18	95,35%	1,68	98,54%	2,18	99,63%	2,68
57,53%	0,19	75,49%	0,69	88,30%	1,19	95,45%	1,69	98,57%	2,19	99,64%	2,69
57,93%	0,2	75,80%	0,7	88,49%	1,2	95,54%	1,7	98,61%	2,2	99,65%	2,7
58,32%	0,21	76,11%	0,71	88,69%	1,21	95,64%	1,71	98,64%	2,21	99,66%	2,71
58,71%	0,22	76,42%	0,72	88,88%	1,22	95,73%	1,72	98,68%	2,22	99,67%	2,72
59,10%	0,23	76,73%	0,73	89,07%	1,23	95,82%	1,73	98,71%	2,23	99,68%	2,73
59,48%	0,24	77,04%	0,74	89,25%	1,24	95,91%	1,74	98,75%	2,24	99,69%	2,74
59,87%	0,25	77,34%	0,75	89,44%	1,25	95,99%	1,75	98,78%	2,25	99,70%	2,75
60,26%	0,26	77,64%	0,76	89,62%	1,26	96,08%	1,76	98,81%	2,26	99,71%	2,76
60,64%	0,27	77,94%	0,77	89,80%	1,27	96,16%	1,77	98,84%	2,27	99,72%	2,77
61,03%	0,28	78,23%	0,78	89,97%	1,28	96,25%	1,78	98,87%	2,28	99,73%	2,78
61,41%	0,29	78,52%	0,79	90,15%	1,29	96,33%	1,79	98,90%	2,29	99,74%	2,79
61,79%	0,3	78,81%	0,8	90,32%	1,3	96,41%	1,8	98,93%	2,3	99,74%	2,8
62,17%	0,31	79,10%	0,81	90,49%	1,31	96,49%	1,81	98,96%	2,31	99,75%	2,81
62,55%	0,32	79,39%	0,82	90,66%	1,32	96,56%	1,82	98,98%	2,32	99,76%	2,82
62,93%	0,33	79,67%	0,83	90,82%	1,33	96,64%	1,83	99,01%	2,33	99,77%	2,83
63,31%	0,34	79,95%	0,84	90,99%	1,34	96,71%	1,84	99,04%	2,34	99,77%	2,84
63,68%	0,35	80,23%	0,85	91,15%	1,35	96,78%	1,85	99,06%	2,35	99,78%	2,85
64,06%	0,36	80,51%	0,86	91,31%	1,36	96,86%	1,86	99,09%	2,36	99,79%	2,86



<b>P(x&lt;Z<sub>a</sub>)</b>	<b>Z<sub>a</sub></b>	<b>P(x&lt;Z<sub>a</sub>)</b>	<b>Z<sub>a</sub></b>	<b>P(x&lt;Z<sub>a</sub>)</b>	<b>Z<sub>a</sub></b>	<b>P(x&lt;Z<sub>a</sub>)</b>	<b>Z<sub>a</sub></b>	<b>P(x&lt;Z<sub>a</sub>)</b>	<b>Z<sub>a</sub></b>	<b>P(x&lt;Z<sub>a</sub>)</b>	<b>Z<sub>a</sub></b>
64,43%	0,37	80,78%	0,87	91,47%	1,37	96,93%	1,87	99,11%	2,37	99,79%	2,87
64,80%	0,38	81,06%	0,88	91,62%	1,38	96,99%	1,88	99,13%	2,38	99,80%	2,88
65,17%	0,39	81,33%	0,89	91,77%	1,39	97,06%	1,89	99,16%	2,39	99,81%	2,89
65,54%	0,4	81,59%	0,9	91,92%	1,4	97,13%	1,9	99,18%	2,4	99,81%	2,9
65,91%	0,41	81,86%	0,91	92,07%	1,41	97,19%	1,91	99,20%	2,41	99,82%	2,91
66,28%	0,42	82,12%	0,92	92,22%	1,42	97,26%	1,92	99,22%	2,42	99,82%	2,92
66,64%	0,43	82,38%	0,93	92,36%	1,43	97,32%	1,93	99,25%	2,43	99,83%	2,93
67,00%	0,44	82,64%	0,94	92,51%	1,44	97,38%	1,94	99,27%	2,44	99,84%	2,94
67,36%	0,45	82,89%	0,95	92,65%	1,45	97,44%	1,95	99,29%	2,45	99,84%	2,95
67,72%	0,46	83,15%	0,96	92,79%	1,46	97,50%	1,96	99,31%	2,46	99,85%	2,96
68,08%	0,47	83,40%	0,97	92,92%	1,47	97,56%	1,97	99,32%	2,47	99,85%	2,97
68,44%	0,48	83,65%	0,98	93,06%	1,48	97,61%	1,98	99,34%	2,48	99,86%	2,98
68,79%	0,49	83,89%	0,99	93,19%	1,49	97,67%	1,99	99,36%	2,49	99,86%	2,99