

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΕΧΝΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ
ΚΑΤΕΥΘΥΝΣΗ: ΤΕΧΝΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ
ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΔΙΟΙΚΗΣΗ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ: ΘΕΩΡΙΑ ΚΑΙ ΜΕΛΕΤΗ
ΠΕΡΙΠΤΩΣΗΣ**

ΠΡΕΒΕΝΑΣ ΑΝΤΩΝΙΟΣ ΜΤΕ-0961
ΕΠΙΒΛΕΠΟΥΣΑ: ΜΑΛΑΜΑΤΕΝΙΟΥ ΦΛΩΡΑ

ΠΕΙΡΑΙΑΣ, ΙΟΥΝΙΟΣ 2012

РАНЕЕЗНАМО ТЕРРА

Περίληψη	5
Abstract	6
Ευχαριστίες	7
ΚΕΦΑΛΑΙΟ 1: ΚΥΚΛΟΙ ΖΩΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ	8
1.1 Εισαγωγή	8
1.2 Φάσεις ανάπτυξης πληροφοριακών συστημάτων	9
1.3 Το μοντέλο του καταρράκτη	11
1.4 Το μοντέλο πρωτυποποίησης	14
1.5 Το μοντέλο λειτουργικής επαύξησης	17
1.6 Το σπειροειδές μοντέλο	19
1.7 Το μοντέλο του πίδακα	23
1.8 Σύγχρονα μοντέλα κύκλου ζωής λογισμικού	25
ΚΕΦΑΛΑΙΟ 2: ΠΡΟΣΕΓΓΙΣΕΙΣ ΑΝΑΠΤΥΞΗΣ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	27
2.1 Εισαγωγή	27
2.2 Ευέλικτες Μέθοδοι Ανάπτυξης (Agile Software Development)	27
2.3 Extreme Programming	33
2.4 Rational Unified Process	39
2.5 Σύγκριση	46
ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΕΣ ΔΙΟΙΚΗΣΗΣ ΕΡΓΩΝ	53
3.1 Εισαγωγή	53
3.2 Prince2	54
3.3 PMBOK	61
3.4 Σύγκριση Prince2 και PMBOK	67
ΚΕΦΑΛΑΙΟ 4: ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΡΓΩΝ	69
4.1 Εισαγωγή	69
4.2 Δομή Ανάλυσης Εργασιών	69
4.3 Μέθοδοι χρονοπρογραμματισμού	71
4.3.1 Δικτυακή Απεικόνιση ενός Έργου	72
4.3.2 Ανάλυση PERT/CPM	75
4.3.2.1 Μέθοδος CPM – Critical Path Method	77
4.3.2.2 Μέθοδος PERT – Programme Evaluation and Review Technique	79
4.3.3 Χρονική Παρακολούθηση Έργων με το Διάγραμμα GANTT	81
4.4 Εργαλεία διοίκησης έργων πληροφορικής	83
4.4.1 GanntProject	84
4.4.2 OmniPlan	85
4.4.3 Open Workbench	85
4.4.4 PertMaster	86
4.4.5 PlanningForce	87
4.4.6 Microsoft Project	88
4.4.7 Achievo	89
ΚΕΦΑΛΑΙΟ 5: ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ	91
5.1 Εισαγωγή	91
5.1 Φάση Αρχικοποίησης – Ορισμός Πεδίου	91
5.1.1 Στόχος του έργου	91
5.1.2 Οι συμμετέχοντες στο έργο	92
5.1.3 Βασικά Παραδοτέα (deliverables)	92
5.1.4 Τεχνικές απαιτήσεις	92
5.1.5 Λειτουργικές απαιτήσεις	93

5.1.6 Business Case.....	93
5.1.6.1 Σκοπός.....	93
5.1.6.2 ΛΕΙΤΟΥΡΓΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ ΥΠΟΣΥΣΤΗΜΑΤΩΝ.....	94
5.2 Φάση σχεδιασμού του έργου	101
5.2.1 Πλάνο διαχείρισης του Πεδίου του Έργου	101
5.2.1.1 Δομή Ανάλυσης Εργασιών(ΔΑΕ).....	101
5.2.2 Πλάνο διαχείρισης του Κινδύνου	103
5.2.3 Πλάνο διαχείρισης Ανθρώπινου Δυναμικού	104
5.2.3.1 Οργανωτικός σχεδιασμός	104
5.2.4 Πλάνο διαχείρισης Επικοινωνίας όσων εμπλέκονται στο έργο.....	105
5.2.5 Πλάνο διαχείρισης Ποιότητας	105
5.2.6 Πλάνο διαχείρισης Κόστους	105
5.2.7 Πλάνο διαχείρισης Προμηθειών	106
5.3 Φάση Εκτέλεσης.....	106
5.4 Φάση Ελέγχου.....	107
5.5 Κλείσιμο Έργου	108
ΣΥΜΠΕΡΑΣΜΑΤΑ	109
Παράρτημα Α GANTT CHART	111
ΒΙΒΛΙΟΓΡΑΦΙΑ	113

Περίληψη

Στην παρούσα διπλωματική εργασία περιγράφονται οι πιο διαδομένες μέθοδοι διαχείρισης έργων όπως είναι η Prince2 και η PMBOK και μία σύγκριση αυτών. Επίσης, παρουσιάζονται οι κύκλοι ζωής ανάπτυξης πληροφοριακών συστημάτων, κάποιες από τις πιο διαδεδομένες προσεγγίσεις ανάπτυξης πληροφοριακών συστημάτων και μεθόδους χρονοπρογραμματισμού έργων. Τέλος, παρουσιάζεται μια μελέτη περίπτωσης πληροφοριακού συστήματος από τον τραπεζικό χώρο που αφορά στη διοίκηση του έργου ανάπτυξης ενός συστήματος διαχείρισης και αξιολόγησης προμηθευτών με χρήση της μεθοδολογίας PMBOK.

Abstract

This thesis presents the widely used project management methods, Prince2 and PMBOK and a comparison between them. Furthermore, the most used information systems' development life cycles and approaches are presented, as well as, methods of project scheduling. Finally, a case study concerning with the development of a banking information system, and more specifically, with the management of the project called "Supplier Management and Evaluation Platform" using the PMBOK methodology is presented.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα την επιβλέπουσα Επίκουρη Καθηγήτρια κυρία Μαλαματένιου Φλώρα που με τις συμβουλές και την καθοδήγηση της συνέβαλε στην περάτωση της εργασίας αυτής.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

ΚΕΦΑΛΑΙΟ 1: ΚΥΚΛΟΙ ΖΩΗΣ ΕΡΓΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ

1.1 Εισαγωγή

Με τον όρο μεθοδολογία ανάπτυξης πληροφοριακού συστήματος αναφερόμαστε στο πλαίσιο που χρησιμοποιείται για να δομηθεί, να σχεδιασθεί και να ελεγχθεί η διαδικασία της ανάπτυξης ενός πληροφοριακού συστήματος. Με τον όρο μεθοδολογία ανάπτυξης πληροφοριακού συστήματος αναφερόμαστε στο πλαίσιο που χρησιμοποιείται για να δομηθεί, να σχεδιασθεί και να ελεγχθεί η διαδικασία της ανάπτυξης ενός πληροφοριακού συστήματος. Μια μεγάλη ποικιλία τέτοιων πλαισίων έχει εξελιχθεί με την πάροδο των ετών. Κάθε μεθοδολογία έχει τα δικά της πλεονεκτήματα και μειονεκτήματα και χρησιμοποιείται ανάλογα με το είδος του έργου. Δεν είναι κατάλληλη η χρήση της ίδιας μεθοδολογίας σε όλα τα έργα. Η επιλογή της κατάλληλης μεθοδολογίας γίνεται βάση τεχνικών και οργανωτικών εκτιμήσεων.

Ο κύκλος ζωής ανάπτυξης πληροφοριακού συστήματος (System Development Life Cycle) είναι μια μεθοδολογία που χρησιμοποιείται για να περιγράψει τη διαδικασία για τη δημιουργία πληροφοριακών συστημάτων, η οποία προορίζεται για την ανάπτυξη πληροφοριακών συστημάτων με σκόπιμο, δομημένο και μεθοδικό τρόπο, επαναλαμβάνοντας κάθε στάδιο του κύκλου ζωής. Ο κύκλος ζωής ανάπτυξης πληροφοριακών συστημάτων, σύμφωνα με τους Έλιοτ & Στράχαν & Radford (2004), "ξεκίνησε στη δεκαετία του 1960, για την ανάπτυξη μεγάλης κλίμακας λειτουργικών επιχειρησιακών συστημάτων σε μια εποχή μεγάλων επιχειρηματικών ομίλων. Οι δράσεις που περιλάμβαναν τα πληροφοριακά συστήματα επικεντρώθηκαν γύρω από την επεξεργασία δεδομένων".

Αρκετά πλαίσια ανάπτυξης πληροφοριακών συστημάτων βασίζονται εν μέρει στον κύκλο ζωής ανάπτυξης πληροφοριακών συστημάτων, όπως η δομημένη ανάλυση και μέθοδος σχεδιασμού συστημάτων (Structured Systems Analysis and Design Method) που δημιουργήθηκε για την κυβέρνηση του Ηνωμένου Βασιλείου Γραφείο Κυβερνητικού Εμπορίου στη δεκαετία του 1980. Από τότε, σύμφωνα με το

Έλιοτ (2004), "οι παραδοσιακές προσεγγίσεις του κύκλου ζωής ανάπτυξης πληροφοριακών συστημάτων ολοένα και αντικαθίστανται με εναλλακτικές προσεγγίσεις και πλαίσια, που προσπάθησαν να ξεπεραστούν ορισμένες από τις εγγενείς αδυναμίες του παραδοσιακού κύκλος ζωή πληροφοριακών συστημάτων " [5].

1.2 Φάσεις ανάπτυξης πληροφοριακών συστημάτων

Ο κύκλος ζωής ανάπτυξης πληροφοριακών συστημάτων παρέχει μία σειρά από δραστηριότητες που οι σχεδιαστές συστημάτων και οι προγραμματιστές καλούνται να ακολουθήσουν. Αποτελείται από μια σειρά βημάτων ή φάσεων στα οποία κάθε φάση χρησιμοποιεί ως δεδομένα εισόδου τα αποτελέσματα της προηγούμενης.

Κάθε εφαρμογή λογισμικού διέρχεται από τέτοιες φάσεις, σε καθεμιά εκ των οποίων πρέπει να γίνονται ορισμένες εργασίες, ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα. Οι γενικές φάσεις είναι που διέρχεται μία εφαρμογή λογισμικού είναι [2]

- σύλληψη,
- κατασκευή,
- χρήση / συντήρηση και
- απόσυρση.



Εικόνα 1: Γενικές φάσεις του κύκλου ζωής του λογισμικού

Ένα μοντέλο κύκλου ζωής λογισμικού είναι μια περιγραφή των δραστηριοτήτων και των επιμέρους φάσεων από τις οποίες διέρχεται μια εφαρμογή λογισμικού, καθώς και των εργασιών που λαμβάνουν χώρα σε καθεμιά από τις φάσεις αυτές. Τα μοντέλα κύκλου ζωής λογισμικού προσδιορίζουν τις διαδικασίες ανάπτυξης οι οποίες λαμβάνουν χώρα κατά τις γενικές φάσεις «κατασκευή» και «χρήση / συντήρηση», προσδιορίζοντας τις επιμέρους φάσεις στις οποίες αυτές αναλύονται, τα προϊόντα που παράγονται σε καθεμιά από αυτές, καθώς και τη σειρά εκτέλεσής τους.

Σε κάθε διαδικασία ανάπτυξης μπορούμε να διακρίνουμε περισσότερες από μία επιμέρους φάσεις, ενώ σε κάθε επιμέρους φάση μπορούμε να διακρίνουμε περισσότερες από μία εργασίες. Οι διαδικασίες ανάπτυξης λογισμικού μπορούν να ταξινομηθούν ως ακολούθως:

- προδιαγραφή, δηλαδή καθορισμός των εργασιών που θα επιτελεί το λογισμικό, καθώς και των περιορισμών και των παραδοχών που ισχύουν.
- ανάπτυξη, δηλαδή κατασκευή του λογισμικού. Εδώ, σε όλα τα μοντέλα κύκλου ζωής μπορούμε να διακρίνουμε τρεις επιμέρους φάσεις: την ανάλυση, τη σχεδίαση και τη συγγραφή του πηγαίου κώδικα, την οποία στη συνέχεια θα ονομάζουμε και κωδικοποίηση.
- επαλήθευση, δηλαδή επιβεβαίωση της ικανοποίησης των προδιαγραφών και της μη ύπαρξης σφαλμάτων.
- εξέλιξη, δηλαδή επαύξηση των λειτουργικών χαρακτηριστικών του λογισμικού ή τροποποίηση υπαρχουσών, προκειμένου να ικανοποιούνται οι μεταβαλλόμενες ανάγκες.

Ένα μοντέλο κύκλου ζωής λογισμικού στοχεύει στην καθοδήγηση του κατασκευαστή προκειμένου αυτός να επιτύχει την καλύτερη δυνατή υλοποίηση των διαδικασιών ανάπτυξης λογισμικού. Τα παραπάνω μπορούν να διαφοροποιούνται ανάλογα με το μέγεθος και το θεματικό πεδίο κάθε εφαρμογής λογισμικού, με την εμπειρία και τα ιδιαίτερα χαρακτηριστικά του κάθε κατασκευαστή και, ασφαλώς, με το εκάστοτε περιβάλλον ανάπτυξης.

Μια σημαντική παράμετρος που καταδεικνύει τη σημασία των μοντέλων κύκλου ζωής είναι το κόστος. Το κόστος αναθεώρησης αποφάσεων ή/και διόρθωσης σφαλμάτων είναι τόσο μεγαλύτερο, όσο μεγαλύτερη είναι και η απαιτούμενη οπισθοδρόμηση της διαδικασίας που αυτή συνεπάγεται. Το κόστος αυτό δεν αφορά μόνο οικονομικούς πόρους που αποδίδονται στο έργο, αλλά και χρόνο καθυστέρησης, που δεν είναι πάντα διαθέσιμος σε πραγματικές συνθήκες. Επίσης, είναι συχνό φαινόμενο οι παρενέργειες στο υπόλοιπο σύστημα λογισμικού, οι οποίες μπορούν να μεταβάλλουν προς το χειρότερο τα ποιοτικά του χαρακτηριστικά και δεν είναι εύκολο να εντοπιστούν από την αρχή.

Υπάρχουν αρκετά μοντέλα κύκλου ζωής, τα οποία διαφοροποιούνται ως προς τη σύλληψη της ιδέας του τρόπου κατασκευής, αλλά και ως προς τις επιμέρους φάσεις που προτείνουν, την επαναληπτικότητα και την εμβέλεια των εργασιών αυτών, τα ενδιάμεσα προϊόντα συστατικά λογισμικού και την περιγραφή τους, τις οικονομικές και επιχειρηματικές πλευρές της χρήσης τους κ.ά. Καθεμία από τις ενέργειες που περιγράφεται σε ένα μοντέλο κύκλου ζωής είναι μια διαδικασία επίλυσης προβλημάτων.

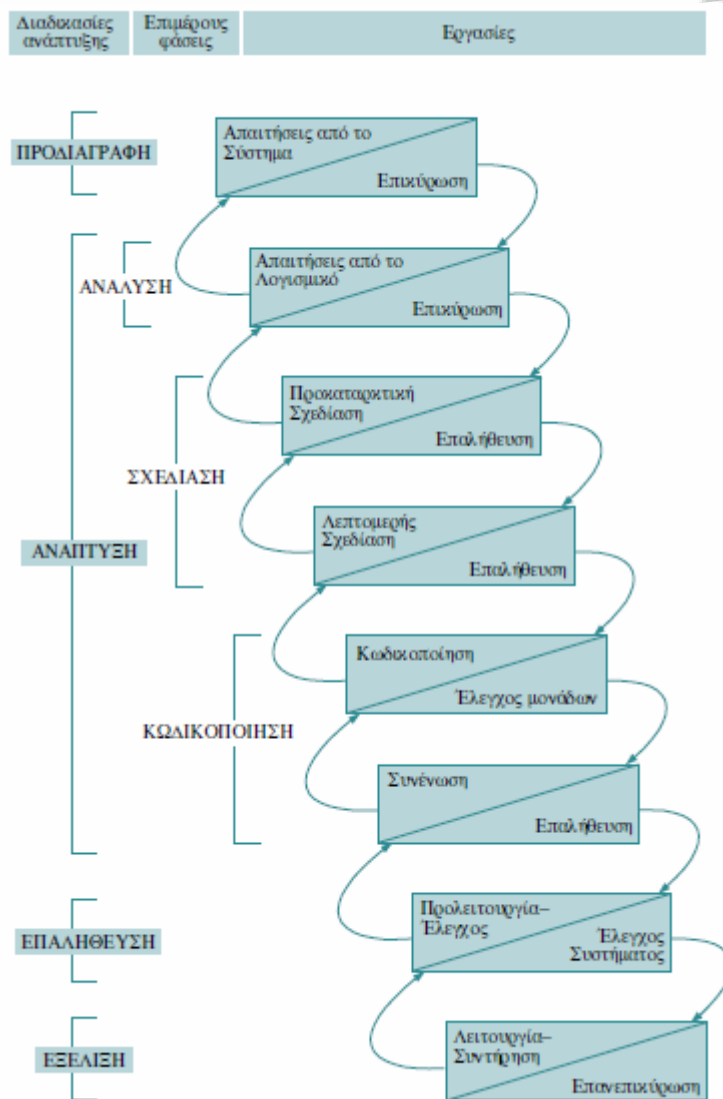
1.3 Το μοντέλο του καταρράκτη

Ένα από τα πιο διαδεδομένα μοντέλα κύκλου ζωής είναι αυτό του καταρράκτη (Waterfall), το οποίο φαίνεται στην παρακάτω εικόνα. Η κεντρική ιδέα του μοντέλου του καταρράκτη ^[7] είναι ότι η εφαρμογή αναπτύσσεται περνώντας ολόκληρη από διαδοχικές επιμέρους φάσεις, καθεμία από τις οποίες θεωρείται περατωμένη με την παραγωγή ορισμένων παραδοτέων. Κάθε επιμέρους φάση ολοκληρώνεται με μια εργασία επαλήθευσης / επικύρωσης των παραδοτέων της, κατά την οποία αποφασίζεται η μετάβαση ή όχι στην επόμενη φάση. Το λογισμικό εμφανίζεται πλήρες, δηλαδή με όλα τα λειτουργικά του χαρακτηριστικά, από την επιμέρους φάση της συνένωσης και μετά. Χαρακτηριστικό του μοντέλου του καταρράκτη είναι ότι, για να ξεκινήσει μια φάση πρέπει να έχει ολοκληρωθεί πλήρως η προηγούμενη φάση. Η ανάπτυξη με τον τρόπο αυτό χαρακτηρίζεται ακολουθιακή, διότι οι επιμέρους φάσεις από τις οποίες διέρχεται είναι διακριτές και ακολουθούν η μία την άλλη.

Αρχικά καθορίζονται οι απαιτήσεις από το λογισμικό. Ακολούθως γίνεται η προκαταρκτική και η λεπτομερής σχεδίαση του λογισμικού, αντίστοιχα. Κατά την προκαταρκτική σχεδίαση καθορίζονται οι μονάδες που θα αποτελούν το λογισμικό, καθώς και οι συσχετίσεις μεταξύ τους. Ο καθορισμός αυτός μπορεί να γίνει σε περισσότερα από ένα επίπεδα λεπτομέρειας, ανάλογα με το μέγεθος και την πολυπλοκότητα του λογισμικού.

Κατά τη λεπτότερη σχεδίαση καθορίζεται η εσωτερική δομή κάθε μονάδας λογισμικού, η οποία αντιστοιχεί πρακτικά σε μονάδες πηγαίου κώδικα προγράμματος. Ο καθορισμός αυτός περιλαμβάνει όλα τα απαραίτητα στοιχεία, ώστε η συγγραφή του πηγαίου κώδικα, που ακολουθεί, να είναι μια διαδικασία

διεκπεραίωσης και μόνο. Ακολουθεί η συνένωση των μονάδων και ο έλεγχος ολόκληρου του συστήματος, η ολοκλήρωση του οποίου επιτρέπει την παράδοση ολόκληρου του προϊόντος στον πελάτη και το πέρασμα στη φάση της λειτουργίας και της συντήρησης.



Εικόνα 2: Το μοντέλο του καταρράκτη

Το μοντέλο του καταρράκτη υπήρξε για μεγάλο διάστημα το πιο διαδεδομένο μοντέλο κύκλου ζωής λογισμικού. Είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπου οι απαιτήσεις από το λογισμικό είναι από την αρχή γνωστές και δε μεταβάλλονται κατά την ανάπτυξη του λογισμικού, και μπορεί να χρησιμοποιηθεί αποτελεσματικά για τη βιομηχανοποίηση της ανάπτυξης τέτοιων εφαρμογών. Για παράδειγμα, τέτοιες είναι

οι εφαρμογές που επιλύουν μεγάλα προβλήματα χρησιμοποιώντας μαθηματικούς υπολογισμούς. Σε πολλές, όμως, περιπτώσεις εφαρμογών οι απαιτήσεις είτε δεν είναι από την αρχή και με σαφήνεια γνωστές είτε ενδέχεται να μεταβληθούν κατά τη διάρκεια της ανάπτυξης.

Τα πλεονεκτήματα αυτής της μεθόδου είναι:

- ιδανικό για ομάδες και υπεύθυνους με μικρή σχετική εμπειρία.
- Η διαδοχική σειρά των σταδίων ανάπτυξης και ο αυστηρός έλεγχος για να διασφαλισθεί η επαρκής τεκμηρίωση διασφαλίζουν την ποιότητα την αξιοπιστία και την συντηρησιμότητα του αναπτυσσόμενου λογισμικού.
- Με αυτό το μοντέλο η πρόοδος της ανάπτυξης του συστήματος είναι μετρήσιμη
- Εξοικονομούνται πόροι.

Και τα μειονεκτήματα

- Άκαμπτη, αργή, δαπανηρή και περίπλοκη λόγω της μεγάλης δομής και των αυστηρών ελέγχων.
- Η υλοποίηση του έργου δεν περιλαμβάνει πολλά βήματα όπου να χρησιμοποιείται η ανάδραση.
- Δεν χρησιμοποιείται η επαναληπτικότητα που μπορεί να μειώσει τον διαχειριστικό κόστος
- Εξαρτάται από τον έγκαιρο εντοπισμό και προδιαγραφή των απαιτήσεων, αλλά οι χρήστες μπορεί να μην είναι σε θέση να ορίσουν με σαφήνεια τι χρειάζονται πριν ξεκινήσει το έργο.
- Νέες απαιτήσεις και νέα στοιχεία του συστήματος, που έχουν σαν αποτέλεσμα νέα ανάπτυξη ανακαλύπτονται κατά τον σχεδιασμό και την κωδικοποίηση
- Τα προβλήματα συχνά ανακαλύπτονται κατά τη φάση της δοκιμής
- Η απόδοση του συστήματος δεν μπορεί να ελεγχθεί μέχρι το σύστημα να είναι έτοιμο και αν υπάρχει πρόβλημα είναι δύσκολο να λυθεί όταν το σύστημα είναι στην παραγωγή.

- Δύσκολο να ανταποκριθεί στις αλλαγές. Αλλαγές που συμβαίνουν αργότερα στον κύκλο ζωής είναι πιο δαπανηρές και, συνεπώς δεν συνίστανται.
- Παράγει υπερβολική τεκμηρίωση και η ενημέρωσή της κατά την πρόοδο του έργου είναι χρονοβόρα.
- Προωθεί το χάσμα μεταξύ των χρηστών και προγραμματιστών, με σαφή διαχωρισμό των αρμοδιοτήτων.

1.4 Το μοντέλο πρωτυποποίησης

Η κεντρική ιδέα του μοντέλου πρωτυποποίησης^[30] είναι η ανάπτυξη του λογισμικού όχι ολόκληρου, αλλά σε τμήματα, που ονομάζονται πρωτότυπα. Οι διαδικασίες ανάπτυξης επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά και, για το λόγο αυτό, το μοντέλο χαρακτηρίζεται ως επαναληπτικό. Κάθε πρωτότυπο περιλαμβάνει τις βασικές από τις λειτουργίες που προορίζεται να εκτελεί το λογισμικό και τίθεται σε δοκιμή από τον πελάτη. Από εκεί συλλέγονται παρατηρήσεις και η διαδικασία κατασκευής νέου πρωτοτύπου επαναλαμβάνεται μέχρις ότου ένα πρωτότυπο να ικανοποιεί τις απαιτήσεις, δηλαδή να εκτελεί τις επιθυμητές λειτουργίες του λογισμικού με τρόπο ικανοποιητικό και να γίνεται αποδεκτό από τον πελάτη. Από το σημείο αυτό και μετά μπορούν να προστεθούν και οι υπόλοιπες λειτουργίες, ώστε το λογισμικό να ολοκληρωθεί.

Ένα σημαντικό πλεονέκτημα του μοντέλου αυτού είναι η δυνατότητα απόκτησης άποψης για την εφαρμογή λογισμικού νωρίτερα απ' ότι στο μοντέλο του καταρράκτη. Αυτό μπορεί να γλιτώσει την ανάπτυξη από καθυστερήσεις ή ακόμη και από ολική αποτυχία, τα οποία θα επέρχονταν, αν ο κατασκευαστής αναγκαζόταν να οπισθοδρομήσει την ανάπτυξη, ενώ αυτή είχε προχωρήσει πολύ. Παράλληλα, ιδιαίτερη σημασία αποκτά η διοίκηση του έργου, η οποία πρέπει να εξασφαλίζει την υλοποιησιμότητα του πρωτοτύπου και την εύκολη τροποποίησή του. Κάθε κατασκευή πρωτοτύπου μπορεί να θεωρηθεί ως ένα μικρό έργο λογισμικού το οποίο κατασκευάζεται με διαδικασίες που μπορούν να ακολουθούν άλλα μοντέλα κύκλου ζωής, όπως αυτό του καταρράκτη.

Με βάση τις παραπάνω παρατηρήσεις, το μοντέλο πρωτυποποίησης χρησιμοποιείται στην ανάπτυξη εφαρμογών λογισμικού όταν δεν υπάρχει βεβαιότητα στην αρχή της ανάπτυξης για τις απαιτήσεις του λογισμικού, οπότε δεν μπορούν να συμφωνηθούν και να παγιωποιηθούν.

Τέτοιες είναι εφαρμογές που κατασκευάζονται για πρώτη φορά ή που είναι στενά εξαρτημένες από τον πελάτη, χωρίς να υπάρχει αποδεκτό προηγούμενο παράδειγμα. Ωστόσο, το μέγεθος των εφαρμογών αυτών δε μπορεί να είναι ιδιαίτερα μεγάλο, διότι ο χρόνος ανάπτυξης κάθε πρωτοτύπου μεγαλώνει και η απαιτούμενη ευελιξία μειώνεται.

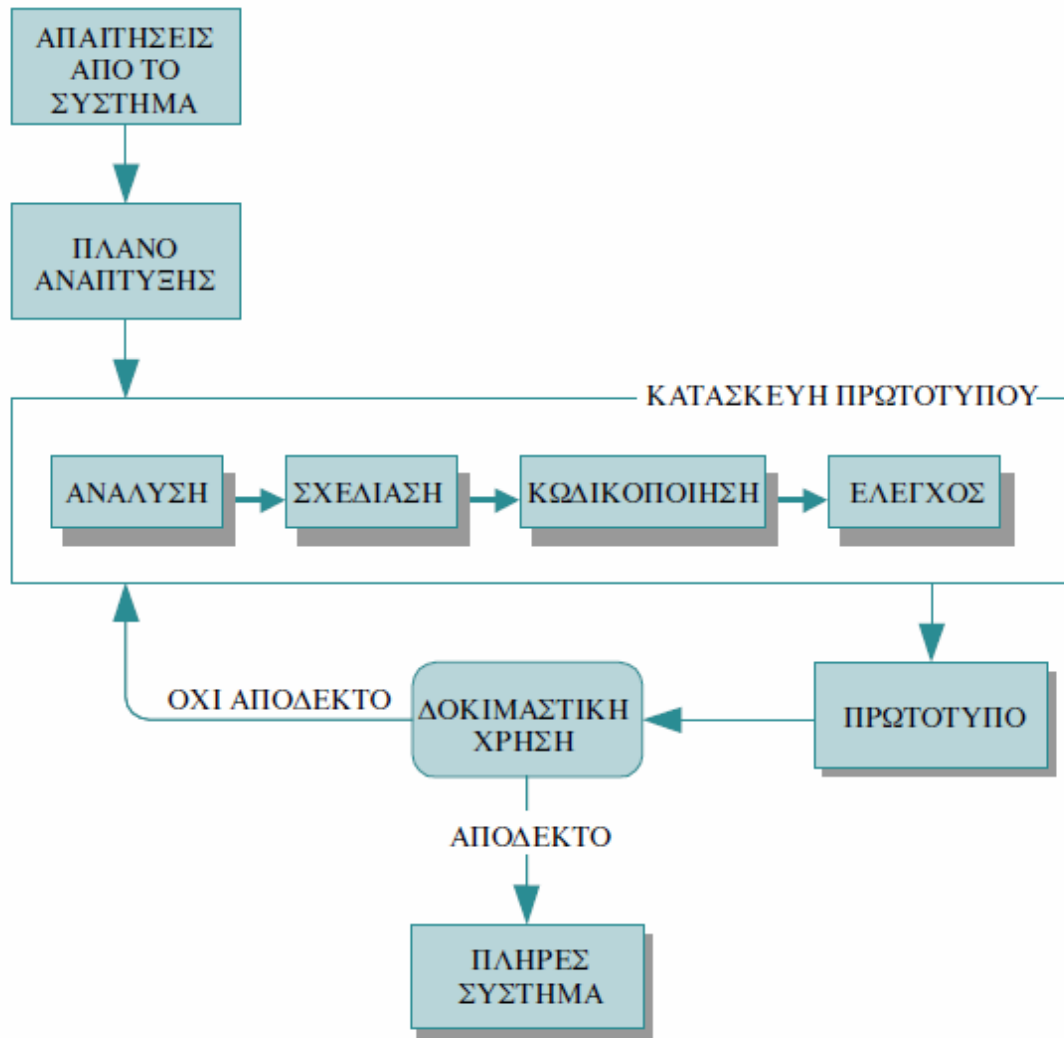
Τα πλεονεκτήματα αυτής της μεθόδου είναι:

- «Αντιμετωπίζει την ανικανότητα πολλών χρηστών να προσδιορίσουν τις πληροφορίες που χρειάζονται και την δυσκολία των αναλυτών συστημάτων να καταλάβουν το περιβάλλον των χρηστών παρέχοντας στους χρήστες ένα δοκιμαστικό σύστημα για διερευνητικούς λόγους όσον το δυνατόν νωρίτερα»^[40]
- “Μπορεί να χρησιμοποιηθεί για να παρουσιάσει ρεαλιστικά σημαντικές πτυχές του συστήματος σε κάθε βήμα τους παραδοσιακού κύκλου ζωής”^[41]
- Βελτιώνει τη συμμετοχή των χρηστών στην ανάπτυξη του συστήματος καθώς και την επικοινωνία μεταξύ των υπευθύνων του έργου
- Ιδιαίτερα χρήσιμη για να διευκρινιστούν ασαφής απαιτήσεις
- Βοηθά να εντοπιστούν μπερδεμένες ή δύσκολες λειτουργίες καθώς και ελλιπής λειτουργικότητα.
- Ενθαρρύνει την καινοτομία και τον ευέλικτο σχεδιασμό
- Παρέχει γρήγορη υλοποίηση μιας ημιτελούς αλλά λειτουργικής εφαρμογής.

Και τα μειονεκτήματα:

- Η διαδικασία έγκρισης και ελέγχου δεν είναι αυστηρή.

- Η ελλιπής ή ανεπαρκής ανάλυση του προβλήματος που είναι πιθανό να προκύψει θα οδηγήσει στην αντιμετώπιση μόνο των πιο προφανών και επιφανειακών αναγκών, με αποτέλεσμα αναποτελεσματικές μέθοδοι να ενσωματωθούν στο νέο σύστημα.
- Οι απαιτήσεις αλλάζουν συχνά και σε μεγάλο βαθμό
- Η ταυτοποίηση μη λειτουργικών στοιχείων είναι δύσκολο να καταγραφεί
- Οι σχεδιαστές μπορούν να πρωτοτυπήσουν πάρα πολύ γρήγορα, χωρίς την επαρκή εκ των προτέρων ανάλυση των αναγκών των χρηστών, με αποτέλεσμα σχεδίαση που περιορίζει τις μελλοντικές δυνατότητες ανάπτυξης του συστήματος
- Οι σχεδιαστές μπορούν να παραμελούν την δημιουργία τεκμηρίωσης, με αποτέλεσμα ανεπαρκή αιτιολόγηση για το τελικό προϊόν και ανεπαρκή στοιχεία για το μέλλον.
- Μπορεί να οδηγήσει σε κακόσχεδιασμένα συστήματα.
- Μπορεί να οδηγήσει σε ψευδείς προσδοκίες, όπου ο πελάτης πιστεύει λανθασμένα ότι το σύστημα «τελείωσε» όταν στην πραγματικότητα δεν έχει ολοκληρωθεί. Το σύστημα φαίνεται καλό και έχει επαρκείς διεπαφές χρήστη, αλλά δεν είναι πραγματικά λειτουργικό.
- Οι επαναλήψεις επιβαρύνουν τους προϋπολογισμούς και τα χρονοδιαγράμματα του έργου, έτσι το πρόσθετο κόστος πρέπει να σταθμιστεί έναντι των δυνητικών οφελών. Πολύ λίγα έργα μπορεί να είναι σε θέση να δικαιολογήσουν το επιπλέον χρόνο και χρήμα, ενώ μόνο τα υψηλού κινδύνου τμήματα των πολύ μεγάλων, πολύπλοκων έργων μπορούν να επωφεληθούν από τη χρήση πρωτοτύπων.



Εικόνα 3: Το μοντέλο πρωτυποποίησης

1.5 Το μοντέλο λειτουργικής επαύξησης

Το μοντέλο της λειτουργικής επαύξησης^[33] συνδυάζει την ακολουθιακή ανάπτυξη του μοντέλου του καταρράκτη με την τμηματική ανάπτυξη του μοντέλου της πρωτυποποίησης. Κεντρική ιδέα είναι η κατάτμηση του υπό κατασκευή λογισμικού σε τμήματα που αναπτύσσονται ανεξάρτητα, ακολουθώντας το καθένα ακολουθιακή ανάπτυξη σύμφωνα με το μοντέλο του καταρράκτη, όπως φαίνεται στην παρακάτω εικόνα. Κατά την αρχική φάση ανάλυσης και σχεδίασης αποφασίζονται τα τμήματα στα οποία θα κατατμηθεί η εφαρμογή, η ανάπτυξη των οποίων γίνεται στη συνέχεια ανεξάρτητα και παράλληλα. Όταν ολοκληρώνεται η ανάπτυξη κάθε τμήματος, αυτό ενσωματώνεται στο σύνολο της εφαρμογής.

Το μοντέλο της λειτουργικής επαύξησης χρησιμοποιείται στην ανάπτυξη μεγάλων εφαρμογών λογισμικού για τις οποίες ισχύουν οι απαιτήσεις του μοντέλου του καταρράκτη, δηλαδή σαφής γνώση και μικρή ή καθόλου μεταβλητότητα των απαιτήσεων κατά την ανάπτυξη.

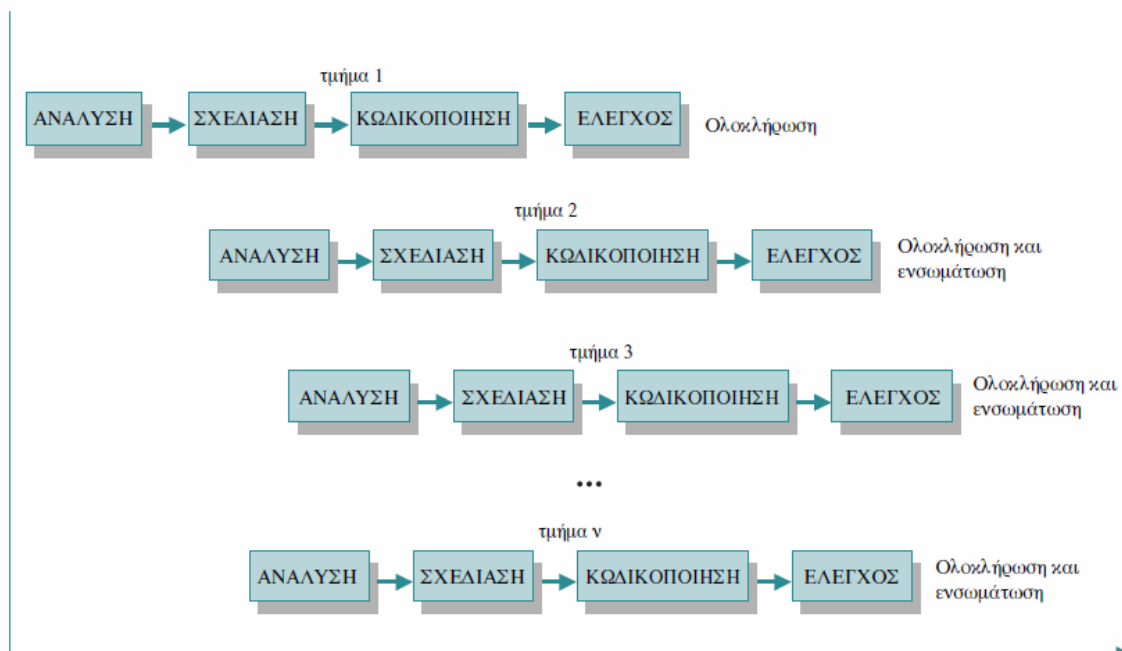
Τα πλεονεκτήματα αυτής της μεθόδου είναι:

- Καθώς εξελίσσεται το έργο υπάρχει η δυνατότητα για αξιοποίηση τεχνογνωσίας που έχει αποκτηθεί σε προηγούμενες φάσεις του έργου.
- Ένας μέτριος έλεγχος υπάρχει σε όλη τη διάρκεια του έργου με τη χρήση εγγράφων και την επίσημη έγκριση και αναθεώρηση από τον χρήστη και τον υπεύθυνο έργου σε συγκεκριμένα ορόσημα του έργου (milestones)
- Οι ενδιαφερόμενοι μπορούν να έχουν συγκεκριμένα στοιχεία για την κατάσταση του έργου καθ' όλη τη διάρκεια του κύκλου ζωής.
- Βοηθά ώστε να μετριαστούν οι σχεδιαστικοί κίνδυνοι καθώς και οι κίνδυνοι κατά την παράδοση του έργου.
- Επιτρέπει την παράδοση μιας σειράς υλοποιήσεων που είναι σταδιακά πιο πλήρης και μπορούν να μπουν στην παραγωγή πιο γρήγορα ως βελτιωμένες εκδόσεις.
- Η σταδιακή εφαρμογή παρέχει τη δυνατότητα να παρακολουθούν την επίδραση των σταδιακών αλλαγών, να απομονώνουν τα προβλήματα και να κάνουν τις αναγκαίες προσαρμογές πριν υπάρξει πρόβλημα
- η δυνατότητα παράλληλης ανάπτυξης τμημάτων του λογισμικού, η οποία τελικά καταλαμβάνει μικρότερο χρόνο

Και τα μειονεκτήματα:

- Κατά τη χρήση μια σειρά από μίνι-Καταρράκτες για ένα μικρό μέρος του συστήματος πριν από τη μετάβαση στην επόμενη αύξηση, υπάρχει συνήθως η έλλειψη συνολικής εξέτασης του επιχειρηματικού προβλήματος και των τεχνικών απαιτήσεων για το συνολικό σύστημα.
- Δεδομένου ότι ορισμένες ενότητες θα ολοκληρωθούν πολύ νωρίτερα από τις άλλες απαιτούνται σαφώς καθορισμένες διεπαφές.

- Δύσκολα σημεία του έργου μετατίθενται στο μέλλον ώστε να παρουσιαστεί στη διοίκηση μια ωραιοποιημένη εικόνα.
- Η αρχική κατάτμηση και γενική σχεδίαση του συστήματος αποκτά ιδιαίτερη βαρύτητα. Σφάλματα σε αυτή μπορούν να έχουν σημαντικές επιπτώσεις στο λογισμικό που θα κατασκευαστεί στη συνέχεια.
- Σε περίπτωση μεταβολής των λειτουργικών απαιτήσεων κατά τη χρήση του ημιτελούς συστήματος, μπορεί η αρχιτεκτονική αυτού να μεταβληθεί σε βαθμό που να κλονιστεί η ανάπτυξη των υπόλοιπων τμημάτων αυτού.



Εικόνα 4: Το μοντέλο της λειτουργικής επαύξεσης

1.6 Το σπειροειδές μοντέλο

Τα μοντέλα κύκλου ζωής που παρουσιάστηκαν μέχρι τώρα αποτελούν παραλλαγές της βασικής ιδέας του μοντέλου του καταρράκτη. Η ανάπτυξη παραμένει, επί της ουσίας, μια ακολουθιακή διαδικασία, η οποία εφαρμόζεται είτε σε ολόκληρο είτε σε ένα μέρος του συστήματος. Στο μοντέλο της προτυποποίησης, καθώς και σε αυτό της λειτουργικής επαύξεσης, η κατάτμηση είναι λίγο ως πολύ αυθαίρετη. Το ρίσκο δεν αποτιμάται, με αποτέλεσμα κάθε οπισθοδρόμηση ή

ανατροπή να κοστίζει σε χρόνο και σε οικονομικούς όρους, συχνά δε σε συνολική αποτυχία των έργων.

Από την άλλη, η αποδοχή των αυστηρών φάσεων που προτείνονται από το μοντέλο του καταρράκτη δεν είναι εφικτό να ακολουθείται σε όλες τις περιπτώσεις και από όλους τους κατασκευαστές, με αποτέλεσμα η ανάπτυξη λογισμικού είτε να γίνεται άναρχα, με βάση τη διαίσθηση των κατασκευαστών, είτε να είναι μια δαπανηρή και στρυφνή διαδικασία, στην οποία πρέπει να ακολουθηθούν κάποια συγκεκριμένα βήματα, ανεξάρτητα από τις εκάστοτε συνθήκες.

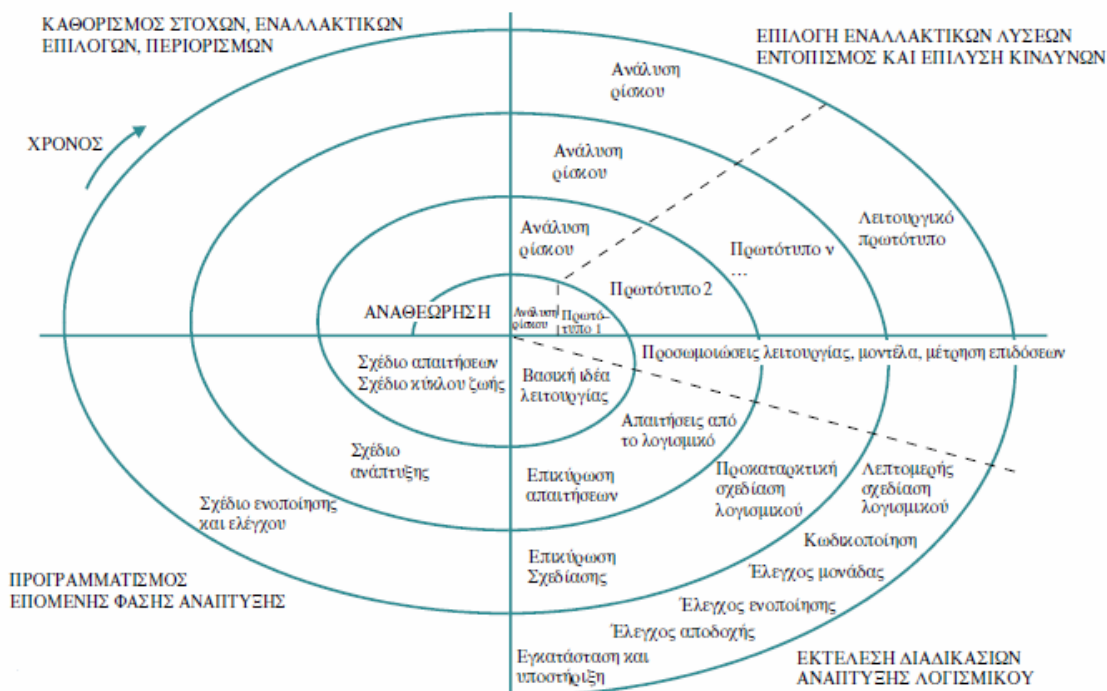
Λύση στα παραπάνω προβλήματα έρχεται να δώσει το σπειροειδές μοντέλο^[7] κύκλου ζωής, το οποίο πήρε το όνομά του από την απεικόνιση σε διάγραμμα, όπως φαίνεται στην παρακάτω εικόνα. Πρόκειται για μια γενίκευση των μοντέλων της λειτουργικής επαύξησης και της πρωτοτυποποίησης, με σημαντικά νέα στοιχεία:

- Οι φάσεις και οι διαδικασίες ανάπτυξης λογισμικού δεν είναι προκαθορισμένες από το μοντέλο, αλλά εξειδικεύονται στο χώρο της εφαρμογής του.
- Η ανάπτυξη ολόκληρου του συστήματος χωρίζεται σε πολλούς κύκλους, σε καθέναν από τους οποίους προστίθενται νέα λειτουργικά χαρακτηριστικά στο σύστημα.
- Πριν από την έναρξη κάθε κύκλου γίνεται μια μελέτη σκοπιμότητας και ανάλυση κινδύνων, από την οποία προκύπτουν, αφενός, οι συγκεκριμένες εργασίες που θα εκτελεστούν μέσα στον κύκλο, αφετέρου, η ίδια η εφικτότητα εκτέλεσης του κύκλου αυτού.

Όπως φαίνεται στην παραπάνω εικόνα, στο σπειροειδές μοντέλο διακρίνονται τέσσερις κατηγορίες εργασιών: προσδιορισμός στόχων, εντοπισμός και επίλυση κινδύνων, εκτέλεση διαδικασιών ανάπτυξης και επαλήθευση, καθώς και εργασίες προγραμματισμού.

- Κατά τον προσδιορισμό στόχων καθορίζονται τα αντικείμενα εργασιών κάθε επανάληψης, καταγράφονται οι περιορισμοί επί του προϊόντος, αλλά και επί της διαδικασίας για την οποία κατασκευάζεται ένα αναλυτικό πλάνο διοίκησης. Επίσης, καταγράφονται οι κίνδυνοι που εμπεριέχει η διαδικασία και οι εναλλακτικές λύσεις, όπου υπάρχουν.

- Κατά τις εργασίες επίλυσης κινδύνων αναλύονται οι κίνδυνοι που έχουν καταγραφεί και αποτιμάται κάθε εναλλακτική λύση. Στο σημείο αυτό λαμβάνονται αποφάσεις για τη συνέχιση ή όχι της ανάπτυξης, για το μοντέλο που θα ακολουθηθεί στη συγκεκριμένη επανάληψη, για την κατασκευή ή όχι πρωτοτύπου κ.ά.



Εικόνα 5: Το σπειροειδές μοντέλο κύκλου ζωής λογισμικού

- Ακολουθεί η εκτέλεση των βημάτων της διαδικασίας ανάπτυξης λογισμικού που έχει επιλεγεί για το τμήμα εκείνο του συστήματος που αφορά η τρέχουσα επανάληψη.
- Τέλος, μετά την επαλήθευση των αποτελεσμάτων, ενδιάμεσων προϊόντων λογισμικού γίνεται προγραμματισμός της συνέχισης της ανάπτυξης.

Το σπειροειδές μοντέλο δεν καθορίζει εκ των προτέρων ποιες ακριβώς είναι οι εργασίες ανάπτυξης λογισμικού που πρέπει να γίνουν ούτε σε ποια έκταση του συστήματος αυτές θα εφαρμοστούν. Διαφορετικές διαδικασίες ανάπτυξης μπορεί να επιλεγούν για διαφορετικά τμήματα του λογισμικού. Αυτό που προτείνει είναι ότι ο καθορισμός των λεπτομερειών υλοποίησης πρέπει να γίνεται συνεχώς κατά την

ανάπτυξη (και όχι μία φορά, όπως συμβαίνει με τα μοντέλα κύκλου ζωής που αναφέρθηκαν μέχρι τώρα) με ευθύνη και με τεκμηρίωση από πλευράς του ίδιου του κατασκευαστή.

Η εφαρμογή του σπειροειδούς μοντέλου στην πράξη δεν είναι πάντα εύκολη υπόθεση. Εισάγονται νέες εργασίες, που δεν ανήκουν καθαρά στις εργασίες ανάπτυξης λογισμικού, αλλά αφορούν την τεκμηρίωση της σκοπιμότητας και τον τμηματικό προγραμματισμό της ανάπτυξης. Οι εργασίες αυτές επιφέρουν ασφαλώς κάποιο κόστος, το οποίο όμως μπορεί να αποσβεστεί από τον έγκαιρο εντοπισμό προβλημάτων και την αποφυγή πιθανού ναυαγίου, κάτι που έχει συμβεί σε αρκετές περιπτώσεις.

Τα πλεονεκτήματα αυτής της μεθόδου είναι:

- Βελτιώνει την αποφυγή του κινδύνου
- Βοηθάει να επιλέξουμε την καλύτερη μέθοδο για την ανάπτυξη ενός συγκεκριμένου μέρους του λογισμικού, με βάση τον κίνδυνο του έργου.
- Μπορεί να ενσωματώσει τα μοντέλα του καταρράκτη, της προτυποποίησης και επαναληπτικών μεθοδολογιών ως ειδικές περιπτώσεις στο πλαίσιο, και να παρέχει καθοδήγηση ως προς το ποιος συνδυασμός αυτών των μοντέλων ταιριάζει για μια συγκεκριμένη επανάληψη του λογισμικού, με βάση το είδος του κινδύνου του έργου. Για παράδειγμα, ένα έργο με χαμηλό κίνδυνο στο να μην πληρήν τις απαιτήσεις των χρηστών αλλά υψηλού κινδύνου στην τήρηση του προϋπολογισμού και του χρονοδιαγράμματος θα ακολουθήσει ουσιαστικά μια γραμμική προσέγγιση καταρράκτη για μια δεδομένη επανάληψη λογισμικού. Αντίθετα, αν οι παράγοντες κινδύνου αναστραφούν, η σπειροειδής μεθοδολογία θα μπορούσε να αποφέρει μια επαναληπτική προσέγγιση.

Και τα μειονεκτήματα:

- Είναι δύσκολο να καθορισθεί η ακριβής σύνθεσή των μεθοδολογιών ανάπτυξης που θα χρησιμοποιηθούν για κάθε επανάληψη γύρω από το σπειροειδές μοντέλο.

- Εξαιρετικά προσαρμοσμένη σε κάθε έργο, και έτσι είναι αρκετά περίπλοκη, περιορίζοντας τις δυνατότητες επαναχρησιμοποίησης.
- Ένα εξειδικευμένος και έμπειρο υπεύθυνος έργου απαιτείται για να καθορίσει πώς να την εφαρμόσει σε ένα έργο.
- Δεν υπάρχουν έλεγχοι για τη μετάβαση από έναν κύκλο ζωής σε έναν άλλο κύκλο. Χωρίς ελέγχους, κάθε κύκλος μπορεί να δημιουργήσει περισσότερη δουλειά για τον επόμενο κύκλο.
- Δεν υπάρχουν αυστηρές προθεσμίες. Οι κύκλοι συνεχίζονται χωρίς σαφή τερματισμό, οπότε ελλοχεύει ο κίνδυνος να ξεφύγει ο προϋπολογισμός και το χρονοδιάγραμμα του έργου.
- Υπάρχει πιθανότητα ότι το έργο θα καταλήξει να υλοποιηθεί με το μοντέλο του καταρράκτη.

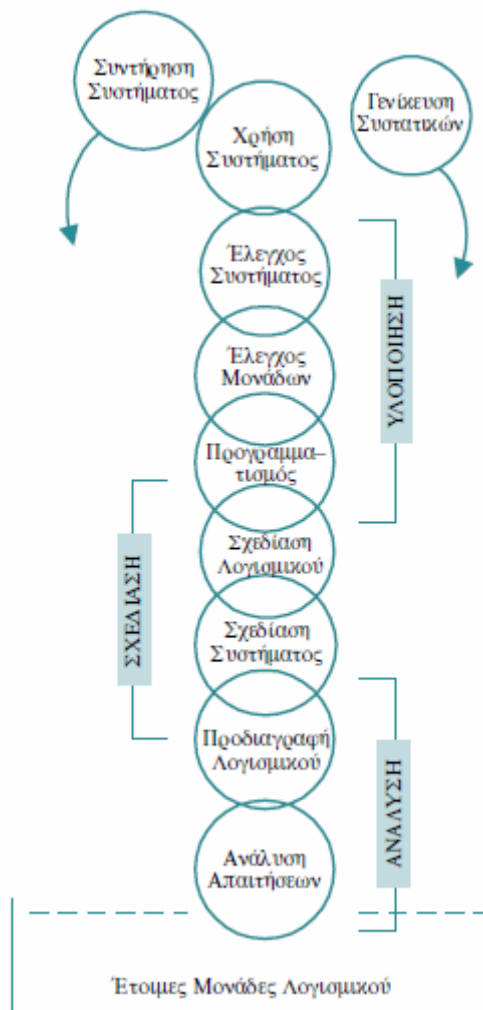
1.7 Το μοντέλο του πίδακα

Αρκετά μοντέλα κύκλου ζωής που έχουν προταθεί αποτελούν παραλλαγές αυτών που αναφέρθηκαν, τα χαρακτηριστικά των οποίων υποβάλλονται από τις μεθοδολογίες ανάπτυξης. Οι πρώτες προσεγγίσεις του θέματος με βάση την αντικειμενοστραφή τεχνολογία διαφοροποίησαν το παραπάνω σχήμα βασιζόμενες σε δύο ιδιαίτερα γνωρίσματά της: πρώτον, ότι οι έννοιες ανάλυση, σχεδίαση και κωδικοποίηση έρχονται στο αντικειμενοστραφές παράδειγμα πολύ πιο κοντά και, δεύτερον, ότι το αποτέλεσμα κάθε διαδικασίας κατασκευής λογισμικού είναι όχι μόνο ένα σύστημα, αλλά και επαναχρησιμοποιήσιμες μονάδες, οι οποίες μπορούν να χρησιμοποιηθούν από τις πρώτες φάσεις της ανάπτυξης μελλοντικών συστημάτων. Με τον τρόπο αυτό προέκυψε το μοντέλο του πίδακα^[2], που φαίνεται στην παρακάτω εικόνα.

Κατά την ανάπτυξη παρατηρούνται επικαλύψεις των φάσεων ανάλυση, σχεδίαση και κωδικοποίηση, οι οποίες φαίνονται με την επικάλυψη των κύκλων στην εικόνα. Κατά το τέλος της ανάπτυξης, ορισμένα από τα συστατικά λογισμικού που

έχουν παραχθεί ενσωματώνονται σε μια δεξαμενή συστατικών και διατίθενται για να χρησιμοποιηθούν στην ανάπτυξη και νέων συστημάτων.

Η ιδέα του μοντέλου κύκλου ζωής του πίδακα τονίζει περισσότερο τα επιθυμητά χαρακτηριστικά της μεθοδολογίας κατασκευής του λογισμικού σύμφωνα με την αντικειμενοστραφή λογική, ήταν δε αρκετά επίκαιρη κατά την έκρηξη ενδιαφέροντος για την αντικειμενοστραφή τεχνολογία στα τέλη της δεκαετίας του '80 και στις αρχές της δεκαετίας του '90.

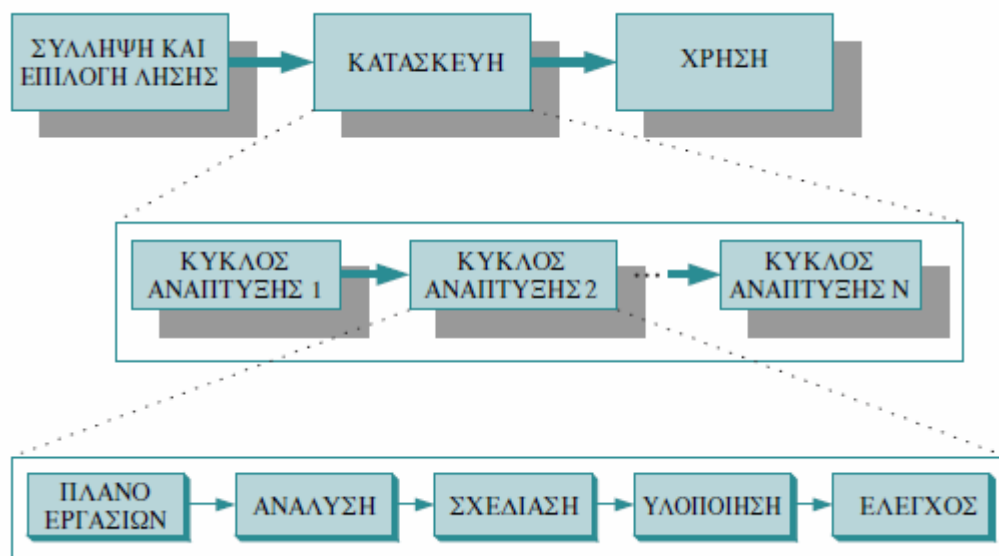


Εικόνα 6: Το μοντέλο κύκλου ζωής του πίδακα

1.8 Σύγχρονα μοντέλα κύκλου ζωής λογισμικού

Μεταγενέστερα μοντέλα κύκλου ζωής λογισμικού προσπαθούν να δώσουν μια γενική κατεύθυνση εφαρμογής των υπαρχουσών ιδεών, αφήνοντας σημαντικούς βαθμούς ελευθερίας στον κατασκευαστή που τα ακολουθεί. Αυτό είναι ιδιαίτερα επιθυμητό, διότι η αυστηρή πειθαρχία που επιχειρήθηκε να εισαχθεί τα πρώτα χρόνια της έκρηξης της χρήσης του λογισμικού δε συμβάδιζε με την ωριμότητα σκέψης που διέθετε η τεχνική κοινότητα την εποχή εκείνη ούτε και μπορούσε να παρακολουθήσει τους υψηλούς ρυθμούς εξελίξεων στο χώρο της πληροφορικής.

Η πειθαρχία αυτή τελικά δεν οδήγησε στην κατασκευή λογισμικού αναμενόμενης ποιότητας. Μια περιγραφή ενός σύγχρονου μοντέλου κύκλου ζωής λογισμικού περιέχει μόνο γενικές κατευθύνσεις, οι οποίες εξειδικεύονται στο εκάστοτε περιβάλλον ανάπτυξης. Επίσης, δεν είναι άρρηκτα συνδεδεμένο με κάποια μεθοδολογία ανάπτυξης λογισμικού, αλλά μπορεί να εξειδικευτεί για την πρακτική του κάθε κατασκευαστή. Ένα τέτοιο μοντέλο φαίνεται στην παρακάτω εικόνα και μπορεί να χαρακτηριστεί ως απόγονος πολλών από τα μοντέλα που προαναφέρθηκαν.



Εικόνα 7: Ένα γενικό μοντέλο κύκλου ζωής το οποίο ενσωματώνει χαρακτηριστικά πολλών από τα μοντέλα που αναφέρθηκαν

Το γενικό πλαίσιο του μοντέλου αυτού περιλαμβάνει τις φάσεις σύλληψης, κατασκευής και λειτουργίας. Καθεμιά από αυτές αναλύεται σε επιμέρους εργασίες, σύμφωνα με τα χαρακτηριστικά του εκάστοτε περιβάλλοντος. Ιδιαίτερα η γενική φάση της κατασκευής αναλύεται σε κύκλους ανάπτυξης, καθένας εκ των οποίων προσθέτει νέα χαρακτηριστικά και λειτουργίες στο υπό κατασκευή λογισμικό.

Τα επιμέρους βήματα μέσα σε κάθε κύκλο ανάπτυξης μοιάζουν με τα βήματα του μοντέλου του καταρράκτη, μόνο που δεν εφαρμόζονται για ολόκληρο το σύστημα, αλλά για το μικρό μέρος του που κατασκευάζεται στον εν λόγω κύκλο, όπως στο μοντέλο της πρωτοτυποποίησης. Για την εκκίνηση κάθε κύκλου ανάπτυξης μπορεί να έχει προηγηθεί ανάλυση ρίσκου και σκοπιμότητας, όπως στο σπειροειδές μοντέλο. Ζητήματα όπως αλληλουχία των ενεργειών, ακριβής καθορισμός των κύκλων ανάπτυξης κ.ά. αφήνονται στη διακριτική ευχέρεια του κάθε κατασκευαστή, από τον οποίο και καθορίζονται σύμφωνα με τις ιδιαιτερότητες κάθε περίπτωσης.

ΚΕΦΑΛΑΙΟ 2: ΠΡΟΣΕΓΓΙΣΕΙΣ ΑΝΑΠΤΥΞΗΣ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

2.1 Εισαγωγή

Η προσέγγιση – μεθοδολογία ανάπτυξης πληροφοριακών συστημάτων είναι ένα οργανωμένο σύνολο διαδικασιών, τεχνικών, εργαλείων και μέσων τεκμηρίωσης που καθοδηγεί τις ενέργειες σε ένα χώρο γνώσης. Περιγράφεται από φάσεις, στάδια, βήματα και δραστηριότητες. Σε αυτό το κεφάλαιο επικεντρωνόμαστε κυρίως στις νέες ευέλικτες προσεγγίσεις ανάπτυξης πληροφοριακών συστημάτων.

Η απογοήτευση από τη χρήση μεθόδων που απαιτούσαν την προσοχή των μηχανικών λογισμικών στην τεκμηρίωση του συστήματος που σχεδιάζοταν, οδήγησε στην δεκαετία του 90' ένα μεγάλο μέρος μηχανικών λογισμικού να προτείνουν νέες και πιο ευέλικτες μεθόδους. Αυτές οι μέθοδοι επιτρέπουν στην ομάδα ανάπτυξης να επικεντρωθεί στη σχεδίαση του λογισμικού. Οι μέθοδοι αυτές ονομάστηκαν ευέλικτες μέθοδοι (agile methods) και βασίζονται εξολοκλήρου σε ένα πλαίσιο, όπου οι διαδικασίες του προσδιορισμού των προδιαγραφών του λογισμικού, της ανάπτυξης και της παράδοσής του, επαναλαμβάνονται.

2.2 Ευέλικτες Μέθοδοι Ανάπτυξης (Agile Software Development)

Ευελιξία στον προγραμματισμό είναι η ικανότητα της προσαρμογής και του επαναπροσδιορισμού ενός αναπτυσσόμενου και συνεχώς εξελισσόμενου συστήματος στην περίπτωση που εμφανίζονται αλλαγές στις αρχικές θεωρήσεις και παραδοχές. Οι οργανισμοί που χρησιμοποιούν ευέλικτες μεθόδους βλέπουν την αλλαγή σαν ευκαιρία βελτίωσης και προόδου και όχι σαν απειλή.

Οι ευέλικτες μέθοδοι είναι ^[23] :

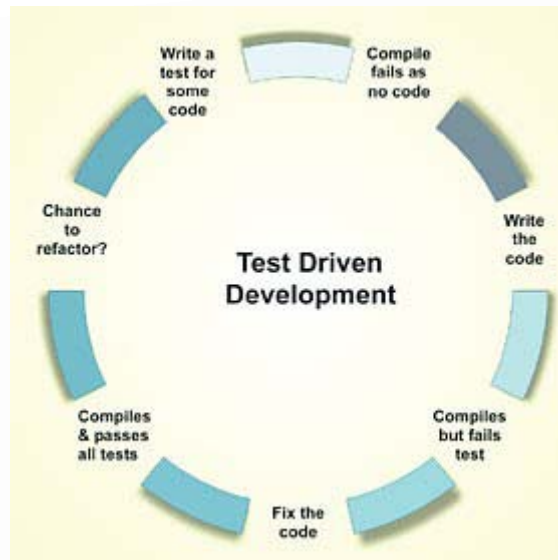
- Επαναληπτικές (iterative): Αρχικά παραδίδεται ένα πλήρες σύστημα και στη συνέχεια γίνονται αλλαγές στη λειτουργία κάθε υποσυστήματος σε κάθε νέα έκδοση.

- Αυξητικές (incremental): Το σύστημα διαιρείται σε υποσυστήματα με βάση τη λειτουργία τους. Νέες λειτουργίες προστίθενται σε κάθε νέα έκδοση.
- Αυτό-διοργανούμενες (self-organising): Η ομάδα έχει την αυτονομία να οργανωθεί κατά βούληση.
- Προκύπτουσες (emergent): Οι απαιτήσεις και η τεχνολογία που θα χρησιμοποιηθεί προκύπτουν κατά τη διάρκεια του κύκλου ανάπτυξης.

Οι ευέλικτες μέθοδοι εμφανίστηκαν τα τελευταία χρόνια ως μία εναλλακτική κατεύθυνση στην τεχνολογία λογισμικού^[14]. Μεταξύ των ευέλικτων μεθόδων είναι και οι eXtreme Programming^[13] και Rational Unified Process^[21] που θα μελετηθούν αναλυτικά στις επόμενες υπό-ενότητες.

Οι διαδικασίες αυτές αρχικά σχεδιάστηκαν για την υποστήριξη της ανάπτυξης επιχειρηματικών λογισμικών, όπου οι απαιτήσεις του συστήματος αλλάζουν συχνά κατά τη διάρκεια της ανάπτυξης. Η μεθοδολογία αυτή δημιουργήθηκε από την προσπάθεια να ξεπεραστούν οι αδυναμίες των παραδοσιακών προσεγγίσεων στις περιπτώσεις που οι απαιτήσεις του έργου αλλάζουν διαρκώς και με ταχύτητα. Κάτι τέτοιο σίγουρα συμβαίνει σε αρκετές κατηγορίες σύγχρονων προϊόντων λογισμικού. Καταρχήν, η μεθοδολογία αυτή δεν δίνει τόσο έμφαση στην ολοκλήρωση του προϊόντος όσο στην συνεχή εξέλιξη του με την προσθήκη καινούργιων χαρακτηριστικών και την βελτίωσή του σε διαδοχικές εκδόσεις.

Σε σύγκριση με παρόμοιες μεθοδολογίες που επίσης προτείνουν την επαναληπτική εξέλιξη του έργου μέσα από διαδοχικές εκδόσεις, η agile φιλοσοφία διαφέρει στο ότι προωθεί την πραγματικά πολύ γρήγορη δημιουργία διαδοχικών εκδόσεων, της τάξης των μερικών εβδομάδων αντί για αρκετών μηνών. Οι φάσεις της διαχείρισης έργου με μια ευέλικτη μεθοδολογία φαίνονται διαγραμματικά στην παρακάτω εικόνα.



Εικόνα 8: Οι φάσεις διαχείρισης έργου με μια agile μεθοδολογία

Πιθανότατα η πιο γνωστή ευέλικτη μέθοδος είναι ο ακραίος προγραμματισμός (extreme programming), ο οποίος περιγράφεται παρακάτω. Ωστόσο, υπάρχουν και άλλες μέθοδοι όπως οι ^[8]:

- Scrum,
- Crystal,
- Adaptive Software Development,
- DSDM,
- Feature Driven Development,
- Iconix,
- Dynamic System Development Methodology (DSDM) και
- Lean Development

Η επιτυχία αυτών των μεθόδων οδήγησε στην ένταξή τους σε πιο παραδοσιακές μεθόδους ανάπτυξης λογισμικού που βασίζονται στην μοντελοποίηση του συστήματος. Παρόλο που αυτές οι μέθοδοι βασίζονται στην ιδέα της αυξανόμενης ανάπτυξης και παράδοσης λογισμικού, προτείνουν διαφορετικές διεργασίες για να το πετύχουν. Ωστόσο, μοιράζονται ένα κοινό σύνολο αρχών και για αυτό έχουν πολλά κοινά σημεία. Αυτές οι αρχές φαίνονται σε παρακάτω πίνακα:

Πίνακας 1: Χαρακτηριστικά μεθοδολογίας Agile Software Development

Αρχή	Περιγραφή
Ανάμειξη του πελάτη	Οι πελάτες θα πρέπει να είναι άμεσα αναμειγμένοι με την διεργασία της ανάπτυξης λογισμικού. Ο ρόλος τους είναι να παρέχουν νέες απαιτήσεις, να τους δίνουν προτεραιότητες και να αξιολογούν τις επαναληπτικές εκδόσεις του συστήματος.
Αυξανόμενη παράδοση	Το λογισμικό αναπτύσσεται σε μέρη και οι πελάτες ορίζουν τις απαιτήσεις που θα περιληφθούν σε κάθε μέρος.
Άνθρωποι, όχι διεργασίες	Οι ικανότητες της ομάδας ανάπτυξης θα πρέπει να αναγνωρίζονται και να εκμεταλλεύονται. Τα μέλη της ομάδας θα πρέπει να έχουν την δυνατότητα να εργάζονται με τον δικό τους τρόπο και όχι στα πλαίσια μιας ορισμένης διεργασίας.
Αποδοχή αλλαγών	Θα πρέπει να αναμένονται αλλαγές στις απαιτήσεις του συστήματος, για αυτό το λόγο ο σύστημα πρέπει να σχεδιαστεί ώστε να δέχεται τέτοιες αλλαγές.
Διατήρηση απλότητας	Θα πρέπει να γίνεται εστίαση στην απλότητα τόσο του λογισμικού που αναπτύσσεται όσο και της διαδικασίας της ανάπτυξης. Όποτε είναι δυνατό, εκτελούνται εργασίες για την εξάλειψη πολυπλοκότητας από το σύστημα.

Επιπλέον, όλες οι ευέλικτες μέθοδοι βασίζονται σε 12 αρχές που περιγράφονται στη διακηρυκτική συνεδρίαση για της ευέλικτες μεθόδους και είναι οι εξής ^[14]:

1. Ικανοποίηση του πελάτη.
2. Συχνή παράδοση λογισμικού.
3. Η αλλαγή είναι ευπρόσδεκτη.
4. Καθημερινή συνεργασία με τον πελάτη.
5. Ικανό προσωπικό και περιβάλλον εμπιστοσύνης στην ομάδα.
6. Διαπροσωπική συζήτηση για την ανταλλαγή πληροφοριών.
7. Σωστή λειτουργία του λογισμικού που κατασκευάζεται.
8. Εξασφάλιση σταθερού ρυθμού ανάπτυξης.
9. Τεχνική αρτιότητας και καλός σχεδιασμός.

10. Υλοποίηση στόχων με σύντομο και αποτελεσματικό τρόπο.
11. Αυτό-διοργανούμενες ομάδες.
12. Επαναπροσδιορισμός της συμπεριφοράς της ομάδας.

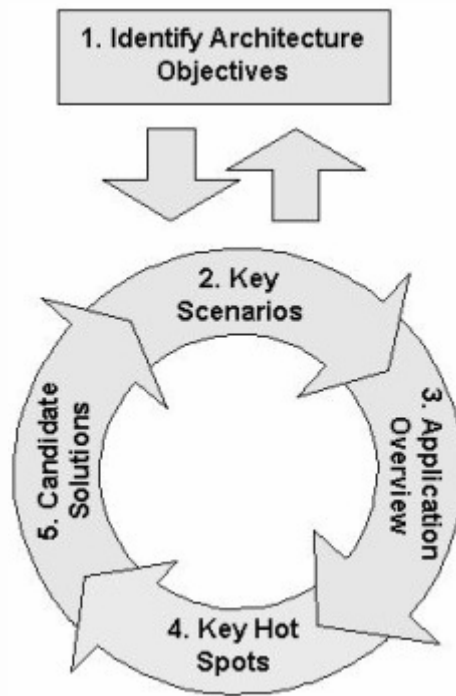
Συνοπτικά τα σημεία στα οποία οι ευέλικτες μέθοδοι δίνουν αξία είναι τα εξής:

- Άτομα και αλληλεπιδράσεις αντί διαδικασίες και εργαλεία.
- Δυναμικός κώδικας αντί γραπτής τεκμηρίωσης.
- Συνεργασία με τον πελάτη αντί αυστηρών συμβολαίων.
- Ανταπόκριση σε αλλαγές αντί ακολουθούμενου σχεδίου.

Στην πραγματικότητα, οι αρχές στις οποίες βασίζονται οι ευέλικτες μέθοδοι είναι πολλές φορές δύσκολο να πραγματοποιηθούν ^[9]:

1. Παρόλο που η ιδέα της ανάμειξης του πελάτη στη διαδικασία ανάπτυξης του λογισμικού είναι ελκυστική, η επιτυχία της εξαρτάται από την επιθυμία και τη δυνατότητα του πελάτη να διαθέσει χρόνο στην ομάδα ανάπτυξης. Συχνά, οι πελάτες έχουν άλλες υποχρεώσεις και δεν μπορούν να έχουν πλήρη συμμετοχή στην ανάπτυξη του λογισμικού.
2. Κάποια άτομα, μέλη μιας ομάδας, μπορεί να μην έχουν τον κατάλληλο χαρακτήρα και προσωπικότητα για να συμμετέχουν εντατικά σε μία ομάδα, κάτι που είναι τυπικό στις ευέλικτες μεθόδους. Για αυτό το λόγο, μπορεί να μη συνεργάζονται σωστά με τα άλλα μέλη της ομάδας.
3. Το να θέτεις προτεραιότητες στις αλλαγές του συστήματος μπορεί να γίνει δύσκολο, ειδικά για συστήματα που υπάρχουν περισσότερα του ενός άτομα που θέτουν τις προτεραιότητες για τις αλλαγές του συστήματος. Συχνά, κάθε άτομο θέτει διαφορετικές προτεραιότητες σε διαφορετικές αλλαγές.
4. Η διατήρηση της απλότητας απαιτεί περισσότερη εργασία. Υπό την πίεση των προθεσμιών παράδοσης του συστήματος, η ομάδα μπορεί να μην έχει το χρόνο να κάνει τις επιθυμητές απλοποιήσεις στο σύστημα.

Οι βασικές αρχές της ευέλικτης μεθοδολογίας αποτυπώνονται και στην παρακάτω εικόνα:



Εικόνα 9: Αρχιτεκτονική της ευέλικτης μεθοδολογίας

Ένα άλλο πρόβλημα, μη τεχνικό, το οποίο είναι ένα πρόβλημα που αντιμετωπίζουν όλες οι μέθοδοι αυξητικής ανάπτυξης είναι το γεγονός ότι ο πελάτης χρησιμοποιεί έναν εξωτερικό οργανισμό για την ανάπτυξη του συστήματος. Το κείμενο του προσδιορισμού των προδιαγραφών του συστήματος είναι το συμβόλαιο που δεσμεύει τα δύο μέρη, δηλαδή τον πελάτη και το οργανισμό που αναπτύσσει το σύστημα. Επειδή, όμως, με τη μεθοδολογία των ευέλικτων μεθόδων, ο όγκος των απαιτήσεων μεγαλώνει ταχύτητα, το γεγονός αυτό είναι ένα πρόβλημα για τη συγγραφή των συμβολαίων.

Συνεπώς, οι ευέλικτες μέθοδοι βασίζονται στα συμβόλαια, στα οποία ο πελάτης πληρώνει για το χρόνο που απαιτείται για την ανάπτυξη του λογισμικού και όχι για ένα συγκεκριμένο σύνολο απαιτήσεων. Εφόσον όλα πάνε καλά, αυτό ωφελεί και τον πελάτη και το δημιουργό του συστήματος. Ωστόσο, αν προκύψουν προβλήματα, θα προκύψουν διαφωνίες, που δύσκολα επιλύονται, για το ποιος ευθύνεται και ποιος πρέπει να πληρώσει για τον επιπλέον χρόνο και τους πόρους που χρειάστηκαν για να επιλυθούν τα προβλήματα.

2.3 Extreme Programming

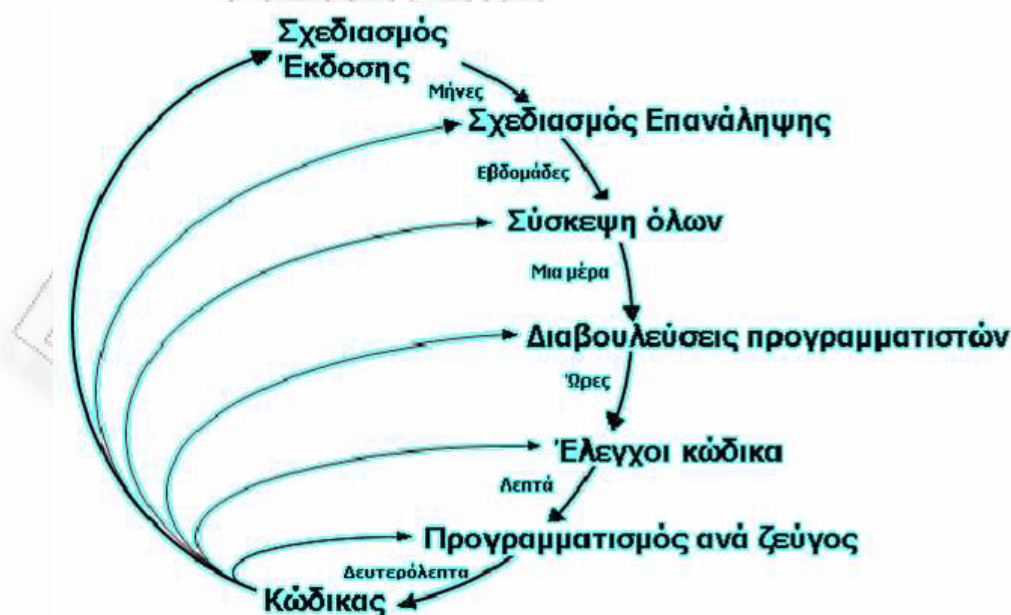
Ο ακραίος προγραμματισμός (Extreme programming - XP) είναι ίσως η πιο γνωστή και η πιο διαδεδομένη από τις ευέλικτες μεθόδους. Ονομάστηκε έτσι από τον Beck, γιατί αυτή η μέθοδος δημιουργήθηκε χρησιμοποιώντας καλές και αναγνωρισμένες πρακτικές, όπως η επαναλαμβανόμενη ανάπτυξη και η ανάμειξη του πελάτη σε υπερβολικό (extreme) βαθμό. Στη μέθοδο XP, όλες οι απαιτήσεις εκφράζονται ως σενάρια, τα οποία υλοποιούνται άμεσα ως μία σειρά εργασιών. Οι προγραμματιστές εργάζονται ανά δύο και δοκιμάζουν κάθε εργασία πριν γράψουν τον κώδικα. Όλες οι δοκιμές πρέπει να ολοκληρωθούν με επιτυχία πριν ενσωματωθεί ο κώδικας τους στο σύστημα. Μεσολαβεί μικρό χρονικό διάστημα μεταξύ των διαφορετικών εκδόσεων του συστήματος. Η μέθοδος XP περιλαμβάνει ένα αριθμό πρακτικών που αντιστοιχούν στις αρχές των ευέλικτων μεθόδων ανάπτυξης:

- Η αυξανόμενη ανάπτυξη στηρίζεται στις μικρές και συχνές εκδόσεις του συστήματος, καθώς και σε μία προσέγγιση όπου η περιγραφή των απαιτήσεων βασίζεται στα σενάρια που παρέχει ο πελάτης και αποτελούν τη βάση για τον σχεδιασμό της διεργασίας.
- Η ανάμειξη του πελάτη στηρίζεται στην πλήρη δέσμευση το πελάτη στην ομάδα ανάπτυξης. Ο εκπρόσωπος του πελάτη ή ο ίδιος ο πελάτης παίρνουν μέρος στην ανάπτυξη του συστήματος και είναι υπεύθυνος στο να καθορίζει αποδεκτές δοκιμές του συστήματος.
- Η άποψη άνθρωποι, όχι διεργασίες στηρίζεται στον προγραμματισμό ανά ζεύγη, τη συλλογική ιδιοκτησία του κώδικα του συστήματος, και τον υποφερτό ρυθμό ανάπτυξης που δεν περιλαμβάνει υπερβολικές ώρες εργασίας.
- Η αλλαγή στηρίζεται σε συχνές εκδόσεις του συστήματος, την ανάπτυξη μετά τον έλεγχο και τη συνεχή συνένωση.
- Η διατήρηση της απλότητας στηρίζεται στις συνεχείς αλλαγές του κώδικα χάριν της βελτίωσης της ποιότητάς του και στη χρήση απλών σχεδιασμών που δεν προσδοκούν μελλοντικές αλλαγές στο σύστημα.

Ο ακραίος προγραμματισμός είναι ένας ελαφρύς, αποτελεσματικός, χαμηλού κινδύνου, ευέλικτος, προβλέψιμος, επιστημονικός και ευχάριστος τρόπος για την ανάπτυξη λογισμικού.

Βασίζεται σε τέσσερις αξίες ^[13]:

1. Απλότητα (simplicity): Μεγιστοποίηση της εργασίας που δεν υλοποιείται, με προτίμηση στον απλό «αφελή» κώδικα.
2. Επικοινωνία (communication): Άμεση και διαπροσωπική για κοινή κατανόηση των προβλημάτων.
3. Ανατροφοδότηση (feedback): Συχνές οπισθοδρομήσεις παραδίδοντας λογισμικό που λειτουργεί αποτελεσματικά. Συγγραφή και χρήση περιπτώσεων ελέγχου (test cases) πριν την παραγωγή κώδικα, καθώς επίσης και ανάπτυξη σε μικρές εκδόσεις και σε μικρότερες επαναλήψεις και σε μικρότερες εργασίες και σε ακόμη μικρότερα τεστ κοκ. Στην παρακάτω εικόνα απεικονίζεται η δυνατότητα που υπάρχει για ανατροφοδότηση κατά τη διάρκεια του κύκλου ανάπτυξης λογισμικού.
4. Κουράγιο (courage): Η καλή ψυχολογία και το κουράγιο απαλλάσσουν την ομάδα ανάπτυξης από το φόβο της αποτυχίας που οδηγεί σε αργοπορημένα αποτελέσματα.



Εικόνα 10: Ανατροφοδότηση κατά τη διάρκεια του κύκλου ζωής της μεθοδολογίας XP

Η μέθοδος XP αποτελεί μια ακραία προσέγγιση της επαναλαμβανόμενης ανάπτυξης. Νέες εκδόσεις του συστήματος μπορεί να δημιουργούνται πολλές φορές κάθε μέρα και πρόσθετες λειτουργίες παραδίδονται στον πελάτη κάθε δύο εβδομάδες. Όταν ο προγραμματιστής κατασκευάζει το σύστημα το οποίο προορίζεται για την κυκλοφορία μιας νέας έκδοσης, θα πρέπει να το ελέγξει χρησιμοποιώντας τις υπάρχουσες αυτοματοποιημένες δοκιμές καθώς και τις δοκιμές για τον έλεγχο των νέων λειτουργιών. Η νέα έκδοση του συστήματος είναι αποδεκτή μόνο όταν όλες οι δοκιμές είναι επιτυχείς.

Μια βασική αρχή την παραδοσιακής τεχνολογίας λογισμικού είναι ότι θα πρέπει να σχεδιάζεις για αλλαγές. Αυτό σημαίνει ότι θα πρέπει να αναμένεις μελλοντικές αλλαγές στο λογισμικό και για αυτό θα πρέπει να σχεδιάζεις το λογισμικό έτσι ώστε αυτές οι αλλαγές να πραγματοποιούνται εύκολα. Ωστόσο, στη μέθοδο XP αυτή η αρχή έχει απορριφθεί με τη λογική ότι ο σχεδιασμός για αλλαγές είναι συχνά σπατάλη προσπάθειας. Οι αλλαγές που αναμένονται συχνά δεν πραγματοποιούνται ποτέ και οι αλλαγές που γίνονται είναι εντελώς διαφορετικές από αυτές που αναμένονταν.

Πίνακας 2: Χαρακτηριστικά της τεχνικής XP

Αρχή	Περιγραφή
Αυξανόμενος προγραμματισμός	Οι απαιτήσεις καταγράφονται σε Story cards και αποφασίζεται σε ποια έκδοση θα συμπεριληφθούν βάσει του χρόνου που θα είναι διαθέσιμος ο ορισμός τους και της σχετικής τους προτεραιότητας. Οι προγραμματιστές μετατρέπουν αυτά τα σενάρια σε εργασίες.
Μικρές εκδόσεις	Το μικρότερο δυνατό σύνολο λειτουργιών είναι το πρώτο που αναπτύσσεται. Είναι συχνή η έκδοση επόμενων εκδόσεων καθώς και η αυξανόμενη προσθήκη λειτουργιών από την πρώτη έκδοση του συστήματος.
Απλός σχεδιασμός	Ο σχεδιασμός που γίνεται είναι αρκετός ώστε να καλύπτει μόνο τις παρούσες απαιτήσεις.
Προγραμματισμός με δοκιμή πρώτα	Ένα αυτοματοποιημένο πλαίσιο εργασίας για έλεγχο μονάδας χρησιμοποιείται για την συγγραφή δοκιμών

Αλλαγή κώδικα	<p>για κάθε νέο κομμάτι λειτουργικότητας, πριν αυτή η λειτουργικότητα υλοποιηθεί.</p> <p>Όλοι οι συγγραφείς κώδικα αναμένεται να αλλάζουν συνεχώς μέρη του κώδικα για να τον βελτιώσουν όσο μπορούν. Αυτό διατηρεί τον κώδικα απλό, και εύκολο προς συντήρηση.</p>
Προγραμματισμός ανά ζεύγη	<p>Οι προγραμματιστές δουλεύουν ανά ζεύγη, ελέγχοντας ο ένας την εργασία του άλλου και προσφέροντας βοήθεια για να πετύχουν το καλύτερο δυνατό αποτέλεσμα.</p>
Συλλογική ιδιοκτησία	<p>Τα ζεύγη των προγραμματιστών δουλεύουν σε όλα τα μέρη του συστήματος, ώστε να μην αναπτυχθεί μεμονωμένη γνώση σε ένα κομμάτι του κώδικα και όλοι οι προγραμματιστές να γνωρίζουν καλά όλο τον κώδικα. Οποιοσδήποτε έχει την ικανότητα να αλλάζει οτιδήποτε.</p>
Συνεχής συνένωση	<p>Μόλις τελειώσει μια εργασία ενσωματώνεται σε όλο το σύστημα. Αφού ενσωματωθεί, πρέπει όλες οι δοκιμές μονάδων να είναι επιτυχείς.</p>
Υποφερτός ρυθμός	<p>Πολλές υπερωρίες δεν είναι αποδεκτές καθώς το φαινόμενο του δικτύου συχνά μειώνει την ποιότητα του κώδικα και την αποδοτικότητα.</p>
Διαθέσιμος πελάτης	<p>Ένας αντιπρόσωπος των τελικών χρηστών θα πρέπει πάντα να είναι διαθέσιμος στην ομάδα του XP. Σε μία διεργασία του XP, ο πελάτης είναι μέλος την ομάδας ανάπτυξης και είναι υπεύθυνος να αποδίδει σε αυτή σωστά τις απαιτήσεις του συστήματος.</p>

Το πρόβλημα με το σχεδιασμό χωρίς να λαμβάνουμε υπόψη τις αλλαγές σε λειτουργικότητα που μπορεί να ζητηθούν, είναι ότι όταν απαιτούνται αλλαγές, υποβαθμίζουν τη δομή του κώδικα και η πραγματοποίηση των αλλαγών με τον καιρό γίνεται δυσκολότερη. Η μέθοδος XP παρακάμπτει αυτό το πρόβλημα προτείνοντας ότι ο κώδικας πρέπει να αλλάζει συνεχώς. Αυτό σημαίνει ότι η ομάδα προγραμματισμού συνεχώς ψάχνει για πιθανές βελτιώσεις του κώδικα και τις υλοποιεί αμέσως μόλις τις εντοπίσει. Για αυτό το λόγο, ο κώδικας πρέπει να είναι κατανοητός και να μπορούν να γίνουν εύκολα αλλαγές.

Ο ακραίος προγραμματισμός ορίζει ένα ενοποιημένο σύνολο από 12 πρακτικές, το οποίο μπορεί να οδηγήσει στην επιτυχία ενός έργου. Επιπρόσθετα η XP μπορεί να χρησιμοποιηθεί από κοινού με άλλα πλαίσια ανάπτυξης, όπως είναι η μεθοδολογία RUP. Ο συνδυασμός αυτός έχει ονομαστεί διαδικασία dX [15]. Οι 12 αυτές πρακτικές είναι οι εξής:

- Το παιχνίδι του σχεδιασμού: Ο γρήγορος καθορισμός του σκοπού της επόμενης έκδοσης του λογισμικού που θα συνδυάζει τις επιχειρησιακές προτεραιότητες και τις τεχνικές εκτιμήσεις. Αν η πραγματικότητα προσπερνάει το σχεδιάγραμμα, τότε το σχεδιάγραμμα πρέπει να ανανεωθεί.
- Προγραμματισμός ανά ζεύγη: Όλος ο κώδικας πρέπει να γράφεται από δύο προγραμματιστές σε έναν υπολογιστή.
- Έλεγχοι πριν την κωδικοποίηση: Οι προγραμματιστές πρέπει συνεχώς να συγγράφουν ελέγχους μονάδων (unit tests), που πρέπει να εκτελούνται άψογα, έτσι ώστε να συνεχίζεται η ανάπτυξη. Οι πελάτες πρέπει να κάνουν ελέγχους για να διαπιστώνουν ότι τα επιθυμητά χαρακτηριστικά έχουν ολοκληρωθεί.
- Ανακατασκευή κώδικα: Οι προγραμματιστές πρέπει να ανακατασκευάζουν τον κώδικα του συστήματος χωρίς αλλαγές στη συμπεριφορά του για να αφαιρέσουν διπλότυπους κώδικες, να απλοποιήσουν τον κώδικα και να βελτιώσουν την επικοινωνία.
- Σταθερές κωδικοποίησης: Οι προγραμματιστές πρέπει να γράφουν όλον τον κώδικα σύμφωνα με ένα πλαίσιο κανόνων που θα δίνει έμφαση στην επικοινωνία μεταξύ διαφορετικών μερών του κώδικα.
- Απλή σχεδίαση: Το σύστημα πρέπει να είναι όσο απλό γίνεται κάθε χρονική στιγμή. Η επιπλέον πολυπλοκότητα πρέπει να αφαιρείται μόλις βρεθεί.
- Μικρές εκδόσεις: Αρχικά, πρέπει να κατασκευαστεί ένα απλό σύστημα και στη συνέχεια να δημιουργούνται νέες εκδόσεις του λογισμικού βασισμένες σε ένα μικρό κύκλο ανάπτυξης.
- Διαρκείς ενοποιήσεις του κώδικα: Πρέπει να υπάρχει διαρκής ενοποίηση και μεταγλώττιση του κώδικα, πολλές φορές την ημέρα και οποτεδήποτε υλοποιείται ένα χαρακτηριστικό.

- Συλλογική ιδιοκτησία κώδικα: Οποιοσδήποτε πρέπει να μπορεί να αλλάξει οποιοδήποτε κώδικα οπουδήποτε οποιαδήποτε στιγμή.
- Διαρκή παρουσία πελάτη: Είναι απαραίτητη η παρουσίαση συνεχώς ενός εργαζόμενου από την πλευρά του πελάτη που να μπορεί να απαντήσει σε κάθε είδους ερώτηση σχετιζόμενη με το λογισμικό.
- Αρχιτεκτονική εικόνα: Όλη η αρχιτεκτονική πρέπει να οδηγείται από μία απλή εικόνα για το πώς πρέπει το ολόκληρο το σύστημα να δουλεύει.
- Υποφερτός ρυθμός εργασίας: Δεν πρέπει οι προγραμματιστές να εργάζονται παραπάνω από 40 ώρες την εβδομάδα. Επίσης, δεν πρέπει οι προγραμματιστές να κάνουν υπερωρίες δύο εβδομάδες συνεχόμενες.

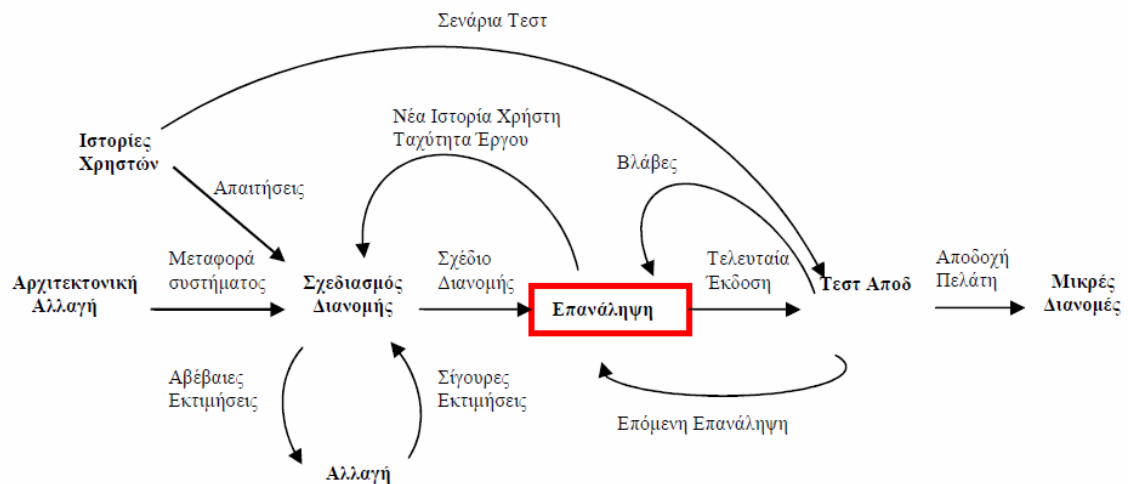
Στην εικόνα που ακολουθεί εμφανίζονται εικονογραφημένα οι 12 πρακτικές της XP.

<p>Το παιχνίδι του σχεδιασμού</p>  <p>=συμβολή του πελάτη</p>	<p>Προγραμματισμός ανά ζεύγη</p>  <p>=λιγότερη λάθη στον κώδικα</p>	<p>Ελεγχος πριν την κωδικοποίηση</p>  <p>=παικτικό λογισμικό</p>	<p>Ανακατασκευή κώδικα</p>  <p>=καθαρός κώδικας</p>
<p>Σταθερές κωδικοποίησης</p>  <p>=καλύτερη επικοινωνία</p>	<p>Απλή Σχεδίαση</p>  <p>=ευελιξία</p>	<p>Μικρές εκδόσεις</p>  <p>=προσαρμοστικότητα</p>	<p>Διαρκείς εντοπισμοί του κώδικα</p>  <p>=διαρκής ανάλυση</p>
<p>Συλλογική ιδιοκτησία κώδικα</p>  <p>=διαμοιρασμός της γνώσης</p>	<p>Διαρκή παρουσία Πελάτη</p>  <p>=εξοικείωση των απαιτήσεων</p>	<p>Αρχιτεκτονική εικόνα</p>  <p>=κατανόηση του συστήματος</p>	<p>Υποφερτός ρυθμός εργασίας</p>  <p>=αποδοτικότητα</p>

Εικόνα 11: Οι δώδεκα πρακτικές της μεθοδολογίας XP

Ολοκληρώνοντας, ο ακραίος προγραμματισμός επικεντρώνεται στην κατασκευή του κώδικα ενθαρρύνοντας τη συγγραφή πολλών τεστ για τον έλεγχο του κώδικα πριν ακόμη από τη συγγραφή του κώδικα, τον προγραμματισμό σε ζεύγη, την ανακατασκευή του κώδικα (refactoring), τη συνεχή επικοινωνία με τους πελάτες και την προσαρμογή στις τρέχουσες αλλαγές. Τέλος, εισάγει τη χρήση των CRC (Class-Responsibility-Collaborator) καρτών όπου γίνεται αποσύνθεση των απαιτήσεων σε κλάσεις (class), αναγράφονται οι υπευθυνότητες (responsibilities) και οι

συνεργαζόμενες κλάσεις (collaborators)^[8]. Στη συνέχεια αναπαριστάται διαγραμματικά ένας κύκλος ζωής έργου.



Εικόνα 3: Κύκλος ζωής έργου με βάση τη μεθοδολογία XP

Ωστόσο, αν και η XP κατατάσσεται στις πολλά υποσχόμενες μεθοδολογίες που αξίζει να μελετηθεί, δε θα πρέπει να χρησιμοποιείται σε μεγάλα έργα. Η XP είναι μια περιορισμένη διαδικασία που χρειάζεται προσθήκες για να ταιριάζει σε ένα ολοκληρωμένο έργο ανάπτυξης. Για μια μικρή ομάδα ανάπτυξης έργου που εργάζεται σε ένα περιβάλλον σχετικά υψηλής εμπιστοσύνης όπου ο χρήστης είναι αναπόσπαστο κομμάτι της ομάδας ανάπτυξης, η XP μπορεί να αποδώσει τα βέλτιστα^[15].

2.4 Rational Unified Process

Το μοντέλο αυτό επιτρέπει σε νέες εκδόσεις του έργου να υλοποιούν και νέες απαιτήσεις που εξελίσσονται όσο το σύστημα υλοποιείται. Καθοδηγείται από μελέτες χρήσης (use cases) και χρησιμοποιεί την UML σαν γλώσσα μοντελοποίησης. Το μοντέλο προσφέρει πλούσιο πλαίσιο υποστήριξης της διαδικασίας.

Το μοντέλο αυτό είναι μια μεθοδολογία ανάπτυξης λογισμικού, που 1998 αναπτύχθηκε από την Rational Software, η οποία σήμερα ανήκει στην IBM. Η πρώτη έκδοση αναπτύχθηκε το 1998 και ήταν ένα αποτέλεσμα της συνεργασίας των Grady Booch, James Rumbaugh και Ivar Jacobson^[19].

Η RUP είναι μια μεθοδολογία έργου που μπορεί να προσαρμοστεί στις συγκεκριμένες απαιτήσεις του πελάτη ^[22]. Είναι κατάλληλη τόσο για μικρές ομάδες όσο και για μεγάλους οργανισμούς ανάπτυξης. Αποτελεί ένα ολοκληρωμένο πλαίσιο εργασίας για την διαδικασία ανάπτυξης λογισμικού με σκοπό τη γρήγορη παραγωγή λογισμικού υψηλής ποιότητας που ικανοποιεί πλήρως τους τελικούς χρήστες και αναπτύσσεται με συγκεκριμένο χρονοδιάγραμμα και κόστος.

Προκειμένου να εξασφαλιστούν τα παραπάνω η RUP παρέχει μια ιεραρχημένη προσέγγιση στην ανάθεση εργασιών και ευθυνών. Προωθείται η συνεργασία των μελών της ομάδας ανάπτυξης, παρέχοντας σε κάθε μέλος εύκολη πρόσβαση σε κοινή βάση γνώσης ενώ παράλληλα παρέχονται οδηγίες και συμβουλευτικά εργαλεία που αυτοματοποιούν μεγάλα τμήματα της διαδικασίας ανάπτυξης. Αυτό που χαρακτηρίζει τη RUP είναι το γεγονός ότι σε αντίθεση με άλλες μεθοδολογίες, οι δραστηριότητές της δεν επικεντρώνονται στην παραγωγή πλεονασματικών εγγράφων αλλά στο χειρισμό μοντέλων που αποτελούν πλούσιες σημασιολογικές αναπαραστάσεις του υπό ανάπτυξη λογισμικού. Η RUP μεταξύ άλλων αποτελεί έναν πολύ καλό οδηγό για την αποτελεσματική χρησιμοποίηση της UML (γλώσσα μοντελοποίησης, αναλύεται σε επόμενη ενότητα του κεφαλαίου).

Ένα ακόμη χαρακτηριστικό της μεθοδολογίας είναι η επαναληπτική της φύση. Σε κάθε κύκλο εξέλιξης ή ανάπτυξης του κύκλου ζωής του λογισμικού και οι τέσσερις φάσεις που προβλέπονται παίρνουν μέρος. Έτσι όλες οι ροές εργασιών της διαδικασίας ανάπτυξης συνεισφέρουν σε κάποιο βαθμό σε κάθε φάση με μια όμως από αυτές να είναι η πιο σημαντική ^[32]. Με τον τρόπο αυτό πετυχαίνεται σημαντική μείωση των κινδύνων.

Οι βασικές αξίες της μεθοδολογίας RUP είναι οι παρακάτω ^[21]

- Σχεδίαση καθοδηγούμενη από τις περιπτώσεις χρήσεις
- Προσαρμογή διαδικασίας
- Υποστήριξη εργαλείων

Η διαδικασία είναι καθοδηγούμενη από τις περιπτώσεις χρήσης και οι περιπτώσεις χρήσης αποτελούν τη βάση για άλλα στοιχεία στη διαδικασία ανάπτυξης. Η πρακτική δουλειά στη μεθοδολογία RUP αποτελείται από τα παρακάτω στάδια:

- Επαναληπτική ανάπτυξη του λογισμικού: Τα επιμέρους τμήματα λογισμικού που αναπτύσσονται επανελέγχονται και συμπληρώνονται συνεχώς καθώς όλο και μεγαλύτερο εύρος του έργου υλοποιείται.
- Διαχείριση των απαιτήσεων: Οι απαιτήσεις οργανώνονται σε θεμελιώδη αυτόνομα τμήματα λογισμικού με καλά καθορισμένη διεπαφή με το σύνολο του έργου.
- Χρησιμοποίηση μιας αρχιτεκτονικής βασισμένης σε συστατικά: Ολοκληρωμένες ανεξάρτητες δομικές μονάδες λογισμικού.
- Διαγραμματική οπτικοποίηση του λογισμικού
- Έλεγχος της ποιότητας του λογισμικού: Στοιχείο απαραίτητο στη μοντέρνα φιλοσοφία ανάπτυξης εφαρμογών.
- Διαχείριση των αλλαγών στο λογισμικού: Στοιχείο απαραίτητο γιατί κανένα λογισμικό δεν έχει ολοκληρωμένο το σύνολο των απαιτήσεων πριν και ίσως κατά τη διάρκεια της ανάπτυξης.
- Καλά ορισμένο πλαίσιο διαχείρισης των αναφορών: Για τα κείμενα της RUP υπάρχουν πρότυπα, τα οποία συμπληρώνονται από τα επιφορτισμένα πρόσωπα και δημιουργούν ένα σύμπλεγμα τεκμηρίωσης που καθιστά εφικτή την διόρθωση, συμπλήρωση και τον έλεγχο ενός προϊόντος λογισμικού από τον οποιοδήποτε είναι «μυημένος» στη RUP ακόμα και αν έχουν παρέλθει έτη από την αποπεράτωση του έργου. Επίσης η τήρηση σωστής τεκμηρίωσης βοηθά την εύκολη εισαγωγή νέων προσώπων στις ομάδες ανάπτυξης και διαχείρισης ενός έργου.

Η μεθοδολογία RUP μπορεί να παρουσιαστεί χρησιμοποιώντας τέσσερα βασικά στοιχεία:

- Ρόλους: ποιος κάνει τι
- Γεγονότα: τι έχει παραχθεί
- Δραστηριότητες: πως επιτυγχάνεται μία εργασία
- Ροές εργασίας: πως επιτυγχάνεται ένας στόχος

Το μοντέλο RUP έχει τέσσερις φάσεις στο χρόνο ^[22]:

1. Έναρξη (Inception)

Ο βασικός και θεμελιώδης στόχος είναι η προκαταρκτική συνεργασία όλων των ενδιαφερομένων ομάδων και πλευρών. Επίσης θα πρέπει να εξασφαλιστεί η συνεργασία όλων των τμημάτων και ομάδων χρηστών που δημιουργούν εκ των πραγμάτων τις απαιτήσεις. Η αρχική καταγραφή γίνεται με τη συνεργασία των χρηστών και με καλά καθορισμένες συναντήσεις. Επίσης οι αναλυτές θα πρέπει εκ των προτέρων να έχουν ανατρέξει σε βιβλιογραφία και σε παράπλευρες διερευνήσεις του αντικειμένου του έργου ώστε να έχουν καλά καθορισμένες ερωτήσεις προς τους χρήστες.

Με το τέλος κάθε συνάντησης οι χρήστες θα πρέπει να λαμβάνουν κείμενο που περιγράφει το περιεχόμενο της συνάντησης καθώς και μια ανάλυση και δόμηση των όσων συζητήθηκαν. Ο χρήστης θα πρέπει να κάνει κριτική στα όσα γράφονται. Η βασική μονάδα κοστολόγησης κάθε εταιρίας είναι ο εργατομήνας. Είναι το μέσο κόστος παρουσίας ενός προσώπου που ανήκει σε ένα επίπεδο αμοιβών σε μία ομάδα. Ο σωστός προϋπολογισμός του έργου προδιαγράφει και το ποιοτικό αποτέλεσμα του έργου. Επίσης στον προϋπολογισμό του έργου θα πρέπει να συμπεριλαμβάνονται και έξοδα μετακίνησης, εκπαίδευσης, μεταφοράς τεχνογνωσίας, αγορών υλικού και λογισμικού.

Η αρχική ανάλυση κινδύνου πρέπει να συμπεριλάβει όλα εκείνα τα σημεία που παρουσιάζουν δυσκολία ανάλυσης, ανάπτυξης αλλά και έχουν υψηλό κίνδυνο αποτυχία λόγω έλλειψης εμπειρίας. Ο καθορισμός του σκοπού του έργου. Έχει σημασία η σωστή προδιαγραφή του έργου ώστε το όλο εγχείρημα να περιοριστεί στα απολύτως απαραίτητα επίπεδα. Ο καθορισμός μιας υποψήφιας πλατφόρμας ανάπτυξης και υλοποίησης. Η πλατφόρμα που θα επιλεγεί θα πρέπει να μπορεί να καλύψει της απαιτήσεις του έργου, αλλά και να υπάρχει εμπειρία (ή να προϋπολογιστεί σωστά η μεταφορά τεχνογνωσίας) από τις ομάδες υλοποίησης.

Η κατασκευή ενός πιλοτικού πρωτοτύπου πιλοτικού μπορεί να παραληφθεί με την επίσπευση του πρώτου iteration το οποίο θα συμπεριλάβει βασικά συστατικά στοιχεία του έργου. Η αρχική περιγραφή των Use Cases βοηθάει πολύ στην κατανόηση του έργου. Σε αυτό το στάδιο θα πρέπει η περιγραφή να περιλαμβάνει βασικά στοιχεία όπως σύντομη περιγραφή και ποιοι οι χρήστες που αλληλεπιδρούν με το συγκεκριμένο Use Case.

2. Επεξεργασία (Elaboration)

Η φάση επεξεργασίας του έργου περιλαμβάνει τους εξής στόχους:

- Ανάλυση των Use Case: Καθορισμός και ανάλυση του 80% των σημαντικότερων use cases και τους αντίστοιχους ρόλους που συμμετέχουν μέχρι το τέλος αυτής της φάσης.
- Δημιουργία των Prototype User Interfaces
- Ιεραρχούμε τα Use Cases μέσα στο Use Case Model μα βάση τη δυσκολία τους αλλά και το ρίσκο που εμπεριέχουν για την ολοκλήρωση του έργου
- Ανάλυση σε βάθος των πιο σημαντικών Use Cases (γράψιμο και διόρθωση τους)
- Προετοιμασία ενός μοντέλου από σημαντικές για το έργο κλάσεις και εκτίμηση της αξίας τους. Δημιουργία ενός βασικού User Interface.

3. Κατασκευή (Construction)

Η φάση κατασκευής επικεντρώνει το ενδιαφέρον της στην κατασκευή του έργου.

Η φάση κατασκευής χαρακτηρίζεται από:

- Σταδιακή ανάπτυξη της λειτουργικότητας του έργου
- Η ανάπτυξη γίνεται σε όλο το βάθος του έργου και όχι επιλεκτικά όπως στις προηγούμενες φάσεις
- Το έργο αποκτά σταδιακά ισορροπία και προσεγγίζει στη μορφή του τελικού παραδοτέου.
- Κατασκευάζεται κάθε επιμέρους λεπτομέρεια που στις προηγούμενες φάσεις δεν αντιμετωπιζόταν λόγω μικρού ρίσκου
- Η ανάλυση συνεχίζεται και βελτιώνονται τα UC αλλά η ανάπτυξη του software παίζει πλέον πρωτεύοντα ρόλο.

Το παραγόμενο αυτής της φάσης είναι έτοιμο προϊόν για να παραδοθεί στους τελικούς χρήστες. Κατ' ελάχιστο το προϊόν αποτελείται από το λογισμικό κατασκευασμένο στις κατάλληλες πλατφόρμες, τα user manuals και μια περιγραφή της εφαρμογής.

Τα σημεία που θα πρέπει να εξασφαλισθούν είναι:

- Είναι το προϊόν σταθερό και αρκετά έτοιμο για παράδοση στους τελικούς χρήστες;
- Είναι το πραγματικό πλέον έργο αρκετά κοντά στο σχεδιαζόμενο;

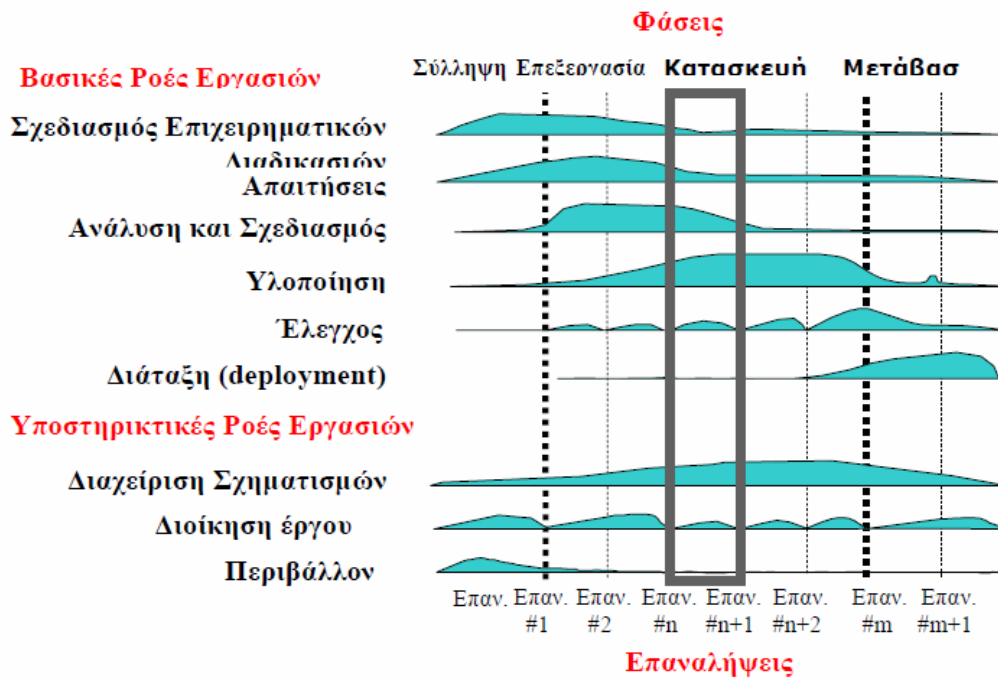
4. Μετάβαση (Transition)

Η φάση αυτή αποτελεί την μετάβαση και εφαρμογή του έργου στο περιβάλλον των τελικών χρηστών. Περιλαμβάνει την τυποποίηση (ανάπτυξη install application), την αποστολή, την εκπαίδευση και την τεχνική υποστήριξη. Η ομάδα των developers μειώνεται σε αριθμό και σε συμμετοχή. Ο έλεγχος του έργου μεταφέρεται στην ομάδα υποστήριξης. ALFA, BETA releases και τελικό προϊόν. Ολοκλήρωση με άλλα λογισμικά και υποδομή της εταιρίας του χρήστη

Αυτή η φάση περιλαμβάνει:

- Beta testing για να αξιολογηθεί το προϊόν σε σχέση με τις προδιαγραφές και τις απαιτήσεις
- Παράλληλη λειτουργία του προϊόντος με τα υπάρχοντα συστήματα ώστε να αξιολογηθεί η ικανότητα του να τα αντικαταστήσει
- Μετατροπή των δεδομένων από την μορφή των παλαιών συστημάτων που είχε η εταιρία (πχ παλαιές databases) στις απαιτήσεις του νέου προϊόντος (νέες databases)
- Εκπαίδευση τόσο των χρηστών όσο και του τμήματος υποστήριξης
- Προώθηση του προϊόντος προς άλλα τμήματα της δικιάς μας εταιρίας (marketing, sales)

Στη συνέχεια αναπαριστάται διαγραμματικά ο κύκλος ζωής ενός έργου.



Εικόνα 4: Κύκλος ζωής έργου με βάση τη μεθοδολογία RUP

Τα στοιχεία που συνθέτουν το πλαίσιο εργασίας της RUP είναι τα εξής ^[15]:

- 4 φάσεις (phases)
- 8 επαναλήψεις (iterations) τουλάχιστον
- 9 ροές εργασιών (workflows)
- 57 δραστηριότητες (activities)
- 270 βήματα δραστηριοτήτων (activity steps) (κατά προσέγγιση)
- 114 αρχέτυπα (artifacts, κύρια μοντέλα της διαδικασίας ανάπτυξης)
- 38 ρόλοι (περισσότερα από 38 άτομα)

Παρατηρώντας την παραπάνω σχηματική απεικόνιση του κύκλου ζωής γίνεται φανερό πως η ροή εργασίας «Διοίκηση έργου» παίζει μεγάλο ρόλο σε όλες τις φάσεις. Αυτό είναι απόλυτα αναμενόμενο καθώς η διοίκηση έργου γενικότερα αποτελείται από όλες τις δραστηριότητες που ασχολούνται με την πραγματοποίηση μιας εργασίας.

Επιπλέον πρέπει να τονιστεί πως η RUP δεν είναι απλά μια μεθοδολογία. Είναι πολύτιμη γιατί περικλείει πολλές από τις βέλτιστες πρακτικές που είναι σήμερα

διαθέσιμες, παρέχοντας έτσι τόσο στους διοικητές έργων όσο και στους αναλυτές, τους ελεγκτές κ.ά. ένα ευέλικτο πλαίσιο για να εφαρμόσουν τις λύσεις τους.

Το σημερινό εξελιγμένο μοντέλο της μεθόδου, Enterprise Unified Process (EUP), περιλαμβάνει μια πέμπτη φάση την «παραγωγή» και δύο ακόμη ροές εργασιών, τις «λειτουργίες και υποστήριξη» και «διαχείριση υποδομής» λύνοντας έτσι το πρόβλημα της διατήρησης και υποστήριξης που είχε η RUP και καλύπτοντας ουσιαστικά όλη τη διαδικασία ανάπτυξης λογισμικού. Κάποια άλλα μειονεκτήματα της μεθοδολογίας RUP είναι η αδυναμία υποστήριξης ταυτόχρονης ανάληψης πολλών έργων αλλά και επαναχρησιμοποίησης σε μεγάλη κλίμακα μέσα σε έναν οργανισμό. Ενώ τέλος δεν μπορεί να υποστηρίξει πλήρως τομείς όπως είναι η διοίκηση μετρήσεων (metrics management), η διοίκηση επαναχρησιμοποίησης και η διοίκηση ανθρώπινου δυναμικού.

2.5 Σύγκριση

Στόχος της ενότητας αυτής είναι η σύγκριση των δύο ευέλικτων μεθοδολογιών που αναλύθηκαν στις προηγούμενες ενότητες: του ακραίου προγραμματισμού και της Rational Unified Process.

Η RUP αναπτύχθηκε για μεγάλα έργα λογισμικού και λειτουργεί συνοδευτικά με την UML. Αντίθετα, η XP είναι μία μεθοδολογία για μικρά και μεσαία έργα λογισμικού. Επίσης, η RUP έχει αναπτυχθεί από τη Rational Software, που πλέον ανήκει στην εταιρία IBM, ενώ η μεθοδολογία XP δεν είναι εμπορική, αλλά αναπτύχθηκε από μία ομάδα ανθρώπων. Παρόλα αυτά, οι δύο μεθοδολογίες έχουν κοινές ρίζες, αφού βασίζονται και οι δύο στην εμπειρία των ανθρώπων που τις ανέπτυξαν στην τεχνολογία λογισμικού. Οι δύο μεθοδολογίες, βέβαια, έχουν διαφορετική φιλοσοφία. Η μεθοδολογία RUP λαμβάνει υπόψη την τεχνική διαχείριση του έργου, ενώ η XP εστιάζει στο προσωπικό ανάπτυξης του έργου.

Παρακάτω παρατίθενται τα πλεονεκτήματα και τα μειονεκτήματα των δύο μεθόδων όσον αφορά τεχνικούς, οικονομικούς και κοινωνικούς παράγοντες:

- **Τεχνικοί παράγοντες**

RUP

Η μεθοδολογία RUP παρέχει στον οργανισμό, που την χρησιμοποιεί, μία μεγάλη πληθώρα από εργαλεία ανάπτυξης και έγγραφα. Επίσης, η RUP μπορεί να μεταφορτωθεί από το διαδίκτυο και να ενημερωθεί αυτόματα από νέες εκδόσεις. Το δυνατό σημείο της είναι τα ενσωματωμένα εργαλεία που προσφέρει το λογισμικό.

Ένα από τα μειονεκτήματα της μεθοδολογίας είναι ότι διαθέτει πολλούς ρόλους και διαδικασίες, οι οποίες για τα περισσότερα έργα λογισμικού, που δεν είναι μεγάλα, πρέπει να μειωθούν σημαντικά.

XP

Από την άλλη μεριά, η μεθοδολογία XP είναι πολύ απλή, που δε συνδέεται με συγκεκριμένα εργαλεία. Η κοινότητα ανάπτυξης της μεθοδολογίας XP έχει αναπτύξει αρκετά τέτοια λογισμικά, όπως για παράδειγμα το λογισμικό JUnit για τον έλεγχο του λογισμικού, και ο χρήστης είναι ελεύθερος να επιλέξει οποιαδήποτε από αυτά θέλει.

Στα μειονεκτήματα της μεθοδολογίας, στην XP υπάρχει ένας μικρός αριθμός από πρακτικές, οι οποίες πρέπει να αναπτυχθούν ανάλογα με τη φύση του έργου.

- **Οικονομικοί παράγοντες**

RUP

Τα οικονομικά θέματα των δύο μεθοδολογιών είναι διαφορετικά τόσο στη διανομή όσο και στην υποστήριξη, αφού η RUP είναι εμπορική, ενώ η XP ανοικτού κώδικα. Η οικονομική δύναμη της RUP της παρέχει τη δύναμη να κάνει περισσότερο μάρκετινγκ και να είναι περισσότερο προβεβλημένη. Επίσης, η RUP ανήκει στην εταιρία IBM, της οποίας η φήμη στη βιομηχανία λογισμικού είναι μεγάλη. Ένα άλλο πλεονέκτημα της μεθοδολογίας RUP είναι ότι παρέχει συνεχείς αναβαθμίσεις, το οποίο διευκολύνει τους χρήστες να είναι σε συνεχή ενημέρωση με τις νέες πρακτικές στις μεθοδολογίες ανάπτυξης έργων λογισμικού.

XP

Από την άλλη μεριά, γιατί κάποιος πρέπει να πληρώσει κάτι το οποίο μπορεί να αποκτήσει δωρεάν; Η μεθοδολογία XO προσφέρει μία δωρεάν λύση της μεθοδολογίας RUP.

- **Κοινωνικοί παράγοντες**

Οι κοινωνικοί παράγοντες των μεθοδολογιών RUP και XP πηγάζουν από το γεγονός ότι το ένα λογισμικό είναι εμπορικό και το άλλο δωρεάν. Οι μεγάλες εταιρίες ανάπτυξης λογισμικού έχουν συνηθίσει να αγοράζουν άδειες λογισμικού για τα εργαλεία που χρησιμοποιούν. Η δωρεάν φύση της μεθοδολογίας XP αντιμετωπίζεται με σκεπτικισμό. Μπορεί κάτι που είναι δωρεάν να είναι καλό;

Η επιλογή της κατάλληλης μεθοδολογίας βέβαια είναι κυρίως τεχνικής και οικονομικής φύσης. Ωστόσο, είναι σημαντικό θέμα και η κοινωνική διάσταση. Μικρότερες επιχειρήσεις δείχνουν μία τάση να χρησιμοποιούν λογισμικό ανοικτού κώδικα, ενώ οι μεγάλες επιχειρήσεις εμπορικά λογισμικά.

Όσον αφορά τις φάσεις το κύκλου ζωής των δύο μεθοδολογιών, η RUP αποτελείται από τέσσερις ακολουθιακούς κύκλους:

- Έναρξη
- Επεξεργασία
- Κατασκευή
- Μετάβαση

Αυτές οι τέσσερις φάσεις δημιουργούν ένα κύκλο ανάπτυξης και παράγουν ένα παραδοτέο για το λογισμικό υπό ανάπτυξη. Μέσα σε κάθε φάση υπάρχει ένας αριθμός από επαναλήψεις και καθεμία φάση εστιάζει το ενδιαφέρον της σε διαφορετικές δραστηριότητες, αν και όλες οι δραστηριότητες εκτελούνται παράλληλα. Η φάση της έναρξης στοχεύει στην επιχείρηση και τις απαιτήσεις, η φάση της επεξεργασίας είναι εστιασμένη στην ανάπτυξη, η φάση της κατασκευής είναι κυρίως υλοποίηση του κώδικα και έλεγχος αυτού και η φάση της μετάβασης εστιάζει στην εγκατάσταση και αλλαγή του λογισμικού.

Από την άλλη μεριά, η μεθοδολογία XP εστιάζει στον παραγόμενο κώδικα, ανεξάρτητα από τη διάσταση του χρόνου. Στην αρχή του έργου, η εστίαση είναι στη βάση του παραγόμενου προϊόντος και στη συνέχεια στα χαρακτηριστικά του, αλλά καθ' όλη τη διάρκεια η εστίαση βρίσκεται στον κώδικα. Η σχεδίαση εξελίσσεται όσο εξελίσσεται και το λογισμικό. Η αξία της απλότητας στη σχεδίαση και στις πρακτικές υπογραμμίζει ότι η σχεδίαση πρέπει να είναι τόσο απλή όσο οι τωρινές ανάγκες και όχι οι μελλοντικές αλλαγές. Οι δραστηριότητες ανάλυσης και σχεδίασης δεν είναι στοχευμένες στην αρχή του έργου, αλλά στην ανάπτυξη της λειτουργίας του σχεδιασμού.

Και οι δύο μεθοδολογίες έχουν μικρές διαδικασίες, αν και η μεθοδολογία XP έχει ακόμα μικρότερες από τη RUP. Οι διαδικασίες στην XP ποικίλουν από δευτερόλεπτα στον προγραμματισμό κατά ζεύξη μέχρι ημέρες και μήνες στο σχεδιασμό. Οι διαδικασίες στη RUP είναι λιγότερο συχνές και μπορεί να λαμβάνουν χώρα ανά εβδομάδα ή και μήνα. Και οι δύο μεθοδολογίες επιμένουν στην αποτελεσματική χρήση των διαθέσιμων πόρων. Μία βασική αρχή της μεθοδολογίας XP είναι ότι πρέπει να αναπτυχθεί ότι είναι απολύτως απαραίτητο.

Όσον αφορά την επικοινωνία και τη γεωγραφική διανομή των μελών ανάπτυξης της ομάδας έργου, η μεθοδολογία RUP στηρίζει τις μεγάλες ομάδες έργου που μπορεί να είναι κατανεμημένες γεωγραφικά και επικοινωνούν βάσει εγγράφων. Αντίθετα, η φιλοσοφία της μεθοδολογίας XP σε αυτόν τον τομέα, υποστηρίζει την άμεση και προφορική επικοινωνία τόσο εσωτερικά στο έργο όσο και εξωτερικά με τους πελάτες. Πρακτικά, οι ομάδες ανάπτυξης σύμφωνα με τη μεθοδολογία XP τείνουν να βρίσκονται ακόμα και στο ίδιο δωμάτιο για να εκμεταλλευτούν στο έπακρο τις βασικές αρχές της μεθοδολογίας αυτής. Ακόμα και η εγκατάσταση των ομάδων σε διαφορετικούς ορόφους ενός κτηρίου έχει δείξει ότι μπορεί να προκαλέσει προβλήματα ^[20].

Όσον αφορά το τεχνικό περιεχόμενο των δύο μεθοδολογιών, μπορούμε να βρούμε ομοιότητες και διαφορές στις δύο μεθοδολογίες σύμφωνα με την έκταση, την ευελιξία, την έμφαση, τις σχέσεις με τους πελάτες, τις διανομές, την τεχνική δουλειά και τα εργαλεία.

- **Έκταση**

Η μεθοδολογία RUP αποτελείται από μία μεγάλη συλλογή από έγγραφα, ρόλους και δραστηριότητες. Η RUP τονίζει την ανάγκη για προσαρμογή σε μία συγκεκριμένη επιχείρηση, το οποίο στις περισσότερες περιπτώσεις αποφέρει μείωση των διαδικασιών. Μία κριτική που δέχεται η RUP είναι ότι είναι περίπλοκη.

Από την άλλη η μεθοδολογία XP είναι απλή τόσο στην παρουσίαση όσο και στην εφαρμογή της. Η διαφορά στην έκταση των δύο μεθοδολογιών μπορεί να φανεί και από τον παρακάτω πίνακα, που δείχνει τους ρόλους που υπάρχουν στη μεθοδολογία XP και τους αντίστοιχους ρόλους της μεθοδολογίας RUP.

Πίνακας 3: Οι ρόλοι στη μεθοδολογία XP και οι αντίστοιχοι ρόλοι στη μεθοδολογία RUP

Ομάδα	Ρόλοι στην XP	Ρόλοι στην RUP
Ομάδα πελάτη	Πελάτης	Υπεύθυνος απαιτήσεων Αναλυτής συστήματος Διαχειριστής έργου
	Ιχνηλάτης	Αναλυτής ελέγχων
	Ελεγκτής	Ελεγκτής Διαχειριστής συστήματος ελέγχων
Ομάδα ανάπτυξης	Προγραμματιστής	Προγραμματιστής Αναλυτής Ενοποιητής
	Προπονητής	Διαχειριστής συστήματος

- **Ευελιξία**

Η ευελιξία υπάρχει στη φιλοσοφία και των δύο μεθοδολογιών. Στην RUP, η ευελιξία σημαίνει κυρίως την προσαρμογή σε διαφορετικές ανάγκες και η εστίαση της στις διαδικασίες. Αντίστοιχα, στην XP η ευελιξία σημαίνει τις

συνεχείς αλλαγές που βασίζονται στην ανατροφοδότηση του πελάτη και το γεγονός ότι οι 12 πρακτικές μπορούν να υλοποιηθούν διαφορετικά σε διαφορετικά έργα λογισμικού. Οι βασικές αξίες είναι σταθερές στη μεθοδολογία XP, όλα τα υπόλοιπα μπορούν να αλλάξουν.

- **Έμφαση**

Η μεθοδολογία RUP οδηγείται από τις περιπτώσεις χρήσεις, δηλαδή τις περιγραφές των χρήσεων του συστήματος που αναπτύσσεται. Η μεθοδολογία XP υλοποιεί το σύστημα βασισμένη στο συνεχή έλεγχο του κώδικα, δηλαδή οι περιπτώσεις ελέγχου έρχονται και υλοποιούνται πριν γραφεί ο κώδικας. Οι περιπτώσεις χρήσης στη μεθοδολογία RUP είναι ολόκληρο το σενάριο της αλληλεπίδρασης του χρήστη και του συστήματος.

Τέλος, όσον αφορά τις αλλαγές, η RUP περιμένει συνεχείς αλλαγές στις απαιτήσεις, ενώ η XP διαχειρίζεται με κάθε λεπτομέρεια μόνο το άμεσο μέλλον.

- **Σχέσεις με τους πελάτες**

Όσον αφορά τις σχέσεις με τους πελάτες και οι δύο μεθοδολογίες κατανοούν τη σημασία της στενής σχέσης με τον πελάτη, αλλά χειρίζονται διαφορετικά το συγκεκριμένο ζήτημα. Η μεθοδολογία XP απαιτεί την άμεση εμπλοκή του πελάτη στην ομάδα για να απαντάει τις ερωτήσεις που προκύπτουν και να θέτει τις προτεραιότητες. Η μεθοδολογία RUP είναι περισσότερο ευέλικτη στη διαχείριση των σχέσεων με τους πελάτες. Δεν απαιτεί ότι πρέπει πάντα να είναι παρών ο πελάτης.

- **Διανομές**

Η μεθοδολογία RUP ορίζει ότι μία διανομή πρέπει να είναι μία σταθερή, εκτελέσιμη έκδοση του προϊόντος και να συνοδεύεται από τα απαραίτητα έγγραφα^[21]. Η μεθοδολογία XP ορίζει ότι μία έκδοση πρέπει να περιέχει της υλοποίηση ενός συνόλου από τις απαιτήσεις του πελάτη^[16]. Ωστόσο και οι δύο μεθοδολογίες ορίζουν ότι οι διανομές μπορεί να είναι τόσο εσωτερικές όσο και εξωτερικές, δηλαδή παραδοτέα στους πελάτες.

- **Τεχνική δουλειά**

Η μεθοδολογία XP εμπλέκει δύο πρακτικές, τη συλλογική ιδιοκτησία κώδικα και την ανακατασκευή κώδικα, που είναι στενά συνδεδεμένες, λόγω της συνεχής

ενοποίησης και των ελέγχων του κώδικα, που προσδίδουν ποιότητα στο λογισμικό. Στη μεθοδολογία RUP, διαφορετικά μέλη της ομάδας έργου είναι υπεύθυνα για διαφορετικά υποσυστήματα.

- **Εργαλεία**

Η διαδικασία RUP καθοδηγείται από ένα λογισμικό και διαθέτει κατάλληλα εργαλεία για διάφορες διαδικασίες, όπως για παράδειγμα τη μοντελοποίηση. Λόγω του ότι η μεθοδολογία είναι εκτενής και περίπλοκη, το λογισμικό αυτό πρέπει να καθοδηγεί απαραίτητα το χρήστη. Αυτός είναι και ένας λόγος για την εμπορική επιτυχία της RUP.

Η μεθοδολογία XP δεν προδικάζει τη χρήση κάποιου συγκεκριμένου εργαλείου. Υπάρχουν πολλά εργαλεία που προσφέρονται από την κοινότητα, όπως το JUnit που αναφέρθηκε και προηγουμένως, αλλά οποιοδήποτε είδους εργαλεία CASE ή διαχείρισης έργων λογισμικού μπορούν να χρησιμοποιηθούν για την XP.

Τελειώνοντας τη σύγκριση των δύο μεθοδολογιών, αξίζει να αναφερθούμε στα χαρακτηριστικά των ανθρώπων και των επιχειρήσεων που χρησιμοποιούν τις δύο μεθοδολογίες. Η μεθοδολογία XP στοχεύει στον ανεξάρτητο σχεδιαστή λογισμικού, εξουσιοδοτώντας το τεχνικό επίπεδο στην επιχείρηση. Βασίζεται στην απευθείας επικοινωνία και απαιτεί κουράγιο, αφού η εμπιστοσύνη των μελών της ομάδας έργου είναι σημαντική.

Από την άλλη μεριά, η μεθοδολογία RUP δεν εστιάζει στο μεμονωμένο σχεδιαστή λογισμικού, αλλά δίνει έμφαση στους ρόλους, που προσαρμόζονται στα συγκεκριμένα έργα λογισμικού. Προϋποθέτει τη συγγραφή αναφορών, που δίνει μία πίεση στο προσωπικό για την παραγωγή των εγγράφων αυτών.

ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΕΣ ΔΙΟΙΚΗΣΗΣ ΕΡΓΩΝ

3.1 Εισαγωγή

Ένα έργο είναι μια προσωρινή, μοναδική προσπάθεια που σκοπό έχει την παραγωγή ενός αποτελέσματος που προκαλεί θετική αλλαγή ή προστιθέμενη αξία στον χρήστη του. Η ιδιότητα αυτή, του προσωρινού και μοναδικού εγχειρήματος, έρχεται σε αντίθεση με τις δραστηριότητες οι οποίες για τον λόγο ότι είναι συστατικά εργασίας που εκτελούνται κατά τη διάρκεια του έργου^[38] είναι μόνιμες ή ημι-μόνιμες προοδευτικές και λειτουργικές εργασίες που παράγουν το ίδιο προϊόν ή υπηρεσία ξανά και ξανά. Έτσι, ο δύσκολος έλεγχος αυτών των δύο αντικρουόμενων εννοιών, έργο και δραστηριότητες, που συχνά απαιτεί διάφορες τεχνικές δεξιότητες και την ανάπτυξη ιδιαίτερης κάθε φορά φιλοσοφίας, οδήγησε στην θεμελίωση ενός νέου όρου τη «Διοίκηση Έργου»

Έτσι λοιπόν, Διοίκηση Έργου (Project Management, PM) καλείται η εφαρμογή γνώσεων, ικανοτήτων, εργαλείων και τεχνικών στις δραστηριότητες του έργου με τέτοιο τρόπο ώστε να ικανοποιούνται οι απαιτήσεις του έργου^[38]

Απώτερος στόχος της διοίκησης έργου είναι η μεγιστοποίηση της πιθανότητας δημιουργίας μιας αλυσίδας από επιτυχημένα διοικούμενα έργα, γεγονός που θα οδηγήσει τελικά σε ένα διατηρήσιμο ανταγωνιστικό πλεονέκτημα^[15]. Ο καλύτερος τρόπος για να συμβεί αυτό είναι μέσω της χρήσης μιας μεθοδολογίας διοίκησης έργου .

Με τον όρο μεθοδολογία (methodology) ονομάζουμε γενικότερα ένα σύνολο από οδηγίες ή αρχές που μπορεί να προσαρμοστεί και να εφαρμοστεί σε μια συγκεκριμένη κατάσταση ή αντικείμενο. Στο χώρο των έργων, αυτές οι οδηγίες μπορεί να περιλαμβάνουν μια λίστα με διάφορες εργασίες που πρέπει να γίνουν. Οι μεθοδολογίες κατατάσσονται σε κατηγορίες ανάλογα με το αντικείμενο που εξυπηρετούν. Στο χώρο των έργων κάνουμε λόγο για τις μεθοδολογίες έργων, ενώ άλλες σχετικές με αυτές σε έναν οργανισμό, είναι οι μεθοδολογίες ανεφοδιασμού, ανάπτυξης, που είναι περισσότερο τεχνικές και αναφέρονται στο σχεδιασμό και την ανάπτυξη των συστημάτων, υποστήριξης και μάρκετινγκ.

Η μεθοδολογία έχει την ευθύνη της διαχείρισης τριών περιοχών, της προόδου του έργου, του κύκλου ζωής του έργου και τέλος των πόρων και των επικοινωνιακών θεμάτων του έργου με άλλα στοιχεία. Μια τυπική μεθοδολογία έργου πρέπει πρώτα από όλα να καθοδηγεί την εργασία όλων των μελών της ομάδας έργου καθ' όλη τη διάρκεια του κύκλου ζωής του. Δεύτερον, πρέπει να ακολουθεί μια δομημένη, αλλά ευέλικτη προσέγγιση που θα προσδίδει ποιότητα και θα διασφαλίζει τη συνεχή εμπλοκή του πελάτη. Επιπλέον, είναι ζωτικής σημασίας να αποσαφηνιστεί ότι μια μεθοδολογία αποτελεί μέρος του συνολικού πλαισίου διοίκησης έργου της εταιρίας όπου πολλές ακόμη συσχετισμένες μεθοδολογίες διεκδικούν τους ίδιους πόρους και ενδιαφέρον με αυτήν. Αυτό είναι απόλυτα φυσικό, καθώς δεν είναι δυνατό μια και μόνο μεθοδολογία να επιλύσει κάθε έργο της εταιρίας. Επίσης, χρήσιμο είναι να τονισθεί ότι η μεθοδολογία έργου δεν είναι μια προσωρινή, γρήγορη λύση ανάγκης αλλά ούτε και συνταγή επιτυχίας για ένα έργο. Η μεθοδολογία μπορεί να θεωρηθεί ως ένα αποτελεσματικό όπλο της εταιρίας για να εξασφαλίσει συνέπεια και τυποποίηση σε όλο το εύρος της κατά τη διοίκηση του έργου ^[15].

Στο παρόν κεφάλαιο παρουσιάζονται οι δύο πιο ευρέως διαδεδομένες μεθοδολογίες διοίκησης έργων η Prince2 και η PMBOK και γίνεται μια σύγκριση μεταξύ τους.

3.2 Prince2

Η μεθοδολογία Prince και η μετεξέλιξη της σε Prince2 ^[11], είναι μια δομημένη μέθοδος με σκοπό την αποτελεσματική διοργάνωση ενός μεγάλου γεγονότος. Η μεθοδολογία Prince2 ^[31] αποτελεί την καταλληλότερη μεθοδολογία διοίκησης έργων, ιδιαίτερα όταν πρόκειται για έργα μεγάλης κλίμακας ή κρίσιμης σημασίας. Η διαπίστωση αυτή οδήγησε το 1989, τη βρετανική κυβέρνηση να υιοθετήσει τη μεθοδολογία Prince (ακρωνύμιο του Projects IN Controlled Environments - «Έργα σε Ελεγχόμενα Περιβάλλοντα») ως επίσημη de facto μεθοδολογία διοίκησης έργων για τα κρατικά έργα στο Ηνωμένο Βασίλειο.

Η μεθοδολογία Prince2 είναι το βρετανικό πρότυπο που υποστηρίζεται από το OGC και είναι ελεύθερα διαθέσιμο. Παρά του ότι το Prince2 είναι μια μεθοδολογία

διαχείρισης έργων, σχετίζει την υλοποίηση ενός έργου με τον φορέα υλοποίησής του τόσο οργανωτικά όσο και επιχειρησιακά ^[17].

Σταδιακά, η πρακτική χρησιμότητα και οι ωφέλειες από τη χρήση της οδήγησαν στη διάδοση πέρα από τον κρατικό, δημόσιο τομέα και στην υιοθέτησή της από τον ιδιωτικό τομέα. Σήμερα αναγνωρίζεται διεθνώς ως βέλτιστη πρακτική διοίκησης έργων, με αποτέλεσμα να παγιώνεται η χρήση της και σε χώρες εκτός Ηνωμένου Βασιλείου, όπως στην Ολλανδία, στο Βέλγιο, στη Γερμανία, στην Ισπανία, στη Νότια Αφρική, στην Αυστραλία και στις ΗΠΑ. Όλο και περισσότερες εταιρείες υιοθετούν αυτήν την μεθοδολογία ως την κύρια προσέγγιση του έργου.

Η μεθοδολογία Prince2 υιοθετεί τις αρχές της δομημένης διαχείρισης, δηλαδή της διοίκησης του έργου με ένα λογικό, κατάλληλα οργανωμένο τρόπο, ακολουθώντας συγκεκριμένα διαδικαστικά «βήματα» συστηματικής προσέγγισης και με μια τεκμηριωμένη περιγραφή. Η μεθοδολογία Prince2 επιδεικνύει ένα θετικότατο απολογισμό στα πάνω από 10 έτη από την υιοθέτηση και χρήση της τόσο στο δημόσιο όσο και στον ιδιωτικό τομέα. Ειδικότερα, η Prince2:

- Προσφέρει μια «κοινή γλώσσα συνεννόησης» μεταξύ όλων των εμπλεκόμενων μερών του έργου συμβάλλοντας πρακτικά στην αποφυγή παρερμηνειών αναφορικά με τους επιδιωκόμενους ενδιάμεσους και τελικούς στόχους του έργου.
- Ενσωματώνει όλη τη διοικητική πληροφορία που παράγεται κατά τη διάρκεια του έργου, ως εισροή ή εκροή των εφαρμοζόμενων διαδικασιών.
- Συμβάλλει στη διαρκή βελτίωση της διοίκησης των έργων ενός οργανισμού. Η «Αναφορά της Καταγεγραμμένης Εμπειρίας» η οποία περιγράφει την εμπειρία (τα «μαθήματα») που αποκομίστηκε κατά την υλοποίηση του έργου, ώστε τόσο τα θετικά όσο και τα αρνητικά αποτελέσματα να τεκμηριώνονται για να αποτελέσουν χρήσιμο οδηγό σε μελλοντικά έργα.

Τα κύρια χαρακτηριστικά της είναι ο σχεδιασμός με βάση το παραγόμενο προϊόν-υπηρεσία, ο διαχωρισμός του έργου σε ελεγχόμενα στάδια και η δυνατότητα προσαρμογής της στο επίπεδο του έργου. Οι κύριες διαδικασίες είναι οι εξής:

- Έναρξη (σωστή αρχή).

- Όρια των φάσεων (δέσμευση για περαιτέρω εργασία μετά τον έλεγχο των μέχρι τώρα αποτελεσμάτων).
- Ad-hoc direction (έλεγχος προόδου, παροχή συμβουλών και οδηγιών).
- Κλείσιμο έργου (επιβεβαίωση των επιθυμητών αποτελεσμάτων του έργου και ελεγχόμενο κλείσιμο του).

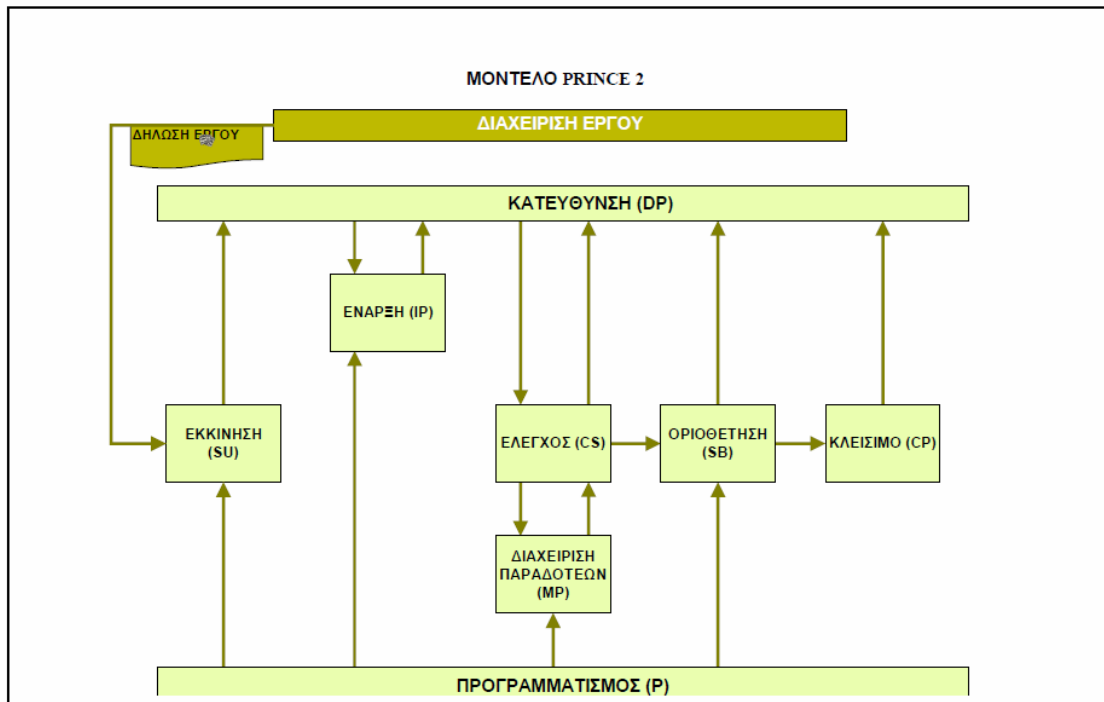
Όπως φαίνονται και στο παρακάτω σχήμα, αναλυτικά περιλαμβάνει τις παρακάτω βασικές λειτουργικές διαδικασίες ενός έργου:

- **Εκκίνηση:** διασφαλίζει τις προϋποθέσεις έναρξης και περιέχει τη γραπτή διακήρυξη – δήλωση του έργου. Η πρώτη διαδικασία της Prince2 είναι διαδικασία πριν από το έργο (preproject) και σχεδιάστηκε για να εξασφαλίσει ότι οι προϋποθέσεις για την έναρξη είναι σε ισχύ. Η διαδικασία αυτή ορίζει, σε αφαιρετικό επίπεδο, τους λόγους για την υλοποίηση του έργου καθώς και ποια είναι τα προσδοκώμενα αποτελέσματα. Η φάση αυτή του έργου περιλαμβάνει τα εξής:
 - Εξασφάλιση ότι οι απαραίτητες πληροφορίες για την ομάδα έργου είναι διαθέσιμες.
 - Σχεδιασμός και συγκρότηση της ομάδας διαχείρισης έργου.
 - Απόφαση για τη μέθοδο που θα υιοθετηθεί μέσα στο έργο για να δοθεί λύση.
 - Να εξασφαλιστεί ότι οι στόχοι του έργου είναι γνωστοί.
 - Τεκμηρίωση και επιβεβαίωση της ύπαρξης business case για το έργο.
 - Η παροχή της κατευθυντήριας γραμμής για τις διαδικασίες λήψης αποφάσεων που απαιτούνται κατά τη διάρκεια ζωής του έργου.
 - Η εξασφάλιση ότι η επένδυση του χρόνου και της προσπάθειας που απαιτούνται για το έργο έγινε σοφά λαμβάνοντας υπ' όψιν τους κινδύνους του έργου.
 - Να προγραμματιστεί η εργασία που απαιτείται για να σχεδιαστεί η Prince2 σύμβαση μεταξύ του πελάτη και του προμηθευτή.

- **Έναρξη:** περιλαμβάνει τις συμφωνίες – συμβόλαια με διευθύνοντες, χορηγούς, διαχειριστές και εγκαθιστά ένα καθεστώς ελέγχου. Οι λειτουργίες που επιτελούνται σε αυτό το στάδιο είναι οι εξής:
 - Καθιέρωση υγιούς διοικητικής βάσης
 - Το διοικητικό προϊόν που δημιουργείται είναι το έγγραφο έναρξης έργου, που είναι η βασική γραμμή έναντι της οποίας αξιολογείται η πρόοδος και η επιτυχία του έργου.
 - Προετοιμάζει τις πληροφορίες που δείχνουν ότι υπάρχει ικανοποιητική αιτιολόγηση για τη συνέχεια του έργου.
 - Δημιουργεί ένα λεπτομερές σχέδιο για το μέρος του έργου που η διοίκηση είναι σε θέση να εγκρίνει.
- **Διαχείριση παραδοτέων:** ικανοποίηση παράδοσης παραδοτέων σύμφωνα με τα προκαθορισμένα ποιοτικά κριτήρια. Στόχος της διαδικασίας αυτής είναι η εξασφάλιση ότι τα προγραμματισμένα προϊόντα δημιουργούνται και παραδίδονται. Αυτό γίνεται με:
 - Εξασφάλιση ότι η εργασία που έχει ανατεθεί στην ομάδα έχει εγκριθεί και συμφωνηθεί.
 - Προγραμματισμός της εργασίας της ομάδας.
 - Εξασφάλιση ότι η εργασία εκτελείται.
 - Τακτική αξιολόγηση της προόδου της εργασίας και των προβλέψεων.
 - Λήψη αποδοχής για τα ολοκληρωμένα προϊόντα.
 - Εξασφάλιση ότι τα προϊόντα ικανοποιούν τα συμφωνημένα κριτήρια ποιότητας.
 - Έκθεση αναφοράς προόδου και ποιότητας προς τον υπεύθυνο έργου.
- **Οριοθέτηση:** οριοθετείται η βιωσιμότητα και εξασφαλίζεται η υλοποίηση των παραδοτέων κάθε ολοκληρωμένου τμήματος του έργου. Αυτή η διαδικασία παρέχει στην ομάδα διοίκησης έργου σημεία απόφασης για την συνέχιση ή όχι του έργου. Οι στόχοι της διαδικασίας είναι:

- Να διασφαλίσει η ομάδα διοίκησης έργου ότι όλα τα προϊόντα που ήταν προγραμματισμένα γι' αυτό το στάδιο έχουν ολοκληρωθεί.
 - Η παροχή των απαραίτητων πληροφοριών στην ομάδα διοίκησης έργου έτσι ώστε να αξιολογήσουν την βιωσιμότητα της συνέχισης του έργου.
 - Η παροχή των απαραίτητων πληροφοριών στην ομάδα διοίκησης έργου για την έγκριση της ολοκλήρωσης της τρέχουσας φάσης και την έναρξη της επόμενης.
 - Καταγραφή μετρήσεων που θα μπορούσαν να βοηθήσουν σε επόμενες φάσεις.
- **Έλεγχος:** παρακολούθηση κι έλεγχος των διαδικασιών διαχείρισης περιλαμβάνοντας την εγγύηση ότι το έργο εξελίσσεται σύμφωνα με τον προγραμματισμό και αντιδρά σε απρόσμενα γεγονότα.
 - **Κλείσιμο:** ελεγχόμενη περάτωση του έργου με ενημέρωση του οργανισμού που το διαχειρίζεται. Στόχος της διαδικασίας είναι η εκτέλεση προγραμματισμένου κλεισίματος του έργου. Αυτό γίνεται με αίτηση του υπεύθυνου έργου προς την επιτροπή διοίκησης έργου να δοθεί η άδεια κλεισίματος του έργου είτε στο φυσικό του τέλος είτε πρόωρα, αν έχει ζητηθεί από την επιτροπή διοίκησης. Ο σκοπός είναι να:
 - Διαπιστωθεί η έκταση ικανοποίησης των αρχικών αντικειμενικών σκοπών του έργου.
 - Επιβεβαιωθεί η ικανοποίηση του πελάτη από τα προϊόντα.
 - Εξασφαλισθεί μέχρι ποιο σημείο όλα τα αναμενόμενα προϊόντα έχουν παραδοθεί και έχουν γίνει αποδεκτά.
 - Εξασφαλιστεί ότι η εμπειρία που αποκτήθηκε καταγράφηκε για να αξιοποιηθεί σε επόμενα έργα.
 - Καταγραφεί αν η δραστηριότητα της διοίκησης του έργου ήταν επιτυχής ή όχι.
 - Γίνουν τυχόν συστάσεις για επόμενες ενέργειες.
 - Επιβεβαιωθεί ότι έχουν διευθετηθεί η συντήρηση και η εγγύηση.

- **Προγραμματισμός:** μια επαναλαμβανόμενη διαδικασία που διαδραματίζει σημαντικό ρόλο στην εκκίνηση και στην εξέλιξη του έργου. Ο προγραμματισμός είναι επαναλαμβανόμενη διαδικασία που χρησιμοποιείται από άλλες διαδικασίες όταν απαιτείται ένα πλάνο. Η διαδικασία χρησιμοποιεί την τεχνική της Prince2 πλάνου βασισμένου σε προϊόν και καλύπτει:
 - Σχεδιασμό πλάνου.
 - Καθορισμός των απαραίτητων δραστηριοτήτων και εξαρτήσεων.
 - Εκτίμηση της απαιτούμενης προσπάθειας.
 - Χρονοπρογραμματισμός των διαθέσιμων πόρων.
 - Ανάλυση του κινδύνου.
 - Επιπλέον περιγραφή του πλάνου, των υποθέσεων του και των βημάτων ποιότητας.
- **Κατεύθυνση:** διενεργείται καθ' όλη την διάρκεια του έργου και απευθύνεται στο Διοικητικό Συμβούλιο. Προορίζεται για τους βασικούς λήπτες αποφάσεων – αρμόδια ομάδα ανώτερης διαχείρισης του έργου. Οι οποίοι είναι συνήθως πολυάσχολοι άνθρωποι και πρέπει να αναμειχθούν μόνο στη διαδικασία λήψης αποφάσεων. Η Prince2 υιοθετεί τη διαχείριση υπό εξαίρεση. Η διαδικασία αυτή καλύπτει τα βήματα που πρέπει να εκτελεστούν από την ομάδα ανώτερης διαχείρισης σε όλο το φάσμα του έργου:
 - Έγκριση προετοιμασίας πλάνου έργου και επιχειρησιακής υπόθεσης (business case) για το έργο.
 - Έγκριση άδειας έναρξης έργου.
 - Έλεγχος ότι το έργο εξακολουθεί να είναι δικαιολογημένο σε βασικά σημεία του κύκλου ζωής του.
 - Παρακολούθηση προόδου και παροχή συμβουλών όπου απαιτείται.
 - Εξασφάλιση ότι το έργο έχει ελεγχόμενο τέλος.



Εικόνα 5: Οι βασικές λειτουργικές διαδικασίες του μοντέλου Prince2

Η μέθοδος αυτή προσφέρει οφέλη τόσο στους διευθύνοντες του έργου όσο και στον ίδιο τον οργανισμό, στο πλαίσιο του οποίου δημιουργείται το έργο, μέσω της ελεγχόμενης χρήσης των πηγών του και της ικανότητας διαχείρισης των εργασιών και των κινδύνων περισσότερο αποτελεσματικά.

Τα πλεονεκτήματα της μεθοδολογίας Prince2 είναι τα εξής:

- Η μέθοδος αυτή εφαρμόζεται ανεξάρτητα από το είδος του έργου (π.χ. IT, marketing).
- Οι υπηρεσίες (εκπαίδευση, συμβουλές και εργαλεία) μπορούν να αποκτηθούν από διάφορους ανεξάρτητους προμηθευτές.
- Η μέθοδος έχει ενεργές ομάδες χρηστών στην Αγγλία και την Ολλανδία.
- Η μέθοδος μπορεί να εφαρμοσθεί σε μικρά και μεγάλα έργα.
- Η μέθοδος εστιάζει στα αποτελέσματα του έργου από την άποψη του καθορισμένου χρόνου , κόστους , ποιότητας και των παραμέτρων λειτουργίας επίσης επικεντρώνεται στη business case και στα οφέλη που αποφέρει το έργο.

- Η μέθοδος περιέχει διαχείριση αλλαγής που ελέγχει το μεταβαλλόμενο περιβάλλον.

Τα μειονεκτήματα της μεθοδολογίας Prince2 είναι τα εξής:

- Είναι μια μέθοδος και όχι μια λύση για οποιαδήποτε έργο. Οι άνθρωποι που χρησιμοποιούν Prince2 πρέπει να συνεχίσουν να σκέφτονται.
- Μερικοί άνθρωποι εφαρμόζουν και ερμηνεύουν τη μεθοδολογία με έναν άκαμπτο τρόπο και δεν την εφαρμόζουν στο πρόγραμμα.

3.3 PMBOK

Η μεθοδολογία PMBOK^[39] ή αλλιώς Project Management Body of Knowledge είναι μια συλλογή από διαδικασίες και γνωστικές περιοχές, που θεωρούνται ως οι πλέον καλύτερες τεχνικές για τη διοίκηση.

Ως ένα διεθνώς αναγνωρισμένο πρότυπο (IEEE Std 1490-2003), ορίζει τα βασικά στοιχεία της διοίκησης έργων, ανεξάρτητα από τον τύπο του έργου, δηλαδή αν πρόκειται για λογισμικό, κατασκευαστικό μηχανικό κλπ.

Η PMBOK (Guide to the Project Management Body of Knowledge) είναι μια συλλογή διεργασιών και κοινά αποδεκτής γνώσης ως καλή πρακτική στο επιστημονικό πεδίο της Διαχείρισης Έργων. Ο χαρακτηρισμός κοινά αποδεκτή γνώση έχει την έννοια πως η σχετική πληροφορία μπορεί να εφαρμοσθεί στα περισσότερα έργα τόσο στον παρόν όσο και στο μέλλον και υπάρχει μια ευρεία ομοφωνία για τη χρησιμότητα της. Επίσης ο όρος καλή πρακτική αναφέρεται στη γενικότερη συμφωνία για τα θετικά αποτελέσματα που μπορεί να έχει σε διάφορα έργα η κατάλληλη εφαρμογή των πρακτικών της PMBOK. Ταυτόχρονα όντας καθιερωμένο ως διεθνές πρότυπο παρέχει τις γενικές βασικές αρχές στη Διαχείριση Έργων, χωρίς να περιορίζεται σε κάποιο είδος συγκεκριμένα, αλλά αφορά διάφορους τύπους έργων, όπως λογισμικό, μηχανική, αυτοματισμοί κ.α. Η PMBOK αναγνωρίζει 5 βασικές φάσεις:

1. **Έναρξη:** επιτρέπει στην εταιρεία να ξεκινήσει την επόμενη φάση του έργου.
2. **Προγραμματισμός:** Ο προγραμματισμός είναι πολύ σημαντικός σε ένα έργο καθώς περιλαμβάνει την πραγματοποίηση κάτι το οποίο δεν έχει γίνει ποτέ πριν. Επομένως υπάρχουν σχετικά περισσότερες διαδικασίες σε αυτόν τον τομέα.

Οι διαδικασίες χωρίζονται σε 2 κατηγορίες: τις Κύριες διαδικασίες (Core processes) και τις διαδικασίες Διευκόλυνσης (Facilitating processes):

- Κύριες διαδικασίες: Ορισμένες διαδικασίες προγραμματισμού έχουν σαφή εξάρτηση και η εκτέλεση τους απαιτείται να γίνει με την ίδια σειρά στα περισσότερα έργα. Οι κύριες διαδικασίες μπορεί να επαναλαμβάνονται πολλές φορές κατά τη διάρκεια μιας φάσης του έργου. Αυτές περιέχουν:
 1. Προγραμματισμό Πεδίου – ανάπτυξη μιας γραπτής δήλωσης πεδίου ως βάση για μελλοντικές αποφάσεις έργου.
 2. Ορισμό Πεδίου – διαίρεση των βασικών προϊόντων έργου σε μικρότερα, πιο εύκολα στη διαχείριση τμήματα.
 3. Ορισμό Δραστηριότητας – προσδιορισμός συγκεκριμένων δραστηριοτήτων που πρέπει να πραγματοποιηθούν για την παραγωγή των διάφορων προϊόντων.
 4. Δραστηριότητα Αλληλουχίας – προσδιορισμός και τεκμηρίωση των εξαρτήσεων αλληλεπίδρασης.
 5. Δραστηριότητα Εκτίμησης Διάρκειας – εκτίμηση του αριθμού των περιόδων εργασίας που θα χρειαστούν για την ολοκλήρωση μεμονωμένων δραστηριοτήτων.
 6. Ανάπτυξη προγράμματος – ανάλυση αλληλουχίας δραστηριοτήτων, διάρκειας δραστηριοτήτων και απαιτούμενων πόρων για τη δημιουργία του προγράμματος έργου.
 7. Προγραμματισμό Πόρων – ορίζει ποιοι πόροι (άνθρωποι, εξοπλισμός) και σε τι ποσότητες από καθέναν θα χρησιμοποιηθεί για την διεξαγωγή των δραστηριοτήτων του έργου.
 8. Εκτίμηση Κόστους – ανάπτυξη μιας εκτίμησης του κόστους των πόρων που είναι απαραίτητοι για την ολοκλήρωση των δραστηριοτήτων του έργου.
 9. Σύνταξη προϋπολογισμού δαπανών – κατανομή του συνολικού εκτιμώμενου κόστους σε μεμονωμένα θέματα εργασίας.

10. Ανάπτυξη πλάνου έργου – συλλογή των αποτελεσμάτων των άλλων διαδικασιών προγραμματισμού και χρήση αυτών για τη σύνταξη ενός ενιαίου και συνεπούς εγγράφου.

- Διαδικασίες Διευκόλυνσης: Εκτελούνται περιοδικά και όπως απαιτείται κατά τη διάρκεια του προγραμματισμού του έργου, δεν είναι προαιρετικές. Αυτές περιέχουν:

1. Προγραμματισμό ποιότητας – προσδιορισμός των standard (προτύπων) ποιότητας που είναι σχετικά με το έργο και ορισμός του πως θα ικανοποιηθούν.
2. Οργανωτικό προγραμματισμό – προσδιορισμός, τεκμηρίωση και ανάθεση ρόλων, ευθυνών και σχέσεων.
3. Απόκτηση Προσωπικού – απόκτηση των απαραίτητων ανθρώπινων πόρων για την ανάθεση και την εργασία πάνω στο έργο.
4. Προγραμματισμό Επικοινωνιών – προσδιορισμός των αναγκαίων πληροφοριών και επικοινωνιών για τους συμμετέχοντες: ποιος χρειάζεται τι πληροφορίες, πότε τις χρειάζεται και πως θα του δοθούν.
5. Προσδιορισμό κινδύνου – προσδιορισμός των κινδύνων που πιθανών να επηρεάσει το έργο.
6. Ποσοτικοποίηση του κινδύνου – αξιολόγηση των κινδύνων και των αλληλεπιδράσεων αυτών.
7. Ανάπτυξη αντίδρασης στον κίνδυνο – βελτιωμένα βήματα για την αντίδραση σε απειλές.
8. Προγραμματισμό Προμηθειών – ορισμός του τι και πότε θα παραχθεί.
9. Προγραμματισμό επιλογής πηγών – καταγραφή απαιτήσεων του προϊόντος και προσδιορισμός πιθανών πόρων.

3. **Εκτέλεση:** Περιέχει τις εξής διαδικασίες:

- Εκτέλεση πλάνων Έργου.
- Επαλήθευση Πεδίου – διαμόρφωση αποδοχής του πεδίου έργου.

- Διασφάλιση Ποιότητας – αξιολόγηση της συνολικής απόδοσης έργου σε μια καθορισμένη βάση για την παροχή εμπιστοσύνης ότι το έργο θα ικανοποιήσει τα σχετικά standard ποιότητας.
 - Ανάπτυξη Ομάδας – ανάπτυξη μεμονωμένων και ομαδικών δεξιοτήτων για τη βελτίωση της απόδοσης του έργου.
 - Κατανομή Πληροφοριών – έγκαιρη διάθεση πληροφοριών στους συμμετέχοντες του έργου.
 - Έρευνα πόρων – μέσω προσφορών και προτάσεων.
 - Επιλογή πόρων – επιλογή από τους πιθανούς προμηθευτές.
 - Διαχείριση συμβολαίου – διαχείριση των σχέσεων με τους προμηθευτές.
4. **Έλεγχος:** Η απόδοση του έργου πρέπει να μετράται τακτικά για την ανίχνευση αποκλίσεων από το πλάνο. Η διαδικασίες αυτές βρίσκονται σε διάφορες περιοχές γνώσης. Στο βαθμό που σημαντικές αποκλίσεις παρατηρούνται (όπως για παράδειγμα αυτές που θέτουν σε κίνδυνο τους στόχους του έργου), οι ρυθμίσεις του πλάνου γίνονται με την επανάληψη των κατάλληλων διαδικασιών προγραμματισμού. Για παράδειγμα, μια δραστηριότητα η οποία δεν ολοκληρώθηκε εγκαίρως μπορεί να απαιτεί ρυθμίσεις στο τρέχων πλάνο επάνδρωσης, στην εμπιστοσύνη στις υπερωρίες, ή στην ανταλλαγή μεταξύ του προϋπολογισμού και του προγραμματισμού στόχων. Ο έλεγχος επίσης περιλαμβάνει τη λήψη προληπτικών μέτρων εν αναμονή πιθανών προβλημάτων.

Οι διαδικασίες ελέγχου περιέχουν τις εξής διαδικασίες:

- Συνολικός Έλεγχος Αλλαγών – συντονισμός αλλαγών σε ολόκληρο το έργο.
- Έλεγχος Αλλαγών Πεδίου – έλεγχος αλλαγών στο πεδίο του έργου.
- Έλεγχος Προγραμματισμού – έλεγχος αλλαγών στον προγραμματισμό του έργου.
- Έλεγχος Κόστους – έλεγχος αλλαγών στον προϋπολογισμό του έργου.
- Έλεγχος Ποιότητας – έλεγχος συγκεκριμένων αποτελεσμάτων έργου για τον προσδιορισμό του αν ακολουθούν τα σχετικά standards ποιότητας και προσδιορισμός τρόπων για την ελαχιστοποίηση αιτιών της μη ικανοποιητικής απόδοσης.

- Αναφορά Απόδοσης – συλλογή και διάδοση πληροφοριών απόδοσης. Αυτή περιέχει αναφορά κατάστασης, μετρήσεις προόδου και προβλέψεις.
- Έλεγχος Αντίδρασης στον Κίνδυνο - αντίδραση στους κινδύνους που εμφανίζονται κατά τη διάρκεια του έργου.

5. **Κλείσιμο:** Περιέχει τις εξής διαδικασίες:

- Ελεγχόμενη Περάτωση – παραγωγή, συλλογή και διάδοση πληροφοριών για τη διαμόρφωση φάσης ή την ολοκλήρωση του έργου.
- Εξωτερικές Συμβάσεις – ολοκλήρωση και τακτοποίησης της σύμβασης συμπεριλαμβανομένων της επίλυσης οποιωνδήποτε ανοικτών θεμάτων.

Οι διαδικασίες αλληλεπιδρούν / συνεργάζονται μέσω ενός έργου ή μιας φάσης. Οι διαδικασίες περιγράφονται με τη βοήθεια των παρακάτω όρων:

- Είσοδος (αρχεία, σχέδια, πλάνα κλπ)
- Εργαλεία και Τεχνικές (μηχανισμοί που εφαρμόζονται στην είσοδο)
- Έξοδος (αρχεία, προϊόντα κλπ)

Οι εννιά θεματικές περιοχές γνώσης, όπου κάθε περιοχή μπορεί να περιέχει ένα μέρος ή ακόμα και όλες τις βασικές διεργασίες, είναι:

1. **Διαχείριση ενοποίησης έργου (Project Integration Management)**

Περιγράφει τις διαδικασίες που απαιτούνται για να εξασφαλισθεί ότι τα διάφορα τμήματα του έργου είναι συντονισμένα. Χρησιμοποιείται η έννοια της ανταλλαγής πόρων (tradeoffs) μεταξύ ανταγωνιστικών στόχων και εναλλακτικών λύσεων, ώστε να καλυφθούν οι ανάγκες και οι προσδοκίες των συμμετεχόντων του έργου.

2. **Διαχείριση πεδίου έργου (Project Scope Management)**

Περιγράφει τις διαδικασίες που απαιτούνται για να εξασφαλισθεί ότι το έργο περιέχει όλη την απαραίτητη εργασία και μόνο αυτή για την επιτυχημένη ολοκλήρωση του έργου. Κυρίως ασχολείται με τον εντοπισμό και τον έλεγχο των στοιχείων που απαιτούνται αλλά και αυτών που δεν απαιτούνται για την ολοκλήρωση του έργου.

3. **Διαχείριση χρόνου έργου (Project Time Management)**

Περιγράφει τις διαδικασίες που απαιτούνται για την εξασφάλιση της έγκαιρης περάτωσης του έργου.

4. Διαχείριση κόστους έργου (Project Cost Management)

Περιγράφει τις διαδικασίες που είναι απαραίτητες για να εξασφαλισθεί ότι το έργο ολοκληρώθηκε στα πλαίσια του εγκεκριμένου προϋπολογισμού.

5. Διαχείριση ποιότητας του έργου (Project Quality Management)

Περιγράφει τις διαδικασίες που είναι απαραίτητες για να εξασφαλισθεί ότι το έργο θα ικανοποιήσει τις ανάγκες για τις οποίες αναλήφθηκε.

6. Διαχείριση ανθρώπινου δυναμικού του έργου (Project Human Resource Management)

Περιγράφει τις διαδικασίες που απαιτούνται για να γίνει η πλέον αποτελεσματική χρησιμοποίηση των ανθρώπων που εμπλέκονται στο έργο.

7. Διαχείριση επικοινωνίας των εμπλεκόμενων στο έργο (Project Communication Management)

Περιγράφει τις διαδικασίες έγκαιρης και κατάλληλης παραγωγής, συλλογής, διάδοσης, αποθήκευσης, και τέλος διάθεσης των πληροφοριών του έργου.

8. Διαχείριση κινδύνου του έργου (Project Risk Management)

Περιγράφει τις διαδικασίες που αφορούν τον προσδιορισμό, την ανάλυση και την αντίδραση στους κινδύνους του έργου.

9. Διαχείριση προμηθειών του έργου (Project Procurement Management)

Περιγράφει τις διαδικασίες που απαιτούνται για την απόκτηση αγαθών και υπηρεσιών από πηγές που βρίσκονται εκτός της ομάδας εκτέλεσης του έργου ή εκτός του οργανισμού. Ενσωμάτωση Έργου (Project Integration Management)

Όλες οι γνωστικές περιοχές αποτελούνται από ορισμένες ή όλες τις τεχνικές διοίκησης. Για παράδειγμα, η γνωστική περιοχή Προμήθεια Έργου αποτελείται από:

- Σχεδιασμό Προμήθειας (Procurement Planning)
- Σχεδιασμό Ζήτησης (Solicitation Planning)
- Ζήτηση (Solicitation)

- Επιλογή υλικών (Source Selection)
- Συμβόλαιο διαχείρισης (Contract Administration)

Οι συμμετέχοντες στο έργο είναι άτομα και οργανισμοί οι οποίοι συμμετέχουν ενεργά στο έργο ή τα συμφέροντά τους μπορούν να επηρεαστούν είτε θετικά είτε αρνητικά από την εκτέλεση του προγράμματος ή την επιτυχή περάτωση του έργου. Η ομάδα διαχείρισης έργου πρέπει να προσδιορίσει τους συμμετέχοντες, να ορίσει τις ανάγκες τους και τις προσδοκίες τους και στη συνέχεια διαχειρίζεται και επηρεάζει αυτές τις προσδοκίες για να εξασφαλισθεί η επιτυχία του έργου. Ο προσδιορισμός των συμμετεχόντων είναι συχνά δύσκολος.

Οι βασικοί συμμετέχοντες σε κάθε έργο είναι:

- Project manager: το άτομο που είναι αρμόδιο για τη διαχείριση του έργου.
- Πελάτης: το άτομο ή ο οργανισμός το οποίο θα χρησιμοποιήσει το προϊόν του έργου.
- Η εταιρεία εκτέλεσης έργου.
- Χρηματοδότης: το άτομο ή η ομάδα που μαζί με την εταιρεία εκτέλεσης έργου παρέχουν οικονομικούς πόρους, σε μετρητά ή σε είδος, για το έργο.

3.4 Σύγκριση Prince2 και PMBOK

Όπως θα περίμενε κανείς, είναι προφανές ότι τα γνωστικά αντικείμενα της PMBOK και οι διαδικασίες και τα συστατικά της Prince2 καλύπτουν πολλά κοινά θέματα. Εκτός από ορισμένες διαφορές στην ορολογία είναι σαφές ότι και οι δύο χρησιμοποιούν «βέλτιστες πρακτικές» με τη διαφορά να βρίσκεται στη υλοποίηση ως μεθοδολογία.

Η Prince2 έχει το πλεονέκτημα ότι (επειδή είναι κάπως περιοριστική) προκαλεί ένα βαθμό τυποποίησης σε έναν οργανισμό. Έτσι ενώ επιτρέπει μικρές διαφοροποιήσεις από έργο σε έργο όλα τα έργα πρέπει να ακολουθήσουν συγκεκριμένα βήματα και συγκεκριμένη ορολογία. Αυτό έχει προφανή οφέλη στην εταιρική διαχείριση των έργων, τα προγράμματα κατάρτισης του προσωπικού των έργων, την απόδοση των έργων και την παρακολούθηση των συστημάτων. Το

μειονέκτημα είναι ότι μπορεί ενδεχομένως να περιορίζει τη δημιουργικότητα στην ποικιλία των μεθόδων που εφαρμόζονται για τη διαχείριση ενός έργου. Η PMBOK από την άλλη στην περιγραφή του έργου εξηγεί ότι «τα έργα συχνά είναι το μέσο για να επιτευχθούν τα στρατηγικά σχέδια ενός οργανισμού». Η PMBOK είναι σχεδιασμένη λαμβάνοντας υπόψη την οπτική γωνία από την πλευρά του κατόχου του έργου και όχι του προμηθευτή. Ως εκ τούτου, η PMBOK καλύπτει περισσότερο έδαφος από ότι η Prince2. Παρ'όλα αυτά, η Prince2 παρέχει μία ισχυρή-ευκολη να εφαρμοστεί μεθοδολογία για χρήση στα περισσότερα έργα.

Συνοψίζοντας μπορούμε να πούμε ότι η PMBOK σε αντίθεση με το Prince2 προσφέρει στον διαχειριστή του έργου μια σημαντική συλλογή από πληροφορίες σχετικά με τις αποδεδειγμένες πρακτικές στον τομέα αυτό και καλεί τον διαχειριστή του έργου να τις εφαρμόσει όταν το κρίνει σκόπιμο, ενώ η μεθοδολογία Prince2 παρέχει στον διαχειριστή του έργου και στις ομάδες του έργου μια πιο καθορισμένη (αν και πιο ευέλικτη) σειρά βημάτων. Η PMBOK είναι η καλύτερη προσέγγιση για να μάθουμε τις πληροφορίες που χρειάζονται για κάθε γνωσιακή περιοχή αλλά δεν είναι τόσο αποτελεσματική όσο η Prince2 στο να παρέχει καθοδήγηση για το πώς θα διαχειριστούμε ένα έργο. Με απλά λόγια σύμφωνα με ορισμένους η Prince2 περιγράφει τι πρέπει να κάνει ο διαχειριστής ενός έργου ενώ η PMBOK περιγράφει τι πρέπει να γνωρίζει.

Πολλοί οργανισμοί έχουν αναγνωρίσει ότι υπάρχουν οφέλη από την συνύπαρξη των μεθοδολογιών PMBOK και Prince2. Η PMBOK παρέχει την ακαδημαϊκή γνώση που είναι χρήσιμη για την ενίσχυση της επαγγελματικής διαχείρισης έργου. Η Prince2 παρέχει ένα εύχρηστο μοντέλο που μπορεί να εφαρμοστεί άμεσα από τους διαχειριστές του έργου και τις ομάδες ανεξάρτητα από το γνωσιακό υπόβαθρο της κάθε ομάδας για να παρέχουν σωστή διαχείριση του έργου και παραδοτέα που πληρούν τις προδιαγραφές που τέθηκαν στην αρχή του έργου.^[37]

ΚΕΦΑΛΑΙΟ 4: ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΡΓΩΝ

4.1 Εισαγωγή

Μόλις ολοκληρωθεί η φάση έναρξης του έργου, η ομάδα έργου, συνήθως ο διαχειριστής του έργου και οι αναλυτές στα πρώτα στάδια, πρέπει να καθορίσουν την προσπάθεια που πρέπει να καταβληθεί για να ολοκληρωθούν οι απαραίτητες εργασίες. Υπάρχουν πολλές διαθέσιμες μέθοδοι για να επιτευχθεί αυτός ο στόχος. Πολλές από αυτές χρησιμοποιούν γραφήματα διαφόρων τύπων, αλλά όλες απαιτούν τις ίδιες βασικές πληροφορίες.

Τα προβλήματα που συναντάει κανείς συνήθως στην εκτέλεση ενός σύνθετου έργου έχουν να κάνουν με το μεγάλο πλήθος των επιμέρους δραστηριοτήτων που απαιτούνται για την ολοκλήρωσή του, καθώς και από την διαπλοκή μεταξύ αυτών των επιμέρους δραστηριοτήτων. Οι δραστηριότητες αυτές διασυνδέονται πάντοτε μεταξύ τους με τεχνολογικές, φυσικές, οικονομικές ή άλλες σχέσεις προτεραιότητας, ενώ υπόκεινται σε διάφορους περιορισμούς. Έχουν λοιπόν αναπτυχθεί μέθοδοι και εργαλεία για την επίλυση των προβλημάτων που ανακύπτουν από τις διασυνδέσεις των επιμέρους δραστηριοτήτων σε ένα σύνθετο έργο, όπως η Μέθοδος της Κρίσιμης Διαδρομής (Critical Path method – CPM) ή η Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (Programme Evaluation and Review Technique – PERT). Μέσα από αυτές τις μεθόδους ωστόσο το ζητούμενο πάντοτε ήταν οι απαντήσεις σε προβλήματα κατά την εκτέλεση σύνθετων έργων (projects), όπως η ελαχιστοποίηση του συνολικού χρόνου εκτέλεσης του έργου, η ελαχιστοποίηση του συνολικού κόστους, η ελαχιστοποίηση του κόστους για ένα δεδομένο ολικό χρόνο, η ελαχιστοποίηση του χρόνου εκτέλεσης για ένα δεδομένο κόστος, η ελαχιστοποίηση των πόρων που αδρανούν.

4.2 Δομή Ανάλυσης Εργασιών

Η ανάπτυξη ενός σχεδίου έργου έχει ως βασική προϋπόθεση ότι έχουμε μια σαφή και λεπτομερή κατανόηση των απαιτούμενων εργασιών, του προβλεπόμενου χρονικού διαστήματος που κάθε δραστηριότητα υπολογίζεται ότι θα διαρκέσει, τις εξαρτήσεις μεταξύ των εργασιών αυτών και τέλος τη σειρά με την οποία οι εν λόγω

εργασίες πρέπει να εκτελεστούν. Επιπλέον, η διαθεσιμότητα των πόρων πρέπει να καθοριστεί, προκειμένου κάθε εργασία ή ομάδα καθηκόντων να ανατεθεί στον κατάλληλο εργαζόμενο. Μία από τις μεθόδους που χρησιμοποιούνται για να αναπτυχθεί η λίστα των εργασιών είναι να δημιουργήσουμε μία Δομή Ανάλυσης Εργασιών (Work Breakdown Structure WBS)

Η Δομή Ανάλυσης Εργασιών (ΔΑΕ) είναι μια ιεραρχική αποσύνθεση ή ανάλυση του έργου ή μίας σημαντικής δραστηριότητας σε μεγάλα διαδοχικά βήματα-επίπεδα, όπου κάθε βήμα-επίπεδο είναι μικρότερο του προηγούμενου. Στην τελική της μορφή μια ΔΑΕ είναι παρόμοια με τη δομή και τη διάταξη σε μια διάρθρωση ενός εγγράφου. Κάθε στοιχείο σε ένα συγκεκριμένο επίπεδο στη ΔΑΕ είναι αριθμημένο διαδοχικά (π.χ. 10, 20, 30, 40, 50). Κάθε στοιχείο στο επόμενο επίπεδο αριθμείται κατά τον αριθμό της θέσης του προηγούμενου βήματος (π.χ. 10,1, 10,2, 10,3, 10,4)

Η ΔΑΕ μπορεί σχεδιαστεί υπό μορφή διαγράμματος (εάν χρησιμοποιήσουμε λογισμικό) ή σε μορφή πίνακα που μοιάζει με περίγραμμα. Η ΔΑΕ ξεκινά με μια εργασία που αντιπροσωπεύει το σύνολο των εργασιών που πρέπει να εκτελεστούν για να ολοκληρωθεί το έργο. Αυτή εργασία γίνεται και το όνομα του σχεδίου ΔΑΕ

Χρησιμοποιώντας μια μεθοδολογία, για τη ανάπτυξη του έργου ως οδηγό, το έργο χωρίζεται σε σημαντικά βήματα. Στην περίπτωσή μας θα χρησιμοποιήσουμε τον ακόλουθο κύκλο ζωής. Η πρώτη φάση είναι η έναρξη του έργου, η δεύτερη σημαντική φάση είναι η ανάλυση και ακολουθείται από τον σχεδιασμό, κατασκευή, δοκιμές, υλοποίηση, και τέλος την αξιολόγηση. Κάθε μία από αυτές τις φάσεις πρέπει να αναλυθεί σε επόμενα επίπεδα και κάθε ένα από αυτά, σε ακόμα μικρότερα επίπεδα, μέχρις ότου να έχουμε ένα διαχωρίσιμο μέγεθος εργασίας. Το πρώτο επίπεδο της Αναλυτικής Δομής Εργασιών για τον κύκλο ζωής απεικονίζεται στον πίνακα 6.1

Οι εργασίες σε κάθε διαδοχικό επίπεδο λεπτομέρειας αριθμούνται ώστε να αντικατοπτρίζουν το έργο από το οποίο προήλθαν. Έτσι, το πρώτο επίπεδο των καθηκόντων θα είχε αρίθμηση 1,0, 2,0, 3,0, και ούτω καθεξής. Κάθε επιμέρους εργασία θα είχε αρίθμηση με δύο αριθμούς. Το πρώτο μέρος αντανακλά την εργασία γονέα και το δεύτερο μέρος είναι το υπό-έργο, π.χ. τον ίδιο αριθμό 1,1, 1,2, 1,3, κλπ. Κάθε μία από αυτές τις υπό-εργασίες μπορούν με τη σειρά τους να αναλυθούν σε επόμενες εργασίες και ούτω καθεξής.

Μια εύκολα διαχωρίσιμη εργασία είναι εκείνη όπου τα αναμενόμενα αποτελέσματα μπορούν εύκολα να προσδιοριστούν, η επιτυχία ή αποτυχία ή η ολοκλήρωση του έργου μπορεί εύκολα να διαπιστωθεί, ο χρόνος για να ολοκληρωθεί το έργο μπορεί εύκολα να εκτιμηθεί και οι απαιτήσεις των πόρων του έργου μπορούν εύκολα να προσδιοριστούν .

Αριθμός	Περιγραφή Εργασίας
1.0	Φάση έναρξης
1.1	Πρόχειρο σχέδιο έργου
2.0	Φάση ανάλυσης
2.1	Σχεδιασμός συνεντεύξεων χρηστών
2.2	Προγραμματισμός συνεντεύξεων χρηστών
3.0	Εξέταση και Δοκιμή
4.0	Σχεδιασμός
5.0	Έλεγχος
6.0	Εκτέλεση
7.0	Αξιολόγηση

Πίνακας 6: Δομή Ανάλυσης Εργασιών

4.3 Μέθοδοι χρονοπρογραμματισμού

Ο σχεδιασμός, ανάλυση, προγραμματισμός & έλεγχος των μεγάλων έργων είναι από τις πιο σύνθετες εργασίες των μηχανικών.

Οι τρεις βασικές μεθοδολογίες που έχουν αναπτυχθεί για το χρονοπρογραμματισμό των έργων είναι οι:

- **PERT** (Project Evaluation & Review Technique): Μεθοδολογία αξιολόγησης & παρακολούθησης έργου

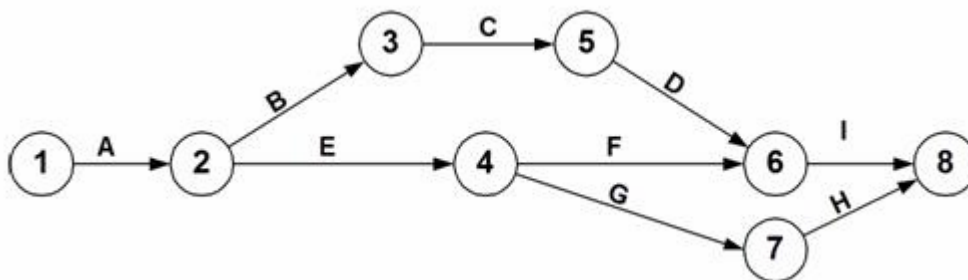
- CPM (Critical Path Method): Μέθοδος Κρίσιμης Διαδρομής
- Διάγραμμα GANTT

Και οι τρεις μεθοδολογίες βασίζονται στην ίδια φιλοσοφία και χρησιμοποιούν τις ίδιες τεχνικές με αποτέλεσμα σήμερα να θεωρούνται σαν μία ενιαία μέθοδος προγραμματισμού και ελέγχου μεγάλων έργων.

4.3.1 Δικτυακή Απεικόνιση ενός Έργου

Για να εφαρμοστεί η μεθοδολογία PERT/CPM χωρίζουμε το έργο σε επί μέρους δραστηριότητες με λεπτομερές χρονοδιάγραμμα έναρξης και περάτωσης κάθε δραστηριότητας καθώς και όλων των διασυνδέσεών τους.

Το όλο έργο απεικονίζεται σε μορφή δικτύου όπου ο κάθε κλάδος είναι μια δραστηριότητα, ενώ ο κάθε κόμβος είναι ένα γεγονός και ταυτόχρονα αποτελεί το σημείο εκκίνησης ή ολοκλήρωσης όσων δραστηριοτήτων ξεκινούν ή καταλήγουν εκεί.



Κατά την απεικόνιση πρέπει να τηρούνται ορισμένοι κανόνες ή περιορισμοί.

- Ένα γεγονός συμβαίνει μόνο όταν όλες οι δραστηριότητες που οδηγούν σε αυτό έχουν ολοκληρωθεί (π.χ.: το γεγονός 6 θα συμβεί μόνο όταν ολοκληρωθούν οι δραστηριότητες D & F).
- Μια δραστηριότητα μπορεί να αρχίσει μόνο όταν συμβεί το γεγονός που προηγείται αυτής (π.χ.: οι δραστηριότητες B & E θα αρχίσουν μόνο όταν συμβεί το γεγονός 2).

- Ένα γεγονός δεν μπορεί να συμβεί δυο φορές δηλαδή οι δραστηριότητες δεν μπορεί να σχηματίζουν βρόγχο (π.χ.: δεν μπορεί να τοποθετηθεί δραστηριότητα που να αρχίζει με το γεγονός 7 και να τελειώνει με το γεγονός 2).
- Κάθε γεγονός έχει προηγούμενη και επόμενη δραστηριότητα εκτός από την έναρξη και το τέλος του έργου.

Η σχεδίαση του δικτύου γίνεται σε τρεις φάσεις.

Στη **πρώτη φάση** αναλύεται το όλο έργο σε επί μέρους δραστηριότητες και καταρτίζεται ο αναλυτικός πίνακας δραστηριοτήτων. Οι δραστηριότητες ομαδοποιούνται σε κατηγορίες (ως προς τα απαιτούμενα μέσα, τοποθεσία, μέθοδο, κλπ.)

Για παράδειγμα, δίνεται ένας απλοποιημένος πίνακας με τις δραστηριότητες κατασκευής ενός κτίσματος:

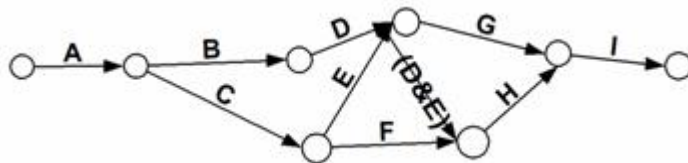
A	Κατασκευή σκελετού
B	Κατασκευή σκεπής
C	Κατασκευή τοιχοποιίας
D	Συλλέκτες νερού σκεπής
E	Υδραυλικές εγκαταστάσεις
F	Ηλεκτρικές εγκαταστάσεις
G	Αποχέτευση
H	Σοβατίσματα
I	Βαψίματα

Στη **δεύτερη φάση** καταρτίζεται ένα πρόχειρο δίκτυο με τη φυσιολογική σειρά των δραστηριοτήτων βάσει του αρχικού πίνακα, των αλληλεξαρτήσεων. Η σειρά αυτή προκύπτει και από τυχόν περιορισμούς λόγω τεχνικών δυνατοτήτων, προσωπικού, υλικών ή πολιτικής.

Στη φάση αυτή μπορεί να επίσης προκύψουν βελτιώσεις του δικτύου είτε με περαιτέρω ανάλυση σύνθετων δραστηριοτήτων ή με τη σύνθεση επί μέρους δραστηριοτήτων σε μία, ή ακόμη και με τη προσθήκη τεχνητών δραστηριοτήτων μηδενικού χρόνου με σκοπό την επισήμανση κάποιας αλληλουχίας.

Για να αποφύγουμε τυχόν παραλείψεις ή λάθη στη σειρά σχεδίασης, για κάθε μια δραστηριότητα θέτουμε τα παρακάτω ερωτήματα:

- A) Ποιες δραστηριότητες πρέπει να έχουν τελειώσει πριν αρχίσει η τρέχουσα;
- B) Ποιες δραστηριότητες πρέπει ή μπορούν να αρχίσουν μόλις τελειώσει η τρέχουσα;
- Γ) Ποιες δραστηριότητες είναι ανεξάρτητες και μπορούν να συμβούν παράλληλα με τη τρέχουσα;

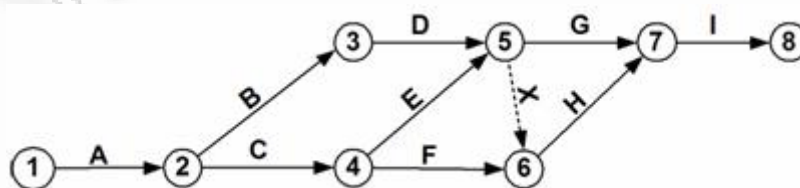


Στη **τρίτη φάση** γίνεται η ανασύνταξη του δικτύου και η βελτίωση της σχεδίασης ώστε να είναι παραστατικό, γίνεται η αρίθμηση των γεγονότων, και γίνεται η σύνταξη του οριστικού πίνακα δραστηριοτήτων και του χρονικού πίνακα που προκύπτει από αυτόν.

Κατά την ανασύνταξη του διαγράμματος αποφεύγονται κατά το δυνατόν οι διασταυρώσεις δραστηριοτήτων, τυχόν καμπύλα τμήματα αντικαθίστανται με ευθύγραμμα και διατάσσονται τα γεγονότα με σαφή χρονική σειρά από αριστερά προς τα δεξιά.

Ο χρονικός πίνακας δραστηριοτήτων περιέχει και τις εκτιμήσεις για τη χρονική διάρκεια όλων των δραστηριοτήτων. Ένα σημαντικό πρόβλημα στη σύνταξη του είναι η χρήση της ίδιας Χρονικής Μονάδας (ώρα, ημέρα, εβδομάδα, μήνας, κλπ.) για όλες τις δραστηριότητες. Σε σύντομα έργα συνήθως χρησιμοποιείται η ώρα ενώ σε μεγαλύτερα έργα χρησιμοποιείται η εβδομάδα.

Για την εκτίμηση της διάρκειας μιας δραστηριότητας λαμβάνουμε υπ' όψιν το διαθέσιμο προσωπικό, τον αριθμό και τη κατάσταση των μηχανημάτων, το είδος των υλικών, τη μέθοδο εργασίας, την τοποθεσία, τις συνθήκες εργασίας, τον καιρό, αλλά και τη πολιτική της διοίκησης.



Τελικό Διάγραμμα και Χρονικός Πίνακας Δραστηριοτήτων

	Δραστηριότητα	Προαπαιτούμενες	Χρον. Διάρκεια (Εβδ.)
A	Κατασκευή σκελετού	-	4
B	Κατασκευή σκεπής	A	2
C	Κατασκευή τοιχοποιίας	A	3
D	Συλλέκτες νερού σκεπής	B	5
E	Υδραυλικές εγκαταστάσεις	C	2
F	Ηλεκτρικές εγκαταστάσεις	C	1
G	Αποχέτευση	D, E	6
X	Τεχνητή Δραστηριότητα	D, E	0
H	Σοβατίσματα	X, F	4
I	Βαψίματα	G, H	3

4.3.2 Ανάλυση PERT/CPM

Την δεκαετία του 1950 αναπτύχθηκαν οι μέθοδοι CPM και PERT που πολύ γρήγορα αποτέλεσαν δημοφιλέστατα εργαλεία στον σχεδιασμό και στον έλεγχο των έργων. Οι μέθοδοι αυτές είναι γραφικές τεχνικές που τις συναντούμε και με τον γενικότερο όρο Προγραμματισμός έργου με την μέθοδο των δικτύων (network modeling) ή ακόμα στην διεθνή βιβλιογραφία κάτω από τον όρο Critical path scheduling. Σε κάθε έργο, τρεις είναι οι παράγοντες οι οποίοι απασχολούν τους διοικούντες ιδιαίτερα: ο χρόνος, το κόστος και η διαθεσιμότητα των πόρων. Οι παραπάνω τεχνικές αναπτύχθηκαν για να ασχοληθούν ακριβώς με τους τρεις αυτούς παράγοντες, ανεξάρτητα αλλά και σε συνδυασμό μεταξύ τους. Οι δύο μέθοδοι CPM και PERT, αν και αναπτύχθηκαν ανεξάρτητα και για διαφορετικά έργα στο τέλος της δεκαετίας του 1950, παρουσιάζουν μεταξύ τους εξαιρετικές ομοιότητες.

Η Μέθοδος της κρίσιμης διαδρομής (μέθοδος CPM – Critical Path Method) αναπτύχθηκε το 1958 από τους J. E. Kelly της Remington Rand και M. R. Walker της

Du Pont για την υποστήριξη του προγραμματισμού των εργασιών κατασκευής και συντήρησης βιομηχανικών συγκροτημάτων παραγωγής χημικών προϊόντων.

Η μέθοδος **PERT (Programme Evaluation and Review Technique)**, που θα μπορούσε να αποδοθεί στα ελληνικά με την έκφραση Τεχνική θεωρημένης αξιολόγησης έργου, αναπτύχθηκε το 1958 από το Γραφείο ειδικών έργων του πολεμικού ναυτικού των ΗΠΑ, για την ανάπτυξη και διαχείριση του προγράμματος «Πύραυλοι Polaris». Οι δύο αυτές μέθοδοι εμφανίζουν το έργο με μια γραφική φόρμα και διασύνδεουν τις συνιστώσες δράσεις του έργου με τρόπο που εστιάζει σ' αυτές που είναι κρίσιμες για την ολοκλήρωσή του. Έτσι υπάρχει η δυνατότητα προγραμματισμού έργων με πολλές διαφορετικές δράσεις που εκτελούνται από διαφορετικά τμήματα και άτομα, προσφέροντας την δυνατότητα άντλησης πληροφοριών τόσο για τον χρόνο εκτέλεσης των δράσεων, όσο και για τη διαδοχή με βάση την οποία πρέπει να εκτελεστούν. Μια σειρά από κρίσιμα ερωτήματα μπορούν να απαντηθούν με την χρήση των μεθόδων CPM και PERT, ερωτήματα που σχετίζονται με την αποτελεσματική παρακολούθηση της πορείας εκτέλεσης των έργων, όπως:

1. Πότε θα ολοκληρωθεί το έργο στο σύνολό του;
2. Πότε έχει προγραμματιστεί να αρχίσουν και να τελειώσουν τα κύρια τμήματα και οι σχετικές δράσεις του έργου;
3. Ποιες είναι οι κρίσιμες δράσεις, δηλαδή εκείνες που δεν πρέπει να καθυστερήσουν για να ολοκληρωθεί το έργο στον καθορισμένο χρονικό ορίζοντα;
4. Ποιες είναι οι μη κρίσιμες δράσεις που μπορούν να καθυστερήσουν χωρίς να καθυστερήσει το συνολικό έργο και πόσο μπορούν να καθυστερήσουν;
5. Ποια είναι η πιθανότητα να ολοκληρωθεί το έργο σε μια συγκεκριμένη χρονική στιγμή;
6. Σε ποιες δραστηριότητες πρέπει ο υπεύθυνος του έργου να δώσει ιδιαίτερη προσοχή σε κάθε χρονική στιγμή του έργου;
7. Πως μπορεί να επιταχυνθεί η διαδικασία εκτέλεσης του έργου και ποιες δράσεις θα επηρεαστούν;
8. Είναι δυνατή η μεταφορά πόρων από μη κρίσιμες σε κρίσιμες δράσεις;

Οι απαντήσεις στα παραπάνω ερωτήματα αλλά και σε άλλα είναι βασικό ζητούμενο για τον ορθολογικό προγραμματισμό ενός έργου και δίνονται πραγματικά εύκολα και συστηματικά με την χρήση των μεθόδων CPM και PERT.

4.3.2.1 Μέθοδος CPM – Critical Path Method

Από τις πολύ πρώιμες ακόμα εφαρμογές της μεθόδου γνώρισε τέτοια απήχηση, που από τότε εφαρμόζεται σε σημαντικό αριθμό έργων διαφόρων τύπων σε παγκόσμια κλίμακα. Για μια απόλυτα επιτυχημένη εφαρμογή της μεθόδου θα πρέπει οι επιμέρους εργασίες (δράσεις ή δραστηριότητες) που σχετίζονται με το έργο, να έχουν τα ακόλουθα χαρακτηριστικά:

- Να είναι καλώς ορισμένες στη διάσταση του χρόνου και η περάτωσή τους να συμπίπτει με το πέρας του συνόλου του έργου,
- Να είναι ανεξάρτητες μεταξύ τους,
- Να ακολουθούν συγκεκριμένη σειρά εκτέλεσης

Χαρακτηριστικές περιπτώσεις που πληρούν τα παραπάνω χαρακτηριστικά είναι η κατασκευή δομικών έργων, η κατασκευή αεροπλάνων, η κατασκευή πληροφοριακών συστημάτων και οι ναυπηγικές βιομηχανίες και έτσι η μέθοδος CPM βρίσκει ευρεία εφαρμογή σ' αυτές. Για την πρακτική εφαρμογή της μεθόδου έχει αναπτυχθεί μια συγκεκριμένη μεθοδολογία που συνίσταται σε γενικές γραμμές από τα παρακάτω βήματα:

1. Καθορισμός των εργασιών ή δράσεων που συνιστούν το σύνολο του έργου.
2. Προσδιορισμός της σειράς (αλληλουχίας) με την οποία πρέπει να εκτελεστούν αυτές.
3. Εκτίμηση του χρόνου ολοκλήρωσης της κάθε επιμέρους εργασίας ή δράσεως.

4. Σχεδιασμός του δικτύου των δράσεων τηρώντας την απαιτούμενη για το έργο αλληλουχία εκτέλεσης.
5. Προσδιορισμός πάνω στο δίκτυο, με διαδρομή από την αρχή προς το τέλος του έργου, της ταχύτερης δυνατής έναρξης και ολοκλήρωσης κάθε δράσης με βάση το δίκτυο και τους χρόνους ολοκλήρωσης.
6. Προσδιορισμός πάνω στο δίκτυο, με διαδρομή από το τέλος προς την αρχή του έργου, του αργότερου χρόνου έναρξης και ολοκλήρωσης κάθε δράσης, με βάση τον ταχύτερο χρόνο ολοκλήρωσης του έργου που προσδιορίστηκε στο προηγούμενο βήμα.
7. Προσδιορισμός του χρόνου που μπορεί να καθυστερήσει κάθε δράση (χρονικό περιθώριο χρόνου) με βάση την διαφορά των χρόνων που βρέθηκαν στα δύο προηγούμενα βήματα.
8. Αναγνώριση και καταγραφή των κρίσιμων δράσεων που είναι εκείνες των οποίων η διαφορά των χρόνων είναι μηδενική και δεν μπορούν κατά συνέπεια να καθυστερήσουν. Αυτές αποτελούν την κρίσιμη διαδρομή.
9. Χρήση των πληροφοριών από τα βήματα 5 και 6 για τον βασικό προγραμματισμό του έργου.

Είναι εύκολα κατανοητό ότι τα σημαντικότερα βήματα θεωρούνται αυτά της εκτίμησης των χρόνων ολοκλήρωσης των επιμέρους δράσεων και του προσδιορισμού της κρίσιμης διαδρομής, του συνόλου δηλαδή των δράσεων που δεν γίνεται να καθυστερήσουν. Έτσι αντιλαμβάνεται κανείς ότι η μέθοδος CPM μπορεί να χρησιμοποιηθεί αποτελεσματικά στις περιπτώσεις που ο χρόνος ολοκλήρωσης κάθε επιμέρους δράσεως του έργου, μπορεί να εκτιμηθεί με σχετική ακρίβεια. Αυτό συμβαίνει όταν υπάρχουν ικανοποιητικά στοιχεία για την προσέγγιση των προαναφερόμενων χρόνων, δηλαδή είτε στατιστικά στοιχεία από παρόμοια έργα, είτε σε έργα όπου τα υπεύθυνα για την εκτέλεση στελέχη μπορούν με βάση τις τεχνικές γνώσεις τους και την εμπειρία τους να προβλέψουν σωστά τους χρόνους ολοκλήρωσης των δράσεων. Με δεδομένους τους χρόνους ολοκλήρωσης των επιμέρους δράσεων, τα αμέσως σημαντικότερα βήματα είναι η σχεδίαση του δικτύου και ο εντοπισμός της κρίσιμης διαδρομής.

4.3.2.2 Μέθοδος PERT – Programme Evaluation and Review Technique

Η μέθοδος PERT αναπτύχθηκε παράλληλα αλλά ανεξάρτητα από την μέθοδο CPM. Ωστόσο πολύ γρήγορα έγινε και αυτή το ίδιο δημοφιλής όσο και η CPM και αποτελεί ισχυρό εργαλείο στην διοίκηση και τον προγραμματισμό των έργων. Η ομοιότητα των δύο μεθόδων είναι εξόχως εντυπωσιακή αν σκεφτεί κανείς ότι αναπτύχθηκαν ανεξάρτητα μεταξύ τους. Αυτό κατά μια έννοια ίσως να οφείλεται στον κοινό τους πρόγονο κατά κάποιον τρόπο, το ευρέως χρησιμοποιούμενο διάγραμμα Gantt. Παρ' όλα αυτά ενώ το διάγραμμα Gantt μπορεί να οπτικοποιήσει τη σχέση των επιμέρους δράσεων με τον χρόνο, είναι δύσκολο να εκφράσει τις αλληλεπιδράσεις μεταξύ τους (αλληλουχία) και ειδικά αν έχουμε δράσεις σε αριθμό πάνω από 20 ή 30 καταντά δύσχρηστο εργαλείο. Επίσης δεν προσφέρει μια εύκολη διαδικασία για τον εντοπισμό της κρίσιμης διαδρομής που βέβαια είναι μεγάλης πρακτικής αξίας. Αντίθετα και η μέθοδος CPM και η μέθοδος PERT μπορούν να διαχειριστούν μεγάλο αριθμό δράσεων μέσα στο έργο αλλά και δίνουν την κρίσιμη διαδρομή. Έτσι λοιπόν είναι προφανής η μεγάλη ομοιότητα μεταξύ των δύο μεθόδων CPM και PERT. Και οι δύο χρησιμοποιούν κόμβους και βέλη για να κτίσουν το δίκτυο δράσεων. Ωστόσο αρχικά ενώ η μέθοδος CPM χρησιμοποιούσε (και χρησιμοποιεί) τους κόμβους για να εκφράσει τις δράσεις και τα βέλη για να δείξει την ακολουθία τους, η μέθοδος PERT χρησιμοποιούσε τα βέλη για να συμβολίσει τις δράσεις και τους κόμβους για να σηματοδοτήσει την αρχή και το τέλος τους. Ωστόσο επειδή με τον καιρό και στην πράξη κρίθηκε σαν πιο αποτελεσματική στην απεικόνιση των δράσεων και των σχέσεων μεταξύ τους, η προσέγγιση της CPM, τείνει πλέον να επικρατήσει αυτός ο συμβολισμός και στην μέθοδο PERT. Στην παρούσα παράγραφο για την μέθοδο PERT ακολουθείται αυτός ο συμβολισμός (οι δράσεις στους κόμβους και η αλληλουχία τους στα βέλη).

Μια άλλη σημαντική διαφορά είναι ότι ενώ η μέθοδος CPM χρησιμοποιεί μία μόνο εκτίμηση για τον απαιτούμενο χρόνο για την ολοκλήρωση των δράσεων (την καλύτερη δυνατή εκτίμηση), η μέθοδος PERT χρησιμοποιεί τρεις εκτιμήσεις την αισιόδοξη, την απαισιόδοξη και την πιθανότερη ή συντηρητική. Αυτή η διαφοροποίηση στην μέθοδο PERT με τις τρεις εκτιμήσεις των χρόνων ολοκλήρωσης

των δράσεων, επιτρέπει χρησιμοποιώντας τα εργαλεία της στατιστικής να βρεθεί η πιθανότητα να ολοκληρωθεί το έργο νωρίτερα από μια ημερομηνία ή αντίθετα την πιθανότητα να ξεφύγει η ολοκλήρωση του έργου πέρα από μια ημερομηνία.

Εκτίμηση χρόνου περάτωσης εργασιών

Στη μέθοδο CPM μια από τις βασικές παραδοχές είναι ότι η διάρκεια κάθε δράσης του έργου είναι σταθερή, δηλαδή δεν χαρακτηρίζεται από τυχαιότητα, η οποία μπορεί να μεταβάλλει το μέτρο της. Αυτό είναι εύλογο όταν υπάρχει απόλυτη βεβαιότητα για την διάρκεια των δράσεων, όταν δηλαδή κάθε μια από αυτές είναι τυποποιημένη και απολύτως ελεγχόμενη, μη εξαρτώμενη δηλαδή από εξωτερικούς παράγοντες. Συχνά όμως στην πράξη δεν συμβαίνει αυτό, ιδίως για εργασίες που εκτελούνται για πρώτη φορά, οπότε δεν υπάρχει εμπειρία ή στατιστικά στοιχεία, ή για εργασίες που υπόκεινται στην επίδραση μεταβλητών εξωτερικών παραγόντων (π.χ. καιρικών ή οικονομικών). Στην περίπτωση αυτή μπορεί να εφαρμοστούν μέθοδοι βασισμένες στην υπόθεση ότι η αβεβαιότητα όσον αφορά τη διάρκεια κάθε δράσης μπορεί να παρασταθεί με την βοήθεια στατιστικών κατανομών ή με κάποιο άλλο τρόπο.

Η μέθοδος PERT στηρίζεται στην υπόθεση ότι ο χρόνος περάτωσης κάθε δράσης του έργου είναι μια στοχαστική μεταβλητή που ακολουθεί την κατανομή βήτα (beta distribution). Αντί μιας σταθερής τιμής για την διάρκεια αυτού του χρόνου δίνονται τρεις εκτιμήσεις για αυτή την τιμή:

Ελάχιστη ή αισιόδοξη εκτίμηση a , που αντιστοιχεί στην πιο αισιόδοξη εκτίμηση της διάρκειας της δράσης, που θα προκύψει υπό τις ευνοϊκότερες συνθήκες εκτέλεσης της. (Υπάρχει μόνο μια πολύ μικρή πιθανότητα, όχι περισσότερο από 1% η δράση να ολοκληρωθεί σε ακόμα μικρότερο χρόνο).

Συντηρητική ή η πλέον πιθανή εκτίμηση m , που είναι η τιμή που θα προέκυπτε συχνότερα, αν η δράση επαναλαμβανόταν πολλές φορές, ή που θα αποτελούσε την εκτίμηση της διάρκειας, αν επρόκειτο να γίνει μια μοναδική τέτοια εκτίμηση.

Μέγιστη ή απαισιόδοξη εκτίμηση b, που θα προκύψει κάτω από τις δυσμενέστερες συνθήκες. (Υπάρχει μόνο μια πολύ μικρή πιθανότητα, όχι περισσότερο από 1% η δράση να ολοκληρωθεί σε ακόμα μεγαλύτερο χρόνο).

Οι παραπάνω εκτιμήσεις στην πράξη γίνονται από έμπειρα και αρμόδια πρόσωπα, που συνήθως είναι και οι υπεύθυνοι για την εκτέλεση κάθε δράσης, παίρνοντας υπ' όψιν τους παράγοντες που μπορεί να επηρεάσουν την διάρκειά της.

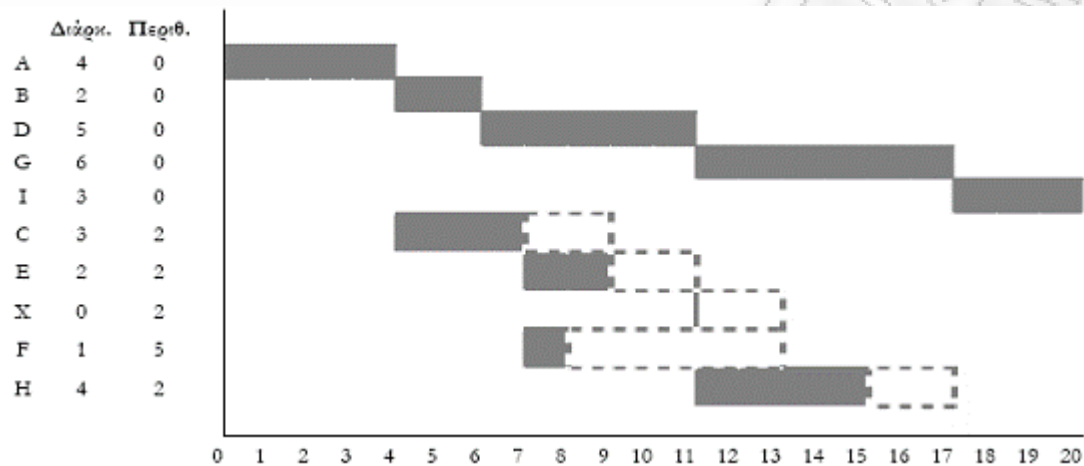
4.3.3 Χρονική Παρακολούθηση Έργων με το Διάγραμμα GANTT

Η επιστήμη της διαχείρισης έργου, για να υποστηρίξει την διαδικασία προγραμματισμού των έργων, έχει αναπτύξει μερικές πολύ χρήσιμες τεχνικές και εργαλεία όπως τις μεθόδους CPM και PERT που έχουν ήδη αναφερθεί.

Μια ωστόσο από τις δημοφιλέστερες τεχνικές προγραμματισμού έργου, δημοφιλής μέσα στην απλότητά της, είναι το **διάγραμμα Gantt**. Ονομάστηκε έτσι από τον Αμερικανό μηχανολόγο μηχανικό Henry Gantt (1869 – 1919), ο οποίος είναι ο πρώτος που το επινόησε και το χρησιμοποίησε. Το διάγραμμα Gantt είναι ένα οριζόντιο ραβδόγραμμα που απεικονίζει στην ουσία την σχέση των διαφορετικών δράσεων του έργου, μέσα στον χρόνο. Στον οριζόντιο άξονα του διαγράμματος τοποθετείται ο χρόνος σε κατάλληλες υποδιαιρέσεις που ταιριάζουν με τις ανάγκες και την χρονική διάρκεια του έργου, ενώ στον κατακόρυφο άξονα τοποθετούνται οι τίτλοι των δράσεων του έργου. Η σειρά τοποθέτησής τους συνήθως είναι προς τα πάνω αυτές που αρχίζουν νωρίτερα και προς τα κάτω αυτές που αρχίζουν αργότερα, χωρίς αυτό να αποτελεί και απαραίτητο κανόνα. Η τοποθέτηση μπορεί να είναι και τυχαία ή να ακολουθεί άλλα κριτήρια χωρίς αυτό να επηρεάζει την ορθότητα του διαγράμματος. Οι δράσεις περιγράφονται είτε με τους τίτλους τους είτε με χρήση κωδικών αριθμών που παραπέμπουν σε συγκεκριμένες εργασίες. Στο κύριο τώρα τμήμα του διαγράμματος τοποθετούνται για κάθε δράση και σε οριζόντια διάταξη οι ράβδοι αποτύπωσης του χρόνου, με μήκος ανάλογο με την χρονική διάρκεια που απαιτείται για την ολοκλήρωσή της. Κάθε ράβδος αρχίζει από το σημείο που στον

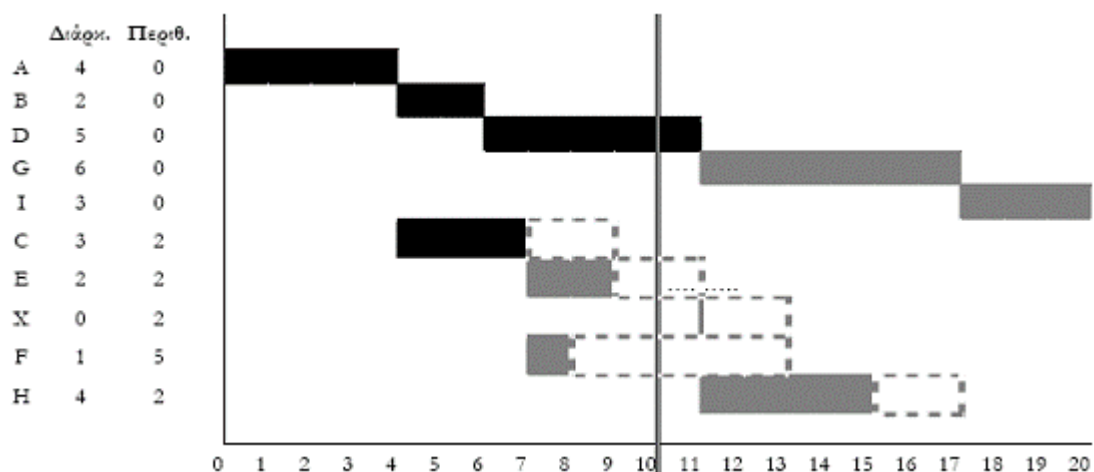
οριζόντιο άξονα αντιστοιχεί με το χρονικό σημείο έναρξης της συγκεκριμένης δράσης.

Στο διάγραμμα GANTT ο οριζόντιος άξονας απεικονίζει το χρόνο με ίδια χρονική μονάδα όπως και στην ανάλυση PERT/CPM. Οι δραστηριότητες παριστάνονται με οριζόντιες ράβδους και με μήκος ανάλογο της διάρκειάς τους.



Από το διάγραμμα GANTT παρακολουθούμε άμεσα τις επιπτώσεις τυχόν καθυστερήσεων ή αλλαγών στην εκτέλεση του έργου.

Για παράδειγμα έστω ότι το έργο βρίσκεται στη 10η εβδομάδα και οι ολοκληρωμένες δραστηριότητες σημειώνονται με μαύρο χρώμα όπως παρακάτω:



Παρατηρούμε ότι η δραστηριότητα D έχει τελειώσει νωρίτερα κατά μια εβδομάδα. Το δεδομένο αυτό θα μπορούσε να σημαίνει μια συντομότερη ολοκλήρωση του έργου. Όμως, η δραστηριότητα E δεν έχει αρχίσει ακόμη και το περιθώριό της είναι

μικρότερο από τη διάρκειά της. Άρα το έργο, αντί να συντομεύσει, τελικά θα καθυστερήσει κατά μια βδομάδα. Για την δραστηριότητα F που έχει καθυστερήσει δεν υπάρχει πρόβλημα γιατί το περιθώριό που της απομένει είναι μεγαλύτερο από τη διάρκειά της. Εύκολα μπορεί να αντιληφθεί κανείς την απλότητα του σχηματικού μοντέλου όσον αφορά τα διαγράμματα Gantt. Τα πλεονεκτήματα από την χρήση της συγκεκριμένης τεχνικής είναι η σαφής απεικόνιση της χρονικής διάρκειας και της αλληλουχίας των δράσεων, η εύκολη και γρήγορη κατασκευή του, αλλά και η ευκολία με την οποία μπορεί να κατανοήσει ακόμα και κάποιο μη εξειδικευμένο άτομο τις πληροφορίες που το διάγραμμα Gantt παρέχει στον χρήστη του. Βέβαια τα διαγράμματα Gantt δεν έχουν μεγάλες δυνατότητες πληροφόρησης και έτσι συνήθως χρησιμοποιούνται σε λιγότερο πολυσύνθετα έργα. Κάποια από τα μειονεκτήματά τους είναι η δυσκολία στην αναπροσαρμογή τους όταν παρουσιάζονται μεταβολές στην χρονική διάρκεια εκτέλεσης κάποιων δράσεων ή δραστηριοτήτων, καθώς επίσης και η δυσκολία της εφαρμογής τους σε έργα με μεγάλο αριθμό δράσεων, λόγω του σημαντικού χώρου που απαιτεί η απεικόνισή τους. Ακόμα υπάρχει αδυναμία στην απεικόνιση των αλληλεξαρτήσεων μεταξύ των δράσεων του έργου και τέλος αδυναμία για την παρουσίαση των κρίσιμων δράσεων ή δραστηριοτήτων για την επιτυχή ολοκλήρωση του συνολικού έργου.

4.4 Εργαλεία διοίκησης έργων πληροφορικής

Στο κεφάλαιο αυτό παρουσιάζεται επτά από τα πιο ευρέως διαδεδομένα πληροφοριακά συστήματα διοίκησης έργου. Μερικά λογισμικά είναι εμπορικά και άλλα ανοικτού κώδικα, ενώ αν και οι κύριες λειτουργίες είναι κοινές σε όλα, το καθένα από αυτά έχει κάποια πρόσθετα χαρακτηριστικά που το κάνουν ξεχωριστό.

Τα 7 λογισμικά που αναλύονται στις επιμέρους υπο-ενότητες του κεφαλαίου αυτού είναι τα:

- GanttProject
- Microsoft Project
- OmniPlan
- Open Workbench

- Pertmaster
- PlanningForce
- Achievo

Στο τέλος του κεφαλαίου υπάρχει και ένας συγκεντρωτικός πίνακας που συγκρίνει τα 7 αυτά λογισμικά με βάση κοινών χαρακτηριστικών που πληρούν τα πληροφοριακά συστήματα διοίκησης έργου.

4.4.1 GanttProject

Το λογισμικό GanttProject^[18] αποτελεί ένα εργαλείο διοίκησης έργου ανοιχτού κώδικα με περιορισμένες δυνατότητες. Επικεντρώνεται στον σχεδιασμό διαγραμμάτων Gantt και διαχείρισης πόρων. Τα βασικά χαρακτηριστικά του είναι η ανάλυση του έργου σε διαδοχικά βήματα, όπου γίνεται ανάθεση πόρων σε κάθε βήμα και καθορίζονται οι εξαρτήσεις μεταξύ των βημάτων. Το Ganttproject αναλύει το κάθε έργο σε διαγράμματα Gantt για τις διεργασίες και σε διαγράμματα ανάθεσης πόρων για τους πόρους. Υπάρχει η δυνατότητα εξαγωγής αναφορών σε HTML και PDF μορφή και ανταλλαγής δεδομένων με λογισμικά, όπως το Microsoft Project και λογισμικά προγραμματιστικών φύλλων.

Τα βασικά χαρακτηριστικά του Ganttproject και τα οποία το διαχωρίζουν τόσο από άλλα εργαλεία ανοιχτού κώδικα όσο και από άλλα πληροφοριακά συστήματα διοίκησης έργων είναι:

- Η βασική έκδοση πληροί τις περισσότερες ανάγκες διοίκησης έργων.
- Είναι ένα εύχρηστο πρόγραμμα, όπου ο χρήστης μπορεί να εκμεταλλευτεί όλες τις δυνατότητες του και να εξοικειωθεί με λίγες ώρες ενασχόλησης ή εκπαίδευσης.
- Τα περισσότερα προγράμματα διοίκησης έργων είναι ακριβά, ενώ το Ganttproject είναι δωρεάν.

- Είναι ανεξάρτητο πλατφόρμας, δηλαδή μπορεί να λειτουργήσει κάτω από οποιοδήποτε λειτουργικό σύστημα το οποίο υποστηρίζει Java.
- Είναι ανοικτού κώδικα, το οποίο σημαίνει ότι είναι πλήρως παραμετροποιήσιμο και μπορεί να ενσωματώσει οποιοδήποτε χαρακτηριστικό μπορεί να χρειαστεί ένας χρήστης όπως υποστήριξη συγκεκριμένων αναφορών.

4.4.2 OmniPlan

Το λογισμικό OmniPlan ^[25] είναι ένα απλό πληροφοριακό σύστημα διοίκησης έργων με δυνατότητες δημιουργίας διαγραμμάτων Gantt, κατανομής πόρων και ελέγχου προϋπολογισμού των έργων. Μπορεί να ορίσει ορόσημα, χρονοδιαγράμματα, κόστη και πόρους. Τα χαρακτηριστικά που υποστηρίζει το λογισμικό είναι:

- Διοίκηση ρόλων, καθηκόντων και λειτουργιών
- Κατανομή πόρων
- Χρονοπρογραμματισμό έργου
- Διαγράμματα Gantt
- Διασύνδεση με άλλα συστήματα (μπορεί να εισάγει XML αρχεία από άλλα προγράμματα διοίκησης έργου όπως το MA project, ενώ παράλληλα μπορεί να εξάγει πληροφορίες σε μορφή csv, Microsoft Project exchange και html)

4.4.3 Open Workbench

Το λογισμικό Open Workbench ^[26] αποτελεί ένα πληροφοριακό σύστημα διοίκησης έργων ανοιχτού κώδικα, το οποίο παρέχει χρονοπρογραμματισμό και διοίκηση έργου και μπορεί να διανεμηθεί δωρεάν. Εφόσον το σύστημα αυτό είναι

ανοιχτού κώδικα έχει όλα τα πλεονεκτήματα των συστημάτων ανοιχτού κώδικα, όπως ότι είναι εύκολα παραμετροποιήσιμα, μπορούν να ενσωματώσουν γρηγορότερα τις τεχνολογικές εξελίξεις και έχουν χαμηλό κόστος ανάπτυξης και συντήρησης. Οι χρήστες του συστήματος θα είναι αποδέκτες μιας σειράς υπηρεσιών προστιθέμενης αξίας από την παγκόσμια κοινότητα των προγραμματιστών και των παρόχων υπηρεσιών. Τα βασικά χαρακτηριστικά που υποστηρίζει το Open Workbench είναι:

- Σχεδιασμός έργου
- Χρονοπρογραμματισμός έργου
- Διαχείριση Πόρων
- Δημιουργία αναφορών προόδου

4.4.4 PertMaster

Το λογισμικό Pertmaster^[27] είναι περισσότερο ένα πρόσθετο σύστημα υποστήριξης αποφάσεων στα ήδη υπάρχοντα συστήματα Microsoft Project και Primavera παρά ένα αυτόνομο σύστημα διοίκησης έργου. Παρέχει τη δυνατότητα καθορισμού διαστημάτων εμπιστοσύνης, όσον αφορά το κόστος του έργου και τον χρονοπρογραμματισμό του, μαζί με εναλλακτικά σχέδια αντιμετώπισης κινδύνου. Τα σχέδια αυτά υπολογίζουν την πιθανότητα απόκλισης του έργου όσον αφορά το χρόνο και το κόστος και αναλύουν την αποτελεσματικότητα των σχεδίων αντιμετώπισης. Αυτός ο συνδυασμός αποτελεί την βάση του χρονοπρογραμματισμού προσαρμοσμένο στον κίνδυνο που σήμερα αποτελεί την βασική νόρμα στις διαδικασίες σχεδιασμού και χρονοπρογραμματισμού.

Το κάθε έργο κατατάσσεται σε κάποια κατηγορία κινδύνου μέσω προκαθορισμένων περιοχών κινδύνου και στη συνέχεια αναλύεται μέσω προσομοιώσεων Monte Carlo. Διαστήματα εμπιστοσύνης και σημεία κινδύνου μπορούν να αναγνωριστούν μέσω ιστογραμμάτων ρίσκου, διαγραμμάτων διασποράς πριν ο κίνδυνος που προκύψει να ενσωματωθεί στον χρονοπρογραμματισμό του έργου. Τα βασικά χαρακτηριστικά του λογισμικού Pertmaster είναι:

- Έλεγχος χρονοπρογραμματισμού
- Ανάλυση ρίσκου
- Ενημέρωση για τον κίνδυνο μέσω αναφορών
- Ολοκλήρωση με άλλα συστήματα διοίκησης έργου όπως το Microsoft Project και το Primavera

4.4.5 PlanningForce

Το λογισμικό PlanningForce^[29] αποτελεί ένα πληροφοριακό σύστημα διοίκησης έργων με δυνατότητες συστήματος διαχείρισης αποφάσεων. Το βασικό χαρακτηριστικό του που το διαχωρίζει από τα άλλα συστήματα είναι το γεγονός ότι είναι ένας συνδυασμός διοίκησης έργου, διαχείρισης πόρων και προγραμματισμού έργων για την επίτευξη συγκεκριμένων στόχων. Ο μηχανισμός προγραμματισμού έργων παράγει εκτιμήσεις όσον αφορά το χρόνο εκτέλεσης του έργου και προγραμμάτων διαχείρισης περιορισμένων πόρων λαμβάνοντας υπόψη στόχους, κινδύνους, χρονικές αποκλίσεις που εξαρτώνται από διακοπές, εποχικότητα, ζήτηση και αποτελεσματικότητα.

Τα χαρακτηριστικά που υποστηρίζει το λογισμικό δίνουν τη δυνατότητα στο χρήστη να εκπονεί το στρατηγικό σχεδιασμό ενσωματώνοντας ευελιξία, εκτίμηση κόστους, χρόνου και πόρων για κάθε είδους έργο. Ο χρήστης μπορεί να ορίσει πόρους, προϋπολογισμό, διάρκεια και το planning force να αυτοματοποιεί όλες τις περαιτέρω διαδικασίες σχεδιασμού και ελέγχου. Βάσει των λειτουργιών του λογισμικού μπορεί να υποστηρίξει:

- Αυτόματο προγραμματισμό έργου.
- Διοίκηση πολλαπλών έργων ταυτόχρονα ανεξάρτητα από την χρονική διάρκεια ή το μέγεθος τους.
- Ανάθεση ρόλων σε κάθε βήμα της διαδικασίας.
- Δημιουργία διαφορετικών σεναρίων και επιλογή αυτών που δίνουν τα καλύτερα αποτελέσματα.

- Διαχείριση πόρων.
- Αυτόματος έλεγχος του έργου με χρήση διαγνωστικών προγραμμάτων και ανίχνευση πιθανών προβλημάτων στο μέλλον από π.χ. ανεπάρκεια πόρων.
- Διαχείριση κόστους, προϋπολογισμού, ταμειακών ροών.
- Διασύνδεση με άλλα συστήματα είτε εισάγοντας είτε εξάγοντας δεδομένα.

4.4.6 Microsoft Project

Το λογισμικό Microsoft Project^[24] παρέχει ισχυρά εργαλεία διοίκησης έργου. Παρέχει τη δυνατότητα ελέγχου κόστους, δημιουργίας αναφορών προόδου, χρήσης έτοιμων προτύπων, χρονοπρογραμματισμού, διαχείρισης κόστους και ανθρώπινων πόρων.

Με την χρήση του Microsoft Project μπορεί να προσδιοριστούν γρήγορα παράγοντες που επηρεάζουν τις ημερομηνίες των εργασιών και να εντοπισθούν πιθανές καθυστερήσεις. Η χρήση προγραμμάτων οδήγησης εργασιών βοηθούν να προσδιορισθεί ο παράγοντας βάσει του οποίου καθορίζεται η ημερομηνία έναρξης της εργασίας, ώστε να παρακολουθούνται όλοι οι παράγοντες, ανακαλύπτοντας την αιτία μίας συγκεκριμένης καθυστέρησης. Μερικά από τα χαρακτηριστικά του λογισμικού είναι τα παρακάτω:

- Ενσωματωμένη ηλεκτρονική βοήθεια
- Κατανόηση και έλεγχος των χρονοδιαγραμμάτων και των οικονομικών στοιχείων των έργων.
- Παρακολούθηση προϋπολογισμού, αντιστοιχία προϋπολογισμών σε έργα και προγράμματα.
- Επισημάνσεις αλλαγών των στοιχείων που υφίστανται τις πιο πρόσφατες αλλαγές.

- Δημιουργία και εκτέλεση υποθετικών σεναρίων.
- Παρακολούθηση και ανάλυση των έργων με ευέλικτο τρόπο. Οργάνωση και ταξινόμηση πληροφοριών του έργου χρησιμοποιώντας προκαθορισμένες ή προσαρμοσμένες ομαδοποιήσεις.
- Δυνατότητα δημιουργίας συγκεντρωτικών πινάκων και γραφημάτων, γραφικών και διαγραμμάτων που βασίζονται σε δεδομένα του Project.

4.4.7 Achievo

Το λογισμικό Achievo^[10] είναι ένα ευέλικτο διαδικτυακό πληροφοριακό σύστημα διοίκησης έργων. Οι δυνατότητες διαχείρισης πόρων και διεργασιών του Achievo βοηθούν τις εταιρείες να υποστηρίξουν τις εταιρικές διαδικασίες με απλό αλλά αποτελεσματικό τρόπο. Τα κύρια χαρακτηριστικά του Achievo είναι:

- Εξ' ολοκλήρου διαδικτυακό σύστημα
- Ανεξάρτητο από την πλατφόρμα εγκατάστασης
- Καμία χρέωση ή περιορισμός στις υπηρεσίες που προσφέρει
- Ελεύθερη πρόσβαση στον κώδικα
- Χρήση πρωτοκόλλων και προτύπων ελεύθερου κώδικα
- Ευέλικτο σύστημα πλήρως παραμετροποιημένο σε επίπεδο πηγαίου κώδικα

Η αρχιτεκτονική του Achievo είναι modularized με μία κεντρική βάση δεδομένων. Συνδυασμένα τα επιμέρους modules διασφαλίζουν τη βέλτιστη χρησιμοποίηση εφαρμογών intranet για εταιρικά περιβάλλοντα.

Υποστηρίζει μόνο εφαρμογές οι οποίες θεωρούνται αξιόπιστες και υποστηρίζονται πλήρως. Μπορεί να χρησιμοποιηθεί-εφαρμοστεί-αναπτυχθεί κάθε περιβάλλον δικτύου. Η λύση είναι server based και ανεξάρτητη από το σύστημα και άρα μπορεί να αναπτυχθεί σε υπάρχοντα δίκτυα.

Πέρα από τα βασικά χαρακτηριστικά που αναφέραμε το Achievo αποτελεί σουίτα εφαρμογών με επίκεντρο την διοίκηση έργου που επιπροσθέτως περιλαμβάνει και δυνατότητες διαχείρισης πελατών και διαχείρισης ανθρωπίνων πόρων. Όσον

αφορά καθαρά τη διοίκηση έργου δίνει την δυνατότητα δημιουργίας Pert και Gantt διαγραμμάτων, στατιστικά στοιχεία παρακολούθησης του έργου, υποστήριξη προτύπων για φάσεις και δραστηριότητες, ορισμό οροσήμων. Υπάρχει η δυνατότητα χρήσης ημερολογίου, είτε προσωπικού είτε ομάδων εργασίας και διαχείρισης επαφών σε σχέση με τα έργα. Επίσης, διοίκηση έργου κατά πελάτη, κατά προμηθευτή, διαχείριση υπαλλήλων, ρόλων και συμβάσεων εργαζομένων. Τέλος, υπάρχει η δυνατότητα εύκολης δημιουργίας αναφορών.

ΚΕΦΑΛΑΙΟ 5: ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ

5.1 Εισαγωγή

Στο Κεφάλαιο αυτό παρουσιάζεται μία μελέτη περίπτωσης που αφορά στο έργο ανάπτυξης ενός συστήματος Διαχείρισης και Αξιολόγησης Προμηθευτών (ΣΔΑΠ) που υλοποιήθηκε από μεγάλο τραπεζικό όμιλο με σκοπό τη διαχείριση και αξιολόγηση των προμηθευτών του ομίλου. Η μεθοδολογία που χρησιμοποιήθηκε για τη διοίκηση του έργου είναι η PMBOK.

Πιο συγκεκριμένα παρουσιάζονται οι φάσεις του έργου και η διαδικασία ολοκλήρωσής του. Η μεθοδολογία διοίκησης του έργου χρησιμοποιήθηκε ώστε το παραγόμενο αποτέλεσμα να είναι όσο το δυνατόν πιο κοντά στο επιθυμητό. Η PMBOK παρέχει ένα σύνολο από πληροφορίες σχετικά με τις αποδεδειγμένες πρακτικές για τη διοίκηση ενός έργου. Έτσι το έργο χωρίστηκε σε φάσεις κατά PMBOK και όποτε χρειάστηκε χρησιμοποιήθηκαν οι βέλτιστες πρακτικές που περιγράφονται από τη μεθοδολογία.

5.1 Φάση Αρχικοποίησης – Ορισμός Πεδίου

Σκοπός της πρότασης είναι η δημιουργία ενός συστήματος το οποίο θα αυτοματοποιήσει τη διαδικασία διαχείρισης και αξιολόγησης προμηθευτών (Σύστημα Διαχείρισης και Αξιολόγησης Προμηθευτών ΣΔΑΠ).

5.1.1 Στόχος του έργου

Το σύστημα αυτό θα χρησιμοποιείται τόσο από το τμήμα προμηθειών όσο και από τις άλλες μονάδες της εταιρείας. Η μελέτη θα επικεντρωθεί σε πρώτη φάση στην αυτοματοποίηση δύο βασικών διαδικασιών πρώτον της διαχείρισης νέων και παλαιών προμηθευτών με τη δημιουργία και συντήρηση ενός μητρώου προμηθευτών και δεύτερο της αξιολόγησης εγκεκριμένων προμηθευτών με την ανάπτυξη και συντήρηση κριτηρίων αξιολόγησης και την αυτοματοποίηση όπου αυτό είναι εφικτό των μεθόδων αξιολόγησης των προμηθευτών

5.1.2 Οι συμμετέχοντες στο έργο

Οι συμμετέχοντες στο έργο καθώς το έργο υλοποιήθηκε με το υπάρχον προσωπικό της εταιρείας είναι.

- Τμήμα που ζήτησε το έργο (Τμήμα Προμηθειών)
- Τμήμα που ανέλαβε το έργο
- Ο project manager και η ομάδα διαχείρισης του έργου
- Η ομάδα που θα αναλάβει την εκπαίδευση και την υποστήριξη των χρηστών
- Οι τελικοί χρήστες του έργου

5.1.3 Βασικά Παραδοτέα (deliverables)

- Απαιτήσεις προσωπικού
- Προσδιορισμός οργανωτικής επιτροπής
- Ομάδα έργου
- Υπόλοιπο προσωπικού
- Χρονοδιάγραμμα
- Απαιτήσεις σε πόρους
- Επιλογή προμηθευτών
- Πλάνο έργου
- Πλάνο επικοινωνίας
- Εφαρμογή
- Milestones

5.1.4 Τεχνικές απαιτήσεις

Λόγω της υπάρχουσας τεχνογνωσίας θα χρησιμοποιηθεί αρχιτεκτονική client/server. Το server κομμάτι θα γραφτεί σε java με χρήση των frameworks

hibernate και spring και για το client κομμάτι θα χρησιμοποιηθεί τεχνολογία Java Server Faces. Οι τεχνολογίες αυτές επιλέγονται γιατί λόγω τεχνογνωσίας περιορίζεται σημαντικά το κόστος και λόγω των δυνατοτήτων που αυτές οι τεχνολογίες παρέχουν. (επεκτασιμότητα, συνδεσιμότητα με άλλα συστήματα)

5.1.5 Λειτουργικές απαιτήσεις

Για τη δημιουργία του συστήματος θα χρησιμοποιήσουμε τις ήδη υπάρχουσες φόρμουλες αξιολόγησης που χρησιμοποιούνται για τους προμηθευτές από την εταιρεία.

5.1.6 Business Case

5.1.6.1 Σκοπός

Το σύστημα θα αποτελείται από δύο διασυνδεδεμένα υποσυστήματα το υποσύστημα διαχείρισης προμηθευτών και το υποσύστημα αξιολόγησης εγκεκριμένων προμηθευτών. Βασικές εργασίες των υποσυστημάτων παρατίθενται στην παρακάτω ενότητα.

A. Υποσύστημα Διαχείρισης Προμηθευτών

- A1. Εκδήλωση ενδιαφέροντος νέων προμηθευτών με σκοπό την ένταξή τους στο μητρώο εγκεκριμένων προμηθευτών
- A2. Αξιολόγηση και ένταξη των εγκεκριμένων προμηθευτών στο εν λόγω μητρώο σε συγκεκριμένες κατηγορίες αγαθών και υπηρεσιών
- A3. Δημιουργία μητρώου εγκεκριμένων προμηθευτών το οποίο θα διαχειρίζεται τους προμηθευτές της εταιρείας
- A4. Αυτοματοποίηση της διαχείρισης του μητρώου εγκεκριμένων προμηθευτών έτσι ώστε να υπάρχει ιστορικότητα σε θέματα αξιολόγησης τους ως προς την ένταξη

B. Υποσύστημα Αξιολόγησης Εγκεκριμένων Προμηθευτών

- B1. Δημιουργία ποιοτικών και ποσοτικών κριτηρίων αξιολόγησης εγκεκριμένων προμηθευτών

B2. Δημιουργία μεθόδων αυτόματης αξιολόγησης εγκεκριμένων προμηθευτών βάσει κριτηρίων και κανόνων που έχουν επιλεγθεί

B3. Διασύνδεση του ΣΔΑΠ με εξωτερικά υποσυστήματα (ERP, Help Desk) με σκοπό την άντληση στοιχείων για:

- a. την αυτόματη εξαγωγή της συνολικής σχέσης του προμηθευτή με την εταιρεία (πχ συνολικός τζίρος προμηθευτή με την εταιρεία, ποσοστό τζίρου προμηθευτή στο σύνολο των προμηθευτών της ίδιας κατηγορίας κλπ).
- b. την αυτοματοποίηση της συλλογής στοιχείων προμήθειας (πχ αποστολή και παραλαβή παραγγελιών κλπ), όπου αυτό είναι εφικτό, για την ενημέρωση συγκεκριμένων κριτηρίων που αφορούν την αξιολόγηση εγκεκριμένων προμηθευτών

B4. Δημιουργία και αποστολή ερωτηματολογίων από το ΣΔΗΠ σε συγκεκριμένους αποδέκτες, για την αντικειμενικότερη βαθμολόγηση κυρίως ποιοτικών χαρακτηριστικών αξιολόγησης

B5. Συντήρηση στοιχείων με σκοπό να μπορεί να αντληθεί εύκολα πληροφόρηση (MIS) αλλά και να διατηρείται η ιστορικότητα της αξιολόγησης.

5.1.6.2 ΛΕΙΤΟΥΡΓΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ ΥΠΟΣΥΣΤΗΜΑΤΩΝ

A. ΥΠΟΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΜΗΘΕΥΤΩΝ

Στην παρούσα ενότητα αναλύονται οι διαδικασίες που είναι απαραίτητες για την ένταξη προμηθευτών στο υποσύστημα διαχείρισης προμηθευτών καθώς και τα εμπλεκόμενα μέρη.

A.1. Διαδικασία εκδήλωσης ενδιαφέροντος από τους Προμηθευτές – Στάδιο 1 (αρχική επαφή)

Κατά την εκδήλωση ενδιαφέροντος ένας νέος προμηθευτής θα έχει τη δυνατότητα να εκδηλώνει το ενδιαφέρον του για συνεργασία με την εταιρεία. Κατά την εκδήλωση

ενδιαφέροντος ο προμηθευτής θα συμπληρώνει τα στοιχεία του σε μία ηλεκτρονική φόρμα η οποία και θα αποστέλλεται στο αρμόδιο τμήμα.

Τα στοιχεία που θα συμπληρώνονται από τον προμηθευτή είναι τα ακόλουθα:

A/A	ΠΕΡΙΓΡΑΦΗ ΣΤΟΙΧΕΙΩΝ	ΣΤΟΙΧΕΙΑ
1	ΕΠΩΝΥΜΙΑ ΕΤΑΙΡΙΑΣ	
2	ΝΟΜΙΚΗ ΜΟΡΦΗ	
3	ΑΝΤΙΚΕΙΜΕΝΟ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ	
4	ΔΙΑΘΕΣΙΜΑ ΠΡΟΪΟΝΤΑ	
5	ΔΙΕΥΘΥΝΣΗ ΕΔΡΑΣ	
6	Α.Φ.Μ.	
7	ΣΧΟΛΙΑ	Προϊόντα, Υπηρεσίες που προωθεί κλπ

Υπεύθυνος διαδικασίας: Διαχειριστής εφαρμογής

Α.2. Διαδικασία εκδήλωσης ενδιαφέροντος από τους Προμηθευτές– Στάδιο 2 (Λήψη στοιχείων για αξιολόγηση)

Το αρμόδιο τμήμα θα παραλαμβάνει τα στοιχεία του προμηθευτή και μετά από μία αρχική αξιολόγηση εάν θεωρεί ότι το αντικείμενο δραστηριότητας ή τα διαθέσιμα προϊόντα του εν λόγω προμηθευτή είναι ενδιαφέροντα τότε θα αποστέλλει στον προμηθευτή ειδικό κωδικό στο site για την εισαγωγή σε ηλεκτρονική μορφή επιπρόσθετων στοιχείων.

Τα στοιχεία αυτά συνοψίζονται στον παρακάτω πίνακα:

ΠΙΝΑΚΑΣ ΣΤΟΙΧΕΙΩΝ ΓΙΑ ΤΟ ΜΗΤΡΩΟ ΣΥΝΕΡΓΑΤΩΝ

A/A	ΠΕΡΙΓΡΑΦΗ ΣΤΟΙΧΕΙΩΝ	ΣΤΟΙΧΕΙΑ
1	ΚΑΤΗΓΟΡΙΑ ΠΡΟΪΟΝΤΩΝ	
2	Συνημμένα Αρχεία που αφορούν προϊόντα	
3	ΣΧΟΛΙΑ	
4	ΥΨΟΣ ΠΩΛΗΣΕΩΝ ΕΤΟΥΣ 200χ	
	» » » 200χ+1	
	» » » 200χ+2	
5	ΑΠΟΤΕΛ. ΠΡΟ ΑΠΟΣΒΕΣ. ΧΡΗΣ. 200χ	

	» » » 200χ+1			
	» » » 200χ+2			
		ΝΑΙ	ΟΧΙ	
6	* ΦΟΡΟΛΟΓΙΚΗ ΕΝΗΜΕΡΟΤΗΤΑ			ΙΣΧΥΣ ΜΕΧΡΙ
7	* ΑΣΦΑΛΙΣΤΙΚΗ ΕΝΗΜΕΡΟΤΗΤΑ			ΙΣΧΥΣ ΜΕΧΡΙ
8	* ΠΙΣΤΟΠΟΙΗΣΗ ΚΑΤΑ ISO 9000			
9	* ΠΙΣΤΟΠΟΙΗΣΗ ΚΑΤΑ ISO 14000			

Με την εισαγωγή των παραπάνω στοιχείων δημιουργείται ειδικός χώρος όπου ο προμηθευτής ανανεώνει τα στοιχεία και τα εμπλουτίζει σε τακτά χρονικά διαστήματα ή όποτε του ζητηθεί.

Τόσο η εκδήλωση ενδιαφέροντος από τους προμηθευτές όσο και η συμπλήρωση των απαραίτητων στοιχείων θα γίνεται ηλεκτρονικά μέσα από το site της ΣΔΗΠ .

Υπεύθυνος διαδικασίας: Διαχειριστής εφαρμογής

Α.3. Διαδικασία αξιολόγησης προμηθευτών για ένταξη στο μητρώο προμηθευτών εταιρείας – Στάδιο 1 (επιλογή προμηθευτών για συνεργασία πχ υποβολή προσφορών σε διαγωνισμούς)

Μετά την εκδήλωση ενδιαφέροντος ενός προμηθευτή διενεργείται αρχική αξιολόγηση του προμηθευτή από το τμήμα προμηθειών με σκοπό την συνεργασία του ομίλου με τον εν λόγω προμηθευτή όπως για παράδειγμα τη συμμετοχή του σε μελλοντικούς διαγωνισμούς. Η αξιολόγηση θα βασίζεται σε έλεγχο των στοιχείων που έχουν εισαχθεί για κάθε προμηθευτή, τα οποία θα εμφανίζονται σε ειδικές καταστάσεις από το σύστημα. Από τη στιγμή που ένας προμηθευτής αξιολογηθεί θετικά ως προς την επιλογή του σε διαγωνισμούς εισάγεται στο σύστημα σε ειδικό πεδίο ένδειξη με την οποία το σύστημα προτάσσει τον προμηθευτή σε διαγωνισμούς της κατηγορίας του.

Υπεύθυνος διαδικασίας: Διαχειριστής εφαρμογής, Τμήμα προμηθειών, Μονάδες

A.4. Διαδικασία αξιολόγησης προμηθευτών για ένταξη στο μητρώο προμηθευτών – Στάδιο 2 (Τελική ένταξη προμηθευτή στο μητρώο προμηθευτών)

Η οριστική ένταξη ενός προμηθευτή στο μητρώο προμηθευτών γίνεται μόνο με απόφαση της επιτροπής αξιολόγησης προμηθευτών. Σε τακτά χρονικά διαστήματα η επιτροπή καλείτε να αξιολογήσει προμηθευτές ως προς την τελική ένταξή τους στο μητρώο προμηθευτών. Η αξιολόγηση βασίζεται τόσο στα στοιχεία του συστήματος όσο και από τις υπηρεσίες που έχει προσφέρει στην εταιρεία (πχ υποβολή προσφορών σε διαγωνισμούς, πώληση προϊόντων υπηρεσιών κλπ).

Υπεύθυνος διαδικασίας: Επιτροπή Προμηθειών

B. ΥΠΟΣΥΣΤΗΜΑ ΑΞΙΟΛΟΓΗΣΗΣ ΠΡΟΜΗΘΕΥΤΩΝ

B.1. ΕΙΚΟΝΑ ΠΡΟΜΗΘΕΥΤΗ

Για τους εγκεκριμένους προμηθευτές που έχουν ενταχθεί στο μητρώο, θα προβλέπεται από το υποσύστημα αξιολόγησης προμηθευτών η εξαγωγή ειδικών καταστάσεων που θα εμφανίζει τη συνολική σχέση του προμηθευτή με τον Όμιλο. Είναι προφανές ότι το υποσύστημα για την εν λόγω δυνατότητα, θα αντλεί στοιχεία από τα υποσυστήματα με τα οποία θα διασυνδεθεί.

Αποδέκτες καταστάσεων: Διαχειριστής εφαρμογής, Τμήμα προμηθειών, Μονάδες

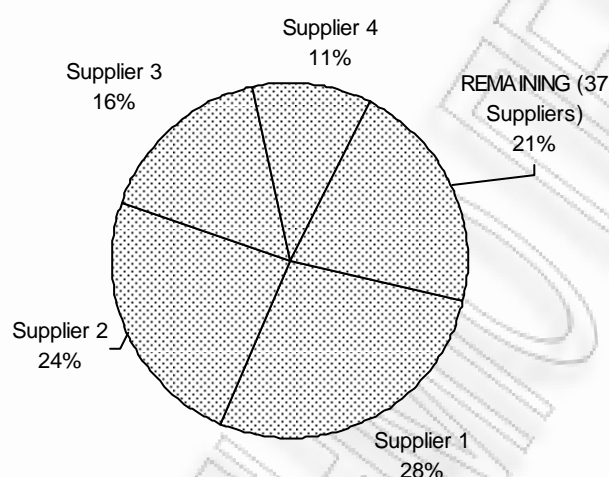
Ενδεικτικά στοιχεία της εικόνας ενός προμηθευτή παρουσιάζονται παρακάτω:

Εικόνα προμηθευτή A

Προμηθευτής	A
Κατηγορία Προμηθευτή*	Γραφική Υλη
Συνολικός Τζίρος έτους	€ 200,000
Αριθμός παραγγελιών	3,000
% Τζίρου προμηθευτή στην κατηγορία του	90%
Στοιχεία συμβάσεως	Αρ. Συμβ. 1234 Έναρξη 01/1/2006 Λήξη 30/06/2006
Κλπ..	...

**Ενδεικτικά γραφήματα εικόνας προμηθευτή:
% Τζίρου προμηθευτή στην κατηγορία του**

**ΚΑΤΗΓΟΡΙΑ ΠΡΟΪΟΝΤΩΝ
ΓΡΑΦΙΚΗ ΥΛΗ**



B.2. ΑΞΙΟΛΟΓΗΣΗ ΠΡΟΜΗΘΕΥΤΗ

Αντικειμενικός σκοπός της εν λόγω διαδικασίας είναι η δημιουργία ενός score card για κάθε προμηθευτή με στόχο:

1. τη παροχή ενός αξιόπιστου εργαλείου αξιολόγησης προμηθευτών του Ομίλου που θα χρησιμοποιηθεί στην αξιολόγηση των προμηθευτών ανεξάρτητα από το προϊόν, υλικό ή κατηγορία.
2. την επικοινωνία των βασικών απαιτήσεων προμηθειών στους προμηθευτές της Εταιρείας.
3. τη διευκρίνιση των στόχων προμηθειών της Εταιρείας στους προμηθευτές και τη θέσπιση αντικειμενικών κριτηρίων με σκοπό:

- βελτίωση των υπηρεσιών
 - μείωση του κόστους
 - καινοτομία και ποιότητα προϊόντων και υπηρεσιών
 - ευελιξία και αποδοτικότητα διαδικασιών
 - αξιοπιστία και συμμόρφωση στις διαδικασίες του Ομίλου
4. τον προσδιορισμό των προμηθευτών που επιθυμούν να συνεχίσουν τη συνεργασία τους με τον όμιλο σε μακροχρόνια βάση

Για τους εγκεκριμένους προμηθευτές θα επιλεγτούν συγκεκριμένα κριτήρια αξιολόγησης τα οποία και θα συμπληρώνονται σε προσυμφωνημένα χρονικά διαστήματα με σκοπό την αξιολόγηση τους. Τα κριτήρια αυτά θα βασίζονται σε ποσοτικά ή ποιοτικά χαρακτηριστικά και θα συμπληρώνονται βάσει διαδικασίας που θα επιλεγεί.

Κριτήρια αξιολόγησης:

- Delivery Performance
- Product Quality
- Responsiveness
- Accuracy/Efficiency
- Υφιστάμενη από μακρού χρόνου καλή συνεργασία (Βαθμολογία προηγούμενου έτους)
- Εισήγηση κατά τον έλεγχο από τον αξιολογητή IN SITE EVALUATION
- Φορολογική ενημερότητα
- Ασφαλιστική ενημερότητα
- Ύπαρξη Δυσμενών στοιχείων

Το σύστημα θα έχει τη δυνατότητα να αποστέλλει ερωτηματολόγια σε χρήστες της εταιρείας, που χρησιμοποιούν τα προϊόντα ή τις υπηρεσίες του κάθε προμηθευτή, με σκοπό τη δημιουργία προτάσεων αξιολόγησης για

ορισμένα κριτήρια, ώστε να αξιολογούνται με αντικειμενικότερο τρόπο οι υπηρεσίες των προμηθευτών. Επίσης το σύστημα, θα έχει τη δυνατότητα μέσα από τη διασύνδεσή του με εξωτερικά συστήματα να παρέχει τις απαραίτητες πληροφορίες για τη συμπλήρωση διαφόρων ποσοτικών κριτηρίων αξιολόγησης (πχ. Delivery Performance κλπ)

Γ. ΔΙΑΓΡΑΦΗ ΠΡΟΜΗΘΕΥΤΗ ΑΠΟ ΜΗΤΡΩΟ ΕΓΚΕΚΡΙΜΕΝΩΝ ΠΡΟΜΗΘΕΥΤΩΝ

Η διαγραφή προμηθευτών από το μητρώο προμηθευτών μπορεί να γίνει με τις παρακάτω διαδικασίες.

A.1. Διακοπή συνεργασίας με προμηθευτή

Εάν για κάποιο λόγο ζητηθεί είτε από τον Διαχειριστή του συστήματος είτε από το τμήμα προμηθειών η διακοπή συνεργασίας με συγκεκριμένο προμηθευτή τότε ο προμηθευτής αξιολογείται με σκοπό τη διακοπή συνεργασίας του με τον όμιλο όπως για παράδειγμα τη συμμετοχή του σε μελλοντικούς διαγωνισμούς.

Υπεύθυνος διαδικασίας: Διαχειριστής εφαρμογής, Τμήμα προμηθειών, Μονάδες

A.1. Οριστική Διαγραφή Προμηθευτή από το Μητρώο

Η οριστική διαγραφή ενός προμηθευτή από το μητρώο προμηθευτών γίνεται μόνο με απόφαση της επιτροπής αξιολόγησης προμηθευτών. Σε τακτά χρονικά διαστήματα η επιτροπή καλείτε να αξιολογήσει προμηθευτές ως προς τη διαγραφή τους από το μητρώο προμηθευτών. Η αξιολόγηση βασίζεται τόσο στα στοιχεία του συστήματος όσο και από τις υπηρεσίες που έχει προσφέρει στον όμιλο (πχ υποβολή προσφορών σε διαγωνισμούς, πώληση προϊόντων υπηρεσιών κλπ).

Υπεύθυνος διαδικασίας: Επιτροπή Προμηθειών

Δ. ΔΙΑΣΥΝΔΕΣΗ ΜΕ ΕΞΩΤΕΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

Η διασύνδεση με εξωτερικά συστήματα (ERP, Help Desk) κρίνεται απολύτως απαραίτητη για την άντληση στοιχείων που αφορούν το υποσύστημα αξιολόγησης προμηθευτών. Αναλυτικές προδιαγραφές των απαιτούμενων διασυνδέσεων θα σχεδιαστούν στη φάση υλοποίησης του συστήματος.

5.2 Φάση σχεδιασμού του έργου

5.2.1 Πλάνο διαχείρισης του Πεδίου του Έργου

Για να διαχειριστούμε το πεδίο του έργου έχουμε δημιουργήσει τις παρακάτω ομάδες έργου με τις ακόλουθες αρμοδιότητες

- Βασική ομάδα υπεύθυνη για τη συλλογή, διαχείριση των απαραίτητων για το έργο πληροφοριών
- Ομάδα παροχής εργασίας και τεχνολογίας υπεύθυνη για τους πόρους και το εργατικό δυναμικό
- Ομάδα ελέγχου ασφαλείας και μυστικότητας της τεχνολογίας πληροφοριών
- Ομάδα διαχείρισης έργου υπεύθυνη για την παρακολούθηση του έργου και επαναπροσδιορισμού του πλάνου αν χρειαστεί
- Ομάδα εκπαίδευσης υπεύθυνη για την εκπαίδευση των χρηστών

5.2.1.1 Δομή Ανάλυσης Εργασιών(ΔΑΕ)

Σύστημα διαχείρισης και αξιολόγησης προμηθευτών

1. Φάση Αρχικοποίησης – Ορισμός Πεδίου

- i. Βασικά Παραδοτέα
- ii. Τεχνικές απαιτήσεις
- iii. Λειτουργικές απαιτήσεις

2. Φάση Σχεδιασμού του έργου

- i. Πλάνο διαχείρισης του Πεδίου του Έργου
- ii. Δομή Ανάλυσης Εργασιών(ΔΑΕ)
- iii. Πλάνο διαχείρισης του Κινδύνου
- iv. Πλάνο διαχείρισης Ανθρώπινου Δυναμικού
- v. Οργανωτικός σχεδιασμός

- vi. Πλάνο διαχείρισης Επικοινωνίας όσων εμπλέκονται στο έργο
- vii. Πλάνο διαχείρισης Ποιότητας
- viii. Πλάνο διαχείρισης Κόστους
- ix. Πλάνο διαχείρισης Προμηθειών
- x. Επιλογή προσωπικού
- xi. Επιλογή προμηθευτή
- xii. Υπογραφή συμβάσεων

3. Φάση Εκτέλεσης

- i. Ανάλυση απαιτήσεων
- ii. Εγκατάσταση Συστημάτων
- iii. Ανάπτυξη συστήματος
 - a. Υλοποίηση μητρώου προμηθευτών
 - b. Υλοποίηση συστήματος αξιολόγησης
 - c. Υλοποίηση διαχειριστικού
- iv. Εκπαίδευση χρηστών
- v. Δοκιμή συστήματος
- vi. Πιλοτική φάση
- vii. Go Live
 - a. Υποστήριξη του προσωπικού
 - b. Συσκέψεις για αντιμετώπιση προβλημάτων κατά τη διάρκεια της εβδομάδας
 - c. Επικοινωνία με προμηθευτές αρμόδια τμήματα

4. Φάση Ελέγχου

- i. Έλεγχος αλλαγής πεδίου
- ii. Ανίχνευση και έλεγχος κινδύνου
- iii. Έλεγχος κόστους
- iv. Αξιολόγηση του συστήματος βάση των προσδοκώμενων ωφελειών

5. Κλείσιμο Έργου

- i. Κλείσιμο συμβάσεων
- ii. Τεκμηρίωση αδειών και εγγυήσεων για τα συστήματα
- iii. Ολοκλήρωση αποδοχής από το αρμόδιο τμήμα
- iv. Κλείσιμο οικονομικών εκκρεμοτήτων
- v. Δημιουργία τελικής αναφοράς
- vi. Αναθεώρηση ολόκληρου του έργου και συλλογή αρχειοθέτηση εγγράφων έργου

(Gaant Chart Παράρτημα 1)

5.2.2 Πλάνο διαχείρισης του Κινδύνου

Η διαχείριση του κινδύνου επειδή υπήρχε εμπειρία από ανάλογα έργα δεν αναπτύχθηκε όπως θα έπρεπε. Επικεντρωθήκαμε στη διαχείριση τριών ειδών κινδύνων τον τεχνολογικό, τον ανθρώπινο και τον διαχειριστικό.

Ο τεχνολογικός κίνδυνος στην περίπτωση μας ήταν η διαχείριση του ζητήματος της ασφαλούς αποθήκευσης δεδομένων. Ο ανθρώπινος κίνδυνος επικεντρώθηκε κυρίως στην ομαλή μετάβαση και την αποδοχή της αλλαγής από τα εμπλεκόμενα τμήματα. Για αυτό το λόγο διοργανώθηκαν εκδηλώσεις για την παρουσίαση του συστήματος με αναφορά στα οφέλη από τη χρήση του καθώς και εκπαιδεύσεις χρηστών. Τέλος ο διαχειριστικός κίνδυνος, που αφορά στη σωστή διαχείριση του έργου, επειδή υπήρχε ομάδα που είχε αναλάβει ανάλογα έργα δεν αναλύθηκε όσο θα έπρεπε.

Απαιτείται μια συστηματική διαδικασία αναγνώρισης, ανάλυσης και ανταπόκρισης σε ενδεχόμενους κινδύνους. Κατ' αυτό τον τρόπο μεγιστοποιείται η πιθανότητα να μην συμβεί το περιστατικό ή και εάν αυτό συμβεί να ελαχιστοποιηθούν οι δυσμενείς επιπτώσεις του.

5.2.3 Πλάνο διαχείρισης Ανθρώπινου Δυναμικού

Για να επιτευχθεί ο κατάλληλος συντονισμός και η άριστη συμβολή του ανθρώπινου παράγοντα στο όλο εγχείρημα, απαιτείται καλή πληροφόρηση, εμπιστοσύνη μεταξύ των μελών, καθένας να γνωρίζει το ρόλο του στο έργο, υψηλό ηθικό και διάθεση για επίτευξη υψηλών στόχων. Η επιβράβευση, τέλος, των μελών της ομάδας έργου συνδέεται με την επίτευξη των στόχων.

Η διαχείριση του Ανθρώπινου δυναμικού περιλαμβάνει τρεις διαδικασίες:

- τον οργανωτικό προγραμματισμό
- την απόκτηση προσωπικού
- την ανάπτυξη ομάδων

Στο συγκεκριμένο έργο απαιτήθηκε μόνο ο οργανωτικός προγραμματισμός καθώς υπήρχε τόσο το προσωπικό όσο και οι ομάδες.

5.2.3.1 Οργανωτικός σχεδιασμός

Για να γίνει ο οργανωτικός σχεδιασμός πρέπει να ανατεθούν οι ρόλοι και οι ευθύνες.

Το προσωπικό είχε ως εξής

- **Οργανωτική επιτροπή** με σκοπό να παρέχει στρατηγική καθοδήγηση, να διασφαλίσει ότι οι κατάλληλοι πόροι θα είναι διαθέσιμοι καθώς και να εξασφαλίσει ότι το έργο κατά την πορεία εκπληρώνει τους στόχους. Επίσης είναι υπεύθυνη για την εφαρμογή των πολιτικών κινδύνου. Η επιτροπή αυτή συνέρχεται τακτικά για την αξιολόγηση της προόδου του έργου.
- **Ομάδα έργου**
 - **Project Manager.** Διαχειρίζεται και επιβλέπει όλες τις πτυχές του έργου
 - **Επιχειρησιακός υπεύθυνος** Είναι υπεύθυνος για οικονομικά και οργανωτικά θέματα κατά την διάρκεια του έργου

- **Τεχνικός υπεύθυνος.** Είναι ο τεχνικός σύμβουλος του έργου που παρέχει την τεχνογνωσία και την καθοδήγηση όσον αφορά τα τεχνικά θέματα

5.2.4 Πλάνο διαχείρισης Επικοινωνίας όσων εμπλέκονται στο έργο

Ένα από τα πιο συχνά προβλήματα σε ένα έργο είναι η έλλειψη συντονισμού. Έτσι στο συγκεκριμένο έργο υπήρχαν οι ακόλουθες διαδικασίες για τη διασφάλιση της καλής επικοινωνίας.

- **Συχνές συσκέψεις** που διασφάλισαν την πλήρη κατανόηση και γρήγορη επίλυση των προβλημάτων
- **Εκθέσεις κατάστασης** με σκοπό τον έλεγχο της προόδου του έργου. Οι εκθέσεις αυτές παράγονταν σε μηνιαία βάση
- **Συντονισμός ομάδας έργου**
- **Επικοινωνία με προμηθευτές** για να τους κρατήσει ενήμερους για τη νέα διαδικασία αξιολόγησης τους

5.2.5 Πλάνο διαχείρισης Ποιότητας

Η διαχείριση της ποιότητας έγινε με χρήση ερωτηματολογίων πριν, κατά τη διάρκεια και μετά το τέλος των εργασιών τόσο για τις προσδοκίες όσο και για την ικανοποίηση των χρηστών από το σύστημα. Ο στόχος είναι να διασφαλιστεί ότι θα καλυφθούν οι ανάγκες που οδήγησαν στη δημιουργία του συστήματος και θα ανακαλυφθούν τυχόν αστοχίες και δυσλειτουργίες που πρέπει να επιλυθούν.

5.2.6 Πλάνο διαχείρισης Κόστους

Για το έργο θα χρειαστούν ανθρώπινοι και υλικοί πόροι. Για τους ανθρώπινους πόρους προϋπολογίστηκε ότι θα χρειαστούν για μισθούς τεχνικού και διοικητικού προσωπικού 1000000. Το κόστος αυτό αφορά μισθούς 1 έτους και αφορούν σε όλη τη διάρκεια του έργου δηλαδή σχεδιασμό, υλοποίηση, εφαρμογή και συντήρηση.

Για υλικούς πόρους θα χρειαστούμε

- 1 sever για παραγωγή 50.000E
- 1 server για preproduction 50.000E
- 1 server για demo 50.000E
- 1 server για disaster 50.000E
- Άδειες χρήσης λογισμικού και συντήρηση 24/7 68.000E
- Λοιπά κόστη domain name, διοικητικές δαπάνες 2000

Συνολικό κόστος 1.270.000E

5.2.7 Πλάνο διαχείρισης Προμηθειών

Στο πλάνο διαχείρισης κόστους περιγράψαμε τους υλικούς πόρους που θα χρειαστούν για το έργο. Για την προμήθεια των παραπάνω ειδών στην καλύτερη δυνατή τιμή ακολουθήθηκαν τα ακόλουθα βήματα.

- Επιλογή αρχικής λίστας προμηθευτών.
- Σύνταξη Request for Proposal όπου ο προμηθευτής περιγράφει τα προϊόντα και τις υπηρεσίες που παρέχει.
- Επιλογή των 4 επικρατέστερων βάση χαρακτηριστικών προϊόντων και αξιολόγησης προμηθευτών
- Διενέργεια ηλεκτρονικού διαγωνισμού για επιλογή της κατάλληλης προσφοράς.
- Διαπραγμάτευση σύμβασης
- Υπογραφή σύμβασης
- Διαδικασία πληρωμής

5.3 Φάση Εκτέλεσης

Η φάση της εκτέλεσης περιλαμβάνει όλες τις διαδικασίες που απαιτούνται για την εκτέλεση όσον περιγράφονται στη φάση σχεδιασμού.

- Ανάπτυξη / Δοκιμή Συστήματος. Για τη τεχνική φάση του έργου χρησιμοποιήθηκε το μοντέλο **πρωτοποποίησης**. Έτσι το σύστημα χωρίστηκε

σε παραδοτέα που παρουσιάζονταν στους χρήστες και με τις παρατηρήσεις τους αν κρινόταν αναγκαίο γυρίζαμε πάλι στη φάση της υλοποίησης για διόρθωση βελτιστοποίηση του παραδοτέου.

- Προετοιμασία λειτουργίας τμημάτων που αντιστοιχούν στις τρέχουσες υπηρεσίες
- Παροχή πρώτου γύρου εκπαίδευσης στους χρήστες που θα βοηθούν στην αξιολόγηση των παραδοτέων.
- Πιλοτική φάση κατά την οποία εκπαιδεύονται οι χρήστες, αξιολογείτε το σύστημα, το εκπαιδευτικό υλικό και αναπτύσσεται το σχέδιο ανάκαμψης από πιθανή καταστροφή.
- Εφαρμογή του συστήματος και συντήρηση

5.4 Φάση Ελέγχου

Σκοπός της φάσης ελέγχου είναι η επιτήρηση της εκτέλεσης του έργου με σκοπό τον έγκαιρο εντοπισμό των προβλημάτων και αν χρειαστεί την εφαρμογή διορθωτικών κινήσεων.

Η φάση αυτή περιλαμβάνει

- **Έλεγχος αλλαγής πεδίου** δηλαδή επαλήθευση ότι το έργο πληροί το σκοπό του έργου κατά τη διάρκεια ζωής του έργου
- **Ανίχνευση και έλεγχος κινδύνου**
- **Έλεγχος κόστους**
 - Αναθεώρηση και τεκμηρίωση του καταμερισμού των ωρών και του πεδίου εργασίας που πρέπει να γίνει.
 - Τεκμηρίωση οποιασδήποτε απόκλισης του εκτιμώμενου σε σχέση με το πραγματικό κόστος.
 - Αναφορά στους συμμετέχοντες όταν το 75% του έργου έχει ολοκληρωθεί.
- **Αναφορά παράδοσης**

5.5 Κλείσιμο Έργου

Σκοπός της φάσης αυτής είναι η ελεγχόμενη περάτωση του έργου με ενημέρωση της ανώτερης διοίκησης για τα αποτελέσματα. Έτσι έγιναν οι παρακάτω ενέργειες που οδήγησαν στο κλείσιμο του έργου

- Κλείσιμο συμβάσεων
- Τεκμηρίωση αδειών και εγγυήσεων για τα συστήματα
- Ολοκλήρωση αποδοχής από το αρμόδιο τμήμα
- Κλείσιμο οικονομικών εκκρεμοτήτων
- Δημιουργία τελικής αναφοράς
- Αναθεώρηση ολόκληρου του έργου και συλλογή αρχειοθέτηση εγγράφων έργου

ΣΥΜΠΕΡΑΣΜΑΤΑ

Κάθε έργο έχει τις δικές του δυσκολίες και κρίσημα σημεία στα οποία πρέπει να δοθεί ιδιαίτερη προσοχή για να επιτύχει. Είναι πολύ σημαντικό να υπάρχουν οι κατάλληλοι άνθρωποι στις κατάλληλες θέσεις που θα εξασφαλίσουν τη σωστή επικοινωνία μεταξύ των εμπλεκόμενων μερών έτσι ώστε να ελαχιστοποιηθεί η πιθανότητα το τελικό παραδοτέο να μην ικανοποιεί τις προσδοκίες.

Στην μελέτη περίπτωσης που παρουσιάστηκε οι δυσκολίες που προέκυψαν σχετίζονταν κυρίως με θέματα επικοινωνίας. Πιο συγκεκριμένα οι απαιτήσεις που δόθηκαν στο τμήμα που θα υλοποιούσε το έργο στην πορεία προέκυψε ότι δεν ήταν πλήρης με αποτέλεσμα να χρειασθεί να διορθωθούν αρκετές φορές με αποτέλεσμα να χρειασθεί περισσότερος χρόνος από ότι στην αρχή είχε υπολογιστεί.

Με τη χρήση της μεθόδου PMBOK για τη διοίκηση του έργου παρέχεται το γνωστικό υπόβαθρο και οι βέλτιστες πρακτικές για τη διοίκηση του έργου αλλά δεν παρέχεται μια αυστηρά καθορισμένη σειρά βημάτων κάτι που παρέχει η μεθοδολογία Prince2. Για την υλοποίηση του πληροφοριακού συστήματος δηλαδή κατά τη φάση εκτέλεσης εφαρμόστηκε το μοντέλο της πρωτυποποίησης. Ο λόγος που χρησιμοποιήθηκε το συγκεκριμένο μοντέλο είναι διότι δεν υπήρχε ξεκάθαρη εικόνα για το σύστημα από την αρχή. Έτσι το έργο χωρίστηκε σε μικρά παραδοτέα που αξιολογούνταν από τη μονάδα που ζήτησε το έργο και έτσι το τελικό αποτέλεσμα ικανοποίησε τις προσδοκίες.

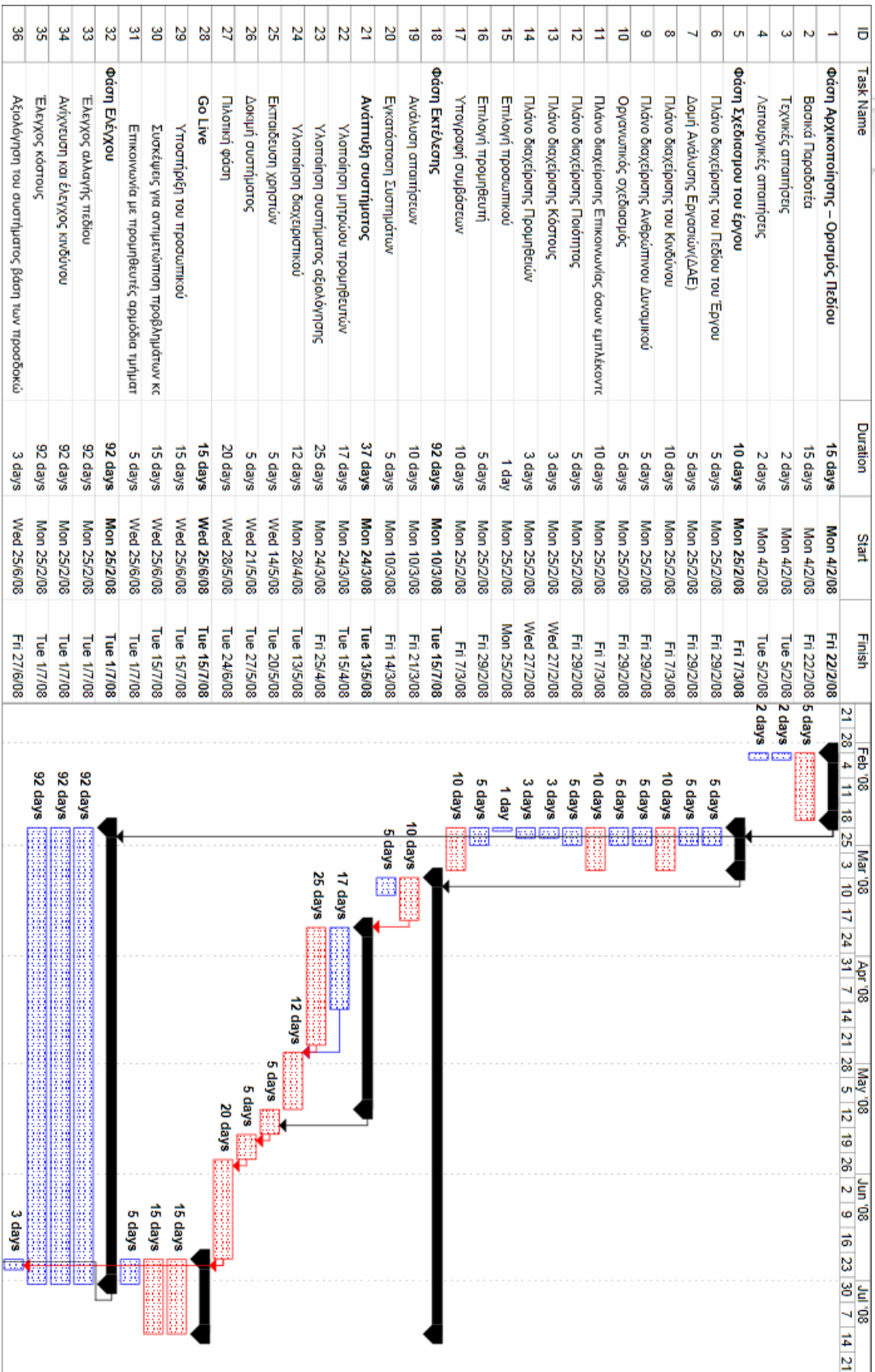
Η ομάδα έργου στη συγκεκριμένη περίπτωση ήταν μια έμπειρη ομάδα που είχε φέρει εις πέρας παρόμοια μεσαίας κλίμακας έργα και για αυτό το λόγο δόθηκε μεγαλύτερη ελευθερία στην ομάδα ανάπτυξης του πληροφοριακού συστήματος. Παρότι κάτι τέτοιο δεν συνίσταται στο συγκεκριμένο έργο αποδείχθηκε ότι λειτούργησε.

Κλείνοντας καταλαβαίνουμε ότι για κάθε έργο πρέπει να επιλέγεται η μεθοδολογία ανάλογα με τους ειδικούς παράγοντες του κάθε έργου όπως είναι η ομάδα έργου, το είδος του έργου, η εμπειρία της ομάδα. Δεν υπάρχει γενικός κανόνας για το ποιος είναι ο καλύτερος τρόπος διοίκησης ενός έργου και γι' αυτό χρειάζεται προσοχή ώστε να επιλέξουμε το καταλληλότερο ώστε το τελικό παραδοτέο όχι μόνο

να ικανοποιεί τις απαιτήσεις αλλά και να ικανοποιεί τους όρους που είχαν τεθεί στην αρχή όπως είναι ο προϋπολογισμός και το κόστος.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Παράρτημα Α GANTT CHART



ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Βασιλακόπουλος Γ., Χρυσικόπουλος Β., “Πληροφοριακά Συστήματα Διοίκησης”, Σταμούλης, Πειραιάς 1990.
2. Βεσκούκης Β., “Τεχνολογία Λογισμικού Ι”, Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα 2000.
3. Γιαννακόπουλος Δ., Παπουτσής Ι., “Πληροφοριακά Συστήματα Διοίκησης”, Έλλην, Περιστέρι 1996.
4. Δημητριάδης Α., “Διοίκηση-Διαχείριση πληροφοριακών συστημάτων”, Εκδόσεις Νέων Τεχνολογιών, Αθήνα 1998.
5. Κιουντούζης Ε., “Ανάλυση και Σχεδιασμός Συστημάτων”, Εκδόσεις Μπένου, Αθήνα 1993.
6. Παιδαγωγικό Ινστιτούτο, “Πληροφοριακά Συστήματα”, Λιβάνη, Αθήνα 2000.
7. Σκορδαλάκης Μ., “Εισαγωγή στην Τεχνολογία Λογισμικού”, Συμμετρία, Αθήνα 1991.
8. Σταμέλος Ι., και Σφέτσος Π., “Agile methods and extreme programming (XP) Ευέλικτες μέθοδοι και Ακραίος Προγραμματισμός”, Θεσσαλονίκη, 2003.
9. Τουρίκη Κ., “Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης”, Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, 2010.
10. Achievo, 2011; <http://www.achievo.org/>
11. Alleman G., “Agile Project Management: Methods for IT Projects. The Story of Managing Projects: A Global, Cross- Disciplinary Collection of Perspectives”, Greenwood Press / Quorum Books, 2002.
12. Andersen E. A., Grude K. V., and Hague T., “Goal Directed Project Management”, Kogan Page, 2004.
13. Beck K., “Extreme Programming Explained”, Addison Wesley, 2000.
14. Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R.

- C., Mellor S., Schwaber K., Sutherland J., and Thomas D., “Manifesto for Agile Software Development”, 2001.
15. Charvat J., “Project Management Methodologies—Selecting, Implementing, and Supporting Methodologies and Processes for Projects”, New Jersey: John Wiley & Sons, INC, 2003.
 16. Chromatic, “Extreme Programming Pocket Guide”, O’Reilly, 2003.
 17. Crawford L., “Global body of project management. Knowledge and standards”, in Morris P. & Pinto J., “The Wiley Guide to Managing Projects”, Wiley, pp. 1150 – 1196, 2004.
 18. GanttProject, 2011: <http://www.ganttproject.biz/>
 19. Jacobson I., Booch G. and Rumbaugh, J, “The Unified Software Development Process”, Addison-Wesley, 1999.
 20. Karlström D. and Runeson P., “Integrating Agile Software Development into Stage-Gate Managed Product Development”, technical report CODEN: LUTEDX (TETS-7203) / 1-34 / (2004) & local 16, 2004.
 21. Kruchten, P., “The Rational Unified Process – An Introduction”, Addison-Wesley 2nd edition, 2000.
 22. Kruchten P, “The Rational Unified Process: An Introduction”, Pearson Education Inc., 2003.
 23. Lindvall M., Basili V., Boehm B., Costa P., Dangle K., Shull F., Tesoriero R., Williams L., and Zelkowitz M., “Empirical Findings in Agile Methods”, Lecture Notes in Computer Science, vol. 2418, pp. 81-92, 2002.
 24. Microsoft Project, 2011: <http://www.microsoft.com/project/en/us/default.aspx>
 25. OmniPlan, 2011: <http://www.omnigroup.com/products/omniplan/>
 26. Open Workbench, 2011: <http://sourceforge.net/projects/openworkbench/>
 27. Pertmaster, 2011:
 28. http://www.2020software.com/products/Primavera_Systems_Pertmaster.asp
 29. PlanningForce, 2011: <http://www.planningforce.com/>

30. Pressman R. S., “Software Engineering: A Practitioners Approach”, McGraw – Hill 4th edition, 2001.
31. Prince2, 2011: www.crazycolour.com
32. Rational (2005) Rational Unified Process: Best practices for software development teams
33. Sommerville I., “Software Engineering”, Addison – Wesley, London 2001.
34. Comparing PRINCE2 with PMBoK,2002, R. Max Wideman. AEW Services, Vancouver, BC, Canada, 2002
35. ΕΔΔΕ, 2006, αναφέρεται στον Εμίρη, 2006
36. Prince 2:a methodology of project management, Christophe Pincemaille,2008
37. Project Management Institute, *A Guide to the Project Management Body of Knowledge:(PMBOK® Guide), 4th edition*, PMI, 2008
38. Weiss, Joseph and Wysocki, Robert, *Five-phase Project Management: A Practical Planning And Implementation Guide*, Basic Books, 1992
39. PMBOK, 2004
40. Janson and Smith, 1985
41. Huffaker, 1986