

ΤΕΙ ΜΕΣΟΛΟΓΓΙΟΥ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ ΚΑΙ
ΣΥΝΕΤΑΙΡΙΣΜΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΣΚΟΠΗΣΗ ΤΕΧΝΙΚΩΝ ΚΑΙ
ΕΡΓΑΛΕΙΩΝ ΔΙΟΙΚΗΣΗΣ ΕΡΓΩΝ ΓΙΑ
ΕΡΓΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΠΟΥΔΑΣΤΗΣ: ΓΡΙΒΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ, ΑΜ: 8509

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΒΑΣΙΛΕΙΑΔΗΣ ΒΑΣΙΛΕΙΟΣ

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	6
1 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΔΙΟΙΚΗΣΗΣ ΔΙΑΧΕΙΡΙΣΗΣ	7
1.1 Ιστορικά Στοιχεία	7
1.1.1 Διάγραμμα Gantt.....	8
1.1.2 Οι τεχνικές PERT/CPM	10
1.1.2.1 CPM (Critical Path Method)	11
1.1.2.2 PERT (Program Evaluation Review Technique)	12
1.2 Βασικές Έννοιες Διοίκησης – Διαχείρισης Έργου	13
1.2.1 Εμπλεκόμενοι στο Έργο	14
1.2.2 Πόροι Έργου.....	16
1.2.3 Υπηρεσίες Έργου.....	16
1.2.4 Παράγοντες που Επιδρούν στο Έργο.....	16
1.2.5 Βασικό Τρίγωνο	17
1.3 Οργανωτικές δομές	19
1.3.1 Οργανωτική δομή «Pure Project».....	20
1.3.2 Οργανωτική δομή «Functional Project»	20
1.3.3 Οργανωτική δομή «Matrix Project»	21
2 Τεχνικές Υλοποίησης Έργων Πληροφορικής.....	24
2.1 Ευέλικτη Ανάπτυξη Λογισμικού.....	24
2.1.1 Ακραίος προγραμματισμός (Extreme programming, XP).....	27
2.1.2 Μέθοδος Scrum (Scrum Method)	31
2.1.3 Μεθοδολογίες κρυστάλλου (Crystal Methods)	34
2.1.4 Ανάπτυξη με βάση τα χαρακτηριστικά (Feature Driven Development).....	35
2.1.5 Μέθοδος Ανάπτυξης Δυναμικών Συστημάτων (Dynamic Systems Development Model Methodology)	36
2.1.6 Lean Development	38
2.2 Μέθοδος ITIL (Information Technology Infrastructure Language).....	39
2.3 Κύκλος ζωής ανάπτυξης πληροφοριακών συστημάτων.....	40
2.3.1 Κοινή Ανάπτυξη Εφαρμογής (Joint Application Methodology).....	41
2.3.2 Μοντέλο του καταρράκτη.....	42
2.3.3 Ταχεία ανάπτυξη εφαρμογής	44
2.3.4 Το Σπειροειδές Μοντέλο	45
2.3.4.1 Επαναληπτική Ενοποιημένη Διαδικασία	46

2.3.5	Το Αυξητικό Μοντέλο	47
2.3.5.1	Το Μοντέλο της Δημιουργίας Πρωτότυπου.....	48
3	Τεχνικές Διοίκησης Έργων.....	50
3.1	Tenstep	50
3.2	Prince2.....	51
3.3	PMBOK.....	58
3.4	Hermes.....	61
3.5	Εργαλεία Διοίκησης Έργων.....	64
3.5.1	GanttProject	64
3.5.2	OmniPlan	65
3.5.3	Open Workbench	66
3.5.4	Pertmaster.....	67
3.5.5	PlanningForce	69
3.5.6	Microsoft Project	70
3.5.7	Achievo	72
4	ΣΥΜΠΕΡΑΣΜΑΤΑ	75
	ΒΙΒΛΙΟΓΡΑΦΙΑ	79
	ΙΣΤΟΣΕΛΙΔΕΣ	80

ΠΕΡΙΕΧΟΜΕΝΑ ΔΙΑΓΡΑΜΜΑΤΩΝ

Εικόνα 1.1 Μορφή του διαγράμματος Gantt.	9
Εικόνα 1.2: Διάγραμμα δικτύου (δείχνει τις λογικές σχέσεις μεταξύ των δραστηριοτήτων).	11
Εικόνα 1.3: Σχηματική αναπαράσταση της σχέσης «Κόστος – Χρόνος – Ποιότητα».	18
Εικόνα 1.4: Οργανωτική δομή «Functional Project».	21
Εικόνα 1.5: Οργανωτική δομή «Matrix Project».	22
Εικόνα 2.3: Διαδικασία Scrum.....	32
Εικόνα 2.2: Η διαδικασία του FFD.	36
Εικόνα 2.1: Διάγραμμα διαδικασίας DSDM.	37
Εικόνα 2.4: Γενικό διάγραμμα του μοντέλου του καταρράκτη.	43
Εικόνα 2.5 : Το σπειροειδές μοντέλο.....	45
Εικόνα 2.6 Διάγραμμα του αυξητικού μοντέλου.....	47
Εικόνα 2.7: Γενικό διάγραμμα του μοντέλου της δημιουργίας πρωτοτύπου.	49
Εικόνα 3.1: Η μεθοδολογία Prince2.	53
Εικόνα 3.2: Προεπισκόπηση των περιοχών γνώσης της διαχείρισης έργου και των διαδικασιών διαχείρισης έργου.	60
Εικόνα 3.3: Οι έξι φάσεις της μεθόδου Hermes	62
Εικόνα 3.4: Οι φάσεις της μεθόδου Hermes ανάλογα με τον τύπο του έργου	62
Εικόνα 3.5: Παράδειγμα όψεων της μεθόδου Hermes	63
Εικόνα 3.6: Παράδειγμα συστήματος GanttProject	65
Εικόνα 3.7: Παράδειγμα συστήματος OmniPlan.	66
Εικόνα 3.8: Παράδειγμα συστήματος OpenWorkbench.....	67
Εικόνα 3.9: Παράδειγμα συστήματος Pertmaster.....	68
Εικόνα 3.10: Παράδειγμα συστήματος PlanningForce.....	70
Εικόνα 3.11: Παράδειγμα συστήματος Microsoft Project.....	72
Εικόνα 3.12: Παράδειγμα συστήματος Achievo.....	74

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

<i>Πίνακας 2.1: Χαρακτηριστικά μεθοδολογίας Agile Software Development.</i>	<i>25</i>
<i>Πίνακας 2.2: Χαρακτηριστικά της τεχνικής XP.</i>	<i>29</i>
<i>Πίνακας 4.1: Γενικά χαρακτηριστικά γνωρίσματα των ευέλικτων μεθόδων διαχείρισης έργων πληροφορικής.</i>	<i>77</i>

ΕΙΣΑΓΩΓΗ

Πρωταρχικός σκοπός της πτυχιακής είναι η επισκόπηση των εργαλείων και των τεχνικών διοίκησης έργων για έργα πληροφορικής. Στοχεύει στην παρουσίαση της αναγκαιότητας εφαρμογής μιας μεθοδολογίας και στην παροχή μιας συνολικής εικόνας της διαχείρισης έργου μέσα από την αναφορά του συνόλου των μεθοδολογιών που υπάρχουν.

Στο πρώτο κεφάλαιο παρουσιάζεται το αντικείμενο που πραγματεύεται η συγκεκριμένη εργασία. Γίνεται μία ιστορική αναδρομή που αφορά στην διοίκηση έργων, αναλύονται οι βασικές έννοιες Διοίκησης – Διαχείρισης Έργου που πρέπει να γνωρίζει ο αναγνώστης και τέλος περιγράφονται οι οργανωτικές δομές που εφαρμόζονται σε ένα έργο για την περαίωση του.

Στα κεφάλαια που ακολουθούν στην συνέχεια αναλύονται οι τεχνικές διοίκησης και υλοποίησης έργων. Οι τεχνικές αυτές, διακρίνονται σε **τεχνικές υλοποίησης έργων πληροφορικής** και σε **τεχνικές διοίκησης έργων**. Η ανάγκη αυτής της διαφοροποίησης προκύπτει από το γεγονός ότι σε πολλές τεχνικές έργων η μεθοδολογία που περιγράφεται δεν εφαρμόζεται σε όλα τα έργα παρά μόνο σε έργα που αφορούν στην πληροφορική και στην ανάπτυξη λογισμικού.

Λίγο πριν κλείσουμε την εργασία παραθέτονται τα συμπεράσματα τα οποία προκύπτουν από την επισκόπηση των εργαλείων και των μεθόδων διοίκησης έργων πληροφορικής.

Τέλος παρουσιάζονται οι πηγές από όπου αντλήσαμε τις πληροφορίες για την υλοποίηση και ανάπτυξη του αντικειμένου της εργασίας αυτής.

ΚΕΦΑΛΑΙΟ 1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΔΙΟΙΚΗΣΗΣ ΔΙΑΧΕΙΡΙΣΗΣ

1.1 Ιστορικά Στοιχεία

Ο Χένρι Γκαντ, Αμερικανός μηχανικός και κοινωνικός επιστήμονας, εισήγαγε τις αρχές του προγραμματισμού και ελέγχου στη διαχείριση έργων και θεωρείται ο πατέρας του γνωστικού πεδίου αυτού. Το διάγραμμα Γκαντ, ένα ραβδόγραμμα που παρουσιάζει τις δραστηριότητες του έργου, πήρε το όνομα του προς τιμή του. Ο Γκάντ μαζί με τον Φρέντερικ Τέιλορ, ο οποίος έβαλε τις βάσεις της επιστημονικής διαχείρισης (*scientific management*), έθεσαν τις θεμέλιες αρχές της διαχείρισης έργων.

Στην δεκαετία του πενήντα, εξελίχθηκαν οι σύγχρονες αρχές της διαχείρισης έργων οι οποίες έκαναν τη διαχείριση έργων όχι μόνο ένα γνωστικό αντικείμενο αλλά και ένα επάγγελμα. Την δεκαετία αυτή διαμορφώθηκαν δύο βασικά μαθηματικά μοντέλα χρονοπρογραμματισμού δραστηριοτήτων, οι μέθοδοι PERT και CPM οι οποίες αποτέλεσαν ορόσημο στη διαχείριση έργων.

Το Ναυτικό των Ηνωμένων Πολιτειών για το έργο για της ανάπτυξης των πυραυλικών συστημάτων Polaris ανέπτυξε την μέθοδο PERT (*Program Evaluation and Review Technique*). Αντίστοιχα οι εταιρείες DuPont Corporation και Remington Rand Corporation ανέπτυξαν την μέθοδο κρίσιμου διαδρομής, CPM (*Critical Path Method*) που είχε σαν στόχο της την διαχείριση έργων συντήρησης. Η εξάπλωση και υιοθέτηση των μεθόδων αυτών έγινε πολύ γρήγορα, έτσι ώστε σήμερα είναι αναπόσπαστο κομμάτι για τη διαχείριση έργων.

Σήμερα ο τομέας της διαχείρισης έργων θεωρείται ιδιαίζόντως αναπτυγμένος και προσελκύει ιδιαίτερο ενδιαφέρον όχι μόνο στον ιδιωτικό, δημόσιο τομέα αλλά και στην ακαδημαϊκή κοινότητα. Πειστήριο του γεγονότος αυτού αποτελεί η ύπαρξη πολλών διεθνών οργανισμών με έντονη δραστηριότητα που έχουν ως βασικό σκοπό την ανάπτυξη του γνωστικού πεδίου της διαχείρισης έργων. Ανάμεσα σε αυτούς

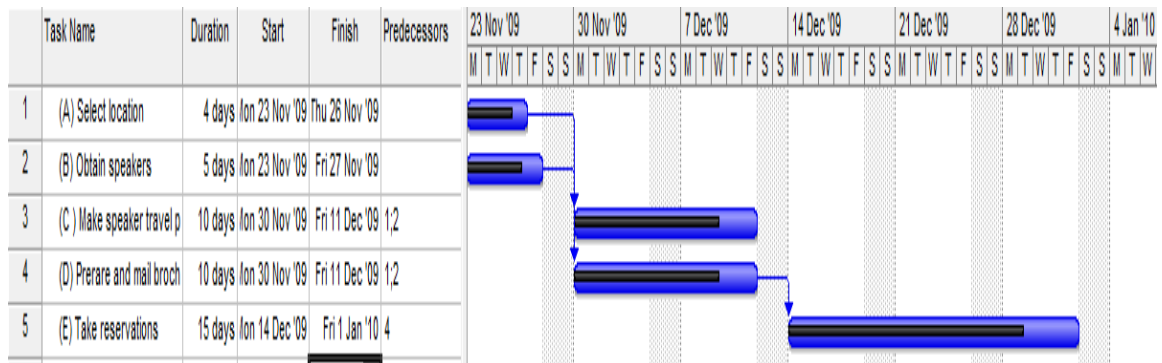
ξεχωριστή θέση κατέχουν οι *International Project Management Association (IPMA)* και το *Project Management Institute (PMI)*.

1.1.1 Διάγραμμα Gantt¹

Παρότι ο *Henry Gantt* θεωρείται πατέρας των τεχνικών σχεδιασμού και ελέγχου, είναι κοινά αποδεκτό ότι η ανάπτυξη συγκεκριμένων τεχνικών παρατηρήθηκε στα πλαίσια των στρατιωτικών και αεροδιαστημικών έργων που υλοποιήθηκαν στις Ηνωμένες Πολιτείες και τη Μεγάλη Βρετανία κατά τις δεκαετίες του 1950 και 1960. Η σύγχρονη ιστορία της διαχείρισης έργου, όμως, αρχίζει με την ανάπτυξη του γραμμικού διαγράμματος στις αρχές τις δεκαετίας του 1900. Η τεχνική της κατασκευής γραμμικών διαγραμμάτων εγκαινιάστηκε κατά τον Πρώτο Παγκόσμιο Πόλεμο, όταν ο Αμερικανός *Henry Gantt (1861-1919)* επινόησε το γραμμικό διάγραμμα ως εποπτικό εργαλείο προγραμματισμού και ελέγχου των ναυπηγικών έργων με τα οποία ασχολούνταν.

Η συμβολή του στη διαχείριση έργου είχε ως αποτέλεσμα τα γραμμικά διαγράμματα προγραμματισμού να ονομάζονται διαγράμματα *Gantt*. Το διάγραμμα *Gantt* αποτελεί μια μορφή γραφικής αναπαράστασης ενός χρονοδιαγράμματος. Η τεχνική σύνταξης του διαγράμματος είναι η εξής: αναλύεται, αρχικά, όλο το έργο σε επί μέρους εργασίες γνωστές με τον όρο δραστηριότητες. Για κάθε μία από τις επί μέρους εργασίες καθορίζονται η μέθοδος εκτέλεσής της, οι απαιτούμενοι πόροι και ο απαιτούμενος χρόνος για την εκτέλεσή της. Οι επιμέρους αυτές εργασίες τίθενται σε ένα διάγραμμα όπως αυτό της παρακάτω εικόνας.

¹ Καντζαρη Μ., «Μαθηματικά των υπολογιστών και των αποφάσεων - μοντέλα για τον χρονοπρογραμματισμό έργων με περιορισμένους πόρους», Πανεπιστήμιο Πατρών, 2010.



Εικόνα 0.1 Μορφή του διαγράμματος Gantt.¹

Στην πρώτη στήλη του διαγράμματος αναγράφονται ιεραρχημένες οι επί μέρους εργασίες ονομαστικά ή κωδικοποιημένες. Στον οριζόντιο άξονα του διαγράμματος σχεδιάζεται η χρονική κλίμακα του έργου (σε ημέρες, εβδομάδες ή μήνες). Ο προγραμματισμός κάθε εργασίας αντιστοιχεί στη χάραξη μιας οριζόντιας ράβδου από την ημερομηνία έναρξης έως την ημερομηνία λήξης της εργασίας. Το μήκος αυτής της.

Ο Gantt διεύρυνε τη χρήση του διαγράμματος ώστε να απεικονίσει και την πρόοδο των εργασιών. Χάραξε μία δεύτερη ράβδο, κατά μήκος της ράβδου που απεικόνιζε την προγραμματισμένη εργασία (γραμμή προγραμματισμού). Η δεύτερη ράβδος (γραμμή προόδου) απεικονίζει το τμήμα της εργασίας που έχει ήδη εκτελεστεί. Η θέση της γραμμής προόδου σε σχέση με τη γραμμή προγραμματισμού υποδεικνύει το ποσοστό ολοκλήρωσης της εργασίας και την υπολειπόμενη διάρκεια μέχρι την αποπεράτωσή της. Η σύγκριση αυτή γίνεται με μία κάθετη γραμμή που υποδηλώνει τη χρονική στιγμή που γίνεται ο έλεγχος προόδου του έργου. Η θέση της γραμμής προόδου σε σχέση με την ένδειξη χρονική στιγμή ελέγχου υποδήλωνε την πρόοδο του έργου που υλοποιήθηκε σε σχέση με τον αρχικό προγραμματισμό.

Το διάγραμμα Gantt, παρά την προφανή χρησιμότητά του, δεν επαρκεί για περίπλοκους σχεδιασμούς έργων, διότι δεν απεικονίζονται οι σχέσεις αλληλεξάρτησης των επί μέρους εργασιών. Δεν είναι δηλαδή εμφανές, ποιες εργασίες πρέπει να αποπερατωθούν ώστε να καταστεί δυνατή η έναρξη εκτέλεσης μιας ορισμένης εργασίας και δεν παρουσιάζει την επίδραση μιας καθυστέρησης ή επίσπευσης σε κάποια φάση του έργου.

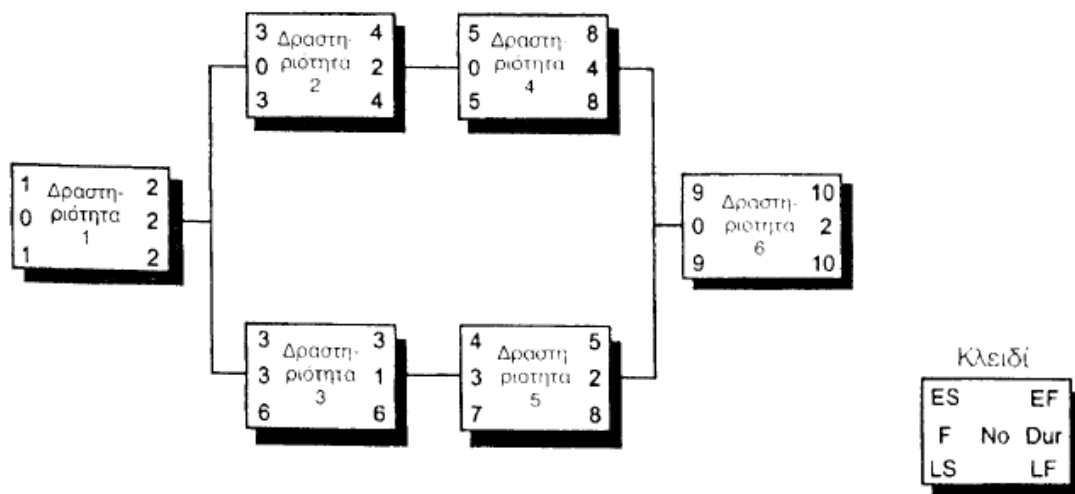
Το διάγραμμα Gantt άντεξε στην δοκιμασία του χρόνου. Έγιναν τροποποιήσεις και διορθώθηκαν αδυναμίες του. Στα διαγράμματα Gantt προστέθηκαν ορόσημα (milestones) που δείχνουν συγκεκριμένα σημεία στο χρόνο (συνήθως ανά εξάμηνο) που πρέπει να έχουν ολοκληρωθεί ορισμένες εργασίες και σταδιοδείκτες που δείχνουν πότε άρχισε και πότε τελείωσε η συγκεκριμένη εργασία. Επάνω στις συμπαγείς ράβδους τοποθετούνται σημεία ενδεικτικά της σημασίας κάθε εργασίας (κρίσιμοι έλεγχοι, αναθεωρήσεις). Στην περίπτωση αυτή οι μονάδες χρόνου αντικαθίστανται από ημερομηνίες. Οι σχέσεις αλληλεξάρτησης των επιμέρους εργασιών μπορούν να απεικονιστούν με βέλη που συνδέουν τις ράβδους (εργασίες) καθιστώντας έτσι το διάγραμμα Gantt δίκτυο. Πολλές φορές, όμως, είναι απλούστερο να έχουμε χωριστά το διάγραμμα Gantt και το δίκτυο συσχέτισης των εργασιών του έργου.

1.1.2 Οι τεχνικές PERT/CPM

Για να είναι εποικοδομητικό το χρονοδιάγραμμα ενός έργου θα πρέπει να λαμβάνει υπόψη του όχι μόνο τον χρόνο που διαρκεί η κάθε δραστηριότητάς του, αλλά και τις λογικές σχέσεις που συνδέουν τις δραστηριότητες μεταξύ τους. Είναι μεγάλης σημασίας ο προσδιορισμός των λογικών σχέσεων έτσι ώστε να είμαστε σε θέση να προβλέψουμε τι επιρροή θα έχει στην εξέλιξη του έργου οποιαδήποτε πιθανή αλλαγή του χρονοδιαγράμματος. Με το πέρασμα του χρόνου το μέγεθος και η πολυπλοκότητα των έργων πολλαπλασιάστηκαν, οπότε διαπιστώθηκε ότι το διάγραμμα Gantt δεν ήταν ικανό να απεικονίσει τέτοιας μορφής σχέσεις.

Είναι σημαντικό να αναφέρουμε, ότι κατά τη διάρκεια της δεκαετίας του 1950, επισημάνθηκαν πολύ συχνές υπερβάσεις χρόνου και κόστους των έργων τόσο στο εμπόριο όσο και στη βιομηχανία. Αρχικά υποστηρίχτηκε ότι αυτό συμβαίνει διότι οι εκτιμήσεις που αφορούσαν στην εξέλιξη των έργων ήταν πολύ αισιόδοξες. Σιγά-σιγά η άποψη αυτή εγκαταλείφτηκε επειδή οι μελετητές των έργων κατέληξαν στο συμπέρασμα ότι οι υπερβάσεις αυτές οφείλονταν στο γεγονός ότι οι διαθέσιμες τεχνικές ελέγχου και προγραμματισμού για τη διαχείριση έργων ήταν ανεπαρκείς.

Οι δυσκολίες που προαναφέρθηκαν είχαν σαν αποτέλεσμα την ανάπτυξη μεθοδολογιών χρονικού προγραμματισμού που μπορούσαν να ενσωματώσουν ως παραμέτρους τους πόρους, τις αγορές και το κόστος. Αυτό διαδραμάτισε σημαντικό ρόλο στη γέννηση της τεχνικής προγραμματισμού που βασίζεται στην κατασκευή λογικών δικτύων. Στη δικτυακή διάταξη κάθε δραστηριότητα απεικονίζεται ως ένα τετράγωνο ή κύκλος. Η λογική αλληλουχία του έργου φαίνεται από τον τρόπο που διατάσσονται τα τετράγωνα ή οι κύκλοι από αριστερά προς τα δεξιά. Αυτό μπορούμε να το δούμε αναλυτικότερα στην παρακάτω εικόνα.



Εικόνα 0.2: Διάγραμμα δικτύου (δείχνει τις λογικές σχέσεις μεταξύ των δραστηριοτήτων).²

Τόσο η PERT όσο και η CPM είναι στενά συνδεδεμένες με τα διαγράμματα δικτύου και είναι αυτές που θα περιγράψουμε περιληπτικά στην συνέχεια αυτής της εργασίας.

1.1.2.1 CPM (Critical Path Method)

Όπως αναφέρθηκε και στην εισαγωγή, η μέθοδος αυτή αναπτύχθηκε για πρώτη φορά την περίοδο μεταξύ Δεκεμβρίου 1956 και Φεβρουαρίου 1959 από τις DuPont Company and Remington Rand Univas, με στόχο την ελαχιστοποίηση του χρόνου επιδιόρθωσης συντήρησης και κατασκευής μεγάλων κτιριακών μονάδων

² Καντζαρη Μ., «Μαθηματικά των υπολογιστών και των αποφάσεων - μοντέλα για τον χρονοπρογραμματισμό έργων με περιορισμένους πόρους», Πανεπιστήμιο Πατρών, 2010.

(εργοστασίων). Η καλύτερη αξιοποίηση του διαθέσιμου υλικού (έμφυχου και άψυχου), ώστε να μην μένουν άτομα ανενεργά μετά την ολοκλήρωση κάποιων δραστηριοτήτων ήταν ο βασικός άξονας πάνω στον οποίο κινήθηκε η ανάπτυξη αυτής της μεθόδου. Η τεχνική CPM είναι ιδιαίτερα χρήσιμη για προγράμματα κατασκευής τεχνικών έργων, αλλά μπορεί να χρησιμοποιηθεί με επιτυχία και σε άλλα προγράμματα όπως για παράδειγμα σε προγράμματα συναρμολογήσεως μεγάλων μηχανολογικών κατασκευών, ανάπτυξης νέων προϊόντων, κλπ. Κάνοντας χρήση της μεθόδου αυτής, υπολογίζεται ένας συγκεκριμένος συντομότερος και αργότερος χρόνος έναρξης και λήξης για κάθε δραστηριότητα, σύμφωνα με την καθορισμένη σειρά αλληλουχίας του δικτύου.

1.1.2.1 PERT (Program Evaluation Review Technique)

Αναπτύχθηκε για πρώτη φορά το 1958 κατά την κατασκευή των οπλικών συστημάτων Polaris από το Αμερικανικό Ναυτικό. Το έργο ήταν σύνθετο και οι συμμετέχοντες αποφάσισαν ότι υπήρχε η ανάγκη ενός ολοκληρωμένου σχεδιασμού που να λαμβάνει υπόψη όχι μόνο την εμπειρία των ειδικών αλλά και πιθανοθεωρητικά φαινόμενα.

Η πρώτη φάση στην ανάπτυξη της τεχνικής αυτής έγινε γνωστή ως PERT-Time, επειδή είχε δοθεί μεγαλύτερο βάρος στον χρόνο που θα διαρκούσε το έργο που σχεδιαζόταν. Η δεύτερη φάση ονομάστηκε PERT/Cost επειδή πρωτεύουσας σημασίας ήταν το κόστος. Η PERT/Cost αναπτύχθηκε μεταγενέστερα, αλλά δεν αποδείχθηκε αποτελεσματική στην πράξη και έχει σχεδόν δεν χρησιμοποιείται.

Η μέθοδος λειτουργεί κάνοντας χρήση ενός σταθμισμένου μέσου της εκτιμώμενης διάρκειας έτσι ώστε να μπορέσει να υπολογίσει την διάρκεια των δραστηριοτήτων. Η PERT διαφέρει από την CPM κυρίως στο ότι θεωρεί ότι ο χρόνος μπορεί να υπολογιστεί από το μέσο μίας κατανομής (μέση διάρκεια) κι όχι από την πιο πιθανή διάρκεια, όπως η CPM.

1.2 Βασικές Έννοιες Διοίκησης – Διαχείρισης Έργου

Η **εργασία (work)**, η οποία παράγεται από έναν **οικονομικό οργανισμό (organization)** είναι αποτέλεσμα δύο παραγόντων: **λειτουργιών (operations)** και **έργων (projects)**.

Ως **έργο (project)** ορίζεται μια χρονικά περιορισμένη προσπάθεια για τη παραγωγή μιας μοναδικής υπηρεσίας ή ενός μοναδικού προϊόντος.

Με τον όρο **λειτουργία (operation)** αναφερόμαστε σε μια χρονικά συνεχή και επαναλαμβανόμενη προσπάθεια .

Παραδείγματος χάρη, έστω ένας κατασκευαστικός οργανισμός ο οποίος έχει ως αντικείμενο απασχόλησης την ανέγερση κτιρίων. Στην περίπτωση αυτή, η ανέγερση κτιρίων αποτελεί λειτουργία μιας και είναι μια χρονικά συνεχής και επαναλαμβανόμενη πράξη. Από την άλλη, η ανέγερση ενός συγκεκριμένου κτιρίου αποτελεί έργο καθότι είναι μία πράξη με ορισμένη χρονική διάρκεια και αποτελεί μοναδικό προϊόν (το κτίριο αυτό είναι διαφορετικό από οποιοδήποτε άλλο κατασκεύασε η εταιρία). Το έργο επομένως αποτελεί μέρος της λειτουργίας.

Ένα έργο απαιτεί πολλές λειτουργίες όπως για παράδειγμα σχεδιασμό, έλεγχο, διαχείριση προσωπικού κτλ.. Δηλαδή το έργο διαθέτει τις δικές του λειτουργίες. Ένα σύνολο έργων που σχετίζονται άμεσα μεταξύ τους και οι επαναλαμβανόμενες προσπάθειες οι οποίες αποτελούν έργα, είναι τα βασικά συστατικά της έννοιας του **προγράμματος (program)**.

Στη σημερινή εποχή, στους οργανισμούς, αναπτύσσεται όλο και περισσότερο η εργοκεντρική αντίληψη διοίκησης, δηλαδή **‘Η Διοίκηση μέσω έργων (Management by project)’,** που βασίζεται στην υποκατάσταση λειτουργιών από έργα.

Για να γίνουμε σαφέστεροι θα μπορούσαμε να πούμε ότι η μηνιαία έκδοση ενός περιοδικού αποτελεί και έργο και επομένως η συνεχής εκδοτική λειτουργία μπορεί να υποκατασταθεί από διοίκηση κατά έργο (μηνιαία προσπάθεια για κάθε τεύχος ξεχωριστά).

Το πιο χαρακτηριστικό παράδειγμα αποτελεί στη βιομηχανία η μέθοδος παραγωγής που ονομάζεται *just in time* - JIT (στο χρόνο που πρέπει). Θεωρούμε ότι μια βιομηχανία παράγει κάποιο προϊόν το οποίο με την κλασική μέθοδο παραγωγής του προϊόντος η παραγωγή αποτελεί μια συνεχής και επαναλαμβανόμενη λειτουργία. Ο διαχωρισμός των προϊόντων σε παρτίδες που στέλνονται σε κάθε πελάτη γίνεται εκ των υστέρων. Σύμφωνα με την μέθοδο JIT, οργανώνεται και παράγεται κάθε φορά ακριβώς μια συγκεκριμένη παρτίδα (φουρνιά) προϊόντων, που απευθύνεται σε συγκεκριμένους πελάτες. Άρα η παραγωγική διαδικασία γίνεται τώρα κατά έργο, δηλαδή τη συγκεκριμένη παρτίδα (φουρνιά).

Σύμφωνα με μελέτη της διεθνούς λογιστικής εταιρίας *Arthur Andersen and Co*, η εφαρμογή της *Just in time* σε αριθμό επιχειρήσεων είχε ως όφελος τη μείωση:

1. Αποθεμάτων ημι – κατεργασμένων προϊόντων κατά 90%
2. Χρόνου παραγωγής ενός προϊόντος κατά 90%
3. Ακατάλληλων και ανακατασκευασμένων προϊόντων κατά 80%
4. Χρόνου προετοιμασίας των μηχανών κατά 75%
5. Χώρου εγκαταστάσεων και αποθηκών κατά 50%
6. Άμεσης και έμμεσης εργασίας κατά 20%

1.2.1 Εμπλεκόμενοι στο Έργο³

Το έργο αποτελεί προϊόν της ανάγκης. Προορισμός του είναι η βελτίωση μιας υπάρχουσας κατάστασης, η οποία δεν κρίνεται ικανοποιητική ή να δημιουργήσει μια ολοκληρωτικά νέα κατάσταση με νέες ευκαιρίες. Σ' αυτήν την προσπάθεια εμπλέκονται διαφορετικοί άνθρωποι και κοινωνικές ομάδες, με συχνά διαφορετικές βλέψεις σε σχέση με το έργο, οι οποίοι όμως αποτελούν την προϋπόθεση για να υπάρξει το έργο. Καλούνται «οι εμπλεκόμενοι στο έργο» ή «τα ενδιαφερόμενα μέρη» και οι σπουδαιότεροι αυτών είναι:

³ Δημητριάδης Α., «Διοίκηση – Διαχείριση Έργου», Εκδόσεις Νέων Τεχνολογιών, Έκδοση 4^η, 2009.

1. **Ο κύριος του έργου (Project Owner)**, δηλαδή ο οργανισμός για τον οποίο γίνεται το έργο, για τον οποίο καταρτίζεται η σύμβαση και ο οποίος θέτει τους όρους εκτέλεσης του έργου.
2. **Ο οργανισμός υλοποίησης ή φορέας υλοποίησης του έργου (Implementing Organization)**, ο οποίος αναλαμβάνει την υλοποίηση του έργου, στον οποίο ανήκουν οι Εργολήπτες ή Ανάδοχοι του έργου (Contractors).
3. **Η αναθέτουσα αρχή (Contracting Authority)**, η οποία αναθέτει το έργο με κάποια διαδικασία (άμεση ανάθεση, διαγωνισμός κτλ..)στο Φορέα υλοποίησης, εφόσον φυσικά πρόκειται για δημόσιο έργο.
4. **Οι χρηματοδότες του έργου (Financiers, Sponsors)**, οι οποίοι πληρώνουν τις δαπάνες του έργου.
5. **Οι ωφελούμενοι ή χρήστες ή πελάτες του έργου (Project users and customers)**, οι οποίοι είτε είναι αποδέκτες του προϊόντος του έργου είτε χρησιμοποιούν το προϊόν του έργου. Το πόσο καλά το προϊόν του έργου καλύπτει τις απαιτήσεις των χρηστών του, εκφράζεται από την Ικανοποίηση των Χρηστών (User Satisfaction), που αποτελεί και δείκτη ποιότητας του προϊόντος του έργου.

Ένα έργο μπορεί να εκτελείται με τα ίδια μέσα του κυρίου του έργου ή να χρησιμοποιεί ανάδοχο. Στην περίπτωση ενός δημόσιου έργου κύριος, αλλά και αναθέτουσα αρχή είναι η πολιτεία με τα αντίστοιχα όργανά της και ωφελούμενοι ή χρήστες του έργου θα πρέπει να είναι το κοινωνικό σύνολο. Χρηματοδότης μπορεί να είναι το δημόσιο, αλλά μπορεί να είναι και ιδιώτες (**Συγχρηματοδοτούμενο έργο – Co financed Project**), ή αποκλειστικά ιδιώτες με αντάλλαγμα την αποκλειστική εκμετάλλευση του προϊόντος του έργου για κάποια χρόνια (πχ αεροδρόμια Ελ. Βενιζέλος). Ανάδοχοι του έργου μπορεί να είναι, είτε αποκλειστικά δημόσιοι φορείς είτε αποκλειστικά ιδιωτικοί φορείς, είτε συνεργασία μεταξύ τους. Ανάδοχος μπορεί να είναι ένας ή περισσότεροι (κοινοπραξία).

1.2.2 Πόροι Έργου

Η διεξαγωγή ενός έργου συνδέεται άμεσα με την έννοια των **Πόρων (Resources)**. Σε κάθε έργο εμπλέκονται περισσότερο ή λιγότερο ανάλογα με το αντικείμενο του έργου:

1. **Ανθρώπινοι πόροι (Human resources)**.
2. **Εξοπλισμός (Equipment)**, δηλαδή οι μηχανές και τα εργαλεία.
3. **Αναλώσιμοι πόροι (Consumable Resources)**, δηλαδή κάθε τι που καταναλώνεται στο έργο και που χρεώνεται άμεσα στο έργο. (πρώτες ύλες, τροφή, ένδυση εργαζομένων κτλ). Ιδιαίτερο ρόλο και σημασία παίζουν τα υλικά, η αλλιώς υλικοί πόροι (*materials*).

1.2.3 Υπηρεσίες Έργου

Οι **υπηρεσίες (services)**, καλύπτουν στο έργο απαιτήσεις, που δεν καλύπτουν οι πόροι του. Το συνολικό κόστος που προκύπτει από την χρήση των πόρων και των υπηρεσιών του έργου αποτελούν το **κόστος του έργου (project cost)** ενώ τα απαιτούμενα κεφάλαια για την υλοποίηση του κόστους του έργου αποτελούν τους **οικονομικούς πόρους (financial resources)**. Η εκ των προτέρων κατά προσέγγιση εκτίμηση του κόστους του έργου και η λογιστική του ανάλυση αποτελεί τον **προϋπολογισμό (budget)** του έργου.

1.2.4 Παράγοντες που Επιδρούν στο Έργο⁴

Ένα από τα συνηθέστερα προβλήματα των έργων είναι οι έντονες αντιπαραθέσεις που δημιουργούνται, είτε το έργο είναι μικρό, ιδιωτικό, είτε μεγάλο. Αυτό θεωρείται απόλυτα φυσικό, δεδομένου ότι με το έργο και το παραγόμενο προϊόν του, δημιουργούνται νέες καταστάσεις με πολυσύνθετες συχνά συνέπειες.

Οι βασικότεροι παράγοντες που, σηματοδοτούν το έργο, τόσο ως προσπάθεια, όσο και ως παραγόμενο προϊόν, είναι οι εξής:

⁴ Πολύζος Σ., «Διοίκηση και Διαχείριση των Έργων – Μέθοδοι και Τεχνικές» Εκδόσεις Κριτική, 1η έκδοση, 2004.

1. **Οικονομικός παράγων**, που αφορά το ύψος του κόστους του έργου. Ποιός θα πληρώσει το έργο, ποιός θα εισπράξει την ωφέλεια του, πως κοστολογήθηκε το έργο κτλ..
2. **Χρονικός παράγων**, που αφορά την χρονική διάρκειά του έργου. Χρόνος και κόστος είναι 2 παράγοντες στενά συνδεδεμένοι. Ο χρόνος είναι συνάρτηση του κόστους και το κόστος συνάρτηση της ποιότητας. Χρόνος – κόστος – ποιότητα , αποτελούν ένα από τα μεγαλύτερα αγκάθια του έργου.
3. **Ποιοτικός παράγων**, που χαρακτηρίζει την ποιότητα του έργου.
4. **Εργασιακός παράγων**, που αποτελεί την ραχοκοκαλιά του έργου.
5. **Κοινωνικός παράγων**, που επηρεάζεται από τους εκάστοτε νόμους, κανόνες, αντιλήψεις, γεωφυσικές συνθήκες κτλ., του εκάστοτε μέρους που λαμβάνει χώρα το έργο.
6. **Θεσμικός και νομικός παράγων**, που εκφράζεται από τους κανονισμούς και νόμους του κράτους και έχουν στενή επαφή με τα πρότυπα του έργου.

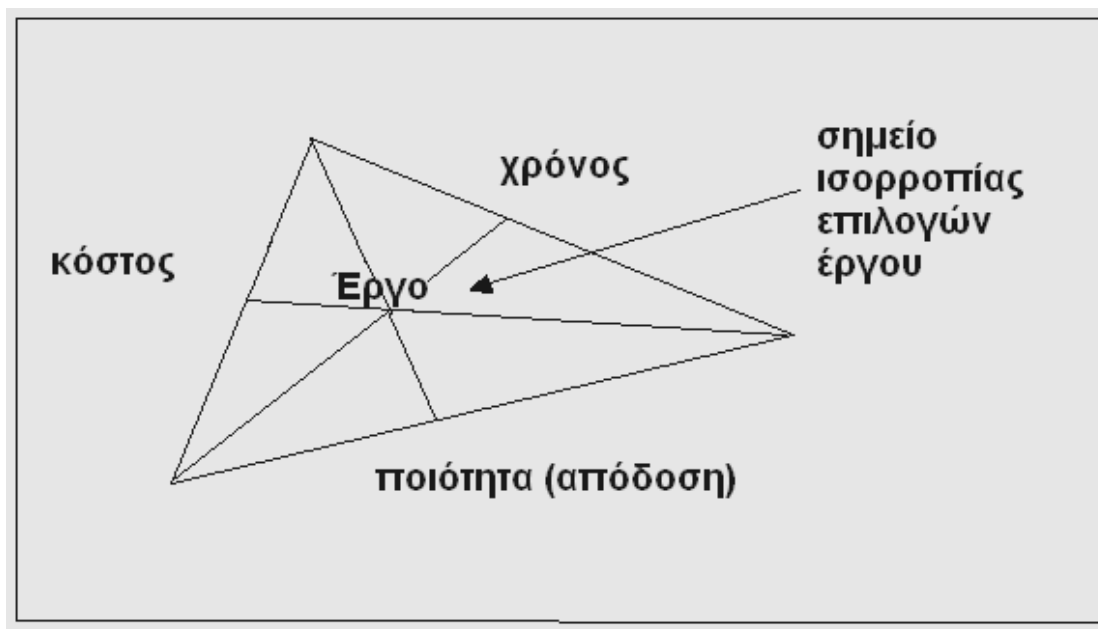
1.2.5 Βασικό Τρίγωνο

Ο δείκτης για το αν το έργο είχε θετικό η αρνητικό αποτέλεσμα αποτελείται από τρία βασικά χαρακτηριστικά και είναι τα εξής:

1. **Χρόνος (time)**. Η τήρηση των χρονικών δεσμεύσεων.
2. **Κόστος (cost)**. Η τήρηση του προϋπολογισμού του έργου.
3. **Ποιότητα (quality)**. Η τήρηση των ποιοτικών προδιαγραφών του έργου.

Αυτοί οι 3 δείκτες είναι συχνά ανταγωνιστικοί. Αυτό σημαίνει ότι διότι η κατά γράμμα τήρηση των χρονοδιαγραμμάτων μπορεί να οδηγήσει σε κοστολογική υπέρβαση ή σε εκπτώσεις στην απόδοση ή η κατά γράμμα τήρηση του προϋπολογισμού μπορεί να οδηγήσει σε μείωση της απόδοσης και σε χρονικές υπερβάσεις παράλληλα. Τέλος η κατά γράμμα τήρηση των ποιοτικών προδιαγραφών μπορεί με την σειρά της να οδηγήσει σε κοστολογική ή και χρονική υπέρβαση.

Σύμφωνα με ορισμένους Project Managers σ' αυτά τα 3 χαρακτηριστικά θα πρέπει να προστεθεί κι άλλο ένα τέταρτο χαρακτηριστικό, το **αντικείμενο (scope)** του έργου.



Εικόνα 0.3: Σχηματική αναπαράσταση της σχέσης «Κόστος – Χρόνος – Ποιότητα».⁵

Αυτό υποχρεώνει τους εμπλεκόμενους στο έργο να καθορίσουν από την αρχή του έργου μια γενική στρατηγική επιλογή, που αφορά αυτούς τους δείκτες. Σε γενικές γραμμές εμφανίζονται οι ακόλουθες περιπτώσεις:

- Καθορισμός ιεραρχικής προτεραιότητας καθενός από τους 3 παράγοντες
Καθορίζεται από την αρχή του έργου η ιεραρχική προτεραιότητα κάθε ενός από τους 3 παράγοντες (πχ κυρίαρχος παράγοντας το κόστος, στη συνέχεια η ποιότητα και τελικά ο χρόνος). Έτσι ο δεύτερος και ο τρίτος σε σημασία παράγοντας θα πρέπει να τροποποιούνται ανάλογα με τις απαιτήσεις του πρώτου σε σημασία παράγοντα, με τον καλύτερο δυνατό τρόπο. Για παράδειγμα σ' ένα έργο που θα πρέπει να είναι έτοιμο σε δεδομένη και μη καθυστερήσιμη ημερομηνία, η ποιότητα και το κόστος θα πρέπει να υποταχθούν στο χρόνο διαφορετικά το έργο δεν έχει νόημα.

⁵ Δημητριάδης Α., «Διοίκηση – Διαχείριση Έργου», Εκδόσεις Νέων Τεχνολογιών, Έκδοση 4^η, 2009.

- Εύρεση χρυσής τομής της αλληλοεπίδρασης των τριών αυτών παραγόντων
Επιδιώκεται η εύρεση της χρυσής τομής της αλληλοεπίδρασης των τριών αυτών παραγόντων. Στην περίπτωση αυτή οι γενικότερες καταστάσεις καθορίζουν ποια σχέση ισορροπίας μεταξύ αυτών των παραγόντων κρίνεται κάθε φορά ως καλύτερη. Με άλλα λόγια στόχος είναι η εκτέλεση του έργου κατά το δυνατόν:
 - Πλησιέστερα στις προθεσμίες.
 - Πλησιέστερα στον προϋπολογισμό.
 - Πλησιέστερα στις ποιοτικές προδιαγραφές.

1.3 Οργανωτικές δομές⁶

Κατά την εκτέλεση των έργων (*projects*) συναντάει κανείς τρεις κύριες οργανωτικές δομές σύμφωνα με τις οποίες οι εταιρείες διεκπεραιώνουν το έργο. Βασική απόφαση της διοίκησης πριν αρχίσει το *project* είναι να αποφασίσει για την δομή που θα διαμορφωθεί για την εκτέλεσή του. Αυτές είναι οι οργανωτικές δομές “*Pure project*”, “*Functional project*” και “*Matrix project*”.

Πάντως όποια δομή και αν επιλεγεί, η σημασία του *project manager* (υπεύθυνος έργου αλλά με την σημασία και του υπεύθυνου για τις επιμέρους δράσεις) είναι μεγάλη για την επαφή με τον «πελάτη». Η επικοινωνία και η ευελιξία είναι σε υψηλό επίπεδο από την στιγμή που ένα άτομο είναι υπεύθυνο για την επιτυχή ολοκλήρωση του έργου.

⁶ «Προγραμματισμός έργων», Διοίκηση Παραγωγής και Συστημάτων Υπηρεσιών, Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης, Εθνικό Μετσόβειο Πολυτεχνείο.

1.3.1 Οργανωτική δομή «Pure Project»

Είναι μια κλασσική ιεραρχική δομή. Υπάρχει ένας κύριος υπεύθυνος όλου του έργου στον οποίο αναφέρονται άμεσα όλοι όσοι εμπλέκονται στο έργο και που τις περισσότερες φορές αποτελούν μια αυτοτελή ομάδα πλήρως απασχολούμενη σ' αυτό.

Αναλυτικότερα τα πλεονεκτήματα και μειονεκτήματα της συγκεκριμένης οργανωτικής δομής δίνονται στην συνέχεια.

Πλεονεκτήματα

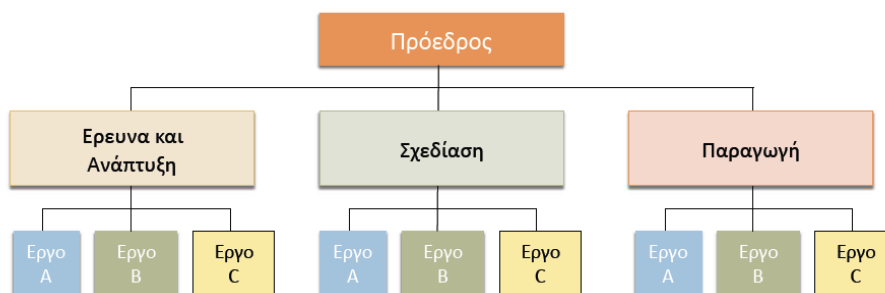
1. Υπάρχει ένας υπεύθυνος με πλήρη έλεγχο όλου του έργου.
2. Τα μέλη της ομάδας έργου αναφέρονται σε ένα υπεύθυνο.
3. Σύντομες «γραμμές» επικοινωνίας.
4. Η δέσμευση και η κινητοποίηση των μελών είναι μεγάλη.

Μειονεκτήματα

1. Πολλαπλή χρήση πόρων.
2. Οι ευρύτεροι οργανωτικοί στόχοι και πολιτικές αγνοούνται.
3. Περιορισμένη «μεταφορά εμπειρίας».
4. Τα μέλη της ομάδας ανησυχούν για την αντιμετώπισή τους μετά το πέρας του έργου.

1.3.2 Οργανωτική δομή «Functional Project»

Στην περίπτωση της δομής αυτής (που θα μπορούσε να ονομαστεί και λειτουργική οργάνωση) υπάρχει η προσαρμογή του έργου σε μια περισσότερο λειτουργική διαίρεση. Διαχωρίζονται δηλαδή οι κύριες λειτουργίες του έργου και με κάθε μία από αυτές συνδέονται συγκεκριμένα τμήματα του έργου. Στην παρακάτω εικόνα δίνεται σχηματικά μια τέτοια δομή.



Εικόνα 0.4: Οργανωτική δομή «Functional Project».⁷

Πλεονεκτήματα

1. Ένα μέλος μπορεί να απασχοληθεί σε πολλά έργα.
2. Η τεχνική γνώση και εμπειρία «μένει» στην ομάδα ακόμη και μετά την αποχώρηση κάποιων μελών.
3. Οι συμμετέχοντες παραμένουν στη «φυσική» τους θέση μετά την περάτωση του έργου.
4. Δημιουργείται «κρίσιμη μάζα» ειδικών για διάφορα πεδία εφαρμογών.
5. Υπάρχει κάθετη επικοινωνία.

Μειονεκτήματα

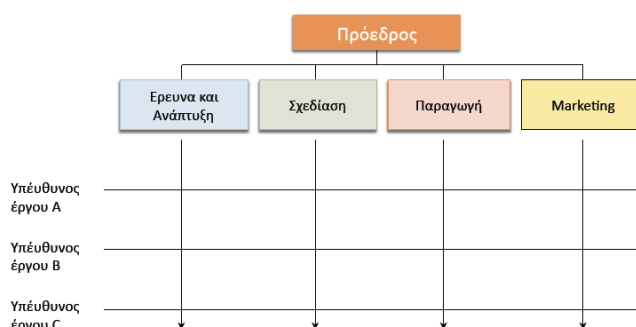
1. Περιορισμένος ενθουσιασμός και δέσμευση των μελών.
2. Πτυχές του έργου, μη σχετικές με τη λειτουργία του τμήματος, δεν αντιμετωπίζονται με ιδιαίτερη προσοχή.
3. Οι ανάγκες του πελάτη θεωρούνται δευτερεύουσες ή ικανοποιούνται με σημαντική καθυστέρηση.

1.3.3 Οργανωτική δομή «Matrix Project»

Στην δομή αυτή διαφορετικά έργα ή δράσεις του ίδιου έργου (οι οριζόντιοι είσοδοι στον πίνακα), χρησιμοποιούν πόρους του συστήματος από ξεχωριστές λειτουργικές

⁷ «Προγραμματισμός έργων», Διοίκηση Παραγωγής και Συστημάτων Υπηρεσιών, Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης, Εθνικό Μετσόβειο Πολυτεχνείο.

περιοχές μεταξύ τους (οι κάθετοι είσοδοι στον πίνακα). Έτσι εδώ υπάρχει μια ώσμωση από ιδιότητες των δομών «Pure project» και «Functional project».



Εικόνα 0.5: Οργανωτική δομή «Matrix Project».⁸

Σε κάθε έργο λοιπόν είναι φανερό ότι χρησιμοποιούνται άνθρωποι από διαφορετικές λειτουργικές περιοχές του συστήματος. Ο υπεύθυνος έργου (*Project manager*) ο οποίος μπορεί να είναι διαφορετικός κάθε φορά, αποφασίζει ποιες δράσεις ή επιμέρους δραστηριότητες πρέπει να γίνουν στα πλαίσια του έργου και πότε. Από την άλλη ο υπεύθυνος για κάθε μια από τις λειτουργικές περιοχές (*Functional manager*) παρέχει το προσωπικό και τον εξοπλισμό που θα χρησιμοποιηθούν στο έργο.

Πλεονεκτήματα

1. Αυξημένη επικοινωνία μεταξύ τμημάτων.
2. Ένας υπεύθυνος αποφασίζει για την πορεία του έργου.
3. Η πολλαπλή χρήση πόρων ελαχιστοποιείται.
4. Οι συμμετέχοντες παραμένουν στη «φυσική» τους θέση μετά την περάτωση του έργου.
5. Οι πολιτικές του οργανισμού τηρούνται εντός του έργου.

Μειονεκτήματα

1. Ύπαρξη δύο αφεντικών (Υπεύθυνος Έργου / Υπεύθυνος Τμήματος).

⁸ «Προγραμματισμός έργων», Διοίκηση Παραγωγής και Συστημάτων Υπηρεσιών, Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης, Εθνικό Μετσόβειο Πολυτεχνείο.

2. Η επιτυχία εξαρτάται σε μεγάλο βαθμό από τα επικοινωνιακά χαρίσματα του.
3. Υπάρχει κίνδυνος μειωμένης απόδοσης της ομάδας καθώς κάθε υπεύθυνος έργου επιδιώκει για λογαριασμό του μεγαλύτερη απασχόληση πόρων.

ΚΕΦΑΛΑΙΟ 2. ΤΕΧΝΙΚΕΣ ΥΛΟΠΟΙΗΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

2.1 Ευέλικτη Ανάπτυξη Λογισμικού

Η δυσαρέσκεια για τις επιβαρυμένες μεθόδους ανάπτυξης λογισμικού οδήγησε, ένα μεγάλο μέρος μηχανικών λογισμικού να προτείνουν νέες, πιο ευέλικτες μεθόδους. Αυτές οι μέθοδοι έδιναν την δυνατότητα στην ομάδα που ήταν υπεύθυνη για την ανάπτυξη να εστιάζει την προσοχή της στο λογισμικό παρά στον σχεδιασμό του και την τεκμηρίωσή του.

Οι ευέλικτες μέθοδοι (*agile methods*) θεμελιώνονται εξολοκλήρου σε μία προσέγγιση όπου:

1. επαναλαμβάνονται οι διαδικασίες του ορισμού προδιαγραφών του λογισμικού, της ανάπτυξης και της παράδοσής του, που έχουν αρχικά οριστεί για την υποστήριξη της ανάπτυξης επιχειρηματικού λογισμικού
2. οι απαιτήσεις του συστήματος αλλάζουν συχνά κατά τη διάρκεια της ανάπτυξης.

Ο σκοπός τους είναι να παραδώσουν γρήγορα λειτουργικό λογισμικό στους πελάτες, οι οποίοι έπειτα μπορούν να ζητήσουν νέες ή αλλαγμένες λειτουργίες που θα ενσωματωθούν στις επόμενες εκδόσεις του συστήματος.

Στις 11 έως 13 Φεβρουαρίου 2001, σε ένα χειμερινό θέρετρο της οροσειράς Wasatch της Utah συγκεντρώθηκαν εκπρόσωποι διάφορων νέων μεθοδολογιών ώστε να συζητήσουν την ανάγκη νέων ελαφρότερων μεθοδολογιών ως εναλλακτικές των παραδοσιακών επιβαρυμένων. Οι συζητήσεις κατέληξαν σε ένα μανιφέστο βασικών αρχών οι οποίες αποτέλεσαν τα θεμέλια της ευέλικτης ανάπτυξης λογισμικού.⁹

⁹ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

Ενδεχομένως η πιο γνωστή ευέλικτη μέθοδος είναι ο «ακραίος προγραμματισμός» (*extreme programming*) ο οποίος περιγράφεται στην συνέχεια. Ωστόσο, υπάρχουν και άλλες μέθοδοι όπως: η *Scrum*, η *Crystal*, η *DSDM*, η *Feature Driven Development* και η *Lean Development*.

Η ευόδωση αυτών των μεθόδων είχε ως αποτέλεσμα την ένταξή τους σε πιο παραδοσιακές μεθόδους ανάπτυξης λογισμικού που έχουν τις βάσεις τους στην μοντελοποίηση του συστήματος.

Μολονότι αυτές οι μέθοδοι στηρίζονται στην ιδέα της αυξανόμενης ανάπτυξης και παράδοσης λογισμικού, προσεγγίζουν τον στόχο τους μέσω διαφορετικών διεργασιών. Παρόλα αυτά, διέπονται από ένα κοινό σύνολο αρχών και κανόνων και αυτός είναι ο λόγος για τον οποίο οι μέθοδοι αυτοί έχουν πολλά κοινά σημεία. Αυτές οι αρχές φαίνονται σε παρακάτω πίνακα:

Πίνακας 0.1: Χαρακτηριστικά μεθοδολογίας Agile Software Development.¹⁰

Αρχή	Περιγραφή
<u>Ανάμειξη του πελάτη</u>	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.
<u>Αυξανόμενη παράδοση</u>	Το λογισμικό αναπτύσσεται σε μέρη και οι πελάτες ορίζουν τις απαιτήσεις που θα περιληφθούν σε κάθε μέρος.
<u>Άνθρωποι, όχι διεργασίες</u>	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.
<u>Αποδοχή αλλαγών</u>	Θα πρέπει να αναμένονται αλλαγές στις απαιτήσεις του συστήματος, για αυτό το λόγο ο σύστημα πρέπει να σχεδιαστεί ώστε να δέχεται τέτοιες αλλαγές.
<u>Διατήρηση απλότητας</u>	Θα πρέπει να γίνεται εστίαση στην απλότητα τόσο του λογισμικού που αναπτύσσεται όσο και της διαδικασίας

¹⁰ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

της ανάπτυξης. Όποτε είναι δυνατό, εκτελούνται εργασίες για την εξάλειψη πολυπλοκότητας από το σύστημα.

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

Παρά το γεγονός ότι η ιδέα της συμμετοχής του πελάτη στην διαδικασία ανάπτυξης του λογισμικού είναι ελκυστική, η επίτευξη της εξαρτάται από δύο παράγοντες: την επιθυμία και την δυνατότητα του πελάτη να διαθέσει χρόνο στην ομάδα ανάπτυξης. Τις περισσότερες φορές παρατηρείται ότι οι πελάτες έχουν άλλες υποχρεώσεις και δεν μπορούν να έχουν πλήρη συμμετοχή στην ανάπτυξη του λογισμικού.

Επίσης, κάποια από τα μέλη μίας ομάδας, μπορεί να μην έχουν τον κατάλληλο χαρακτήρα και προσωπικότητα για συμμετέχουν εντατικά σε αυτήν, κάτι που απαιτείται κατά κόρον στις ευέλικτες μεθόδους. Για αυτό το λόγο, μπορεί να μην συνεργάζονται σωστά με τα άλλα μέλη της ομάδας.

Εκτός αυτού, το να θέτεις προτεραιότητες στις αλλαγές του συστήματος μπορεί να γίνει δύσκολο, ειδικά όταν αναφερόμαστε σε συστήματα όπου υπάρχουν περισσότερα του ενός άτομα που θέτουν τις προτεραιότητες για τις αλλαγές του συστήματος. Συχνά κάθε άτομο θέτει διαφορετικές προτεραιότητες σε διαφορετικές αλλαγές.

Δεν θα πρέπει να ξεχνάμε ότι η διατήρηση της απλότητας απαιτεί περισσότερη εργασία και ότι υπό την πίεση των προθεσμιών παράδοσης του συστήματος, η ομάδα μπορεί να μην έχει το χρόνο να κάνει τις επιθυμητές απλοποιήσεις.

Ένα άλλο πρόβλημα, μη τεχνικό, το οποίο είναι γενικό πρόβλημα των μεθόδων της αυξανόμενης ανάπτυξης και παράδοσης, προκύπτει όταν ο πελάτης χρησιμοποιεί έναν εξωτερικό οργανισμό για την ανάπτυξη του συστήματος. Το κείμενο των προδιαγραφών του συστήματος συνήθως αποτελεί μέρος του συμβολαίου μεταξύ του πελάτη και του οργανισμού που το αναπτύσσει. Λόγω του ότι ο αυξανόμενος ορισμός των απαιτήσεων αποτελεί μέρος των ευέλικτων

μεθόδων, η συγγραφή συμβολαίων για αυτό τον τύπο ανάπτυξης μπορεί να είναι δύσκολη.¹¹

Συνεπώς οι ευέλικτες μέθοδοι βασίζονται στα συμβόλαια, στα οποία ο πελάτης πληρώνει για τον 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 τον δημιουργό του συστήματος. Ωστόσο αν προκύψουν προβλήματα, θα προκύψουν διαφωνίες που δύσκολα επιλύονται, όσον αφορά το ποιος ευθύνεται και ποιος πρέπει να πληρώσει για τον επιπλέον χρόνο και τους πόρους που χρειάστηκαν για να επιλυθούν τα προβλήματα.

2.1.1 Ακραίος προγραμματισμός (Extreme programming, XP)

Ο ακραίος προγραμματισμός (Extreme programming, XP) είναι ίσως η πιο γνωστή και η πιο διαδεδομένη από τις ευέλικτες μεθόδους. Πήρε την ονομασία του από τον Beck γιατί αυτή η μέθοδος δημιουργήθηκε χρησιμοποιώντας καλές και αναγνωρισμένες πρακτικές, όπως η επαναλαμβανόμενη ανάπτυξη, και η συμμετοχή του πελάτη στην ανάπτυξη του λογισμικού σε υπερβολικό (extreme) βαθμό.

Στην μέθοδο XP, όλες οι απαιτήσεις αντιπροσωπεύονται από σενάρια, τα οποία υλοποιούνται άμεσα ως μία αλληλουχία εργασιών. Οι προγραμματιστές εργάζονται σε μικρές ομάδες και δοκιμάζουν κάθε εργασία πριν γράψουν τον κώδικα. Πριν ενσωματωθεί ο κώδικας τους στο σύστημα όλες οι δοκιμές πρέπει να ολοκληρωθούν με επιτυχία. Το χρονικό διάστημα που μεσολαβεί μεταξύ των διαφορετικών εκδόσεων του συστήματος είναι πολύ μικρό. Η μέθοδος XP περιλαμβάνει ένα αριθμό πρακτικών που αντιστοιχούν στις αρχές των ευέλικτων μεθόδων ανάπτυξης:¹²

¹¹ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

¹² Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

- 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.

Η μέθοδος XP αποτελεί μια ακραία προσέγγιση της επαναλαμβανόμενης ανάπτυξης. Νέες εκδόσεις του συστήματος μπορεί να υλοποιούνται πολλές φορές καθημερινά και πρόσθετες λειτουργίες του λογισμικού 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. Με άλλα λόγια, ότι θα πρέπει να αναμένεις

μελλοντικές αλλαγές στο λογισμικό και για αυτό το λόγο θα πρέπει να σχεδιάζεις το λογισμικό έτσι ώστε να είναι ευέλικτο και οι αλλαγές αυτές να πραγματοποιούνται εύκολα.

Ωστόσο, στην 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 0.2: Χαρακτηριστικά της τεχνικής XP.¹³

Αρχή	Περιγραφή
Αυξανόμενος προγραμματισμός	Οι απαιτήσεις καταγράφονται σε Story cards και αποφασίζεται σε ποια έκδοση θα συμπεριληφθούν βάση 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 για κάθε νέο κομμάτι λειτουργικότητας, πριν αυτή η λειτουργικότητα υλοποιηθεί.
Αλλαγή κώδικα	Όλοι οι συγγραφείς κώδικα αναμένεται να αλλάζουν συνεχώς μέρη του κώδικα για να τον βελτιώσουν όσο

¹³ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

	μπορούν. Αυτό διατηρεί τον κώδικα απλό, και εύκολο προς συντήρηση.
Προγραμματισμός ανά ζεύγη	Οι προγραμματιστές δουλεύουν ανά ζεύγη, ελέγχοντας ο ένας την εργασία του άλλου και προσφέροντας βοήθεια για να πετύχουν το καλύτερο δυνατό αποτέλεσμα.
Συλλογική ιδιοκτησία	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.
Συνεχής συνένωση	Μόλις τελειώσει μια εργασία ενσωματώνεται σε όλο το σύστημα. Αφού ενσωματωθεί, πρέπει όλες οι δοκιμές μονάδων να είναι επιτυχείς.
Υποφερτός ρυθμός	Πολλές υπερωρίες δεν είναι αποδεκτές καθώς το φαινόμενο του δικτύου συχνά μειώνει την ποιότητα του κώδικα και την αποδοτικότητα.
Διαθέσιμος πελάτης	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.

Το πρόβλημα με το σχεδιασμό χωρίς να λαμβάνουμε υπόψη τις αλλαγές σε λειτουργικότητα που μπορεί να ζητηθούν, είναι ότι όταν απαιτούνται αλλαγές, υποβαθμίζουν την 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.

2.1.2 Μέθοδος Scrum (Scrum Method)

Η μέθοδος Scrum⁵ αναπτύχθηκε από τον Ken Schwaber. Είναι μία μέθοδος ανάπτυξης **επαναληπτική** (το προϊόν παράγεται κατά την διάρκεια μικρών κύκλων) και **αυξητική** (οι λειτουργίες του προϊόντος αυξάνονται προσθέτοντας νέα ποιοτικά χαρακτηριστικά και ιδιότητες). Το Scrum είναι μια προσέγγιση διοίκησης του προγράμματος που στηρίζεται στις αυτορρυθμιζόμενες ανεξάρτητες ομάδες ανάπτυξης για την εφαρμογή ενός προγράμματος λογισμικού στους κύκλους 30-ημέρων.

Στην μέθοδο αυτή υπάρχει μία ομάδα έργου που αποτελείται από τον ιδιοκτήτη του προϊόντος - ScrumMaster και την ομάδα - Scrumteam. Και οι δύο, έχουν διαφορετικά καθήκοντα να φέρουν εις πέρας. Ο Scrummaster δεν είναι ο αρχηγός της ομάδας. Ρόλος του είναι να απομακρύνει τα εμπόδια που σχετίζονται με την ικανότητα της ομάδας.

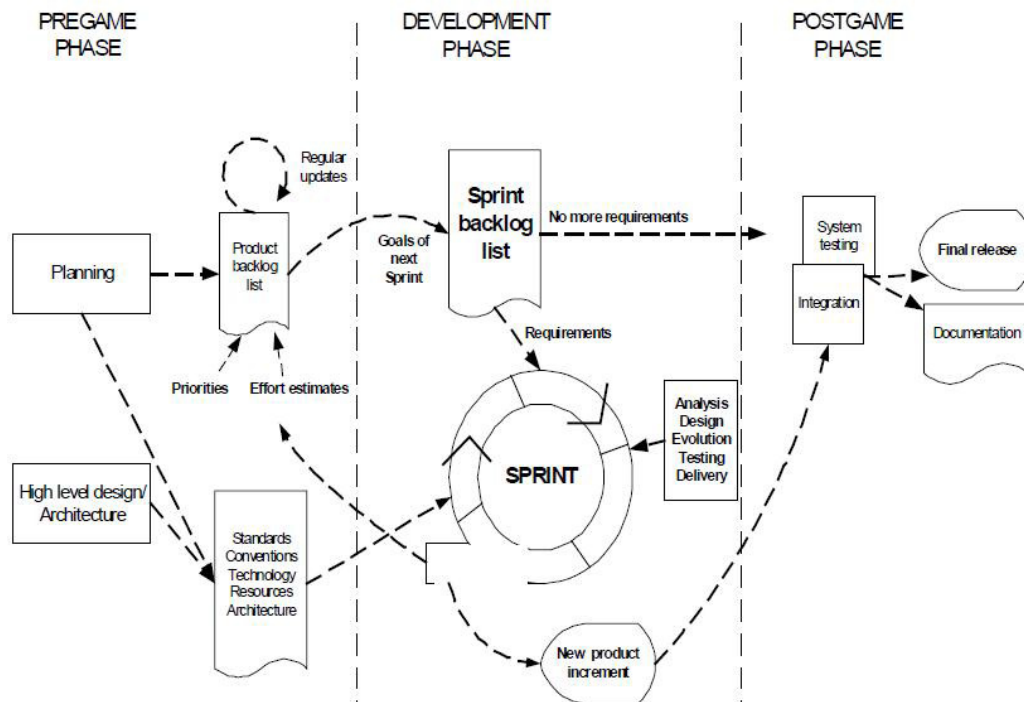
Το Scrum προτείνεται για οποιοδήποτε μέγεθος προγράμματος, αλλά η ομάδα του Scrum μπορεί να είναι μεταξύ πέντε και εννιά ατόμων. Για τα μεγάλα προγράμματα σχηματίζονται πολλαπλές ομάδες Scrum.

Η διαδικασία Scrum περιλαμβάνει τρεις φάσεις: πριν το παιχνίδι (Pre game), το παιχνίδι (game) και μετά το παιχνίδι (Post game).¹⁴

Η φάση πριν το παιχνίδι περιλαμβάνει δύο υπό-φάσεις:

1. Τον προγραμματισμό (Planning) και
2. Τον σχεδιασμό υψηλού επιπέδου/ της αρχιτεκτονικής (Architecture/High level design)

¹⁴ Schwaber K., and Beedle M., "Agile Software Development With Scrum", Upper Saddle River, NJ, Prentice-Hall, 2008.



Εικόνα 0.1: Διαδικασία Scrum.¹⁵

Ο προγραμματισμός περιλαμβάνει τον καθορισμό του συστήματος που θα αναπτυχθεί.

Ένας κατάλογος ανεκτέλεστών προϊόντων (*Product Backlog list*) της παραγγελίας δημιουργείται. Στον κατάλογο αυτό περιέχονται όλες τις απαιτήσεις που είναι γνωστές την παρούσα περίοδο. Οι απαιτήσεις αυτές μπορεί να προέρχονται από διαφορετικά μέρη: από τον πελάτη, από τις πωλήσεις και το τμήμα μάρκετινγκ, από την υποστήριξη πελατών ή από τους ίδιους τους υπεύθυνους για την ανάπτυξη λογισμικού.

Στην συνέχεια οι απαιτήσεις κατατάσσονται ανάλογα με την προτεραιότητά τους και υπολογίζεται η προσπάθεια που απαιτείται για την εφαρμογή τους. Ο κατάλογος ανεκτέλεστων προϊόντων είναι δυναμικός και ενημερώνεται διαρκώς με νέα και πιο λεπτομερή στοιχεία, καθώς επίσης με ακριβέστερες εκτιμήσεις για την αναγκαία προσπάθεια και με νέες προτεραιότητες.

¹⁵ Schwaber K., and Beedle M., "Agile Software Development With Scrum", Upper Saddle River, NJ, Prentice-Hall, 2008.

Ο προγραμματισμός εκτός των άλλων περιλαμβάνει επίσης:

- τον καθορισμό της ομάδας προγράμματος,
- τον καθορισμό των εργαλείων και άλλων πόρων,
- την αξιολόγηση του κινδύνου,
- τις ανάγκες εκπαίδευσης της ομάδας και των χρηστών,
- και την διοικητική έγκριση.

Σε κάθε επανάληψη, ο ενημερωμένος κατάλογος ανεκτέλεστης παραγγελίας αναθεωρείται από τη ομάδα Scrum έτσι ώστε να αποκτηθεί η δέσμευσή τους για την επόμενη επανάληψη.

Στη φάση αρχιτεκτονικής και του σχεδιασμού υψηλού επιπέδου η αρχιτεκτονική προγραμματίζεται βασισμένη στα τρέχοντα στοιχεία που βρίσκονται στον κατάλογο ανεκτέλεστων εργασιών. Σε περίπτωση βελτίωσης ενός υπάρχοντος συστήματος, οι αλλαγές που απαιτούνται για την εφαρμογή των στοιχείων του καταλόγου προσδιορίζονται μαζί με τα προβλήματα που μπορούν να προκαλέσουν. Μια συνεδρίαση για την αναθεώρηση του σχεδίου πραγματοποιείται για να ελέγξει τις προτάσεις για την εφαρμογή και να λάβει αποφάσεις. Επιπλέον, τα προκαταρκτικά σχέδια για το περιεχόμενο τις επικείμενης νέας έκδοσης προετοιμάζονται.

Η φάση παιχνιδιού (επίσης αποκαλούμενη η φάση ανάπτυξης) αποτελεί το ευέλικτο μέρος της προσέγγισης Scrum. Αυτή η φάση αντιμετωπίζεται ως «μαύρο κουτί» όπου όλα τα απρόβλεπτα αναμένονται. Οι διαφορετικές περιβαλλοντικές και τεχνικές μεταβλητές (όπως το χρονικό πλαίσιο, η ποιότητα, οι απαιτήσεις, οι πόροι, οι τεχνολογίες και τα εργαλεία εφαρμογής που προσδιορίζονται στο Scrum και που μπορεί να αλλάξουν κατά τη διάρκεια της διαδικασίας), παρατηρούνται και ελέγχονται μέσω των διάφορων πρακτικών κατά τη διάρκεια των ορμών (Sprints) της φάσης ανάπτυξης. Η μέθοδος Scrum αντί να λάβει υπόψη αυτά τα θέματα μόνο στην αρχή της ανάπτυξης του έργου, στοχεύει στο να ελέγχει τα παραπάνω, για να είναι σε θέση συνεχώς να προσαρμόζεται στις αλλαγές.

Στη φάση ανάπτυξης το σύστημα αναπτύσσεται **σε ορμές (Sprints)** δηλαδή σε επαναληπτικούς κύκλους όπου η λειτουργία αναπτύσσεται ή εμπλουτίζεται για να

παραγάγει μικρά νέα κομμάτια (προσαυξήσεις). Κάθε ορμή περιλαμβάνει τις παραδοσιακές φάσεις ανάπτυξης λογισμικού:

1. προδιαγραφές,
2. ανάλυση,
3. σχέδιο,
4. εξέλιξη
5. και παράδοση.

Η αρχιτεκτονική και το σχέδιο του συστήματος εξελίσσονται κατά τη διάρκεια του σπριντ. Ένα σπριντ προγραμματίζεται για να διαρκέσει από μια εβδομάδα έως έναν μήνα. Μπορεί να υπάρξουν, παραδείγματος χάριν, τρία έως οκτώ σπριντ σε μια διαδικασία ανάπτυξης προτού το σύστημα να είσαι έτοιμο για τη διανομή. Επίσης εάν το έργο είναι μεγάλο και πολύπλοκο, μπορεί να υπάρξουν περισσότερες από μία ομάδες που δουλεύουν παράλληλα για την ολοκλήρωση της ίδιας προσαύξησης στο σύστημα .

Η **φάση μετά το παιχνίδι** περιέχει την ολοκλήρωση της νέας έκδοσης. Σε αυτή τη φάση εισερχόμαστε όταν συμφωνηθεί ότι οι μεταβλητές του περιβάλλοντος καθώς και οι απαιτήσεις έχουν ολοκληρωθεί. Σε αυτήν την περίπτωση, δεν μπορούν να βρεθούν άλλα στοιχεία και ζητήματα ούτε κάποια νέα να εισαχθούν. Το σύστημα είναι τώρα έτοιμο για την έκδοση του και η προετοιμασία για αυτό γίνεται κατά τη διάρκεια της φάσης «μετά το παιχνίδι», συμπεριλαμβανομένων των εργασιών όπως η προσαρμογή της προσαύξησης στο υπάρχον σύστημα (*integration*), η δοκιμή του συστήματος (*system testing*) και η τεκμηρίωση (*documentation*).

2.1.3 Μεθοδολογίες κρυστάλλου (Crystal Methods)

Η μέθοδος αυτή αναπτύχθηκε από τον Alistair Cockburn. Οι κύριοι άξονες στους οποίους βασίστηκε η εξέλιξη αυτής της μεθόδου είναι οι άνθρωποι, οι μεταξύ τους αλληλεπιδράσεις, οι ικανότητες, το ταλέντο και η επικοινωνία. Η μέθοδος

υποστηρίζει ότι τα στοιχεία που προαναφέρθηκαν είναι ο παράγοντες που έχουν την κύρια επίδραση στην επίδοση των έργων.

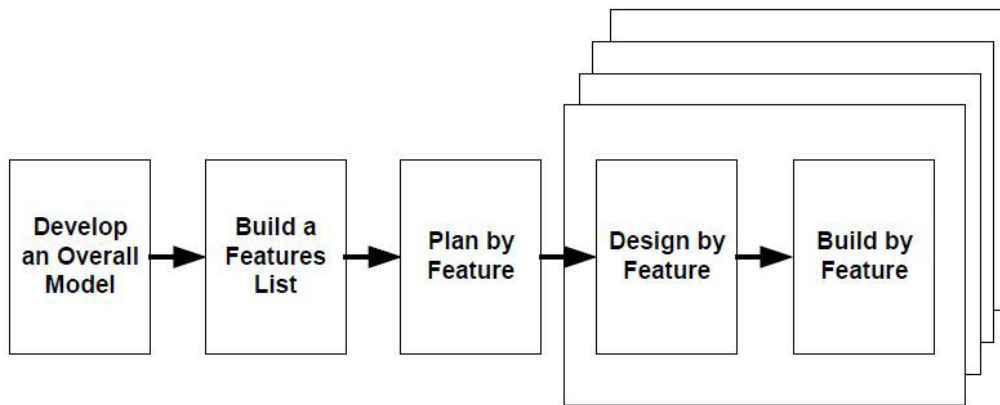
Η φιλοσοφία της μεθοδολογίας είναι ότι κάθε ομάδα έχει συγκεκριμένες δυνατότητες. Κατά συνέπεια, πρέπει να χειρίζεται διεργασίες που προορίζονται μόνο για αυτήν. Η κάθε διαδικασία λοιπόν, χωρίζεται σε εξειδικευμένες υποδιεργασίες έτσι ώστε κάθε ομάδα να μπορεί να ανταπεξέλθει στις ανάγκες μίας συγκεκριμένης διεργασίας. Η λέξη «crystal» αναφέρεται στις πολλές διαφορετικές πλευρές που έχει μία αποστολή-διεργασία.

Η οικογένεια των «Κρυστάλλινων» μεθοδολογιών περιλαμβάνει έναν αριθμό από διαφορετικές μεθοδολογίες για να επιλέγεται κάθε φορά η καταλληλότερη μεθοδολογία για κάθε πρόγραμμα. Εκτός από τις μεθοδολογίες, η προσέγγιση του «Κρυστάλλου» περιλαμβάνει επίσης τις αρχές για να τροποποιηθεί η μεθοδολογία τις ποικίλες περιστάσεις των διαφορετικών προγραμμάτων.

2.1.4 Ανάπτυξη με βάση τα χαρακτηριστικά (Feature Driven Development)

Η FDD αποτελείται από πέντε διαδοχικές διαδικασίες κατά τη διάρκεια των οποίων πραγματοποιούνται ο σχεδιασμός και η οικοδόμηση του συστήματος (Εικόνα 2.2). Η ευέλικτη ανάπτυξη του συστήματος με γρήγορες προσαρμογές στις πρόσφατες αλλαγές στις απαιτήσεις και τις επιχειρησιακές ανάγκες υποστηρίζεται με το επαναληπτικό μέρος των διαδικασιών FDD (σχέδιο και κατασκευή). Χαρακτηριστικά, μια επανάληψη ενός χαρακτηριστικού γνωρίσματος περιλαμβάνει μια περίοδο μιας έως τριών εβδομάδας εργασίας για την ομάδα¹⁶.

¹⁶ Palmer S. R., and Felsing J. M., "A Practical Guide to Feature-Driven Development", Upper Saddle River, NJ, Prentice-Hall, 2002.



Εικόνα 0.2: Η διαδικασία του FFD.¹³

Τα κυριότερα χαρακτηριστικά της μεθόδου είναι ¹⁷:

1. Απαιτείται σύστημα για την κατασκευή άλλων συστημάτων.
2. Μια απλή αλλά καλά ορισμένη διαδικασία δουλεύει καλύτερα.
3. Τα βήματα των διαδικασιών πρέπει να είναι λογικά και η χρησιμότητά τους πρέπει να είναι φανερή στην ομάδα.
4. Οι διεργασίες μετακινούνται στο παρασκήνιο ώστε τα μέλη της ομάδας να εστιάζουν στα αποτελέσματα.
5. Οι κύκλοι ζωής πρέπει να είναι μικροί, επαναληπτικοί.

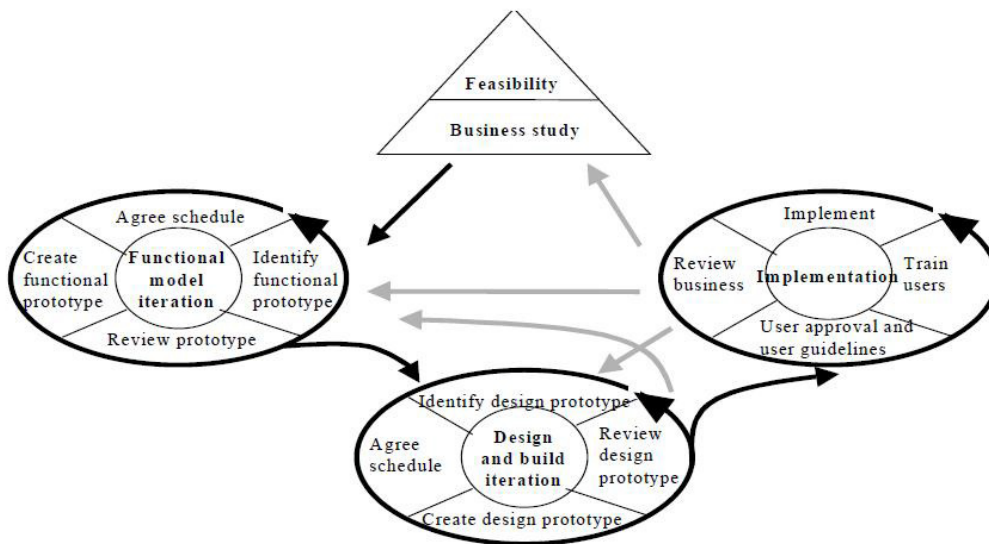
2.1.5 Μέθοδος Ανάπτυξης Δυναμικών Συστημάτων (Dynamic Systems Development Model Methodology)

Η μέθοδος αυτή, *Dynamic Systems Development Model Methodology*, αναπτύχθηκε στην Βρετανία, στα μέσα της δεκαετίας του '90 και αποτελεί την εξέλιξη της μεθόδου RAD. Η φιλοσοφία της μεθόδου είναι ότι τίποτα δεν

¹⁷ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

φτιάχνεται καλά από την πρώτη φορά. Οι εννέα αρχές της μεθόδου είναι οι ακόλουθες¹⁸:

1. Συνεχής συμμετοχή χρήστη.
2. Δυνατές ομάδες οι οποίες αναλαμβάνουν πρωτοβουλίες και παίρνουν αποφάσεις.
3. Εστίαση στη συχνή μεταφορά προϊόντων.
4. Επαναληπτική ανάπτυξη ώστε να επιβεβαιωθεί η σύγκλιση στις επιχειρησιακές λύσεις.
5. Αναστρέψιμες αλλαγές κατά τη διάρκεια ανάπτυξης.
6. Απαιτήσεις υψηλού επιπέδου.
7. Ολοκληρωμένος έλεγχος κατά τον κύκλο ζωής.
8. Συνεργασία μεταξύ όλων των εμπλεκομένων.



Εικόνα 0.3: Διάγραμμα διαδικασίας DSDM.¹⁹

¹⁸ Somerville I., «Βασικές Αρχές Τεχνολογίας Λογισμικού», Εκδόσεις Κλειδάριθμος, 2009.

¹⁹ Stapleton J., “Dynamic systems development method – The method in practice”, Addison Wesley, 1997.

Το DSDM όπως μπορούμε να παρατηρήσουμε και στην παραπάνω εικόνα αποτελείται από πέντε φάσεις:

1. τη μελέτη σκοπιμότητας (*feasibility study*),
2. την επιχειρησιακή μελέτη (*business study*),
3. την λειτουργική πρότυπη επανάληψη (*functional model iteration*),
4. το σχέδιο και τη δομή της επανάληψης (*design and build iteration*),
5. και την εφαρμογή (*implementation*).

Οι πρώτες δύο φάσεις είναι διαδοχικές, και εκτελούνται μόνο μια φορά. Οι τελευταίες τρεις φάσεις, κατά τη διάρκεια των οποίων πραγματοποιείται το πραγματικό έργο ανάπτυξης, έχουν δύο κύρια χαρακτηριστικά: είναι επαναληπτικές και επαυξητικές.

Το DSDM προσεγγίζει τις επαναλήψεις ως χρονικά παράθυρα. Ένα χρονικό παράθυρο διέπεται από δύο κανόνες:

1. διαρκεί για μια προκαθορισμένη χρονική περίοδο,
2. η επανάληψη πρέπει να τελειώσει μέσα αυτό.

Ο χρόνος που επιτρέπεται σε κάθε επανάληψη είναι καθορισμένος εκ των προτέρων. Το ίδιο συμβαίνει και για τα αποτελέσματα που η επανάληψη εγγυάται για τα προϊόντα. Στο DSDM, μια χαρακτηριστική διάρκεια των παραθύρων είναι από μερικές ημέρες σε μερικές εβδομάδες.

2.1.6 Lean Development

Η τεχνική LD εστιάζει στη δημιουργία λογισμικού με υψηλό βαθμό ανοχής σε αλλαγές. Στόχος είναι η κατασκευή λογισμικών με το 1/3 της ανθρώπινης προσπάθειας, των ωρών ανάπτυξης και επένδυσης που η κλασική μέθοδος θα επιτύγχανε. Μερικές από τις αρχές της είναι²⁰:

1. Η ικανοποίηση του πελάτη είναι η μεγαλύτερη προτεραιότητα.

²⁰ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

2. Η επιτυχία εξαρτάται από τη συμμετοχή του πελάτη.
3. Κάθε LD έργο είναι μία ομαδική προσπάθεια.
4. Όλα είναι ευμετάβλητα.
5. Ολοκλήρωση έργου και όχι καταστροφή.
6. Προτιμούμε ένα 80% λύσεων σήμερα παρά ένα 100% λύσεων αύριο.
7. Ο μινιμαλισμός είναι απαραίτητος.
8. Η ανάπτυξη προϊόντος είναι ανάπτυξη των χαρακτηριστικών του και όχι του μεγέθους του.
9. Ποτέ μη πιέζεις την LD μεθοδολογία πέρα από τα όριά της.

2.2 Μέθοδος ITIL (Information Technology Infrastructure Language)

Η μεθοδολογία *Information Technology Infrastructure Language*, συλλέγοντας τις καλύτερες τεχνικές στοχεύει στο να βελτιώσει και να συντηρήσει ένα ορισμένο επίπεδο στην ποιότητα υπηρεσιών πληροφορικής. Η μέθοδος καλύπτει τις IT απαιτήσεις ενός οργανισμού μέσω ενός συνόλου διεργασιών που οργανώνουν τις IT λειτουργίες του. Σχεδιάστηκε για το δημόσιο αλλά και για τον ιδιωτικό τομέα διεθνώς.

Περιγράφει πώς οι IT πόροι πρέπει να οργανωθούν ώστε να αποφέρουν επιχειρηματική επιτυχία. Μία ολοκληρωμένη ITIL μεθοδολογία χωρίζεται σε επτά κατηγορίες²¹:

1. Μεταφορά Υπηρεσιών (*Service Delivery*).
2. Υποστήριξη Υπηρεσιών (*Service Support*).

²¹ Sarantis, D., Charalabidis, Y., and Askounis, D. "A Goal-Driven Management Approach based on Knowledge Exploitation for e-Government Projects", *International Journal of Electronic Government Research*, Volume 6, Issue 3, July 2010.

3. Σχεδιασμός και Υλοποίηση Υπηρεσιών (*Planning and Implement service Management*).
4. Διοίκηση Ασφαλείας (*Security Management*).
5. Κατασκευή Τεχνολογιών Πληροφορικής και Επικοινωνιών (*Information and Communications Technologies- ITC*).
6. Οι προοπτικές της επιχείρησης και η εφαρμογή (*Business Perspective and Application Management*).

2.3 Κύκλος ζωής ανάπτυξης πληροφοριακών συστημάτων

Το πρώτο μοντέλο διαδικασίας ανάπτυξης λογισμικού (*Systems Development Life Cycle*) που δημοσιεύτηκε, προήλθε από γενικότερες πρακτικές της τεχνολογίας συστημάτων. Τα κύρια στάδια αυτού του μοντέλου αντιστοιχούν στις θεμελιώδεις δραστηριότητες ανάπτυξης²²:

1. *Ανάλυση και καθορισμός απαιτήσεων.* Οι υπηρεσίες, οι περιορισμοί και οι στόχοι του συστήματος προσδιορίζονται μέσω συζήτησης με τους χρήστες ως προδιαγραφές του συστήματος.
2. *Σχεδιασμός συστήματος και λογισμικού.* Η διαδικασία του σχεδιασμού του συστήματος χωρίζει τις απαιτήσεις σε συστήματα υλικού και λογισμικού. Αυτό καταλήγει σε μία συνολική αρχιτεκτονική του συστήματος. Ο σχεδιασμός λογισμικού περιλαμβάνει τον προσδιορισμό και την περιγραφή των θεμελιωδών αφηρημένων τμημάτων του συστήματος, καθώς και των σχέσεων μεταξύ τους.
3. *Υλοποίηση και δοκιμές υπομονάδων.* Κατά το στάδιο αυτό, ο σχεδιασμός λογισμικού υλοποιείται ως ένα σύνολο προγραμμάτων ή προγραμματιστικών υπομονάδων. Οι δοκιμές υπομονάδων συνίσταται

²² Somerville I., «Βασικές Αρχές Τεχνολογίας Λογισμικού», Εκδόσεις Κλειδάριθμος, 2009.

στην επαλήθευση ότι κάθε υπομονάδα ικανοποιεί τις προδιαγραφές της.

4. *Ενοποίηση και δοκιμές συστήματος.* Οι μεμονωμένες προγραμματιστικές υπομονάδες ή προγράμματα ενοποιούνται και δοκιμάζονται ως ένα ενιαίο σύστημα, για να διασφαλιστεί ότι οι απαιτήσεις λογισμικού έχουν εκπληρωθεί. Μετά τη δοκιμή, το σύστημα λογισμικού παραδίδεται στον πελάτη.
5. *Λειτουργία και συντήρηση.* Η φάση αυτή είναι μια μακροχρόνια φάση του κύκλου ζωής. Το σύστημα εγκαθίσταται και τίθεται σε χρήση. Η συντήρηση συνίσταται στη διόρθωση σφαλμάτων που δεν εντοπίστηκαν στα προηγούμενα στάδια του κύκλου ζωής.

2.3.1 Κοινή Ανάπτυξη Εφαρμογής (Joint Application Methodology)

Η μέθοδος αυτή (Joint Application Methodology) έχει σαν πρωτεύον στόχο να εμπλέξει τον πελάτη με τη σχεδίαση και ανάπτυξη της εφαρμογής μέσω μίας σειράς από συναντήσεις που ονομάζονται *Joint Sessions*.

Κύρια σημεία της μεθόδου JAD²³:

- Επιτρέπει στον συντονιστή, τους χρήστες και τους προγραμματιστές (10-20 πρόσωπα) να δουλέψουν μαζί προκειμένου να ορίσουν τις απαιτήσεις του συστήματος
- Μπορεί να μειώσει το «να βγούμε εκτός θέματος» κατά 50%
- Αποτρέπει τον ορισμό απαιτήσεων που είναι υπερβολικά συγκεκριμένες ή υπερβολικά γενικές (και ασαφείς)

Οι βασικοί ρόλοι είναι οι εξής:

²³ Τζίτζικας Γ., Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων, «Συλλογή Απαιτήσεων», Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών.

- *Facilitator* (μεσολαβητής): Ειδικευμένος στις τεχνικές ανάλυσης και σχεδίασης συστημάτων και στις διαδικασίες της ομάδας. Ο ίδιος καθοδηγεί τη συζήτηση αλλά δεν έχει το ρόλο του συμμετέχοντα. Επίσης ορίζει την ατζέντα, βοηθά την επικοινωνία της ομάδας με το να αποσαφηνίζει την τεχνική ορολογία παρατηρεί την ομάδα και βοηθά την επίλυση των ζητημάτων που προκύπτουν.
- *Scribes* (1 ή 2): Βοηθούν τον μεσολαβητή (*facilitator*), κρατούν τα πρακτικά, κάνουν αντίγραφα, κλπ.

Τα χαρακτηριστικά της συνεδρίας είναι:

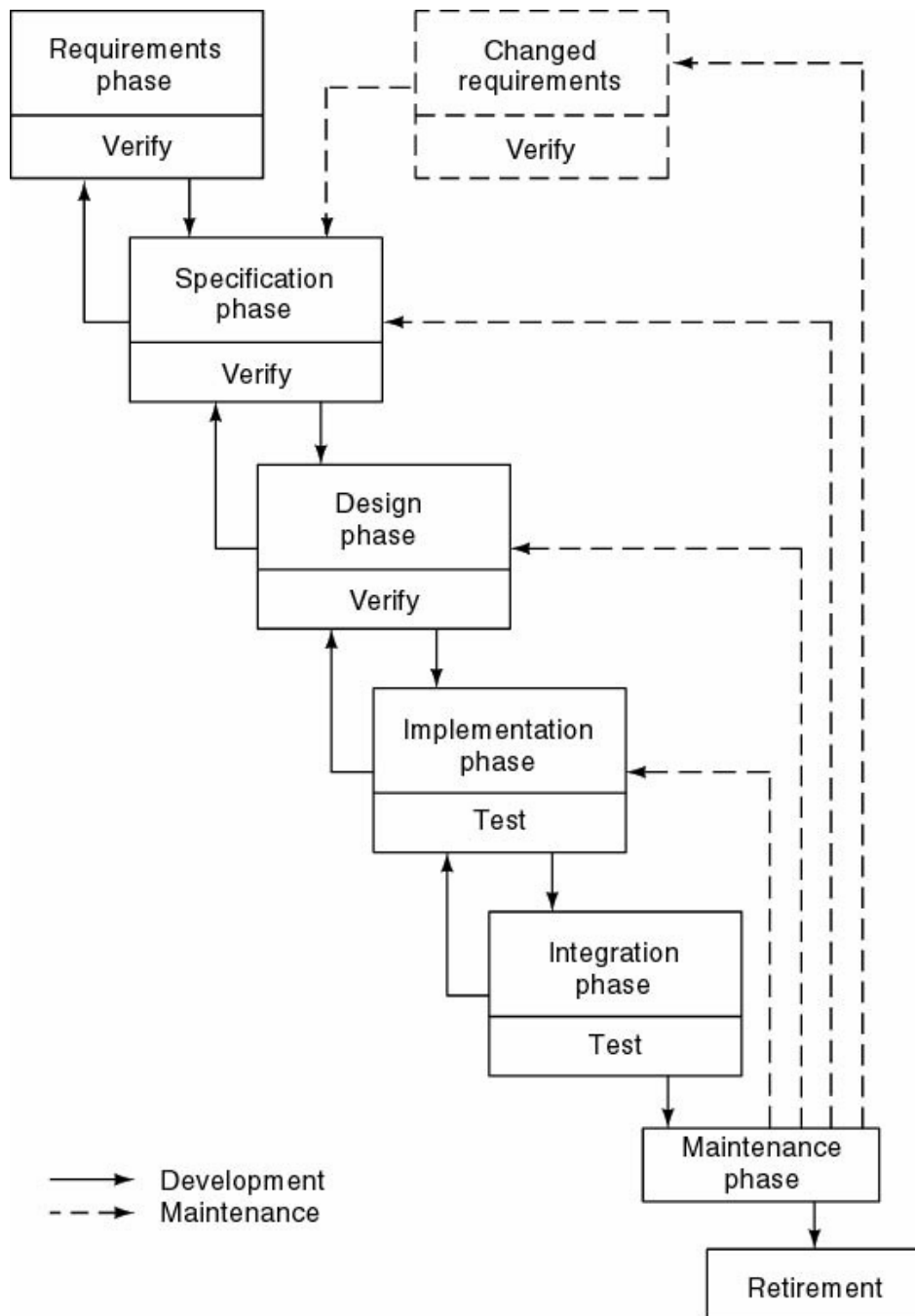
- Συνήθως διαρκεί 5 με 10 μέρες σε ένα διάστημα 3 εβδομάδων.
- Υπάρχει τυπική ατζέντα και διαδικαστικοί κανόνες.
- Γίνεται προετοιμασία ερωτημάτων (όπως στις συνεντεύξεις).
- Υπάρχει συνέχεια (*Post-session follow-up*). Όπως η Αναφορά Συνέντευξης που συντάσσεται στο τέλος μιας συνέντευξης.

2.3.2 Μοντέλο του καταρράκτη

Χαρακτηρίζεται από²⁴ :

1. Σειριακά βήματα (*phases*)
2. Ανάδραση ανάμεσα σε δύο γειτονικά βήματα
3. Βασίζεται στην δημιουργία προδιαγραφών σε κάθε βήμα

²⁴ Πάγκαλος Γ., Δομημένος Προγραμματισμός, «Αρχές και Τεχνικές του Δομημένου Προγραμματισμού», Σημειώσεις Παραδόσεων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.



Εικόνα 0.4: Γενικό διάγραμμα του μοντέλου του καταρράκτη.²⁵

Πιο συγκεκριμένα, η δημιουργία του λογισμικού βάσει του μοντέλου του καταρράκτη γίνεται σε τρεις διακριτές φάσεις:

1. Τη φάση της Ανάλυσης (κατανόηση προβλήματος – ανάλυση απαιτήσεων)

²⁵ Πάγκαλος Γ., Δομημένος Προγραμματισμός, «Αρχές και Τεχνικές του Δομημένου Προγραμματισμού», Σημειώσεις Παραδόσεων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

2. Τη φάση του Σχεδιασμού (σχεδιασμός μιας λειτουργικής / βέλτιστης λύσης)
3. Τη φάση της Υλοποίησης (ανάπτυξη εκτελέσιμου κώδικα)

Τα προτερήματα του μοντέλου είναι η παραγωγή προδιαγραφών και η διευκόλυνση στη συντήρηση, ενώ τα μειονεκτήματα είναι:

4. προδιαγραφές που δεν μπορούν να αλλάξουν στη πορεία.
5. δεν είναι ρεαλιστική παραδοχή, ο χρήστης να συμμετέχει μόνο στην αρχή.
6. σειριακή και πλήρης ολοκλήρωση κάθε βήματος δεν είναι πάντα ενδεδειγμένη.
7. διαδικασία είναι δύσκολο να ελεγχθεί.
8. Ο χρήστης βλέπει το προϊόν πολύ αργά στη διάρκεια της διαδικασίας.

2.3.3 Ταχεία ανάπτυξη εφαρμογής

Η μέθοδος αυτή (*Rapid Application Development*) προτείνει τα προϊόντα να αναπτύσσονται γρηγορότερα και με καλύτερη ποιότητα με τους εξής τρόπους:

1. Χρησιμοποιώντας ομάδες ώστε να συλλεχθούν οι απαιτήσεις.
2. Πρωτοτυποποίηση και έλεγχο των σχεδίων από το χρήστη.
3. Επαναχρησιμοποιώντας τα λογισμικά συστατικά.
4. Κρατώντας μυστική την επικοινωνία της ομάδας.

Μπορούμε να πούμε ότι είναι η μέθοδος κύκλου ζωής με έμφαση σε έναν πολύ μικρό κύκλο ανάπτυξης. Προσανατολίζεται δε στις παρακάτω φάσεις:

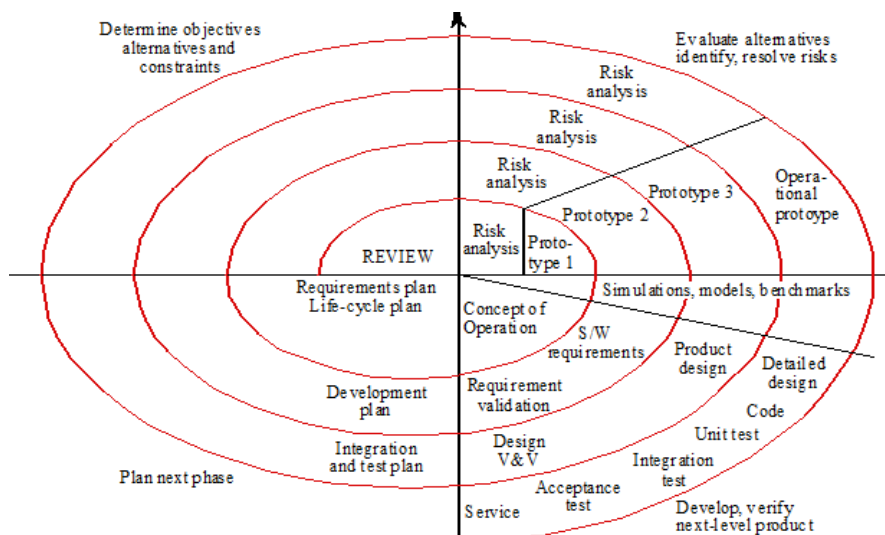
- επιχειρησιακή διαμόρφωση,
- διαμόρφωση στοιχείων,

- διαμόρφωση διαδικασίας,
- υλοποίηση εφαρμογής,
- δοκιμές και επαλήθευση.

Θα πρέπει να σημειωθεί ότι η μέθοδος αυτή προϋποθέτει την δημιουργία αποτελεσματικών ομάδων εργασίας καθώς και την δέσμευση των δύο πλευρών για άμεσες ενέργειες αναγκαίες για την συμπλήρωση του συστήματος.

2.3.4 Το Σπειροειδές Μοντέλο

Το σπειροειδές μοντέλο, είναι το μοντέλο πρωτοτυποποίησης όπου κάνουμε έλεγχο σκοπιμότητας και ανάλυση ρίσκο στο τέλος κάθε βήματος. Εάν η ανάλυση ρίσκου αποτύχει τότε το έργο δεν συνεχίζεται. Παρόλα αυτά αξίζει να αναφερθεί ότι αν το έργο έχει ήδη προχωρήσει πολύ είναι δύσκολο να τερματιστεί ακόμη και αν η ανάλυση ρίσκου αποτύχει Το σπειροειδές μοντέλο είναι καταλληλότερο για μεγάλα έργα μιας και το κόστος διαχείρισής του είναι αυξημένο.



Εικόνα 0.5 : Το σπειροειδές μοντέλο.²⁶

Κάθε βρόχος του σπειροειδούς μοντέλου διαιρείται σε τέσσερις τομείς²⁷:

²⁶ Somerville I., «Βασικές Αρχές Τεχνολογίας Λογισμικού», Εκδόσεις Κλειδάριθμος, 2009.

²⁷ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

1. **Καθορισμός στόχων.** Τίθενται καθορισμένοι στόχοι για τη δεδομένη φάση του έργου. Προσδιορίζονται περιορισμοί για τη διαδικασία και για το προϊόν, και καταρτίζεται ένα λεπτομερές πλάνο διαχείρισης. Προσδιορίζονται οι κίνδυνοι του έργου. Μπορεί να προγραμματιστούν εναλλακτικές στρατηγικές, ανάλογα με τους κινδύνους.
2. **Αξιολόγηση και περιορισμός κινδύνων.** Για κάθε προσδιορισμένο κίνδυνο του έργου, διενεργείται λεπτομερής ανάλυση και λαμβάνονται μέτρα για τον περιορισμό του κινδύνου.
3. **Ανάπτυξη και επικύρωση.** Αφού γίνει η εκτίμηση των κινδύνων, επιλέγεται ένα μοντέλο που θεωρείται κατάλληλο για την ανάπτυξη του συστήματος.
4. **Προγραμματισμός.** Γίνεται μία επισκόπηση του έργου και αποφασίζεται αν θα προχωρήσει σε επόμενο βρόχο του σπειροειδούς. Αν αποφασιστεί η συνέχεια του έργου, καταρτίζονται πλάνα για την επόμενη φάση.

2.3.4.1 Επαναληπτική Ενοποιημένη Διαδικασία

Η διαδικασία αυτή (*Rational Unified Process*) 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 RUP έχει τέσσερις φάσεις στο χρόνο²⁸:

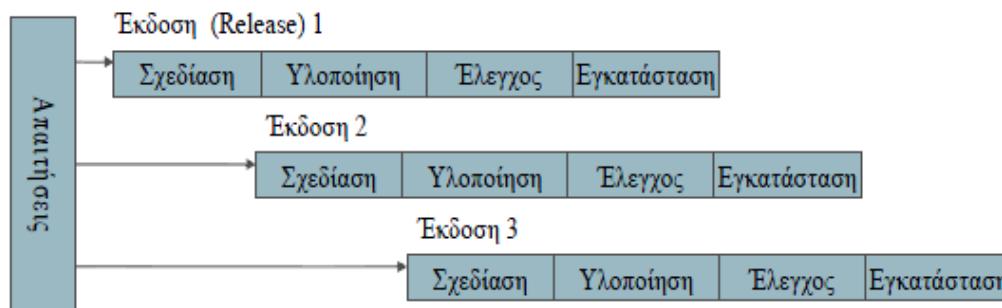
1. Έναρξη (*Inception*) – Ορισμός του έργου και της έκτασής του.

²⁸ Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού», Πανεπιστήμιο Πατρών, 2009.

2. Επεξεργασία (Elaboration) – Κατάστρωση μεθόδου υλοποίησης του έργου, μοντελοποίηση χαρακτηριστικών του έργου, ορισμός της αρχιτεκτονικής του συστήματος.
3. Κατασκευή (Construction) – Υλοποίηση του έργου.
4. Μετάβαση (Transition) – Ανάπτυξη του συστήματος στο περιβάλλον χρήσης του.

2.3.5 Το Αυξητικό Μοντέλο ²⁹

Στο μοντέλο αυτό η ανάπτυξη του λογισμικού γίνεται σε επάλληλες (διαδοχικές) εκδόσεις, όπου σε κάθε έκδοση του λογισμικού προσθέτουμε και νέες λειτουργίες / ποιοτικά χαρακτηριστικά, από ένα προκαθορισμένο σύνολο απαιτήσεων. Το Μοντέλο αυτό μπορεί να περιγραφεί με το παρακάτω γενικό διάγραμμα:



Εικόνα 0.6 Διάγραμμα του αυξητικού μοντέλου.³⁰

Βασικά προτερήματα του μοντέλου:

- Σε κάθε έκδοση έχουμε ένα σύστημα σε λειτουργία
- Καλύτερη διανομή κόστους στο χρόνο

Βασικά μειονεκτήματα του μοντέλου:

²⁹ Πάγκαλος Γ., Δομημένος Προγραμματισμός, «Αρχές και Τεχνικές του Δομημένου Προγραμματισμού», Σημειώσεις Παραδόσεων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

³⁰ Πάγκαλος Γ., Δομημένος Προγραμματισμός, «Αρχές και Τεχνικές του Δομημένου Προγραμματισμού», Σημειώσεις Παραδόσεων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

- Οι απαιτήσεις δεν πρέπει να αλλάζουν

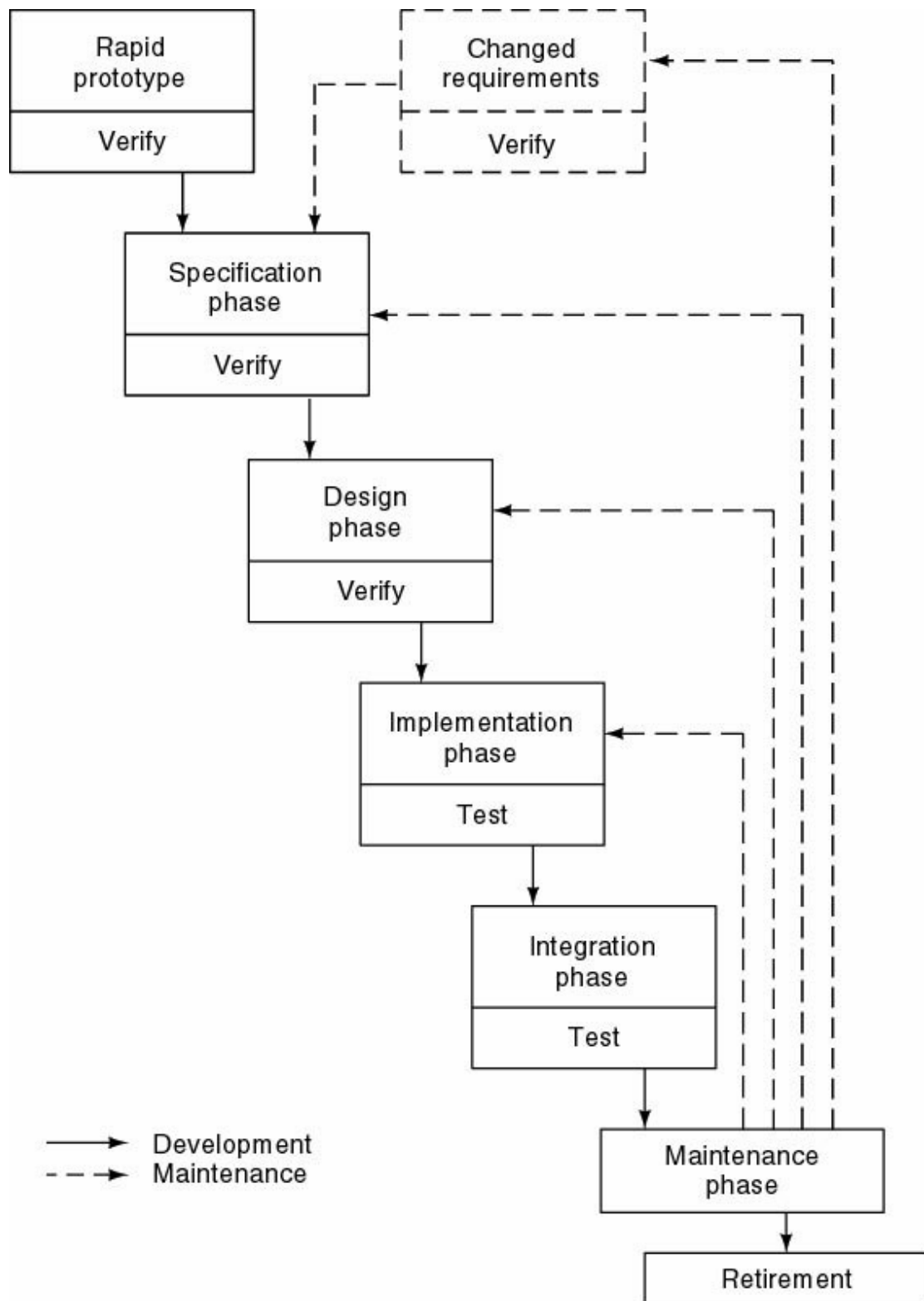
2.3.5.1 Το Μοντέλο της Δημιουργίας Πρωτότυπου

Είναι μια τεχνική ανάπτυξης λογισμικού κατά την οποία δημιουργούνται ελλιπείς εκδόσεις του συνολικού προγράμματος που θα αναπτυχθεί. Ένα πρωτότυπο μιμείται μερικές πτυχές των χαρακτηριστικών γνωρισμάτων του ενδεχόμενου προγράμματος, και μπορεί να είναι απολύτως διαφορετικό από την τελική εφαρμογή.

Ο τυποποιημένος σκοπός ενός πρωτοτύπου είναι να δοθεί στους χρήστες του λογισμικού η ευκαιρία να αξιολογήσουν τις προτάσεις των υπεύθυνων για την ανάπτυξη με το να δοκιμάζουν στην πράξη ένα προσχέδιο του προϊόντος, παρά να πρέπει να ερμηνεύσουν και να αξιολογήσουν ένα σχέδιο βασισμένο στις περιγραφές.

Η διαμόρφωση πρωτοτύπου μπορεί επίσης να χρησιμοποιηθεί από τους τελικούς χρήστες. Στην περίπτωση αυτή στόχος της είναι να περιγράψει τις απαιτήσεις που οι υπεύθυνοι για την ανάπτυξη δεν έχουν εξετάσει. Έτσι «ο έλεγχος του πρωτοτύπου» μπορεί να είναι ένας βασικός παράγοντας στην εμπορική σχέση μεταξύ των προμηθευτών, της λύσης και των πελατών τους.

Στην μέθοδο αυτή, μετά την αρχική διεύρυνση ξεκινά ένας κύκλος προσδιορισμού των προδιαγραφών, του σχεδιασμού του συστήματος, της κωδικοποίησης και των δοκιμών. Κατά την διάρκεια του κύκλου αυτού δημιουργείται το πρωτότυπο. Στην συνέχεια, αφού δημιουργηθεί και αξιολογηθεί το πρωτότυπο, υλοποιείται το λογισμικό. Στην τελευταία φάση μένει μόνο η συντήρηση του λογισμικού.



Εικόνα 0.7: Γενικό διάγραμμα του μοντέλου της δημιουργίας πρωτοτύπου.³¹

³¹ Πάγκαλος Γ., Δομημένος Προγραμματισμός, «Αρχές και Τεχνικές του Δομημένου Προγραμματισμού», Σημειώσεις Παραδόσεων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

ΚΕΦΑΛΑΙΟ 3. ΤΕΧΝΙΚΕΣ ΔΙΟΙΚΗΣΗΣ ΕΡΓΩΝ

3.1 Tenstep

Η τεχνική διοίκησης έργων της TenStep είναι μια μεθοδολογία η οποία διαχειρίζεται την εργασία σε ένα έργο. Κύριος στόχος της είναι να είναι τόσο ευέλικτη όσο χρειάζεται ώστε να διοικηθεί αποτελεσματικά το έργο.

Παραδείγματος χάριν, μπορεί να επιβεβαιωθεί ότι δεν είναι απαραίτητο να σπαταληθεί ένα μέρος του χρόνου στη διαχείριση κινδύνου σε ένα έργο που απαιτεί 500 ώρες προσπάθειας, ενώ έχει πολλές ομοιότητες με πολλά άλλα έργα που έχουν ήδη ολοκληρωθεί πριν από αυτό.

Αυτό δεν υποδηλώνει ότι πρέπει να αγνοηθούν οι πιθανοί κίνδυνοι, αλλά το ότι δεν χρειάζεται να πολύτιμος χρόνος ο οποίος θα μπορούσε να επενδυθεί για ένα άλλο, βαρύνουσα σημασίας, έργο (όπως για παράδειγμα, αυτό της ενσωμάτωσης νέας τεχνολογίας στην επιχείρηση).

Τα βήματα που ακολουθεί είναι 10 και δίνονται συνοπτικά παρακάτω:

1. Καθορισμός της εργασίας.
2. Διαμόρφωση του προγράμματος και του προϋπολογισμού.
3. Διαχείριση του προγράμματος και του προϋπολογισμού.
4. Διαχείριση των ζητημάτων.
5. Διαχείριση της αλλαγής.
6. Διαχείριση της επικοινωνίας.
7. Διαχείριση κινδύνου.
8. Διαχείριση ανθρώπινου δυναμικού.
9. Διαχείριση ποιότητας.

10. Διαχείριση μετρικών.

3.2 Prince2

Η PRINCE2 αποτελεί την καταλληλότερη μεθοδολογία διοίκησης έργων, κυρίως όταν πρόκειται για έργα μεγάλης κλίμακας ή κρίσιμης σημασίας. Το 1989 η Βρετανική κυβέρνηση υιοθέτησε τη μεθοδολογία PRINCE (Projects IN Controlled Environments - «Έργα σε Ελεγχόμενα Περιβάλλοντα») ως επίσημη μεθοδολογία διοίκησης έργων για τα κρατικά έργα στο Ηνωμένο Βασίλειο.

Σιγά σιγά, η πρακτική χρησιμότητα και τα οφέλη από τη χρήση της μεθόδου αυτής οδήγησαν στη διάδοση της πέρα από τον κρατικό και στον ιδιωτικό τομέα.

Στις μέρες μας η PRINCE2 είναι διεθνώς αναγνωρισμένη ως η 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.

Η μεθοδολογία PRINCE2 υιοθετεί τις αρχές της δομημένης διαχείρισης, δηλαδή της διοίκησης του έργου με έναν λογικό και κατάλληλα οργανωμένο τρόπο. Η μέθοδος ακολουθεί συγκεκριμένα «βήματα» συστηματικής προσέγγισης 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 PRINCE2³²:

- Προσφέρει μια «κοινή γλώσσα συνεννόησης» μεταξύ όλων των εμπλεκόμενων μερών του έργου συμβάλλουν πρακτικά στην αποφυγή παρερμηνειών αναφορικά με τους επιδιωκόμενους ενδιάμεσους και τελικούς στόχους του έργου.
- Ενσωματώνει όλη την διοικητική πληροφορία που παράγεται κατά τη διάρκεια του έργου, ως εισροή ή εκροή των εφαρμοζόμενων διαδικασιών.

³² <http://www.gatech.gr/pm.html>

- 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.

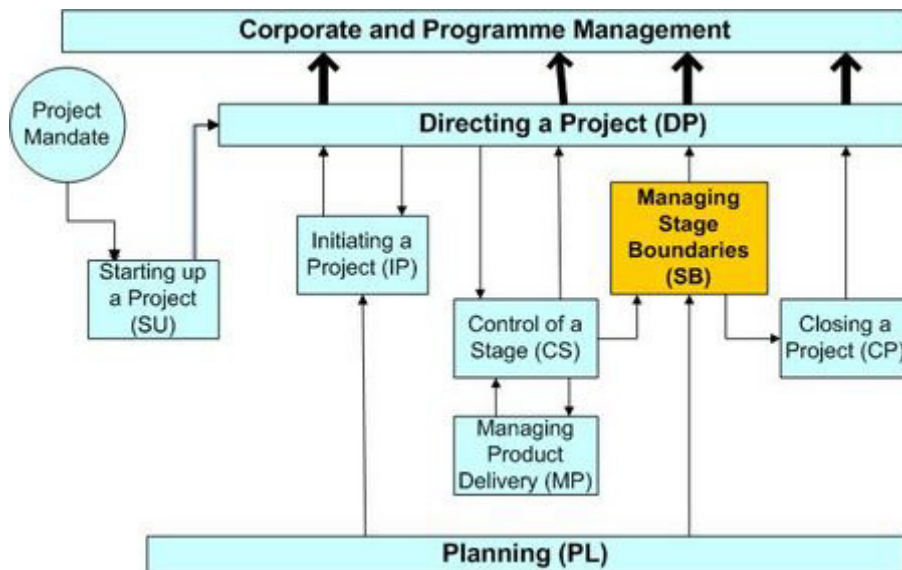
Μερικά από τα πολλά χαρακτηριστικά της είναι:

1. Μια καθορισμένη δομή διαχείρισης έργου.
2. Ευέλικτα σημεία λήψης αποφάσεων.
3. Ένα σύνολο διαδικασιών ελέγχου.
4. Εστίαση στα προϊόντα που προορίζονται για τον πελάτη.
5. Εστίαση στα προϊόντα που παράγονται κατά τη διάρκεια του έργου.

Αυτή η μεθοδολογία μπορεί να χρησιμοποιηθεί με το ίδιο ποσοστό επιτυχίας και σε έργα που δεν σχετίζονται με πληροφορική, όπως για παράδειγμα στον κατασκευαστικό τομέα. Επίσης η εφαρμογή της είναι απλή και δεν χρειάζεται ιδιαίτερη εκπαίδευση.

Η χρήση της είναι διαδεδομένη στον ιδιωτικό και στο δημόσιο τομέα. Ένα χαρακτηριστικό της PRINCE2 που δεν υπάρχει σε άλλες μεθοδολογίες είναι η έννοια της "εξασφάλισης της προόδου" μέσα από τις τρεις συνδεδεμένες μεταξύ τους φάσεις.

Η παρακάτω εικόνα δείχνει ότι η PRINCE2 αποτελεί μια προσέγγιση βασισμένη στις διαδικασίες για τη διαχείριση του έργου. Κάθε διαδικασία έχει τις εισόδους και τις εξόδους της και συσχετισμένες εργασίες που αλληλεπιδρούν και πρέπει να διεξαχθούν.



Εικόνα 0.1: Η μεθοδολογία Prince2.³³

Η μεθοδολογία μας δείχνει ότι το έργο τμηματοποιείται σε φάσεις που μπορούμε να διαχειριστούμε δίνοντας μας έτσι την ευκαιρία για έναν αποτελεσματικό έλεγχο. Για παράδειγμα εξετάζουμε τη φάση project planning (σχεδιασμός έργου). Αυτή η φάση επικεντρώνεται στα αποτελέσματα παρά στο σχεδιασμό του πότε οι δραστηριότητες θα πραγματοποιηθούν.

Η PRINCE2 οδηγείται από το περιβάλλον της εταιρείας, το οποίο περιγράφει τον τρόπο σκέψης και τα κίνητρα για το έργο. Το ίδιο πράγμα ισχύει για όλες τις φάσεις που παρουσιάζονται στο σχήμα.

Οι κύριες διαδικασίες είναι οι εξής :

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

³³ http://en.wikipedia.org/wiki/PRINCE2#Description_of_the_PRINCE2_method Τελευταία πρόσβαση: Αύγουστος 2011.

Οι φάσεις είναι:

Εκκίνηση έργου (SU)

Η πρώτη διαδικασία της PRINCE2 είναι preproject διαδικασία 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 :

1. Εξασφάλιση ότι οι απαραίτητες πληροφορίες για την ομάδα έργου είναι διαθέσιμες .
2. Σχεδιασμός και συγκρότηση της ομάδας διαχείρισης έργου.
3. Απόφαση για τη μέθοδο που θα υιοθετηθεί μέσα στο έργο για να δοθεί λύση.
4. Να εξασφαλιστεί ότι οι στόχοι του έργου είναι γνωστοί.
5. Τεκμηρίωση και επιβεβαίωση της ύπαρξης business case για το έργο.
6. Η παροχή της κατευθυντήριας γραμμής για τις διαδικασίες λήψης αποφάσεων που απαιτούνται κατά τη διάρκεια ζωής του έργου.
7. Η εξασφάλιση ότι η επένδυση του χρόνου και της προσπάθειας που απαιτούνται για το έργο έγινε σοφά λαμβάνοντας υπ' όψη τους κινδύνους του έργου.
8. Να προγραμματιστεί η εργασία που απαιτείται για να σχεδιαστεί η PRINCE2 "σύμβαση" μεταξύ του πελάτη και του προμηθευτή.

Αρχικοποίηση έργου (IP)

1. Καθιέρωση υγιούς διοικητικής βάσης.
2. Το διοικητικό προϊόν που δημιουργείται είναι το "έγγραφο έναρξης έργου" η βασική γραμμή έναντι της οποίας αξιολογείται η πρόοδος και η επιτυχία του έργου.
3. Προετοιμάζει τις πληροφορίες που δείχνουν ότι υπάρχει ικανοποιητική αιτιολόγηση για τη συνέχεια του έργου.

4. Δημιουργεί ένα λεπτομερές σχέδιο για το μέρος του έργου που η διοίκηση είναι σε θέση να εγκρίνει.

Διαχείριση ορίων σταδίου (SB)

Αυτή η διαδικασία παρέχει στην ομάδα διοίκησης έργου (*project board*) σημεία απόφασης για την συνέχιση ή όχι του έργου. Οι στόχοι της διαδικασίας είναι:

1. Να διασφαλίσει η ομάδα διοίκησης έργου ότι όλα τα προϊόντα που ήταν προγραμματισμένα γι' αυτό το στάδιο έχουν ολοκληρωθεί.
2. Η παροχή των απαραίτητων πληροφοριών στην ομάδα διοίκησης έργου έτσι ώστε να αξιολογήσουν την βιωσιμότητα της συνέχισης του έργου.
3. Η παροχή των απαραίτητων πληροφοριών στην ομάδα διοίκησης έργου για την έγκριση της ολοκλήρωσης της τρέχουσας φάσης και την έναρξη της επόμενης.
4. Καταγραφή μετρήσεων που θα μπορούσαν να βοηθήσουν σε επόμενες φάσεις.

Έλεγχος σταδίου (CS)

Αυτή η διαδικασία περιγράφει τις δραστηριότητες παρακολούθησης και ελέγχου του *project manager* που συμμετέχει στην εξασφάλιση ότι μια φάση προχωρά σύμφωνα με τον προγραμματισμό και αντιδρά σε απροσδόκητα γεγονότα.

Η διαδικασία διαμορφώνει τον πυρήνα της προσπάθειας του *project manager* για το έργο και χειρίζεται τους καθημερινούς στόχους και δραστηριότητες διαχείρισης του έργου. Στη διάρκεια κάθε φάσης υπάρχει ένας κύκλος που αποτελείται από:

1. Έγκριση της εργασίας που πρέπει να γίνει.
2. Συλλογή πληροφοριών προόδου σχετικά με αυτήν την εργασία.
3. Επαγρύπνηση για αλλαγές.
4. Ανασκόπηση της κατάστασης.
5. Έκδοση αναφορών.
6. Λήψη απαραίτητης ενέργειας.

Διαχείριση παράδοσης προϊόντος (MP)

Στόχος της διαδικασίας αυτής είναι η εξασφάλιση ότι τα προγραμματισμένα προϊόντα δημιουργούνται και παραδίδονται. Αυτό γίνεται με:

1. Εξασφάλιση ότι η εργασία που έχει ανατεθεί στην ομάδα έχει εγκριθεί και συμφωνηθεί.
2. Προγραμματισμός της εργασίας της ομάδας.
3. Εξασφάλιση ότι η εργασία εκτελείται.
4. Τακτική αξιολόγηση της προόδου της εργασίας και των προβλέψεων.
5. Λήψη αποδοχής για τα ολοκληρωμένα προϊόντα.
6. Εξασφάλιση ότι τα προϊόντα ικανοποιούν τα συμφωνημένα κριτήρια ποιότητας.
7. Έκθεση αναφοράς προόδου και ποιότητας προς τον project manager.

Κλείσιμο έργου (CP)

Στόχος της διαδικασίας είναι η εκτέλεση προγραμματισμένου κλεισίματος του έργου. Αυτό γίνεται με αίτηση του project manager 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:

1. Διαπιστωθεί η έκταση ικανοποίησης των αρχικών αντικειμενικών σκοπών του έργου.
2. Επιβεβαιωθεί η ικανοποίηση του πελάτη από τα προϊόντα.
3. Εξασφαλισθεί μέχρι ποιο σημείο όλα τα αναμενόμενα προϊόντα έχουν παραδοθεί και έχουν γίνει αποδεκτά.
4. Εξασφαλιστεί ότι η εμπειρία που αποκτήθηκε καταγράφηκε για να αξιοποιηθεί σε επόμενα έργα.
5. Καταγραφεί αν η δραστηριότητα της διοίκησης του έργου ήταν επιτυχής ή όχι.
6. Γίνονται τυχόν συστάσεις για επόμενες ενέργειες.
7. Επιβεβαιωθεί ότι έχουν διευθετηθεί η συντήρηση και η εγγύηση.

Προγραμματισμός (PL)

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:

1. Σχεδιασμό πλάνου.
2. Καθορισμός των απαραίτητων δραστηριοτήτων και εξαρτήσεων.
3. Εκτίμηση της απαιτούμενης προσπάθειας.
4. Χρονοπρογραμματισμός των διαθέσιμων πόρων.
5. Ανάλυση του κινδύνου.
6. Επιπλέον περιγραφή του πλάνου, των υποθέσεων του και των βημάτων ποιότητας.

Διεύθυνση έργου (DP)

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 DP καλύπτει τα βήματα που πρέπει να εκτελεστούν από την ομάδα ανώτερης διαχείρισης σε όλο το φάσμα του έργου:

1. Έγκριση προετοιμασίας πλάνου έργου και επιχειρησιακής υπόθεσης (business case) για το έργο.
2. Έγκριση άδειας έναρξης έργου.
3. Έλεγχος ότι το έργο εξακολουθεί να είναι δικαιολογημένο σε βασικά σημεία του κύκλου ζωής του.
4. Παρακολούθηση προόδου και παροχή συμβουλών όπου απαιτείται.
5. Εξασφάλιση ότι το έργο έχει ελεγχόμενο τέλος.

Πλεονεκτήματα της PRINCE2

- ❏ Η μέθοδος αυτή εφαρμόζεται ανεξάρτητα από το είδος του έργου.

- ❑ 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, του κόστους, της 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.

Μειονεκτήματα της PRINCE2

- ❑ Είναι μια μέθοδος και όχι μια "πανάκεια".
- ❑ Οι άνθρωποι που χρησιμοποιούν PRINCE2 πρέπει να συνεχίσουν να σκέφτονται.
- ❑ Μερικοί άνθρωποι ερμηνεύουν τη μεθοδολογία και την εφαρμόζουν στο έργο τους με έναν άκαμπτο τρόπο που σαν αποτέλεσμα έχει την αποτυχία της μεθόδου.

3.3 PMBOK

Η μέθοδος PMBOK ή αλλιώς *Project Management Body of Knowledge* είναι μια συλλογή από διαδικασίες και γνωστικές περιοχές, που θεωρούνται ως οι πλέον καλύτερες τεχνικές για την διοίκηση.

Ως ένα διεθνώς αναγνωρισμένο πρότυπο (*IEE 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 αναγνωρίζει 5 βασικές αρχές:

1. Έναρξη.
2. Σχεδιασμό.
3. Εκτέλεση.
4. Έλεγχος.
5. Κλείσιμο.

Οι διαδικασίες αλληλεπιδρούν και 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.).

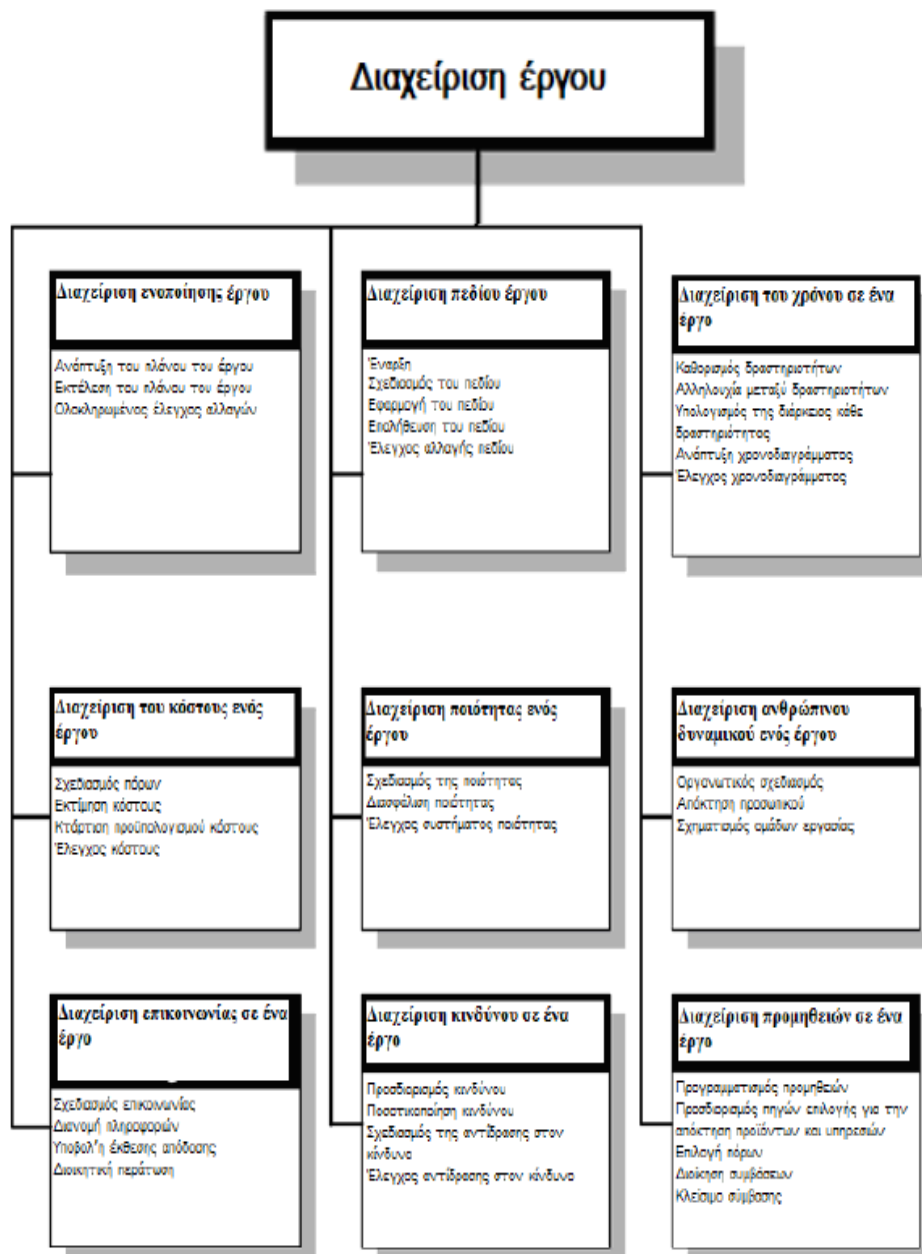
Οι εννέα γνωστικές περιοχές είναι:

1. Ενσωμάτωση έργου (*Project Integration Management*).
2. Προοπτικές έργου (*Project Scope Management*).
3. Χρονοπρογραμματισμός έργου (*Project Time Management*).
4. Προϋπολογισμός έργου (*Project Cost Management*).
5. Ποιότητα έργου (*Project Quality Management*).
6. Πόροι ανθρωπίνου δυναμικού έργου (*Project Human Resource Management*).
7. Επικοινωνίες έργου (*Project Communications Management*).
8. Κίνδυνος έργου (*Project Risk Management*).
9. Προμήθεια έργου (*Project Procurement 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).



Εικόνα 0.2: Προεπισκόπηση των περιοχών γνώσης της διαχείρισης έργου και των διαδικασιών διαχείρισης έργου.

3.4 Hermes

Η μέθοδος Hermes³⁴ χρησιμοποιείται για διοίκηση έργων στον τομέα της τεχνολογίας, της πληροφορικής και των επικοινωνιών (Information and Communication Technologies-ICT). Η μέθοδος αυτή είναι ένα ανοιχτό πρότυπο που αναπτύχθηκε από την Ελβετική Κυβέρνηση, και είναι σε θέση να αντιμετωπίζει όλες τις πραγματικές απαιτήσεις ενός έργου ICT.

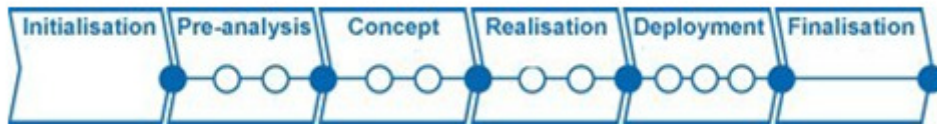
Τα περισσότερα έργα χρησιμοποιούν στοχοκεντρική προσέγγιση, εκτέλεση και έλεγχο. Η μέθοδος Hermes συνιστά διαδικασίες οι οποίες ακολουθούν όχι μόνο την προσέγγιση στόχου αλλά και του αποτελέσματος. Ιδιαίτερη σημασία δίνεται στα ενδιαφέροντα και στα καθήκοντα των αγοραστών, του διευθυντή καθώς επίσης και των συνεργατών έργου.

Αυτό έχει σαν αποτέλεσμα να δημιουργεί τις κατάλληλες συνθήκες συνεργασίας μεταξύ όλων των εμπλεκόμενων στο έργο. Η μέθοδος δομεί την ανάπτυξη και την εκτέλεση του έργου, ορίζοντας τα αποτελέσματα και τις φάσεις από τα οποία δημιουργούνται οι απαραίτητες δραστηριότητες και ευθύνες.

Για την καλύτερη οργάνωση του έργου, η μέθοδος ορίζει έξι φάσεις που είναι οι εξής:

1. Έναρξη (Initialization)
2. Προ-ανάλυση (Pre-analysis)
3. Ιδέα (Concept)
4. Πραγμάτωση (Realization)
5. Ανάπτυξη (Deployment)
6. Οριστικοποίηση (Finalization)

³⁴ http://www.hermes.admin.ch/ict_project_management/the-hermes-method



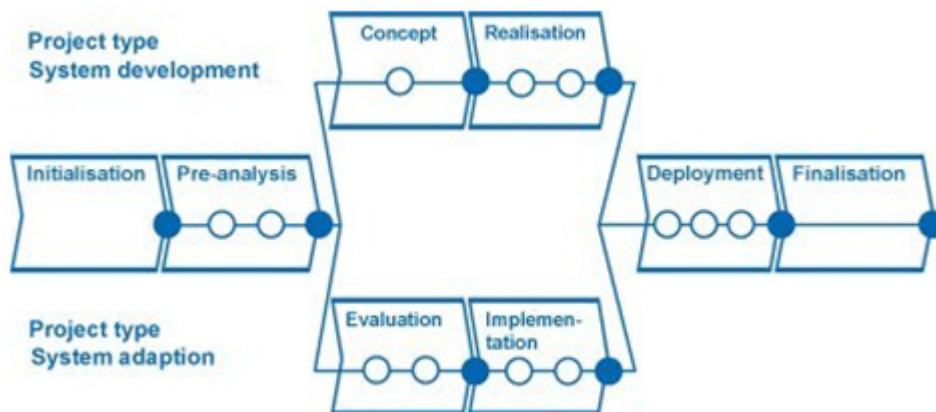
Εικόνα 0.3: Οι έξι φάσεις της μεθόδου Hermes³⁵

Οι φάσεις του έργου οργανώνονται ανάλογα με τα αποτελέσματα που παράγονται, και τις ανάγκες για αποφάσεις που προκύπτουν από αυτές.

Η μέθοδος Hermes διακρίνει δύο είδη έργων:

1. Ανάπτυξη συστήματος όπου η υλοποίηση αρχίζει από την αρχή.
2. Συστήματα προσαρμογής για λόγους πώλησης.

Το μοντέλο φάσεων είναι διαφορετικό για κάθε είδος.



Εικόνα 0.4: Οι φάσεις της μεθόδου Hermes ανάλογα με τον τύπο του έργου¹⁶

Οι διεργασίες περιγράφονται στη δομή των υπομοντέλων και είναι ίδιες για κάθε είδος έργου. Εφαρμόζονται στα περισσότερα έργα πληροφορικής και επικοινωνιών (ICT):

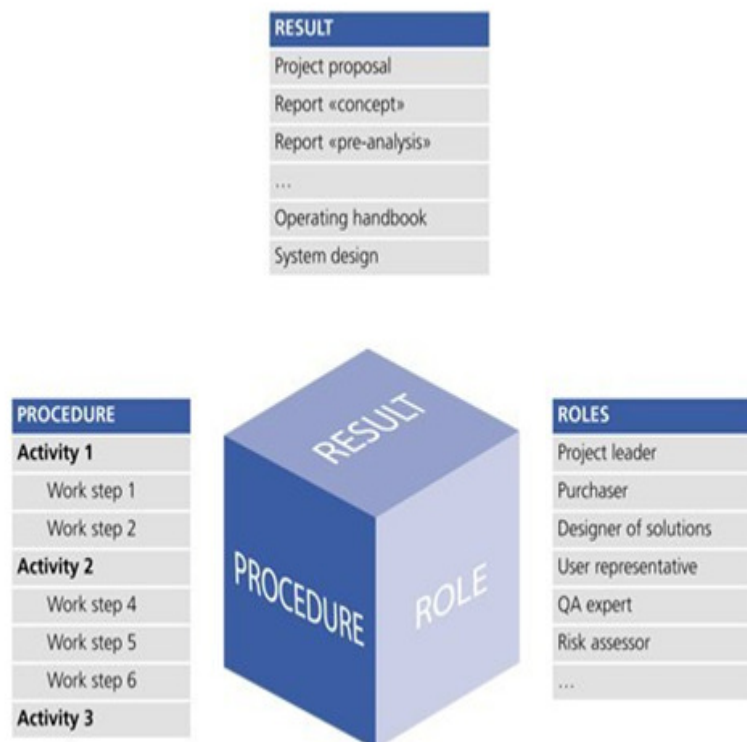
- Διοίκηση έργου (project management)
- Εγγύηση ποιότητας (quality assurance)

³⁵ http://www.hermes.admin.ch/ict_project_management/the-hermes-method

- Διοίκηση κινδύνου (*risk management*)
- Διοίκηση σύνθεσης/διαμόρφωσης (*configuration management*)
- Προώθηση έργου (*project marketing*)

Σ' ένα έργο τα συμφωνηθέντα παραδοτέα πρέπει να παραχθούν, οι διεργασίες πρέπει να είναι φανερές και οι ρόλοι πρέπει να είναι ενεργή. Γι' αυτό το λόγο, η μέθοδος *Hermes* υποστηρίζει τρεις όψεις:

- Όψη στις διεργασίες: πως το έργο εκτελείται;
- Όψη στα παραδοτέα: τι παράγεται;
- Όψη στους ρόλους: ποιος κάνει τι;



Εικόνα 0.5: Παράδειγμα όψεων της μεθόδου *Hermes*³⁶

³⁶ http://www.hermes.admin.ch/ict_project_management/the-hermes-method

3.5 Εργαλεία Διοίκησης Έργων

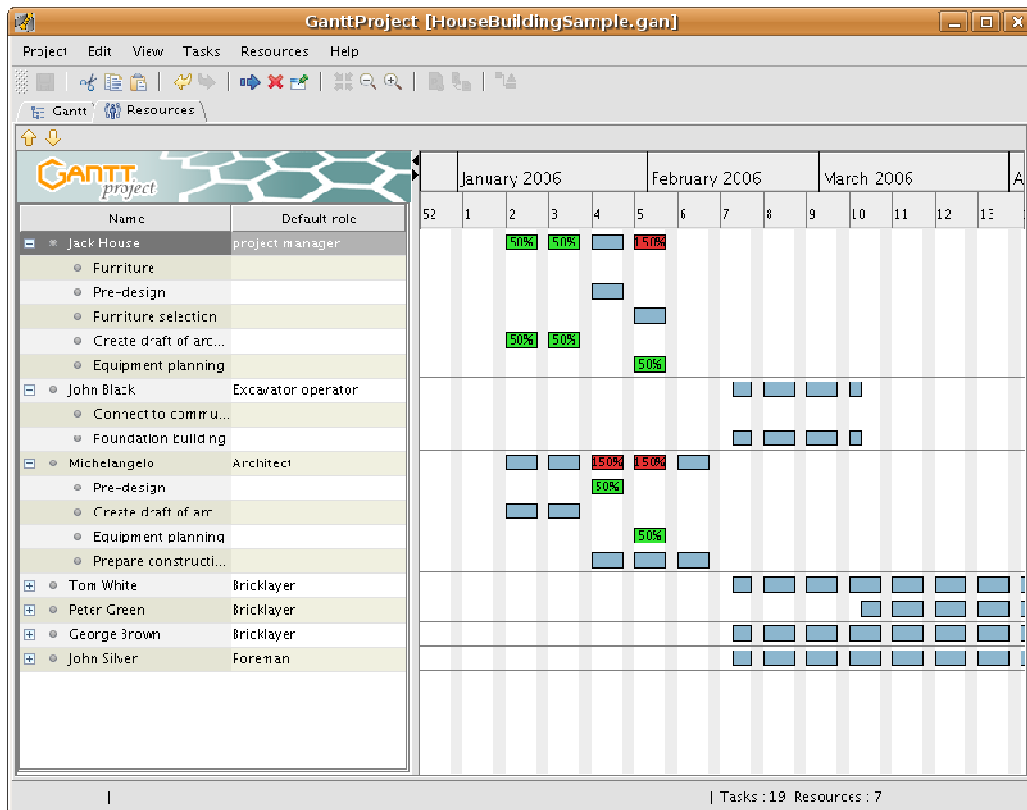
3.5.1 GanttProject

Αποτελεί ένα εργαλείο διοίκησης έργου ανοιχτού κώδικα με περιορισμένες δυνατότητες, παρόλα αυτά καλύπτει τις περισσότερες ανάγκες διοίκησης έργων. Επικεντρώνεται κυρίως 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 Gantt διάγραμμα για τις διεργασίες και σε διάγραμμα ανάθεσης πόρων για τους πόρους. Υπάρχει η δυνατότητα δημιουργίας αναφορών σε HTML και PDF μορφή και ανταλλαγής δεδομένων με το Microsoft Project και προγραμματιστικών φύλλων. Το περιβάλλον του συστήματος φαίνεται στην εικόνα.

Τα βασικά χαρακτηριστικά του GanttProject και τα οποία το ξεχωρίζουν 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. Ο χρήστης μπορεί να εκμεταλλευτεί όλες τις δυνατότητες που του παρέχει το πρόγραμμα και να εξοικειωθεί με λίγες ώρες ενασχόλησης ή εκπαίδευσης.
- Το GanttProject είναι δωρεάν.
- Μπορεί να λειτουργήσει κάτω από οποιοδήποτε λειτουργικό σύστημα το οποίο υποστηρίζει Java.
- Είναι πλήρως παραμετροποιήσιμο και μπορεί να ενσωματώσει οποιοδήποτε χαρακτηριστικό μπορεί να χρειαστεί ένας χρήστης όπως υποστήριξη συγκεκριμένων αναφορών.



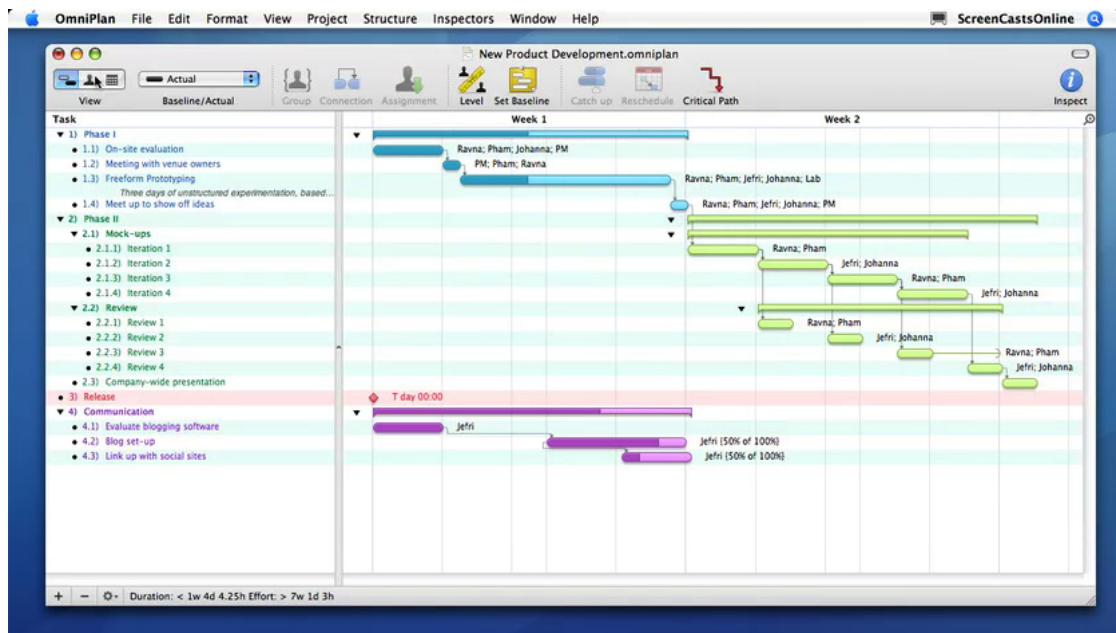
Εικόνα 0.6: Παράδειγμα συστήματος GanttProject³⁷

3.5.2 OmniPlan

Είναι ένα απλό πληροφοριακό σύστημα διοίκησης έργων που δίνει στους χρήστες αυτού την δυνατότητα δημιουργίας διαγραμμάτων Gantt, κατανομής πόρων και έλεγχου προϋπολογισμού. Μέσα στις ικανότητες του είναι ο ορισμός οροσήμων, χρονοδιαγραμμάτων, κοστών και πόρων. Το περιβάλλον του συστήματος φαίνεται στην εικόνα 3.7. Τα χαρακτηριστικά που υποστηρίζει 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, html.)

³⁷ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.



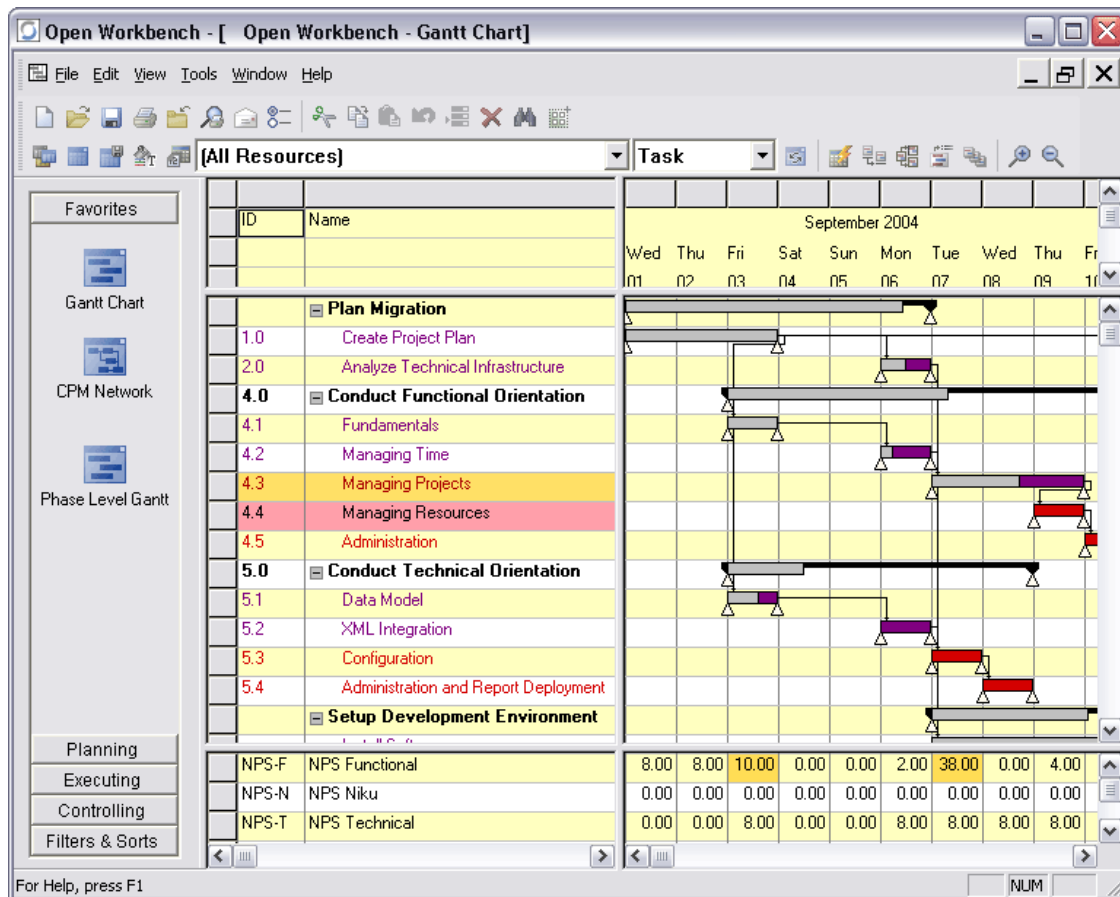
Εικόνα 0.7: Παράδειγμα συστήματος OmniPlan.³⁸

3.5.3 Open Workbench

Αποτελεί ένα πληροφοριακό σύστημα διοίκησης έργων ανοιχτού κώδικα που δίνει την δυνατότητα χρονοπρογραμματισμού και διοίκησης έργου. Εφόσον το σύστημα είναι ανοιχτού κώδικα έχει όλα τα πλεονεκτήματα αυτών συστημάτων όπως για παράδειγμα εύκολη παραμετροποίηση, γρήγορη ενσωμάτωση των τεχνολογικών εξελίξεων, 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. Το περιβάλλον του συστήματος φαίνεται στην εικόνα 3.8. Τα βασικά χαρακτηριστικά που υποστηρίζει το open workbench είναι:

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

³⁸ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.



Εικόνα 0.8: Παράδειγμα συστήματος OpenWorkbench.³⁹

3.5.4 Pertmaster

Θα μπορούσαμε να πούμε ότι μοιάζει περισσότερο σαν ένα *add-on* σύστημα υποστήριξης αποφάσεων στα ήδη υπάρχοντα συστήματα *Microsoft Project* και *Primavera*, παρά ένα αυτόνομο σύστημα διοίκησης έργου. Παρέχει την δυνατότητα καθορισμού διαστημάτων εμπιστοσύνης, όσον αφορά το κόστος του έργου και τον χρονοπρογραμματισμό του, μαζί με εναλλακτικά σχέδια αντιμετώπισης κινδύνου.

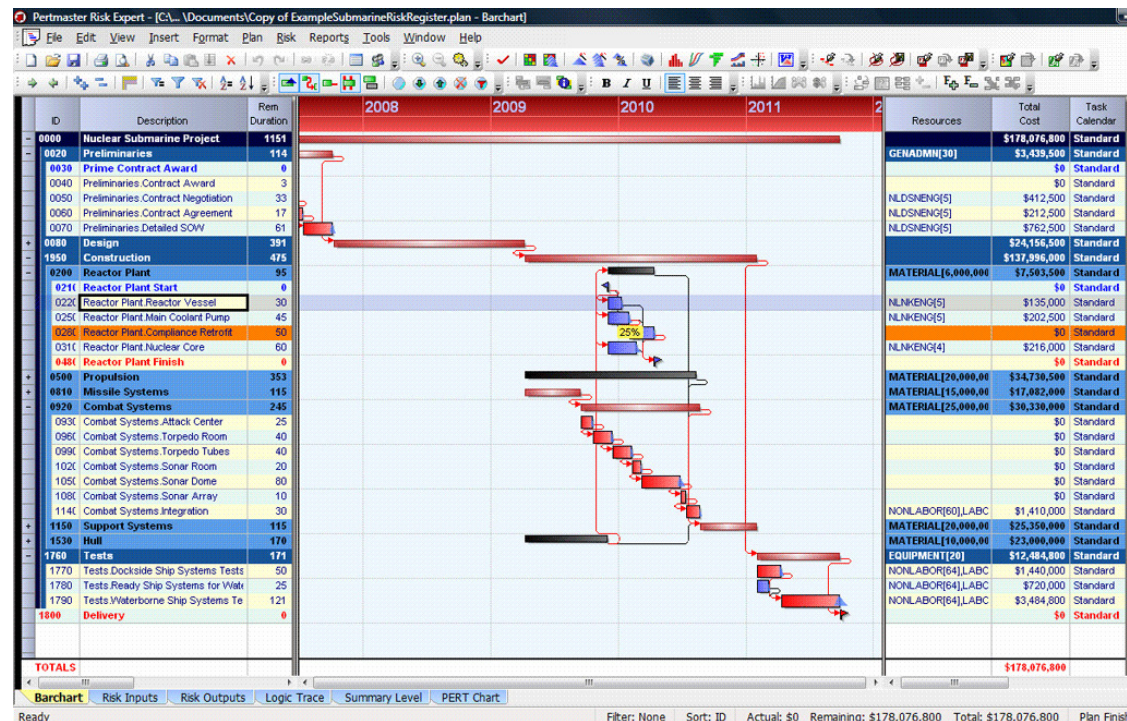
Τα σχέδια αυτά υπολογίζουν την πιθανότητα απόκλισης του έργου όσον αφορά το χρόνο και το κόστος, και αναλύουν την αποτελεσματικότητα των σχεδίων αντιμετώπισης. Αυτός ο συνδυασμός αποτελεί την βάση του χρονοπρογραμματισμού και είναι προσαρμοσμένος στον κίνδυνο, που σήμερα

³⁹ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

αποτελεί τον κύριο κανόνα στις διαδικασίες σχεδιασμού και χρονοπρογραμματισμού.

Το κάθε έργο μέσω προκαθορισμένων περιοχών κινδύνου κατατάσσεται σε κάποια κατηγορία και στην συνέχεια αναλύεται μέσω προσομοίωσης με την μέθοδο Monte Carlo. Διαστήματα εμπιστοσύνης και σημεία κινδύνου μπορούν να αναγνωριστούν μέσω ιστογραμμάτων ρίσκου και διαγράμματα διασποράς πριν ο κίνδυνος που προκύψει ενσωματωθεί στον χρονοπρογραμματισμό του έργου. Το περιβάλλον του συστήματος φαίνεται στην εικόνα 3.9. Τα βασικά χαρακτηριστικά του *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*.



Εικόνα 0.9: Παράδειγμα συστήματος *Pertmaster*.⁴⁰

⁴⁰ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

3.5.5 PlanningForce

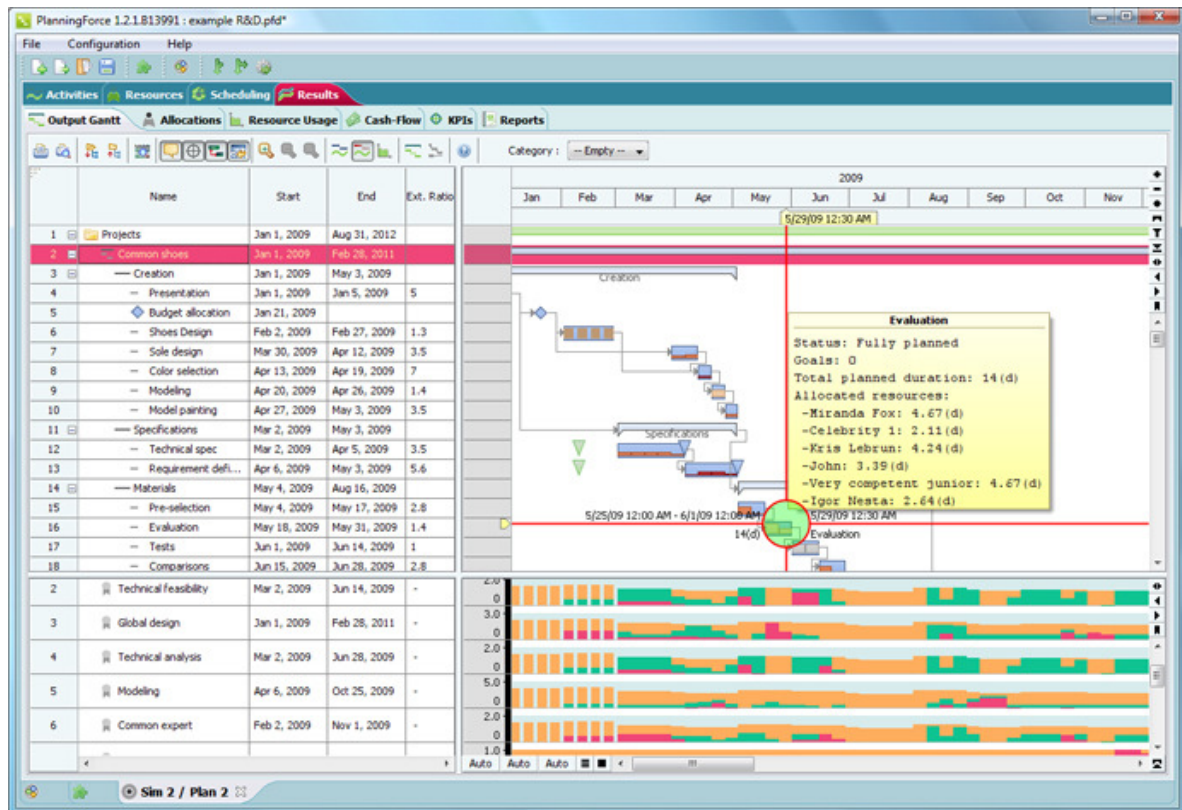
Είναι ένα πληροφοριακό σύστημα διοίκησης έργων με δυνατότητες συστήματος διαχείρισης αποφάσεων με βασικό χαρακτηριστικό γνώρισμα τον συνδυασμό διοίκησης έργου, διαχείρισης πόρων και προγραμματισμού έργων για την επίτευξη συγκεκριμένων στόχων.

Ο μηχανισμός προγραμματισμού έργων εκτιμά τον χρόνο εκτέλεσης του έργου λαμβάνοντας υπόψη στόχους, κινδύνους, χρονικές αποκλίσεις που εξαρτώνται από διακοπές, εποχικότητα, ζήτηση και αποτελεσματικότητα.

Τα χαρακτηριστικά που υποστηρίζει δίνουν την δυνατότητα στο χρήστη να εκπονεί το στρατηγικό σχεδιασμό ενσωματώνοντας ευελιξία, εκτίμηση κόστους, χρόνου και πόρων για κάθε είδους έργου. Ο χρήστης μπορεί να ορίσει πόρους, προϋπολογισμό και διάρκεια και το πρόγραμμα να αυτοματοποιεί όλες τις περαιτέρω διαδικασίες σχεδιασμού και ελέγχου. Βάσει των λειτουργιών του 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.

Στην εικόνα 3.10 φαίνεται το περιβάλλον του συστήματος.



Εικόνα 0.10: Παράδειγμα συστήματος PlanningForce.⁴¹

3.5.6 Microsoft Project

Το πρόγραμμα αυτό είναι εξοπλισμένο με εργαλεία διοίκησης έργου που προσφέρουν στον χρήστη τη δυνατότητα: ελέγχου κόστους, δημιουργίας αναφορών προόδου, χρήσης έτοιμων προτύπων, χρονοπρογραμματισμού καθώς και 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.

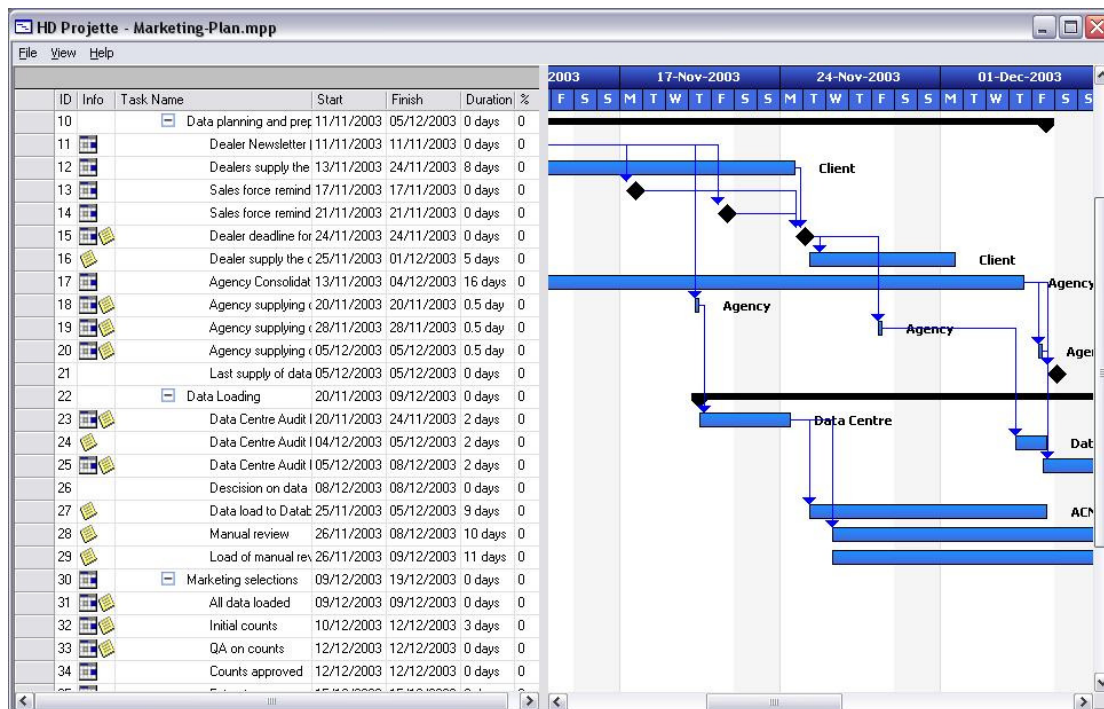
Η χρήση προγραμμάτων οδήγησης εργασιών διευκολύνουν το να προσδιορισθεί ο παράγοντας βάσει του οποίου καθορίζεται η ημερομηνία έναρξης της εργασίας,

⁴¹ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

ώστε να παρακολουθούνται όλοι οι παράγοντες δίνοντας έτσι τη δυνατότητα ανακάλυψης της αιτίας μίας συγκεκριμένης καθυστέρησης. Μερικά από τα χαρακτηριστικά του 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.*

Ακολουθεί εικόνα, όπου απεικονίζεται το περιβάλλον του συστήματος.



Εικόνα 0.11: Παράδειγμα συστήματος Microsoft Project.⁴²

3.5.7 Achievo

Είναι ένα web-based πληροφοριακό σύστημα διοίκησης έργων πολύ ευέλικτο. Οι δυνατότητες διαχείρισης πόρων και διεργασιών του διευκολύνουν τις εταιρείες να υποστηρίξουν τις εταιρικές διαδικασίες με ένα τρόπο απλό αλλά συνάμα αποτελεσματικό. Τα κύρια χαρακτηριστικά του είναι:

- Εξολοκλήρου web based 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.

⁴² Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

Η αρχιτεκτονική του Achieno είναι τμηματοποιημένη με μία κεντρική βάση δεδομένων. Συνδυασμένα τα επιμέρους τμήματα διασφαλίζουν τη βέλτιστη χρησιμοποίηση εφαρμογών intranet για εταιρικά περιβάλλοντα.

Υποστηρίζει μόνο εφαρμογές οι οποίες θεωρούνται αξιόπιστες και μπορεί να χρησιμοποιηθεί-εφαρμοστεί-αναπτυχθεί σε κάθε περιβάλλον δικτύου. Το πρόγραμμα Achieno είναι βασισμένο σε διακομιστή και ανεξάρτητο από το σύστημα, συνεπώς μπορεί να αναπτυχθεί και σε ήδη υπάρχοντα δίκτυα.

Πέρα από τα κύρια χαρακτηριστικά που αναφέραμε, το Achieno αποτελεί ένα σύνολο εφαρμογών με επίκεντρο την διοίκηση έργου που επιπροσθέτως περιλαμβάνει και δυνατότητες διαχείρισης πελατών και ανθρωπίνων πόρων. Όσον αφορά καθαρά τη διοίκηση έργου, το Achieno δίνει την δυνατότητα δημιουργίας Pert και Gantt διαγραμμάτων, στατιστικά στοιχεία παρακολούθησης του έργου, υποστήριξη προτύπων για φάσεις και δραστηριότητες, καθώς και ορισμό οροσήμων.

Επίσης δύναται η χρήση ημερολογίου, είτε προσωπικού είτε ομάδων εργασίας, και η διαχείριση επαφών που σχετίζονται με τα έργα. Επιπροσθέτως, μπορεί να υπάρξει διοίκηση έργου κατά πελάτη ή κατά προμηθευτή, διαχείριση υπαλλήλων, ρόλων και συμβάσεων εργαζομένων. Τέλος, υπάρχει η δυνατότητα εύκολης δημιουργίας αναφορών. Το περιβάλλον του συστήματος φαίνεται στην εικόνα που ακολουθεί.

Home | User preferences | Logout | Search

Achievo

Logged in as: demo1

- Time registration
- Scheduler
- Reports
 - Time Survey
 - Employee week report
 - Projectstatus
- Projects
- Sales
- Organizations
- Employees
- Setup
- Support

Parameters

Select which hours you want to view:

Name: User 1, Demo
Functionlevel: All functionlevels

Period: Monday 28 September 2009 until Friday 9 October 2009
Status: All


Customise the output:
Output type: Standard output
Show details:
Order by: Period

Refresh

Weekreport: In Period 2009-09-28 T/m 2009-10-11

Name: User 1, Demo
Email:

Period	Normal	Sunday	Overtime compensation	Total	Contract	Overtime	Balance	Status
--------	--------	--------	-----------------------	-------	----------	----------	---------	--------



Weekreport - Report

Εικόνα 0.12: Παράδειγμα συστήματος Achievo.⁴³

⁴³ Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι ευέλικτες μέθοδοι μοιράζονται ένα παρόμοιο σύνολο βασικών αξιών και αρχών και πιστεύουν στην ύπαρξη μίας μόλις ικανοποιητικά προσδιορισμένης μεθοδολογίας.

Παρόλα αυτά, κάθε μέθοδος προσεγγίζει τα προβλήματα που αντιμετωπίζονται στην διοίκηση έργων πληροφορικής από μια διαφορετική οπτική σκοπιά. Ο πίνακας 4.1 συνοψίζει κάθε μέθοδο χρησιμοποιώντας τρεις πτυχές της:

1. τα κύρια σημεία,
2. τα ειδικά χαρακτηριστικά γνωρίσματα
3. και τις αναγνωρισμένες ελλείψεις.

Όπως είναι εμφανές, τα κύρια σημεία αναφέρονται στις κύριες πτυχές και στις λύσεις των μεθόδων. Τα ειδικά χαρακτηριστικά γνωρίσματα περιγράφουν διεξοδικά μια ή περισσότερες πτυχές των μεθόδων που τις διαφοροποιούν από άλλες. Τέλος, οι αναγνωρισμένες ελλείψεις αφορούν μερικές πτυχές μιας μεθόδου που λείπουν ή άλλες πτυχές που δεν έχουν τεκμηριωθεί επαρκώς.

Ο ακραίος προγραμματισμός είναι μία μέθοδος που προσανατολίζεται ως επί το πλείστον στην πρακτική εφαρμογή. Περιέχει διάφορες εμπειρικά επικυρωμένες από τους επαγγελματίες πρακτικές που είναι χρήσιμες. Προσφέρεται για ομάδες μικρού μεγέθους, μεταξύ τριών και είκοσι ατόμων. Απαραίτητη προϋπόθεση για την εφαρμογή του ΧΡ είναι και το κατάλληλο εργασιακό κλίμα διότι ακόμη και μια μικρή αντίσταση στην αλλαγή είτε από την διοίκηση, είτε από τους συμμετόχους ή τον πελάτη μπορεί να οδηγήσει το πρόγραμμα σε αποτυχία.

Επίσης, το ΧΡ δεν προτείνεται όταν η χρησιμοποιούμενη τεχνολογία δεν μπορεί να υποστηρίξει ομαλά τις αλλαγές ή όταν χρειάζεται πολύ καιρός για να έχουμε αποτελέσματα και ανατροφοδότηση.

Το Scrum είναι μια προσέγγιση διοίκησης του προγράμματος που στηρίζεται στις αυτορυθμιζόμενες ανεξάρτητες ομάδες ανάπτυξης για την εφαρμογή ενός προγράμματος λογισμικού στους κύκλους 30-ημέρων. Η διαδικασία Scrum περιλαμβάνει τρεις φάσεις: πριν το παιχνίδι (Pre game) , το παιχνίδι (game) και μετά το παιχνίδι (Post game)

Το Scrum προτείνεται για οποιοδήποτε μέγεθος προγράμματος, αλλά η ομάδα του Scrum μπορεί να είναι μεταξύ πέντε και εννιά ατόμων. Για τα μεγάλα προγράμματα σχηματίζονται πολλαπλές ομάδες Scrum.

Η οικογένεια των «Κρυστάλλινων» μεθοδολογιών περιλαμβάνει έναν αριθμό από διαφορετικές μεθοδολογίες για να επιλέγεται κάθε φορά η καταλληλότερη μεθοδολογία για κάθε πρόγραμμα. Οι κύριοι άξονες στους οποίους βασίστηκε η εξέλιξη αυτής της μεθόδου είναι οι άνθρωποι, οι μεταξύ τους αλληλεπιδράσεις, οι ικανότητες, το ταλέντο και η επικοινωνία. Η μέθοδος υποστηρίζει ότι τα στοιχεία που προαναφέρθηκαν είναι ο παράγοντες που έχουν την κύρια επίδραση στην επίδοση των έργων.

Το FDD στρέφεται σε μια απλή προσέγγιση πέντε-βημάτων, η οποία είναι βασισμένη στον προσδιορισμό, το σχεδιασμό και την εφαρμογή των χαρακτηριστικών γνωρισμάτων. Το FDD προϋποθέτει ότι κάποια εργασία για το πρόγραμμα έχει γίνει ήδη και με τον τρόπο αυτό δεν καλύπτει τις πρώτες φάσεις ενός προγράμματος.

Το DSDM ξεχωρίζει από τις άλλες μεθόδους λόγω του ότι χρησιμοποιεί την διαμόρφωση πρωτοτύπου. Το μειονέκτημα της μεθόδου αυτής είναι η απαίτηση να ανήκει κανείς στην κοινοπραξία του DSDM προκειμένου να έχει μια πρόσβαση στη βιβλιογραφία που συζητά τις διαφορετικές πτυχές της μεθόδου. Η DSDM μπορεί να χρησιμοποιηθεί το ίδιο εύκολα σε μικρά και σε μεγάλα έργα υπό τον όρο ότι αυτά τμηματοποιούνται. Επιπλέον συνίσταται και για πολύπλοκες επιχειρησιακές εφαρμογές.

Πίνακας 0.1: Γενικά χαρακτηριστικά γνωρίσματα των ευέλικτων μεθόδων διαχείρισης έργων πληροφορικής.

	Κύρια σημεία	Ειδικά χαρακτηριστικά γνωρίσματα	Αναγνωρισμένες ελλείψεις
XP	<ul style="list-style-type: none"> • Προσαρμοσμένη ανάπτυξη στις ανάγκες του πελάτη, • μικρές ομάδες, • καθημερινές προσαυξήσεις του κώδικα . 	Αναδόμηση - ο συνεχής επανασχεδιασμός του συστήματος για να βελτιώσει τη απόδοση του και την ανταπόκρισή του στις αλλαγές.	Ενώ οι μεμονωμένες πρακτικές είναι κατάλληλες για πολλές καταστάσεις, στο σύνολο του συστήματος και στις διοικητικές πρακτικές δίνεται η λιγότερη προσοχή.
Scrum	<ul style="list-style-type: none"> • Ανεξάρτητες , μικρές, αυτορυθμιζόμενες ομάδες ανάπτυξης, • κύκλοι απελευθέρωσης 30-ημέρων. 	Επιβάλει τη μετατόπιση από τον «καθορισμένο και επαναλαμβανόμενο» τρόπο ανάπτυξης στο «νέο τρόπο ανάπτυξης προϊόντων, Scrum.»	Ενώ το Scrum δίνει συγκεκριμένες λεπτομέρειες για το πώς διαχειρίζεται η απελευθέρωση των 30-ημέρων , η ολοκλήρωση και δοκιμές αποδοχής δεν είναι λεπτομερής.
Crystal	<ul style="list-style-type: none"> • Μια οικογένεια μεθόδων. Καθεμία έχει τον ίδιο πυρήνα από αρχές αλλά οι τεχνικές, οι ρόλοι, τα εργαλεία και τα πρότυπα ποικίλλουν. 	Δυνατότητα να επιλέξει κανείς την καταλληλότερη μέθοδο βασισμένη μέγεθος προγράμματος και την κρισιμότητα του.	Πάρα πολύ νωρίς για να αναθρέφουν: Μόνο δύο από τις τέσσερις προτεινόμενες μεθόδους υπάρχουν.
FDD	<ul style="list-style-type: none"> • Διαδικασία πέντε-βημάτων, • αντικειμενοστραφής ανάπτυξη βασισμένη στα χαρακτηριστικά γνωρίσματα. • Πολύ σύντομες επαναλήψεις: από ώρες έως 2 εβδομάδες. 	Η απλότητα της μεθόδου, ο σχεδιασμός και η εφαρμογή του συστήματος από τα χαρακτηριστικά γνωρίσματα, η διαμόρφωση αντικειμένων.	Το FDD εστιάζει μόνο επάνω στο σχεδιασμό και την εφαρμογή. Υπάρχει ανάγκη και για άλλες ενισχυτικές προσεγγίσεις.
DSDM	<ul style="list-style-type: none"> • Εφαρμογή για τον έλεγχο του RAD, • χρήση της τμηματοποίησης του χρόνου, • ενεργή κοινοπραξία για την ανάπτυξη μεθόδου 	Πρώτη αληθινά ευέλικτη μέθοδος ανάπτυξης λογισμικού , χρήση των διαμορφώσεων πρωτοτύπου, αρκετοί νέοι ρόλοι: «πρεσβευτής», «οραματιστής» και «σύμβουλος».	Ενώ η μέθοδος είναι διαθέσιμη, μόνο μέλη της κοινοπραξίας έχουν πρόσβαση στα έγγραφα που δηλώνουν την πραγματική χρήση της μεθόδου.

Η μεθοδολογία Information Technology Infrastructure Language, συλλέγοντας τις καλύτερες τεχνικές στοχεύει στο να βελτιώσει και να συντηρήσει ένα ορισμένο επίπεδο στην ποιότητα υπηρεσιών πληροφορικής. Η μέθοδος καλύπτει τις IT απαιτήσεις ενός οργανισμού μέσω ενός συνόλου διεργασιών που οργανώνουν τις IT

λειτουργίες του. Σχεδιάστηκε για το δημόσιο αλλά και για τον ιδιωτικό τομέα διεθνώς.

Τώρα σε ότι αφορά τις τεχνικές διοίκησης έργων που πέρα από τα έργα πληροφορικής μπορούν να εφαρμοστούν και σε άλλους τομείς καταλήγουμε στα εξής συμπεράσματα:

Η μέθοδος-διαδικασία διοίκησης έργων της TenStep είναι μια μεθοδολογία για τη διαχείριση της εργασίας σε ένα έργο. Έχει ως σκοπό να είναι τόσο ευέλικτη όσο χρειάζεται ώστε να διοικηθεί αποτελεσματικά ένα έργο.

Η PRINCE2 αποτελεί την καταλληλότερη μεθοδολογία διοίκησης έργων, ιδιαίτερα όταν πρόκειται για έργα μεγάλης κλίμακας ή κρίσιμης σημασίας. Η μεθοδολογία PRINCE2 υιοθετεί τις αρχές της δομημένης διαχείρισης, δηλαδή της διοίκησης του έργου με έναν λογικό, κατάλληλα οργανωμένο τρόπο, ακολουθώντας συγκεκριμένα διαδικαστικά «βήματα» συστηματικής προσέγγισης και με μια τεκμηριωμένη περιγραφή.

Η μέθοδος PMBOK ή αλλιώς Project Management Body of Knowledge είναι μια συλλογή από διαδικασίες και γνωστικές περιοχές, που θεωρούνται ως οι πλέον καλύτερες τεχνικές για την διοίκηση. Ως ένα διεθνώς αναγνωρισμένο πρότυπο (IEE 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 μέθοδος Hermes χρησιμοποιείται για να διοικεί έργα στον τομέα της τεχνολογίας, της πληροφορικής και των επικοινωνιών (Information and Communication Technologies-ICT). Η μέθοδος αυτή είναι ένα ανοιχτό πρότυπο και αντιμετωπίζει όλες τις πραγματικές απαιτήσεις ενός έργου ICT. Τα περισσότερα επιτυχημένα έργα χρησιμοποιούν στοχοκεντρική προσέγγιση, εκτέλεση και έλεγχο. Η μέθοδος Hermes προτείνει διαδικασίες οι οποίες ακολουθούν προσέγγιση στόχου αλλά και αποτελέσματος.

Συμπεραίνουμε λοιπόν ότι για οποιοδήποτε έργο πληροφορικής, μικρό ή μεγάλο είναι απαραίτητη η χρήση κάποιας μεθόδου διαχείρισης και διοίκησης είτε ευέλικτη είτε παραδοσιακή έτσι ώστε να επιτευχθεί ο στόχος μας με όσο το δυνατό μεγαλύτερη ακρίβεια, ευκολία και χαμηλό κόστος για τους επενδυτές.

BIBΛΙΟΓΡΑΦΙΑ

1. Τμήμα Μηχανικών Η/Υ και Πληροφορικής, *Κατανεμημένα Συστήματα II, Σημειώσεις «Μεθοδολογίες ταχείας ανάπτυξης λογισμικού»*, Πανεπιστήμιο Πατρών, 2009.
2. Somerville I., «Βασικές Αρχές Τεχνολογίας Λογισμικού», Εκδόσεις Κλειδάριθμος, 2009.
3. arantis D., Charalabidis Y., and Askounis D., “A Goal-Driven Management Approach based on Knowledge Exploitation for e-Government Projects”, *International Journal of Electronic Government Research, Volume 6, Issue 3, July 2010.*
4. Τζιτζικας Γ., *Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων, «Συλλογή Απαιτήσεων»*, Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών.
5. Πάγκαλος Γ., *Δομημένος Προγραμματισμός, «Αρχές και Τεχνικές του Δομημένου Προγραμματισμού»*, Σημειώσεις Παραδόσεων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.
6. Stapleton J., “Dynamic systems development method – The method in practice, Addison Wesley”, 1997.
7. Palmer S. R., and Felsing J. M., “A Practical Guide to Feature-Driven Development”, Upper Saddle River, NJ, Prentice-Hall, 2002.
8. Schwaber K., and Beedle M., “Agile Software Development With Scrum”, Upper Saddle River, NJ, Prentice-Hall, 2008.
9. Pekka, Salo, Outi, Ronkainen, Jussi & Warsta, Juhani, “Agile software development methods”, VVT Publications, 2002.
10. Δημητριάδης Α., «Διοίκηση – Διαχείριση Έργου», Εκδόσεις Νέων Τεχνολογιών, Έκδοση 4^η, 2009.
11. Καντζαρη Μ., «Μαθηματικά των υπολογιστών και των αποφάσεων - μοντέλα για τον χρονοπρογραμματισμό έργων με περιορισμένους πόρους», Πανεπιστήμιο Πατρών, 2010.
12. Πολύζος Σ., «Διοίκηση και Διαχείριση των Έργων – Μέθοδοι και Τεχνικές» Εκδόσεις Κριτική, 1η έκδοση, 2004.
13. «Προγραμματισμός έργων», Διοίκηση Παραγωγής και Συστημάτων Υπηρεσιών, Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης, Εθνικό Μετσόβειο Πολυτεχνείο.

14. Τουρίκη Α., «Ανάπτυξη Εργαλείου Διοίκησης Έργων Ηλεκτρονικής Διακυβέρνησης», Εθνικό Μετσόβειο Πολυτεχνείο, 2010.

ΙΣΤΟΣΕΛΙΔΕΣ

1. http://el.wikipedia.org/wiki/Διοίκηση_και_Διαχείριση_Έργου
2. <http://www.gatech.gr/pm.html> Τελευταία πρόσβαση: Ιούνιος 2011.
3. <http://www.hermes.admin.ch/ictΔprojectΔmanagement/the-hermes-method>
Τελευταία πρόσβαση: Ιούνιος 2011.
4. http://en.wikipedia.org/wiki/PRINCE2#Description_of_the_PRINCE2_method
Τελευταία πρόσβαση: Αύγουστος 2011.