

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΩΝ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΟΙΚΟΝΟΜΙΑ
ΤΜΗΜΑ: ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

Πτυχιακή Εργασία

«Διαχείριση Έργων Πληροφορικής – Μελέτη των σύγχρονων Τεχνικών και Εργαλείων»



Όνομ/μο Φοιτήτριας: Παπαδοθωμάκου Ελένη

Επιβλέπων Καθηγητής: ΣΠ. Β. ΒΑΣΙΛΕΙΑΔΗΣ

ΠΑΤΡΑ 2010

**Την παρούσα εργασία την
αφιερώνω στην οικογένειά μου**



Πρόλογος

Είναι γνωστό ότι οι πρώτες εμπορικές εφαρμογές ξεκίνησαν στα τέλη της δεκαετίας του 50, όπου και οι υπολογιστές εκτελούσαν επαναληπτικές, μεγάλου όγκου, υπολογίσιμες πράξεις που αφορούσαν τις στοιχειώδεις κινήσεις υποστήριξης των βασικών επιχειρησιακών διαδικασιών, δηλαδή το επίπεδο λειτουργίας ενός οργανισμού. Στην πορεία οι διαδικασίες αυτές ονομάστηκαν *δοσοληψίες- συναλλαγές*. Μια κατηγορία πληροφοριακών συστημάτων που παρέχουν διαδικασίες για την καταγραφή και την παραγωγή πληροφοριών σχετικών με κάποιες δοσοληψίες είναι τα Συστήματα Επεξεργασίας Συναλλαγών. Τυπικά συστήματα της κατηγορίας αυτής είναι η διαχείριση αποθήκης, επεξεργασία λογιστικών εφαρμογών, η διαχείρισης ανθρώπινων πόρων και άλλα. Τα συστήματα επεξεργασίας συναλλαγών υποστηρίζουν τη συλλογή, αποθήκευση επεξεργασία και διασπορά των βασικών επιχειρησιακών συναλλαγών. Επίσης παρέχουν δεδομένα εισόδου για πολλές εφαρμογές, όπως είναι τα Συστήματα υποστήριξης Αποφάσεων συλλέγουν τας δεδομένα συνήθως σε καθημερινή βάση ή σε πραγματικό χρόνο, τα οποία αποθηκεύουν σε εταιρικές βάσεις δεδομένων για επεξεργασία. Συχνά σε ένα οργανισμό υπάρχουν πολλά συστήματα επεξεργασίας συναλλαγών, τα οποία θεωρούνται κεντρικής σημασίας για την επιτυχή λειτουργία του οργανισμού εφόσον υποστηρίζουν βασικές λειτουργίες.

Η μείωση του κόστους των υπολογιστών και η αύξηση της υπολογιστικής ισχύος κατέστησαν δυνατή τη χρήση της τεχνολογίας της πληροφορικής για λιγότερο επαναληπτικές πράξεις. Τη δεκαετία του 60' (εξήντα) αναπτύχθηκε μια νέα γενιά πληροφοριακών συστημάτων που μπορεί κανείς να έχει πρόσβαση, να οργανώνει την εργασία του, να ομαδοποιεί καταστάσεις παρουσιάζοντας τις ανάλογες πληροφορίες που απαιτούνται για τη στήριξη της λήψης αποφάσεων αναφορικά με τις βασικές λειτουργίες οργανισμό. Πρόκειται για τα Πληροφοριακά Συστήματα Διοίκησης (M.I.S) που απευθύνονται στα μεσαία διοικητικά στελέχη και χαρακτηρίζονται από την ικανότητά τους να παράγουν περιοδικές αναφορές για μεγέθη που αντικατοπτρίζονται με την αποδοτικότητα, την αποτελεσματικότητα και την παραγωγικότητα του οργανισμού. Αρχικά τα MIS, είχαν ιστορικό προσανατολισμό περιγράφοντας τα γεγονότα που έχουν συμβεί, ενώ αργότερα



χρησιμοποιήθηκαν για να προβλέψουν τάσεις, να υποστηρίξουν αποφάσεις και απαντήσεις σε ερωτήματα.

Τα υποστηρικτικά συστήματα άρχισαν να κάνουν την εμφάνιση τους στα τέλη της δεκαετίας του εξήντα και στις αρχές της δεκαετίας του εβδομήντα όταν αρχίζει να επικρατεί η ηλεκτρονική επικοινωνία και να εξαπλώνεται η διαδικτυακή υποδομή. Η ηλεκτρονική επικοινωνία είναι μόνο το ένα χαρακτηριστικό των συστημάτων που είναι γνωστά ως Συστήματα Αυτοματισμού Γραφείου (Office Automation Systems O.S.S). Άλλο ένα, χαρακτηριστικό είναι τα συστήματα επεξεργασίας κειμένου, που άρχισαν να εξαπλώνονται στους οργανισμούς της δεκαετίας του εβδομήντα ενώ, παράλληλα, άρχισε η εισαγωγή των υπολογιστών στον κατασκευαστικό τομέα με εφαρμογές ρομποτικής, σχεδίου και κατασκευών.

Μέχρι το τέλος της δεκαετίας του 70, μεγάλωναν οι απαιτήσεις για νέους τύπους εφαρμογών που υποστηρίζονται από τις Τεχνολογίες της Πληροφορίας. Οι αυξημένες δυνατότητες των υπολογιστών και τα μειωμένα κόστη δικαιολογούν την ανάδειξη εφαρμογών διαφορετικής φύσεως όπως τα συστήματα στήριξης αποφάσεων. Η εξάπλωση των προσωπικών υπολογιστών, που μπορούσαν εύκολα να προγραμματιστούν κατέστησε δυνατή την κατασκευή εφαρμογών της υποστήριξης αποφάσεων, ακόμη και από άτομα που δεν γνωρίζουν προγραμματισμό. Τα συστήματα αυτά αξιοποιούν δεδομένο και μαθηματικά μοντέλα και βοηθούν στην επίλυση μη δομημένων και ημιδομημένων προβλημάτων που εμφανίζονται στην κορυφή της διοικητικής πυραμίδας.

Η υποστήριξη αποφάσεων επεκτάθηκε προς δύο κατευθύνσεις. Πρώτον, σχεδιάστηκαν συστήματα που απευθύνονται σε υψηλόβαθμα στελέχη, τα οποία εν συνεχεία επεκτάθηκαν για να υποστηρίξουν όλους τους μάνατζερ μέσα στον οργανισμό και ονομαστικά Επιχειρησιακά Πληροφοριακά Συστήματα. Η δεύτερη κατεύθυνση αφορά την υποστήριξη ανθρώπων που εργάζονται σε ομάδες και συμμετέχουν στη λήψη αποφάσεων και μάλιστα μπορεί να βρίσκονται σε διαφορετικές τοποθεσίες. Τα συστήματα αυτής της κατηγορίας ονομάστηκαν Συστήματα Υποστήριξης Ομάδων.



Γύρω στα μέσα της δεκαετίας του 80 άρχισε η εμπορευματοποίηση των εφαρμογών τεχνητής νοημοσύνης. Εφαρμογές τεχνητής νοημοσύνης που παρουσιάζουν ιδιαίτερο ενδιαφέρον για τους οργανισμούς είναι τα έμπειρα συστήματα, τα οποία είναι συστήματα που δεν στηρίζονται στη συναλλαγή αν και χρησιμοποιούν τη συναλλαγή με δεδομένα. Στηρίζοντας στην προσπάθεια «άντλησης» γνώσεων από την εμπειρία και την δεξιοτεχνία ενός ή περισσότερων εμπειρογνομόνων ενός γνωστικού χώρου, με τέτοιο τρόπο ώστε να είναι δυνατή η εξαγωγή λογικών συμπερασμάτων. Τα συστήματα αυτά παρέχουν την συσσωρεύομε γνώση των εμπειρογνομόνων στους μη εμπειρογνώμονες ώστε να μπορούν να λύνουν δύσκολα προβλήματα.

Τα συστήματα διαχείρισης γνώσης είναι τα συστήματα που σχεδιάστηκαν ειδικά για να δημιουργούν, συγκεντρώνουν, οργανώνουν και διαχέουν την γνώση ενός οργανισμό σε αντίθεση με τα δεδομένα και τις πληροφορίες. Το λογισμικό που υποστηρίζει αυτά τα συστήματα συνδέει ανθρώπους με έγγραφα και πληροφορίες και ανθρώπους με ανθρώπους. Επίσης, συλλαμβάνει νέες πληροφορίες και της μετατρέπει σε καινούργια γνώση που ωθεί τον οργανισμό προς τους στόχους του. Τα συστήματα διαχείρισης γνώσης δεν αποτελούν απλά το εσωτερικό και το εξωτερικό περιβάλλον ενός οργανισμού, αλλά επικεντρώνονται στη συλλογή, οργάνωση και αποτελεσματική διανομή της συσσωρευμένης γνώσης του οργανισμού.



Πίνακας Περιεχομένων

| | |
|---------------|---|
| Εισαγωγή..... | 8 |
|---------------|---|

Κεφάλαιο Πρώτο

Εισαγωγικές Έννοιες Διαχείρισης Έργων (Project Management)

| | |
|--|----|
| 1.1 Ανάλυση Βασικών Ορισμών..... | 10 |
| 1.2 Ιστορία Διαχείρισης Έργου..... | 11 |
| 1.3 Περιεχόμενο Διαχείρισης Έργων..... | 13 |

Κεφάλαιο Δεύτερο

Διαχείριση Έργου

| | |
|--|----|
| 2.1 Εισαγωγικά | 18 |
| 2.2 Σχεδιασμός έργων..... | 19 |
| 2.3 Σχεδιασμός Έργου..... | 20 |
| 2.4 Η Σπουδαιότητα του Χρονικού Προγραμματισμού στην Διαχείριση Έργων Πληροφορικής..... | 22 |
| 2.5 Διαχείριση Κινδύνων..... | 24 |

Κεφάλαιο Τρίτο

Σχεδιασμός – Έλεγχος – Ανάπτυξη Έργων Πληροφορικής

| | |
|--|----|
| 3.1 Αρχιτεκτονική Σχεδίασης..... | 30 |
| 3.2 Έλεγχος και Αναφορές | 33 |
| 3.2.1 Αρχιτεκτονική Αναφοράς..... | 35 |
| 3.3 Ανάπτυξη | 36 |
| 3.3.1 Γρήγορη Ανάπτυξη..... | 39 |
| 3.3.2 Κατασκευή Προτύπων Λογισμικού..... | 40 |



Κεφάλαιο Τέταρτο

Επαλήθευση και Επικύρωση

| | | |
|-----|--|----|
| 4.1 | Σχεδιασμός Επαλήθευσης & Επικύρωσης..... | 45 |
| 4.2 | Δοκιμή Λογισμικού..... | 47 |
| 4.3 | Επικύρωση Κρίσιμων Συστημάτων | 48 |

Κεφάλαιο Πέμπτο

Ο ρόλος της Διαχείρισης

| | | |
|-----|---------------------------|----|
| 5.1 | Διαχείριση Ανθρώπων..... | 52 |
| 5.2 | Κόστος..... | 54 |
| 5.3 | Διαχείριση Ποιότητας..... | 57 |
| 5.4 | Διαχείριση Ασφάλειας..... | 58 |

Κεφάλαιο Έκτο

Έρευνα – Συμπεράσματα

| | | |
|-----|--------------------|----|
| 6.1 | Έρευνα..... | 62 |
| 6.2 | Συμπεράσματα | 72 |



Πίνακας Εικόνων – Σχημάτων

| | |
|--|----|
| Εικόνα 1: Διάγραμμα Gantt..... | 13 |
| Εικόνα 2: Δομή ισορροπημένου πίνακα..... | 15 |
| Εικόνα 3: Πλάνο Έργου..... | 22 |
| Εικόνα 4: Χρονοπρογραμματισμός έργου με διαφορετικά project..... | 24 |
| Εικόνα 5: Block Diagrams | 32 |
| Εικόνα 6: Η αρχιτεκτονική του μοντέλου OSI..... | 36 |
| Εικόνα 7: Τηλεδιάσκεψη | 55 |
| | |
| Πίνακας 1: Τύποι Πλάνων..... | 21 |
| Πίνακας 2: Μερικά παραδείγματα των παραπάνω κινδύνων..... | 26 |
| | |
| Σχήμα 1: Ορόσημα στη διαδικασία των απαιτήσεων..... | 23 |
| Σχήμα 2: Η διαδικασία διαχείρισης κινδύνων..... | 25 |
| Σχήμα 3: Το μοντέλο ελέγχου κλήσης – επιστροφής..... | 35 |
| Σχήμα 4: Περιβάλλον γρήγορης ανάπτυξης..... | 40 |
| Σχήμα 5: Συγκριτική Δοκιμή..... | 43 |
| Σχήμα 6: Η διαδικασία μέτρησης της αξιοπιστίας..... | 50 |



Εισαγωγή

Στις μέρες μας σχεδόν όλοι οι οργανισμοί, δημόσιοι ή ιδιωτικοί, χρησιμοποιούν διάφορους τύπους πληροφοριακών συστημάτων, συμπεριλαμβανομένου το ηλεκτρικό εμπόριο για να υποστηρίξουν τις λειτουργίες τους. Αυτό συμβαίνει γιατί η τεχνολογία της πληροφορικής αποτελεί σημαντικό καταλύτη για τις επιχειρηματικές δραστηριότητες παγκοσμίως. Η τεχνολογία της πληροφορικής θα είναι επίσης καταλύτης θεμελιωδών αλλαγών στη δομή, τις λειτουργίες για τη διαχείριση των οργανισμών. Οι δυνατότητες αυτές υποστηρίζουν τους πέντε επιχειρηματικούς αντικειμενικούς στόχους: (1) Βελτίωση παραγωγικότητας, (2) μείωση διάφορων τύπων κόστους, (3) βελτίωση λήψης αποφάσεων, (4) ενίσχυση σχέσεων με πελάτες, (5) ανάπτυξη νέων στρατηγικών εφαρμογών.

Στην παρούσα πτυχιακή εργασία με τίτλο «Διαχείριση έργων πληροφορικής. Μελέτη των σύγχρονων τεχνικών και εργαλείων», πρόκειται να γίνει μια εκτενής ανάλυση των έργων πληροφορικής.

Η εργασία αποτελείται από έξι κεφάλαια. Στο πρώτο κεφάλαιο ο αναγνώστης εισάγεται στις βασικές έννοιες διαχείρισης έργων. Το δεύτερο κεφάλαιο αναφέρεται στις δραστηριότητες διαχείρισης, τον σχεδιασμό, τον χρονικό προγραμματισμό και την διαχείριση κινδύνων. Το τρίτο κεφάλαιο ασχολείται με την αρχιτεκτονική σχεδίασης των έργων πληροφορικής, τον έλεγχο και τις αναφορές που γίνονται στο σύστημα και την διαδικασία της ανάπτυξης. Η επαλήθευση και η επικύρωση (E&E) είναι από τα βασικά κομμάτια της διαχείρισης έργων και αναλύονται στο τέταρτο κεφάλαιο. Επίσης, αναλύονται και οι διαδικασίες του σχεδιασμού, της δοκιμής αλλά και την διαδικασία της επικύρωσης των κρίσιμων συστημάτων. Το πέμπτο κεφάλαιο με τίτλο 'ο ρόλος της διαχείρισης', ασχολείται με την διαχείριση του προσωπικού που λαμβάνει μέρος στα έργα πληροφορικής, τον ρόλο του κόστους και της ποιότητας. Τέλος, στο έκτο κεφάλαιο γίνεται μια έρευνα σε μια επιχείρηση και εξάγονται τα συμπεράσματα.



Κεφάλαιο Πρώτο

Εισαγωγικές Έννοιες Διαχείρισης Έργων (Project Management)



Κεφάλαιο Πρώτο

Εισαγωγικές Έννοιες Διαχείρισης Έργων (Project Management)

1.1 Ανάλυση Βασικών Ορισμών

Στο πρώτο κεφάλαιο με τίτλο «Εισαγωγικές έννοιες διαχείρισης έργων (project management)», πρόκειται να αναφερθούν οι βασικότερες έννοιες που σχετίζονται με την έννοια της διαχείρισης έργων.

Η έννοια **έργο (Project)** και διαχείριση έργου (Project Management) χρησιμοποιούνται ευρέως από επιχειρήσεις κάθε κλάδου (βιομηχανία, εμπόριο, κατασκευές). Η **έννοια του έργου** αφορά σε κάθε δραστηριότητα μιας επιχείρησης ή ενός οργανισμού και έχει ξεκάθαρο στόχο την παραγωγή ή ολοκλήρωση ενός μοναδικού προϊόντος ή υπηρεσίας, σε συγκεκριμένη χρονική διάρκεια και ημερομηνίες αρχής και τέλους (Σπύρογλου, 2005).

Ένας ορισμός που θα μπορούσε να δοθεί για τη διαχείριση έργου είναι ο ακόλουθος:

Διαχείριση Έργου (Project Management - PM) μπορεί να οριστεί μια οργανωμένη προσέγγιση για το χειρισμό της διαδικασίας εκτέλεσης και ολοκλήρωσης διαφόρων τύπων έργων

Για να γίνει πιο κατανοητή η έννοια του έργου, παρατίθενται κάποια παραδείγματα έργων.

- Ο σχεδιασμός και η παραγωγή ενός νέου μοντέλου αυτοκινήτου (Βιομηχανία)
- Η εκστρατεία διαφήμισης και προβολής μιας εταιρείας (Επιχειρήσεις)
- Η κατασκευή ενός κτιρίου ή εγκαταστάσεων (Κατασκευές)
- Η εγκατάσταση ενός νέου πληροφοριακού συστήματος σε έναν οργανισμό ή εταιρεία (Πληροφορική)

Πιο αναλυτικά παρουσιάζεται ως παράδειγμα έργου η Κατασκευή της γέφυρας Ρίου –Αντιρίου:



Κατασκευή Γέφυρας Ρίου- Αντιρρίου

Χρονοπρογραμματισμός

(Πότε;)

Κοστολόγηση

(Πόσο;)

Ανάθεση σε ανθρώπινο δυναμικό

(Ποιος θα κάνει τι;)

Διαχείριση Πόρων

(τι θα χρησιμοποιηθεί που;)

Έλεγχος Ποιότητας

(αυτό που κατασκευάζουμε πληροί τις προδιαγραφές;)

Διαχείριση Κινδύνου

(αν πάει κάτι λάθος πως θα το αντιμετωπίσουμε;)



Ένας άλλος ορισμός που θα μπορούσε να δοθεί στην έννοια του έργου είναι ο ακόλουθος:

«Ως **έργο** ορίζεται ένα προσωρινό εγχείρημα που στοχεύει στη δημιουργία ενός μοναδικού προϊόντος, υπηρεσίας ή αποτελέσματος»

Ένα έργο μπορεί να χωρίζεται σε υπο- έργα. Το υπο-έργο: αναφέρεται σε ειδικές περιοχές όπως για παράδειγμα η κατασκευή Αεροπλάνου έχει ως υπο- έργα την άτρακτος του αεροσκάφους, τους κινητήρες, το λογισμικό, τους εσωτερικοί χώροι κλπ

1.2 Ιστορία Διαχείρισης Έργου

Όπως αναφέρθηκε και παραπάνω η διοίκηση και διαχείριση έργων (project management) αναπτύχθηκε ως ξεχωριστό γνωστικό πεδίο από την εφαρμογή των αρχών της διοίκησης αλλά και της επιχειρησιακής έρευνας σε διάφορους τομείς εφαρμογής, όπως αυτός των κατασκευών, της μηχανολογίας, των μεγάλων



Henry Gantt
1861 - 1919



στρατιωτικών προγραμμάτων.

Πατέρας του γνωστικού πεδίου της διαχείρισης έργων θεωρείται ο Χένρι Γκαντ, Αμερικανός μηχανικός και κοινωνικός επιστήμονας, ο οποίος εισήγαγε τις αρχές του προγραμματισμού και ελέγχου στη διαχείριση έργων. Το γνωστό διάγραμμα Gantt, είναι ένα ραβδόγραμμα που παρουσιάζει τις δραστηριότητες του έργου παίρνοντας την ονομασία του από τον Henry Gantt . Ο Γκαντ μαζί με τον Φρέντερικ Τέιλορ έθεσαν τις θεμέλιες αρχές της διαχείρισης έργων και ο Τέιλορ τις αρχές της επιστημονικής διαχείρισης (scientific management).



**Frederick Winslow
Taylor
1856 - 1915**

Οι σύγχρονες αρχές της διαχείρισης έργων οι οποίες έκαναν τη διαχείριση έργων ένα διακριτό γνωστικό αντικείμενο αλλά και ένα επάγγελμα αναπτύχθηκαν την δεκαετία του 1950.

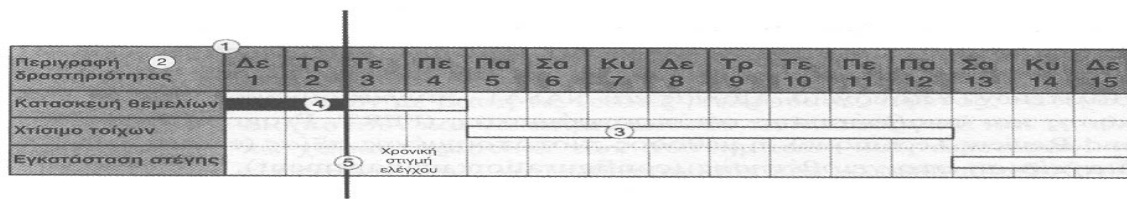
Την δεκαετία αυτή αναπτύχθηκαν δύο βασικά μαθηματικά μοντέλα χρονοπρογραμματισμού δραστηριοτήτων, οι μέθοδοι PERT και CPM οι οποίες αποτέλεσαν σταθμό στη διαχείριση έργων.

Η μέθοδος PERT (Program Evaluation and Review Technique) αναπτύχθηκε από το Ναυτικό των Ηνωμένων Πολιτειών. Αντίστοιχα η μέθοδος CPM (Critical Path Method) γνωστή στα ελληνικά και ως μέθοδος κρίσιμου διαδρομής αναπτύχθηκε από τις εταιρείες DuPont Corporation και Remington Rand Corporation με σκοπό την διαχείριση έργων συντήρησης. Η διάδοση και αποδοχή των μεθόδων αυτών έγινε με ταχύτατο τρόπο έτσι ώστε σήμερα αποτελούν βασικές μεθόδους για τη διαχείριση έργων.

Σήμερα ο χώρος της διαχείρισης έργων θεωρείται ιδιαίτερα αναπτυσσόμενος και προσελκύει ιδιαίτερο ενδιαφέρον τόσο στον ιδιωτικό, δημόσιο τομέα όσο και στην ακαδημαϊκή κοινότητα. Απόδειξη του γεγονότος αυτού αποτελεί η ύπαρξη πολλών και ιδιαίτερα δραστήριων διεθνών οργανισμών που έχουν ως στόχο την ανάπτυξη του γνωστικού πεδίου της διαχείρισης έργων. Μεταξύ αυτών ξεχωρίζουν οι International Project Management Association



Εικόνα 1: Διάγραμμα Gantt



Πηγή: Αδάμ Κ., 2007

1.3 Περιεχόμενο Διαχείρισης Έργων

Η διοίκηση έργων λειτουργεί σε ένα γενικότερο περιβάλλον. Είναι σημαντικό για τον υπεύθυνο κάθε έργου, να κατανοεί το περιβάλλον αυτό ώστε να μπορεί να λειτουργήσει αποτελεσματικότερα. Ο υπεύθυνος πρέπει να έχει πάντα υπόψη στο σχεδιασμό τα παρακάτω :

- Φάσεις του έργου, κύκλος ζωής του, παραδοτέα
- Εμπλεκόμενοι στο έργο (Πελάτες, συνεργάτες, προμηθευτές, μέλη της ομάδας)
- Εταιρική κουλτούρα και δομή (της εταιρείας που αναλαμβάνει αλλά και αυτής που αναθέτει το έργο)
- Ικανότητες διοίκησης και οργάνωσης
- Κοινωνικοοικονομικές επιδράσεις (εξωτερικά γεγονότα)

Στάδια έργου

Η διαδικασία της διοίκησης έργων μπορεί να οργανωθεί σε 5 μεγάλες ομάδες ενεργειών και δραστηριοτήτων βρίσκοντας εφαρμογή τόσο συνολικά στο έργο όσο και σε κάθε φάση αυτού.

Οι ομάδες αυτές είναι:

- *Εκκίνηση: διαδικασία έναρξης του έργου. Διαπίστωση αναγκών, αποδοχή, δέσμευση στους στόχους.*
- *Σχεδιασμός: δημιουργία οργανωτικού πλαισίου και σχεδίου ενεργειών.*
- *Εκτέλεση: συντονισμός ανθρώπων και άλλων πόρων για την ολοκλήρωση.*
- *Έλεγχος: διασφάλιση ποιότητας και τήρηση των υποχρεώσεων, διορθωτικές ενέργειες αν είναι απαραίτητες.*



- *Κλείσιμο: ουσιαστικό και τυπικό κλείσιμο με επίσημη παράδοση και αποδοχή.*

Ο υπεύθυνος του έργου όσο πιο ρεαλιστικά και πραγματοποιήσιμα προγραμματίσει τις επιμέρους φάσεις τόσο πιο πιθανό είναι το έργο να ολοκληρωθεί με επιτυχία και εντός χρόνου και κόστους.

Παράδειγμα η εγκατάσταση ενός ERP σε μια Α.Ε. στον κλάδο των τροφίμων με τζίρους άνω των 4 δισ., προσωπικό 300 ατόμων, 2 εργοστάσια, απομακρυσμένα σημεία πώλησης, οργανωμένα τμήματα (λογιστήριο, παραγωγή, μηχανογράφηση) είναι κάτι εντελώς διαφορετικό από την εγκατάσταση του ιδίου ERP σε μια μικρή Ο.Ε. (οικογενειακού χαρακτήρα) στον κλάδο ανταλλακτικών αυτοκινήτων με τζίρο 500 εκ., 1 αποθήκη – έδρα, και προσωπικό 4 – 5 ατόμων.

Εμπλεκόμενοι στο έργο

Ως εμπλεκόμενοι στο έργο μπορούν να θεωρηθούν εταιρείες και το προσωπικό υλοποίησης του έργου. Ο πελάτης, ο υπεύθυνος έργου, η ομάδα υλοποίησης, η εταιρεία η οποία ανέλαβε την υλοποίηση, οι χρηματοδότες (αν είναι εξωτερικοί οργανισμοί) έχουν όλοι ανάμιξη σε κάποια από τις φάσεις του έργου.

Στην εγκατάσταση ενός πληροφοριακού συστήματος μπορούμε να διαχωρίσουμε τον πελάτη (Η εταιρεία ή οργανισμός η οποία και αναθέτει το έργο), τον χρηματοδότη – ελεγκτή (εποπτεύει την σωστή απορρόφηση των κονδυλίων), τον οργανισμό υλοποίησης (η διοίκηση της εταιρείας) και τον υπεύθυνο ή την ομάδα εγκατάστασης.

Εταιρική κουλτούρα και δομή

Τα έργα που εκτελούνται μέσα σε μια εταιρεία αποτελούν μέρος της ευρύτερης δραστηριότητας της. Η «εταιρική κουλτούρα» μιας επιχείρησης είναι ένα δύσκολο κομμάτι για κάθε επιχείρηση – οργανισμό μιας και διαμορφώνεται στην πορεία. Αν και για τα στελέχη οι κουλτούρα δεν είναι εύκολη υπόθεση παρόλα αυτά κάθε στέλεχος της εταιρείας θα πρέπει να γνωρίζει αυτή την κουλτούρα και να συμμορφώνεται προς αυτή.

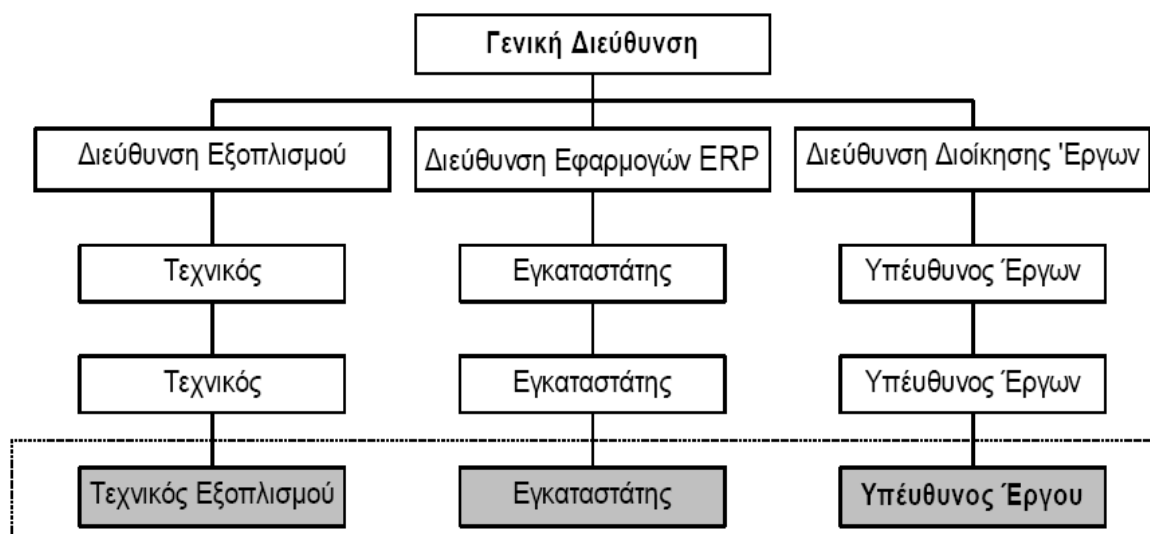


Παράδειγμα της εταιρικής κουλτούρας είναι και ο τρόπος συμπεριφοράς και συνεργασίας μεταξύ των στελεχών της. Σε μια μικρότερη και ευέλικτη επιχείρηση πληροφορικής το κλίμα είναι συνήθως πιο χαλαρό και φιλικό από ότι σε έναν μεγάλο επιχειρηματικό όμιλο όπου εκ των πραγμάτων επικρατούν πιο αυστηροί και γραφειοκρατικοί κώδικες επικοινωνίας.

| Δομή εταιρείας | Κατά Λειτουργίες (Functional) | Δομή πίνακα (matrix) | | | Κατά έργο (Projectized) |
|--|--------------------------------------|-----------------------------------|----------------------------------|------------------|-------------------------|
| | | Χαλαρός | Ισορροπημένος | Ισχυρός | |
| Διοίκηση έργων | | | | | |
| Εξουσία Υπευθύνου έργου | Μικρή ή καθόλου | Περιορισμένη | Μέτρια | Μέτρια ως αρκετή | Αρκετή ως πλήρης |
| Προσωπικό σε απασχόληση κατά έργο (επί συνόλου προσωπικού) | 0 | 0-25% | 15-60% | 50-95% | 85-100% |
| Απασχόληση υπευθύνου έργου | Μερική | | Πλήρης | | |
| Τίτλος | Συντονιστής έργου / Επικεφαλής | Υπεύθυνος έργου | Υπεύθυνος προγράμματος | | |
| Αγγλικός Όρος | Project Coordinator / Project Leader | Project Manager / Project Officer | Project Manager/ Program Manager | | |

Οι περισσότερες επιχειρήσεις πληροφορικής ακολουθούν μια ισορροπημένη δομή πίνακα, όπου υπάρχουν τα τμήματα βάση των λειτουργιών και των δραστηριοτήτων της επιχείρησης. Στο ακόλουθο οργανόγραμμα φαίνεται μια τυπική δομή μιας επιχείρησης πληροφορικής.

Εικόνα 2: Δομή ισορροπημένου πίνακα



Πηγή: Σπυρογλου, 2005



Ικανότητες Διοίκησης και Οργάνωσης

Ο υπεύθυνος έργου πρέπει να καταφέρει να συνδυάσει τις απαιτήσεις με τις δεσμεύσεις και υποχρεώσεις της εταιρείας ώστε να προκύψει το καλύτερο αποτέλεσμα.

Οι Ικανότητες και γνώσεις που είναι απαραίτητες για έναν υπεύθυνο έργων πληροφορικής είναι οι παρακάτω:

- Γνώσεις και εμπειρία της λειτουργίας των επιχειρήσεων (Οικονομικά, Λογιστικά, Εργασιακά)
- Ικανότητα ανάλυσης προβλημάτων και σχεδιασμού
- Επικοινωνιακές ικανότητες (προφορικά και γραπτά)
- Γνώσεις και ικανότητα οργάνωσης προσωπικού και παρακίνησης ομάδας
- Ικανότητες διαχείρισης κρίσεων, διαπραγμάτευσης και γρήγορης λήψης αποφάσεων
- Γνώσεις και ικανότητα του σύγχρονου περιβάλλοντος στην πληροφορική και τις επιχειρήσεις
- Ηγετικές ικανότητες



Κεφάλαιο Δεύτερο

Διαχείριση Έργου



Κεφάλαιο Δεύτερο

Διαχείριση Έργου

2.1 Εισαγωγικά

Η διαχείριση έργων λογισμικού θα μπορούσαμε να πούμε ότι είναι ένα ουσιώδες μέρος της τεχνολογίας λογισμικού. Η καλή διαχείριση δεν μπορεί να εγγυηθεί την επιτυχία ενός έργου αντίθετα η κακή διαχείριση συνήθως οδηγεί στην αποτυχία του έργου.

Οι διευθυντές έργων λογισμικού είναι υπεύθυνοι για το σχεδιασμό και το χρονικό προγραμματισμό της ανάπτυξης ενός έργου. Επιβλέπουν την εργασία ώστε να διασφαλίσουν ότι πραγματοποιείται σύμφωνα με τις επιθυμητές προδιαγραφές και παρακολουθούν την πρόοδο ώστε να εξασφαλίσουν ότι η ανάπτυξη πραγματοποιείται μέσα στα καθορισμένα χρονικά και οικονομικά πλαίσια.

Ο ρόλος της διαχείρισης είναι πολύ σημαντικός επειδή η επαγγελματική παραγωγή λογισμικού υπόκειται πάντοτε σε οργανωτικούς περιορισμούς προϋπολογισμού και χρόνου. Η δουλειά του διευθυντή ενός έργου λογισμικού είναι να διασφαλίσει ότι το έργο λογισμικού θα ικανοποιεί αυτούς τους περιορισμούς και θα παραδώσει λογισμικό που συμβάλλει στους στόχους της εταιρίας που το αναπτύσσει. Οι διευθυντές έργων λογισμικού κάνουν την ίδια δουλειά με τους διευθυντές άλλων έργων. Αντίθετα η τεχνολογία λογισμικού διαφέρει από τους άλλους τεχνικούς κλάδους σε ορισμένα σημεία. Μερικές από τις διαφορές είναι:

- § Το προϊόν δεν είναι απτό
- § Δεν υπάρχουν τυποποιημένες διαδικασίες παραγωγής λογισμικού
- § Τα μεγάλα έργα λογισμικού συχνά είναι έργα «της μίας φοράς»

Εξαιτίας αυτών των προβλημάτων, δεν αποτελεί έκπληξη το γεγονός ότι μερικά έργα λογισμικού καθυστερούν, υπερβαίνουν τον προϋπολογισμό και παραβιάζουν το χρονοδιάγραμμα.



Στο δεύτερο κεφάλαιο πρόκειται να αναλυθούν οι βασικές δραστηριότητες διαχείρισης, ο σχεδιασμός, η σπουδαιότητα του χρονικού προγραμματισμού και οι κίνδυνοι διαχείρισης.

2.2 Σχεδιασμός έργων

Η τυπική περιγραφή της εργασίας του διευθυντή έργου λογισμικού είναι αδύνατος να διατυπωθεί. Η εργασία διαφέρει ριζικά ανάλογα με τον οργανισμό και το έργο λογισμικού που αναπτύσσεται. Ωστόσο, οι περισσότεροι διευθυντές έργων σε κάποιο στάδιο αναλαμβάνουν την ευθύνη για ορισμένες ή για όλες τις ακόλουθες δραστηριότητες:

- § Σύνταξη πρότασης
- § Σχεδιασμός και χρονικός προγραμματισμός έργου
- § Κοστολόγηση έργου
- § Παρακολούθηση έργου και επιθεωρήσεις
- § Επιλογή και αξιολόγηση προσωπικού
- § Σύνταξη αναφορών και παρουσιάσεις

Σύνταξη πρότασης

Το πρώτο στάδιο ενός έργου λογισμικού μπορεί να περιλαμβάνει τη σύνταξη μιας πρότασης με σκοπό τη σύναψη σύμβασης για την πραγματοποίηση της εργασίας. Η πρόταση περιγράφει τους σκοπούς του έργου και το πώς θα υλοποιηθούν. Συνήθως περιλαμβάνει εκτιμήσεις κόστους και χρόνου και αιτιολογεί γιατί η σύμβαση του έργου θα πρέπει να παραχωρηθεί σε ένα συγκεκριμένο οργανισμό ή ομάδα.

Σχεδιασμός και χρονικός προγραμματισμός έργου

Ο σχεδιασμός έργου αφορά τον προσδιορισμό των δραστηριοτήτων, των ορόσημων, και των παραδοτέων προϊόντων ενός έργου. Καταστρώνεται ένα πλάνο που θα καθοδηγεί την ανάπτυξη προς τους στόχους του έργου

Κοστολόγηση έργου

Η εκτίμηση του κόστους είναι μια σχετική δραστηριότητα που αφορά την εκτίμηση των πόρων που απαιτούνται για την υλοποίηση του πλάνου του έργου.



Παρακολούθηση έργου και επιθεωρήσεις

Η παρακολούθηση έργου είναι μια συνεχής δραστηριότητα. Ο διευθυντής του έργου πρέπει να παρακολουθεί την πρόοδο του και να συγκρίνει πραγματική πρόοδο και το πραγματικό κόστος με τα προβλεπόμενα. Κατά τη διάρκεια ενός έργου, συνηθίζεται να γίνονται μια σειρά από τυπικές της διαχείρισης του έργου. Αυτές σχετίζονται με την πρόοδο και της τεχνική ανάπτυξη του έργου, οι οποίες ελέγχουν επίσης ανήκουν στο έργο και οι στόχοι του οργανισμού που πληρώνει για το λογισμικό παραμένουν σε συμφωνία. Το αποτέλεσμα μιας επιθεώρησης μπορεί να είναι μια απόφαση ακύρωσης του έργου.

Επιλογή και αξιολόγηση προσωπικού

Οι διευθυντές έργων συνήθως είναι υπεύθυνοι να επιλέγουν τα άτομα που θα εργαστούν στο έργο τους. Στην ιδανική περίπτωση, θα υπάρχει διαθέσιμο ειδικευμένο προσωπικό με κατάλληλη πείρα για να δουλέψει στο έργο.

Σύνταξη αναφορών και παρουσιάσεις

Οι διευθυντές έργων είναι συνήθως υπεύθυνοι για την υποβολή αναφορών για το έργο τόσο στον πελάτη - οργανισμό όσο και στον ανάδοχο οργανισμό. Πρέπει να γράφουν λιτά, κατανοητά έγγραφα, που να αναδεικνύουν τις κρίσιμες πληροφορίες μέσω από τις λεπτομερείς αναφορές του έργου.

2.3 Σχεδιασμός Έργου

Η αποτελεσματική διαχείριση ενός έργου λογισμικού εξαρτάται από τον εκτενή σχεδιασμό της προόδου του έργου. Ο διευθυντής του έργου παίζει τον βασικότερο ρόλο καθώς πρέπει να προβλέπει τα προβλήματα που μπορεί να ανακύψουν και να ετοιμάζει ενδεχόμενες λύσεις γι' αυτά τα προβλήματα. Ένα πλάνο που έχει καταστρωθεί στην αρχή του έργου θα πρέπει να χρησιμεύει ως οδηγός για το έργο (εγχειρίδιο). Αυτό το αρχικό πλάνο θα πρέπει να είναι το καλύτερο δυνατό με δεδομένες τις διαθέσιμες πληροφορίες, και θα εξελίσσεται καθώς προχωρά το έργο και γίνονται διαθέσιμες καλύτερες πληροφορίες.



Ο διευθυντής έργου λογισμικού εκτός από το πλάνο έργου θα πρέπει να καταστρώσει και άλλους τύπους πλάνων. Οι τύποι αυτού περιγράφονται συνοπτικά:

Πίνακας 1: Τύποι Πλάνων

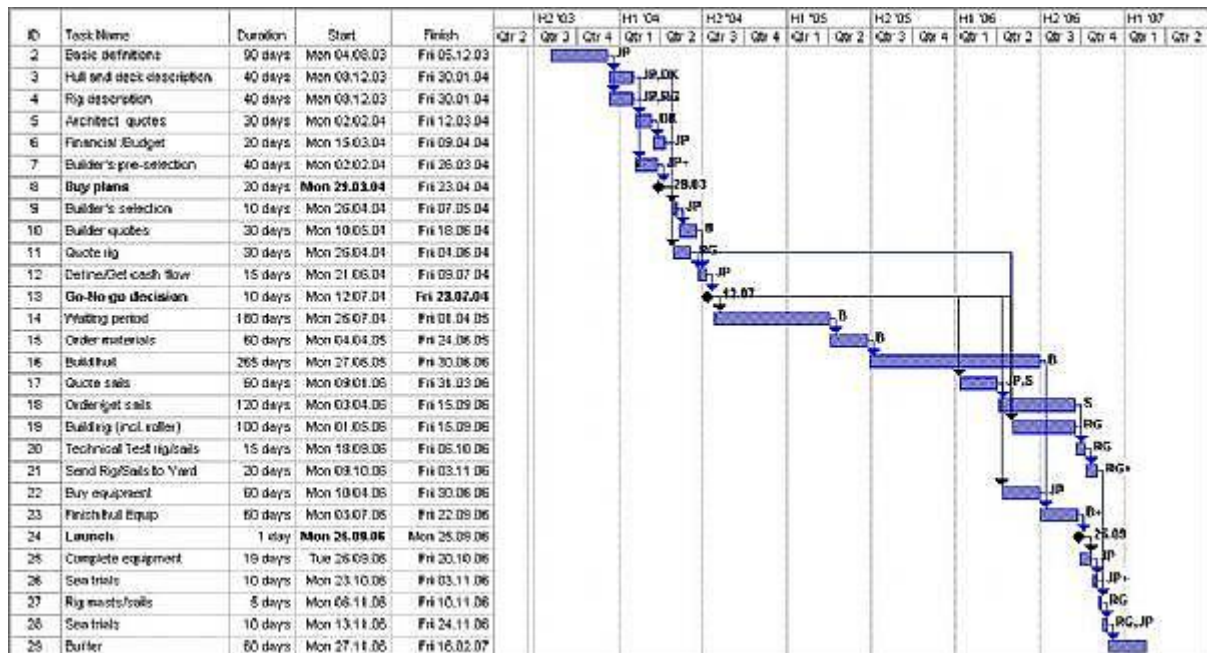
| Τύποι πλάνων | |
|---------------------------------------|--|
| Πλάνο ποιότητας | Περιγράφει τις διαδικασίες και τα πρότυπα ποιότητας που θα χρησιμοποιηθούν σε ένα έργο |
| Πλάνο επικύρωσης | Περιγράφει την προσέγγιση, τους πόρους, και το χρονοδιάγραμμα που χρησιμοποιούνται για την επικύρωση του συστήματος. |
| Πλάνο διαχείρισης διευθετήσεων | Περιγράφει τις διαδικασίες και τις δομές της διαχείρισης διευθετήσεων που θα χρησιμοποιηθούν |
| Πλάνο συντήρησης | Προβλέπει τις απαιτήσεις συντήρησης του συστήματος, το κόστος συντήρησης, και την απαιτούμενη εργασία. |
| Πλάνο ανάπτυξης προσωπικού | Περιγράφει πώς θα αναπτυχθούν οι δεξιότητες και η πείρα των μελών της ομάδας του έργου. |

Πλάνο έργου

Το πλάνο έργου (project plan) περιγράφει αναλυτικά τους πόρους που είναι διαθέσιμοι για το έργο, την κατάτμηση της εργασίας, καθώς και ένα χρονοδιάγραμμα για την πραγματοποίηση του έργου. Σε ορισμένους οργανισμούς, το πλάνο έργου είναι ένα μόνο έγγραφο που περιλαμβάνει τους διάφορους τύπους επιμέρους πλάνων.



Εικόνα 3: Πλάνο Έργου



Πηγή: <http://www.pslint.com/images/Project%20Plan.jpg>

Οι λεπτομέρειες του πλάνου έργου ποικίλλουν ανάλογα με τον τύπο του έργου και τον οργανισμό. Ωστόσο, τα περισσότερα πλάνα περιλαμβάνουν οπωσδήποτε τις ακόλουθες ενότητες:

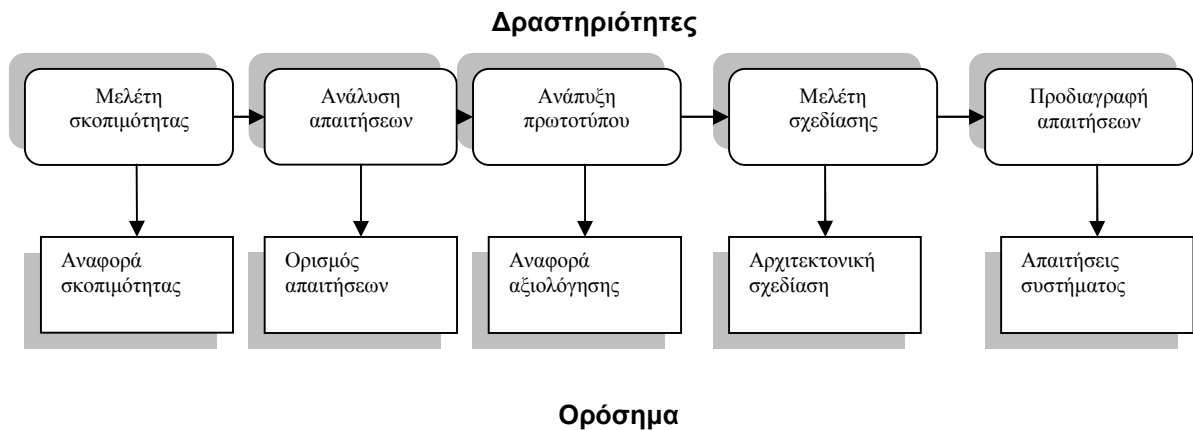
- § Εισαγωγή
- § Οργάνωση έργων
- § Ανάλυση κινδύνων
- § Απαιτήσεις σε πόρους υλικού και λογισμικού
- § Κατάμηση εργασίας
- § Χρονοδιάγραμμα έργου
- § Μηχανισμοί παρακολούθησης και αναφορών

2.4 Η Σπουδαιότητα του Χρονικού Προγραμματισμού στην Διαχείριση Έργων Πληροφορικής

Ο χρονικός προγραμματισμός έργου (project scheduling) είναι από τις δυσκολίες εργασίες για ένα διευθυντή έργου. Ο διευθυντής έργου εκτιμά το χρόνο και τους πόρους που απαιτούνται για την ολοκλήρωση των δραστηριοτήτων και τους οργανώνει σε μία συνεπή ακολουθία.



Σχήμα 1: Ορόσημα στη διαδικασία των απαιτήσεων



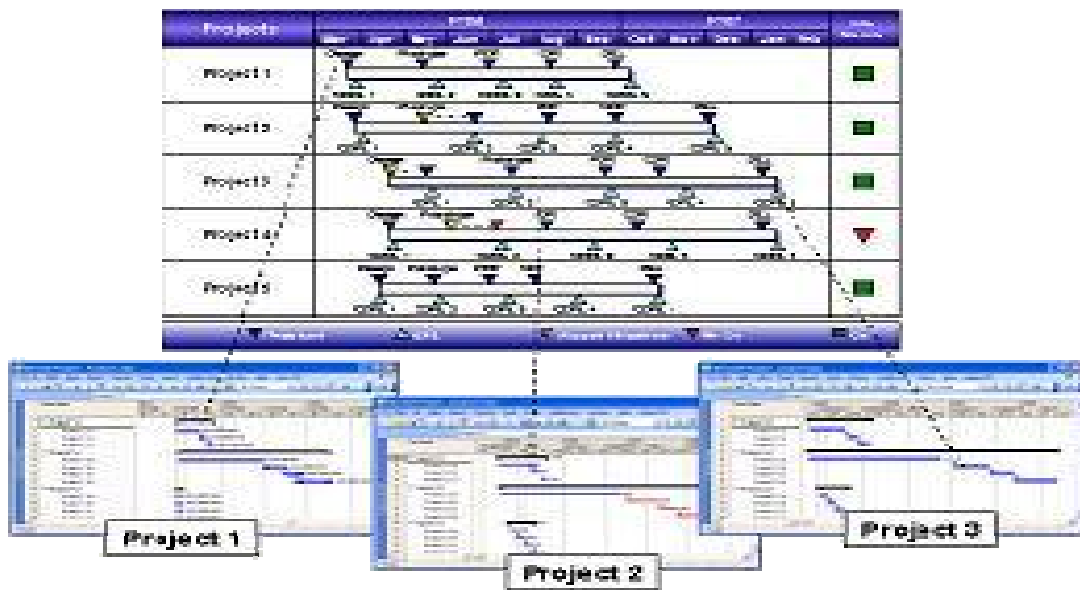
Εκτός από την περίπτωση όπου το προγραμματιζόμενο έργο είναι όμοιο με κάποιο προηγούμενο έργο, οι προηγούμενες εκτιμήσεις δεν είναι αξιόπιστες ως βάση για το χρονοπρογραμματισμό του νέου έργου. Οι εκτιμήσεις ενός χρονοδιαγράμματος μπορεί να περιπλέκονται ακόμα περισσότερο από το γεγονός ότι τα διαφορετικά έργα μπορεί να χρησιμοποιούν διαφορετικές μεθόδους σχεδίασης και γλώσσες υλοποίησης.

Αν το έργο είναι υψηλού τεχνολογικού επιπέδου, οι αρχικές εκτιμήσεις θα είναι σχεδόν σίγουρα αισιόδοξες ακόμη και αν προσπαθήσετε να λάβετε υπόψη σας όλα τα ενδεχόμενα. Από αυτή την άποψη, ο χρονοπρογραμματισμός του λογισμικού δε διαφέρει από το χρονοπρογραμματισμό οποιουδήποτε άλλου μεγάλου έργου υψηλού τεχνολογικού επιπέδου.

Ο προγραμματισμός έργου περιλαμβάνει το διαχωρισμό της συνολικής εργασίας του έργου σε ξεχωριστές δραστηριότητες και την εκτίμηση του χρόνου που απαιτείται για την ολοκλήρωση αυτών των δραστηριοτήτων. Συνήθως, κάποιες από αυτές τις δραστηριότητες εκτελούνται παράλληλα.



Εικόνα 4: Χρονοπρογραμματισμός έργου με διαφορετικά project



Πηγή: <http://www.engineering-intelligence.gr/admin/fckfiles/image/sem29.png>

2.5 Διαχείριση Κινδύνων

Η διαχείριση κινδύνων (risk management) θεωρείται ως μία από τις κύριες εργασίες του διευθυντή έργου. Ουσιαστικά Αφορά την πρόβλεψη των κινδύνων που μπορεί να επηρεάσουν το χρονοδιάγραμμα του έργου ή την ποιότητα του λογισμικού που αναπτύσσεται, καθώς και τις ενέργειες που χρειάζεται να γίνουν για την αποφυγή αυτών των κινδύνων (Hall, 1998 & Ould, 1999). Τα αποτελέσματα της ανάλυσης των κινδύνων θα πρέπει να τεκμηριώνονται στο πλάνο του έργου, μαζί με μια ανάλυση των συνεπειών των κινδύνων αν γίνουν πραγματικότητα. Η αποτελεσματική διαχείριση κινδύνων κάνει ευκολότερη την αντιμετώπιση των προβλημάτων και διασφαλίζει ότι δε θα οδηγήσουν σε μη αποδεκτές αποκλίσεις είτε του προϋπολογισμού είτε του χρονοδιαγράμματος.

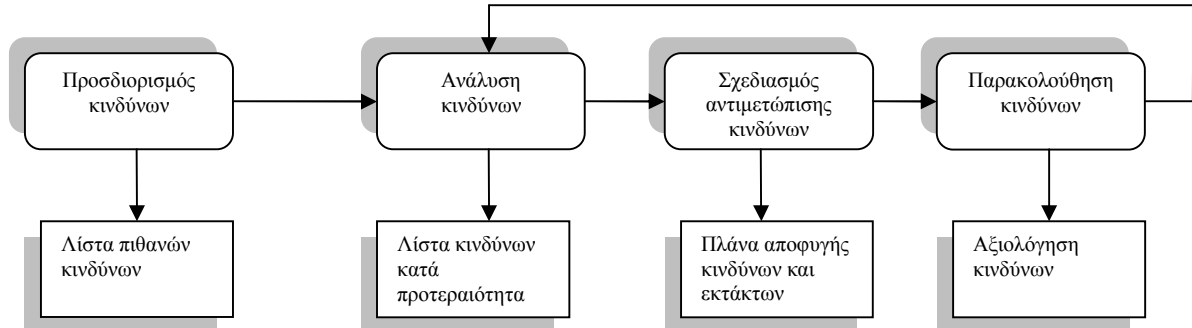
Ως κίνδυνος θεωρείται κάτι που θα ήταν προτιμότερο να μη συμβεί. Κίνδυνοι μπορεί να απειλούν το έργο, το λογισμικό που αναπτύσσεται, ή τον οργανισμό. Οι κατηγορίες κινδύνων είναι οι ακόλουθες:

1. Κίνδυνοι έργου
2. Κίνδυνοι προϊόντος
3. Επιχειρηματικοί κίνδυνοι



Μιας και οι κίνδυνοι είναι πιθανό να 'τινάξουν στον αέρα' το έργο θα πρέπει να γίνεται σωστή διαχείριση.

Σχήμα 2: Η διαδικασία διαχείρισης κινδύνων



Η διαχείριση κινδύνων είναι ιδιαίτερα σημαντική στα έργα λογισμικού, λόγω των εγγενών αβεβαιοτήτων που αντιμετωπίζουν τα περισσότερα έργα. Αυτές οι αβεβαιότητες προκύπτουν από χαλαρά ορισμένες απαιτήσεις, δυσκολίες στην εκτίμηση του χρόνου και των πόρων που απαιτούνται για την ανάπτυξη του λογισμικού, εξάρτηση από τις ικανότητες μεμονωμένων ατόμων, και αλλαγές απαιτήσεων που οφείλονται σε αλλαγές των αναγκών των πελατών.

Η διαδικασία διαχείρισης των κινδύνων, όπως και ο σχεδιασμός του έργου, είναι μια επαναληπτική διαδικασία η οποία συνεχίζεται σε όλη τη διάρκεια του έργου. Αφού καταστρωθούν τα αρχικά πλάνα, παρακολουθείται η κατάσταση. Καθώς γίνονται διαθέσιμες περισσότερες πληροφορίες για τους κινδύνους, οι κίνδυνοι χρειάζεται να αναλυθούν εκ νέου και να τεθούν νέες προτεραιότητες. Τα πλάνα αποφυγής κινδύνων και τα πλάνα εκτάκτων καταστάσεων μπορεί να τροποποιηθούν καθώς αναδύονται νέες πληροφορίες για τους κινδύνους.

Θα πρέπει να τεκμηριώνετε τα αποτελέσματα της διαδικασίας διαχείρισης κινδύνων σε ένα πλάνο διαχείρισης κινδύνων. Αυτό θα πρέπει να περιλαμβάνει μια περιγραφή των κινδύνων που αντιμετωπίζει το έργο, ανάλυση αυτών των κινδύνων, καθώς και τα πλάνα που απαιτούνται για τη διαχείρισή τους. Όπου χρειάζεται, θα πρέπει επίσης να εντάσσετε στο πλάνο και αποτελέσματα της διαδικασίας διαχείρισης κινδύνων, όπως συγκεκριμένα πλάνα εκτάκτων καταστάσεων που θα ενεργοποιηθούν αν πραγματοποιηθεί ο κίνδυνος.



Προσδιορισμός Κινδύνων

Ο προσδιορισμός των κινδύνων είναι το πρώτο στάδιο της διαχείρισης κινδύνων. Αφορά την ανακάλυψη των πιθανών κινδύνων για το έργο. Στο στάδιο αυτό δε θα πρέπει να γίνεται αξιολόγηση των κινδύνων ή να τίθενται προτεραιότητες, αν και στην πράξη οι κίνδυνοι με πολύ μικρές επιπτώσεις ή με πολύ μικρή πιθανότητα συνήθως δε λαμβάνονται υπόψη.

Ο προσδιορισμός των κινδύνων μπορεί να γίνει με τη μορφή ομαδικής προσπάθειας, με ανταλλαγή ιδεών, ή μπορεί απλώς να βασιστεί στην πείρα. Για διευκόλυνση της διαδικασίας, μπορεί να χρησιμοποιηθεί μια λίστα επιλογής που περιέχει διάφορους τύπους κινδύνων. Υπάρχουν τουλάχιστον έξι τύποι κινδύνων που μπορεί να ανακύψουν:

1. Τεχνολογικοί κίνδυνοι
2. Ανθρώπινοι κίνδυνοι
3. Οργανωτικοί κίνδυνοι
4. Κίνδυνοι εργαλείων
5. Κίνδυνοι απαιτήσεων
6. Κίνδυνοι εκτιμήσεων

Πίνακας 2: Μερικά παραδείγματα των παραπάνω κινδύνων είναι τα ακόλουθα:

| Τύπος Κινδύνου | Πιθανοί κίνδυνοι |
|----------------|--|
| Τεχνολογία | Η βάση δεδομένων που χρησιμοποιείται από σύστημα δεν μπορεί να επεξεργαστεί τόσες συναλλαγές ανά δευτερόλεπτο όσες αναμενόταν. Συστατικά στοιχεία λογισμικού που πρέπει να επαναχρησιμοποιηθούν έχουν ελαττώματα που περιορίζουν τη λειτουργικότητά τους |
| Άνθρωποι | Είναι αδύνατο να προσληφθεί προσωπικό με τις απαιτούμενες δεξιότητες. Σημαντικά άτομα από το προσωπικό |



| | |
|------------|---|
| | <p>είναι ασθενή και μη διαθέσιμα σε κρίσιμες χρονικές περιόδους. Η απαιτούμενη εκπαίδευση του προσωπικού δεν είναι διαθέσιμη</p> |
| Οργανισμός | <p>Γίνεται αναδόμηση του οργανισμού και αναλαμβάνουν την ευθύνη του έργου διαφορετικοί διευθυντές. Οικονομικά προβλήματα του οργανισμού επιβάλλουν μειώσεις του προϋπολογισμού του έργου.</p> |
| Εργαλεία | <p>παραγόμενος κώδικας από τα εργαλεία CASE είναι μη αποδοτικός. Δεν μπορεί να γίνει ολοκλήρωση των εργαλείων CASE</p> |
| Απαιτήσεις | <p>Προτείνονται αλλαγές στις απαιτήσεις, οι οποίες απαιτούν ριζική επανασχεδίαση. Οι πελάτες δεν μπορούν να αντιληφθούν τις επιπτώσεις των αλλαγών των απαιτήσεων.</p> |
| Εκτίμηση | <p>Ο απαιτούμενος χρόνος για την ανάπτυξη του λογισμικού έχει υποτιμηθεί. Η συχνότητα επιδιόρθωσης των ελαττωμάτων έχει υποτιμηθεί.</p> |

Ο Boehm, 1988 συνιστά τον προσδιορισμό και παρακολούθηση των «10 κορυφαίων» κινδύνων. Ο αριθμός των κινδύνων που χρειάζεται να παρακολουθούνται θα πρέπει να εξαρτάται από το έργο. Μπορεί να είναι 5 ή 15. Πάντως ο αριθμός των κινδύνων που θα επιλέξετε για παρακολούθηση θα πρέπει να είναι πρόσφορος για τη διαχείριση. Ένας υπερβολικά μεγάλος αριθμός κινδύνων θα απαιτούσε τη συγκέντρωση πολύ μεγάλου όγκου πληροφοριών.



Σχεδιασμός και αντιμετώπιση κινδύνων

Η διαδικασία του σχεδιασμού αντιμετώπισης κινδύνων εξετάζει κάθε έναν από τους κρίσιμους κινδύνους που έχουν προσδιοριστεί και ορίζει τις στρατηγικές για τη διαχείριση του κάθε κινδύνου. Και πάλι, δεν υπάρχει κάποια απλή διαδικασία που να μπορεί να ακολουθηθεί για τη δημιουργία πλάνων διαχείρισης κινδύνων.

Για την καλύτερη αντιμετώπιση των κινδύνων χρειάζεται σωστές στρατηγικές. Οι στρατηγικές ανήκουν σε τρεις κατηγορίες που είναι οι ακόλουθες:

1. Στρατηγικές αποφυγής (π.χ. αντιμετώπιση των ελαττωματικών στοιχείων)
2. Στρατηγικές ελαχιστοποίησης (π.χ. απουσία προσωπικού λόγω ασθένειας)
3. Πλάνα εκτάκτων καταστάσεων (π.χ. στρατηγική για την αντιμετώπιση των οικονομικών προβλημάτων)



Κεφάλαιο Τρίτο

Σχεδιασμός – Έλεγχος – Ανάπτυξη Έργων Πληροφορικής



Κεφάλαιο Τρίτο

Σχεδιασμός – Έλεγχος – Ανάπτυξη Έργων Πληροφορικής

3.1 Αρχιτεκτονική Σχεδίασης

Τα μεγάλα συστήματα έργων πληροφορικής πάντοτε αποσυντίθενται σε υποσυστήματα που παρέχουν κάποιο σχετικό σύνολο υπηρεσιών. Η διαδικασία της αρχικής σχεδίασης για να προσδιοριστούν αυτά τα υποσυστήματα και να καταλήξουμε σε ένα πλαίσιο εργασίας για τον έλεγχο και την επικοινωνία των υποσυστημάτων ονομάζεται αρχιτεκτονική.

Ο Bass, 2003 εξετάζει τρία πλεονεκτήματα της ρητής σχεδίασης και τεκμηρίωσης μιας αρχιτεκτονικής λογισμικού:

1. **Επικοινωνία ενδιαφερομένων:** Η αρχιτεκτονική είναι μια υψηλού επιπέδου παρουσίαση του συστήματος, που μπορεί να αποτελέσει το επίκεντρο της συζήτησης μεταξύ των διαφόρων ενδιαφερομένων.
2. **Ανάλυση συστήματος:** Η ρητή διατύπωση της αρχιτεκτονικής του συστήματος σε ένα πρώιμο στάδιο της ανάπτυξης του απαιτεί κάποια ανάλυση. Οι αποφάσεις της αρχιτεκτονικής σχεδίασης έχουν καθοριστική επίδραση στο αν το σύστημα θα μπορέσει να ικανοποιήσει κρίσιμες απαιτήσεις, όπως η απόδοση, η αξιοπιστία και η συντηρησιμότητα.
3. **Επαναχρησιμοποίηση σε μεγάλη κλίμακα:** Ένα μοντέλο αρχιτεκτονικής συστήματος είναι μια συμπαγής και διαχειρίσιμη περιγραφή του πώς είναι οργανωμένο το σύστημα και πώς συνεργάζονται τα συστατικά του στοιχεία. Η αρχιτεκτονική συστήματος είναι συχνά ίδια μεταξύ συστημάτων με παρόμοιες απαιτήσεις, γι' αυτό μπορεί να υποστηρίξει την επαναχρησιμοποίηση λογισμικού σε μεγάλη κλίμακα.

Ο Hofmeister, (2000) αναφέρει το πώς το στάδιο της αρχιτεκτονικής σχεδίασης υποχρεώνει τους σχεδιαστές λογισμικού να εξετάσουν ορισμένες καί-



ριες πλευρές της σχεδίασης του λογισμικού από νωρίς στη διαδικασία. Υποστηρίζει ότι η αρχιτεκτονική λογισμικού μπορεί να παίξει το ρόλο πλάνου της σχεδίασης, το οποίο χρησιμοποιείται στη διαπραγμάτευση των απαιτήσεων του συστήματος, και επίσης να χρησιμοποιηθεί ως μέσο δόμησης των συζητήσεων με τους πελάτες, τους κατασκευαστές και τους διευθυντές.

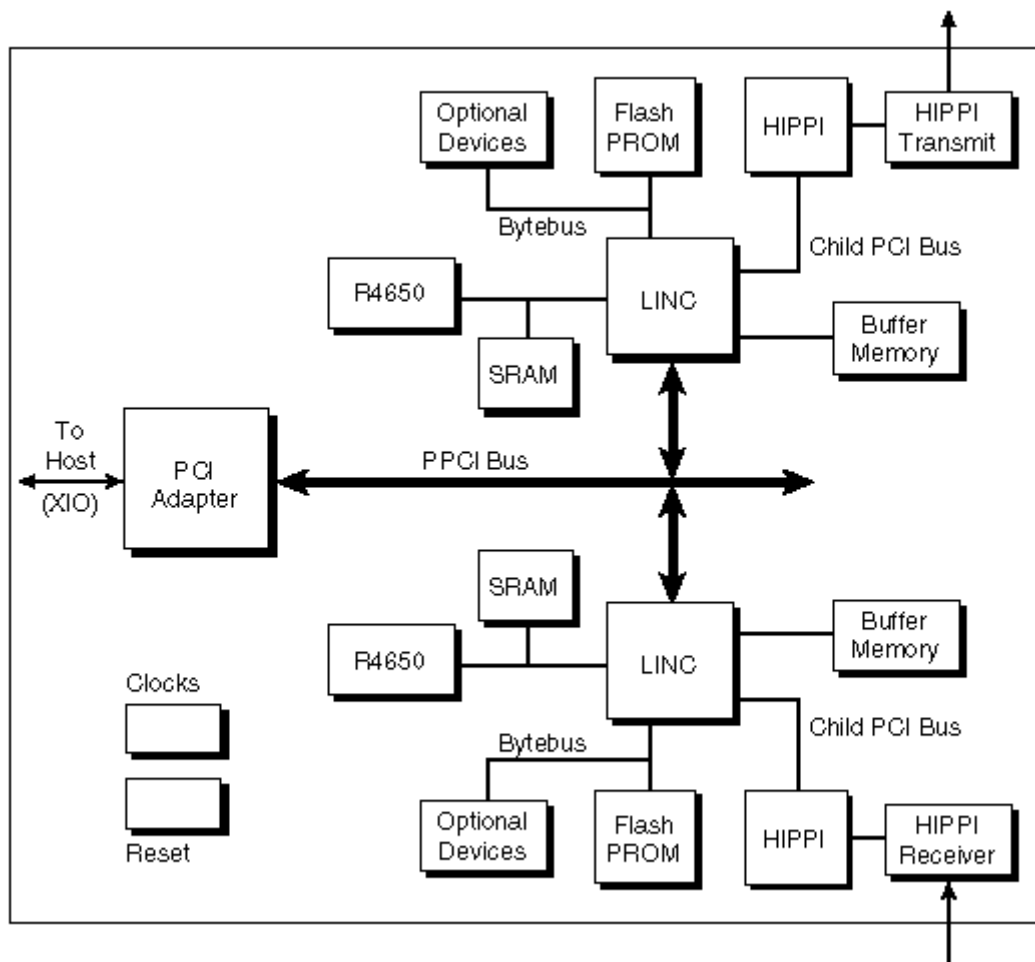
Ο Hofmeister, (2000) επισημαίνει επίσης ότι είναι απαραίτητο εργαλείο για τη διαχείριση της πολυπλοκότητας, καθώς αποκρύπτει λεπτομέρειες και επιτρέπει στους σχεδιαστές να εστιάσουν την προσοχή τους στις θεμελιώδεις αφαιρετικές έννοιες το» συστήματος. Η αρχιτεκτονική του συστήματος επηρεάζει την απόδοση, την ανθεκτικότητα, τη δυνατότητα κατανομής και τη συντηρησιμότητα ενός συστήματος (Bosch, 2000). Το ιδιαίτερο στυλ και η δομή που θα επιλεγεί για μία εφαρμογή μπορεί επομένως να εξαρτάται από μη λειτουργικές απαιτήσεις του συστήματος:

1. Απόδοση
2. Ασφάλεια
3. Ασφαλής λειτουργία
4. Διαθεσιμότητα
5. Συντηρησιμότητα

Η σχεδίαση υποσυστημάτων είναι μια αφηρημένη αποσύνθεση ενός συστήματος σε χονδρομερή συστατικά στοιχεία, κάθε ένα από τα οποία μπορεί να είναι σημαντικό σύστημα από μόνο του. Για την περιγραφή μιας σχεδίασης υποσυστημάτων συχνά χρησιμοποιούνται δομικά διαγράμματα (βλέπε Εικόνα 18), όπου κάθε πλαίσιο αντιπροσωπεύει ένα υποσύστημα. Τα πλαίσια μέσα σε πλαίσια υποδηλώνουν ότι και το ίδιο το υποσύστημα έχει αποσυντεθεί σε υποσυστήματα. Τα βέλη σημαίνουν ότι δεδομένα ή/και σήματα ελέγχου μεταβιβάζονται από υποσύστημα σε υποσύστημα κατά τη κατεύθυνση που δείχνουν. Τα δομικά διαγράμματα παρουσιάζουν μια υψηλού επιπέδου εικόνα της δομής του συστήματος, την οποία μπορούν εύκολα να καταλάβουν άτομα διαφόρων ειδικοτήτων που εμπλέκονται στη διαδικασία ανάπτυξης του συστήματος.



Εικόνα 5: Block Diagrams



Πηγή:

http://techpubs.sgi.com/library/dynaweb_docs/0650/SGI_Developer/books/OrOn2_Theops/sgi_html/figures/LINK.block.diagram.gif

Ο Bass, (2003) υποστηρίζει ότι τα απλά διαγράμματα που αποτελούνται μόνο από γραμμές και πλαίσια δεν είναι χρήσιμες αρχιτεκτονικές αναπαραστάσεις, επειδή δεν δείχνουν ούτε τη φύση των σχέσεων μεταξύ των συστατικών στοιχείων του συστήματος ούτε τις εξωτερικά ορατές ιδιότητες των στοιχείων. Από την άποψη του σχεδιαστή λογισμικού, αυτό είναι απόλυτα σωστό. Ωστόσο, τα μοντέλα αυτού του τύπου είναι αποτελεσματικά για την επικοινωνία με τους ενδιαφερόμενους του συστήματος και για το σχεδιασμό του έργου, επειδή δεν είναι υπερφορτωμένα με λεπτομέρειες. Οι ενδιαφερόμενοι μπορούν να τα καταλάβουν και να σχηματίσουν μια αφηρημένη αντίληψη για το σύστημα. Το μοντέλο προσδιορίζει τα βασικά υποσυστήματα που πρόκειται να αναπτυχθούν ανεξάρτητα, ώστε οι διευθυντές να μπορούν να αρχίσουν να αναθέτουν σε άτομα



το σχεδιασμό της ανάπτυξης αυτών των συστημάτων. Τα διαγράμματα πλαισίων-γραμμών σίγουρα δε θα πρέπει να είναι η μοναδική αρχιτεκτονική αναπαράσταση που χρησιμοποιείται είναι όμως ένα από τα πολλά χρήσιμα αρχιτεκτονικά μοντέλα.

Το γενικό πρόβλημα του προσδιορισμού του τρόπου με τον οποίο θα αποσυντεθεί ένα σύστημα σε υποσυστήματα είναι δύσκολο. Φυσικά, οι απαιτήσεις του συστήματος είναι σημαντικός παράγοντας, και θα πρέπει να προσπαθείτε να δημιουργήσετε μια σχεδίαση όπου υπάρχει στενή αντιστοιχία μεταξύ απαιτήσεων και υποσυστημάτων. Αυτό σημαίνει ότι, αν αλλάξουν οι απαιτήσεις, οι απαιτούμενες αλλαγές θα είναι τοπικές και δε θα εκτείνονται σε πολλά υποσυστήματα.

3.2 Έλεγχος και Αναφορές

Τα μοντέλα για τη δόμηση ενός συστήματος ασχολούνται με το πώς αποσυντίθεται το σύστημα σε υποσυστήματα. Για να λειτουργήσουν ως σύστημα, τα υποσυστήματα πρέπει να ελέγχονται έτσι ώστε οι υπηρεσίες τους να παρέχονται στο σωστό τόπο και χρόνο. Τα δομικά μοντέλα δεν περιλαμβάνουν πληροφορίες ελέγχου. Αντίθετα, ο αρχιτέκτονας θα πρέπει να οργανώνει τα υποσυστήματα σύμφωνα με κάποιο μοντέλο ελέγχου που συμπληρώνει το δομικό μοντέλο που χρησιμοποιείται. Τα μοντέλα ελέγχου σε αρχιτεκτονικό επίπεδο ασχολούνται με τον έλεγχο ροής μεταξύ των υποσυστημάτων.

Υπάρχουν δύο γενικά στυλ ελέγχου που χρησιμοποιούνται στα συστήματα λογισμικού:

1. Συγκεντρωτικός έλεγχος: Ένα υποσύστημα έχει τη συνολική ευθύνη για τον έλεγχο και εκκινεί και σταματά τα άλλα υποσυστήματα.
2. Έλεγχος βάσει συμβάντων: Αντί να ενσωματώνονται πληροφορίες ελέγχου σε ένα υποσύστημα, κάθε υποσύστημα μπορεί να αποκρίνεται σε συμβάντα που παράγονται εξωτερικά. Τα συμβάντα αυτά μπορεί να προέρχονται από άλλα υποσυστήματα ή από το περιβάλλον του συστήματος.



Τα στυλ ελέγχου χρησιμοποιούνται σε συνδυασμό με τα δομικά στυλ. Όλα τα δομικά στυλ που έχουμε εξετάσει μπορούν να υλοποιηθούν με τη χρήση συγκεντρωτικού ελέγχου ή ελέγχου βάσει συμβάντων.

Ανάλυση Μοντέλων

Συγκεντρωτικός Έλεγχος

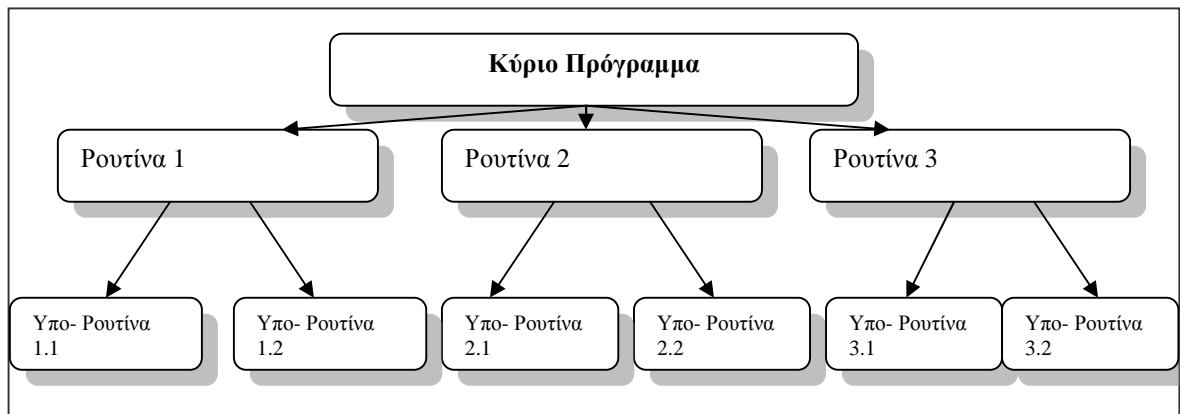
Σε ένα μοντέλο συγκεντρωτικού ελέγχου, ένα υποσύστημα αναλαμβάνει το ρόλο του ελεγκτή του συστήματος και έχει την ευθύνη για τη διαχείριση της εκτέλεσης των άλλων υποσυστημάτων. Τα μοντέλα συγκεντρωτικού ελέγχου εμπίπτουν σε δύο κατηγορίες, ανάλογα με το αν τα ελεγχόμενα υποσυστήματα εκτελούνται σειριακά ή παράλληλα.

1. **Το μοντέλο κλήσης-επιστροφής:** Πρόκειται για το γνωστό μοντέλο υπο ρουτινών «από πάνω προς τα κάτω», όπου ο έλεγχος ξεκινά από την κορυφή μιας ιεραρχίας υπο ρουτινών και, μέσω κλήσεων υπο ρουτινών, μεταφέρεται στα χαμηλότερα επίπεδα του δένδρου. Το μοντέλο υπο ρουτινών έχει εφαρμογή μόνο στα σειριακά συστήματα.
2. **Το μοντέλο διαχειριστή:** Έχει εφαρμογή στα ταυτόχρονα συστήματα. Ένα συστατικό στοιχείο του συστήματος αναλαμβάνει το ρόλο του διαχειριστή του συστήματος και ελέγχει την εκκίνηση, την παύση, και το συντονισμό των άλλων διεργασιών του συστήματος. **Διεργασία** είναι ένα υποσύστημα ή υπό μονάδα που μπορεί να εκτελείται παράλληλα με άλλες διεργασίες. Μια μορφή αυτού του μοντέλου μπορεί επίσης να εφαρμοστεί στα σειριακά συστήματα, όπου μία ρουτίνα διαχείρισης καλεί συγκεκριμένα υποσυστήματα ανάλογα με τις τιμές κάποιων μεταβλητών κατάστασης. Αυτό συνήθως υλοποιείται με μια εντολή περιπτώσεων (case).

Ένα μοντέλο κλήσης επιστροφής παρουσιάζεται παρακάτω (βλέπε σχήμα 3)



Σχήμα 3: Το μοντέλο ελέγχου κλήσης – επιστροφής



3.2.1 Αρχιτεκτονική Αναφοράς

Εκτός από αυτά τα γενικά μοντέλα, μπορούν να χρησιμοποιηθούν αρχιτεκτονικά μοντέλα ειδικά για το συγκεκριμένο πεδίο εφαρμογής. Αν και τα συγκεκριμένα συστήματα διαφέρουν στις λεπτομέρειες, η κοινή αρχιτεκτονική δομή μπορεί να επαναχρησιμοποιηθεί για την ανάπτυξη νέων συστημάτων. Αυτά τα αρχιτεκτονικά μοντέλα ονομάζονται ειδικές αρχιτεκτονικές πεδίων.

Υπάρχουν δύο τύποι μοντέλων ειδικών αρχιτεκτονικών πεδίου:

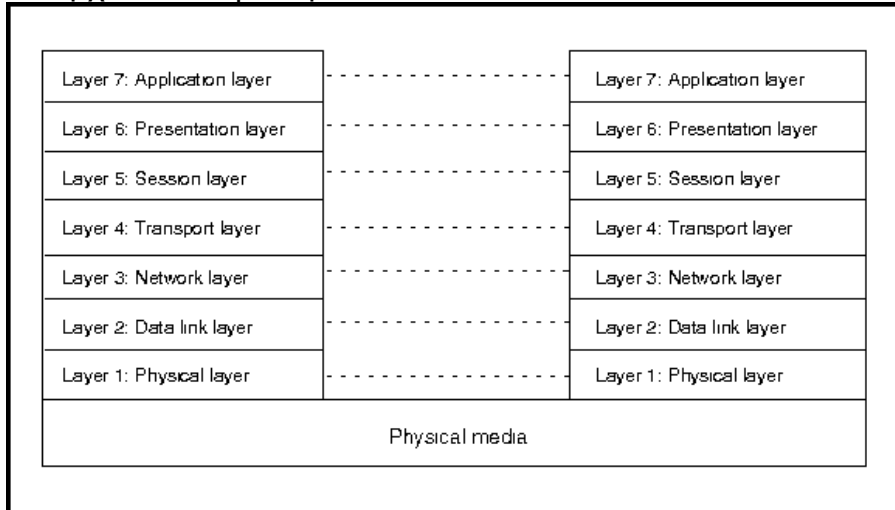
1. **Τα γενικά μοντέλα:** είναι αφαιρέσεις που αντιπροσωπεύουν ένα σύνολο πραγματικών συστημάτων.
2. **Τη μοντέλα αναφοράς:** είναι πιο αφηρημένα και περιγράφουν μια ευρύτερη κατηγορία συστημάτων.

Τα γενικά μοντέλα μπορούν και αυτά να παίξουν το ρόλο μοντέλων αναφοράς. Ο λόγος για τον οποίο κάνουμε τη διάκριση είναι ότι τα γενικά μοντέλα μπορούν να επαναχρησιμοποιηθούν άμεσα σε μια σχεδίαση.

Οι αρχιτεκτονικές αναφοράς κανονικά δε θεωρούνται δρόμος προς την υλοποίηση. Ο κύριος ρόλος τους είναι να παρέχουν ένα μέσο για τη συζήτηση ειδικών αρχιτεκτονικών για το πεδίο εφαρμογής και για τη σύγκριση διαφόρων συστημάτων μέσα σε αυτά τα πλαίσια. Το μοντέλο αναφοράς παρέχει ένα λεξιλόγιο για συγκρίσεις. Παρέχει μια βάση πάνω στην οποία μπορούν να αξιολογούνται τα συστήματα.



Εικόνα 6: Η αρχιτεκτονική του μοντέλου OSI



Πηγή:

http://www.islab.demokritos.gr/gr/html/ptixiakes/ATM_IPv6_&SecurityConsiderations/images/image72.gif

Αναφορά στο μοντέλο OSI

Σύμφωνα με την ηλεκτρονική εγκυκλοπαίδεια Wikipedia, 2010 το μοντέλο αναφοράς Ανοικτής Διασύνδεσης Συστημάτων, ή μοντέλο αναφοράς OSI (αγγλ. *OSI reference model*) «είναι μια διαστρωματωμένη, αφηρημένη περιγραφή για τη σχεδίαση τηλεπικοινωνιακών και δικτυακών πρωτοκόλλων η οποία καθορίστηκε από την πρωτοβουλία Ανοικτή Διασύνδεση Συστημάτων – OSI. Είναι γνωστό και ως μοντέλο των επτά επιπέδων».

Ο βασικός σκοπός του μοντέλο OSI υποδιαιρεί τις λειτουργίες ενός τηλεπικοινωνιακού δικτύου σε μια «κατακόρυφη» στοίβα από επτά επίπεδα. Το κάθε επίπεδο μπορεί να οριστεί κάποιο πρωτόκολλο σε μία συγκεκριμένη υλοποίηση αξιοποιώντας κατάλληλα τις λειτουργίες του κατώτερου του επιπέδου στη στοίβα επιπέδου.

3.3 Ανάπτυξη

Οι επιχειρήσεις σήμερα λειτουργούν σε ένα παγκόσμιο περιβάλλον το οποίο ως γνωστών μεταβάλλεται ταχύτατα. Έτσι λοιπόν πρέπει να ανταποκρίνονται σε νέες ευκαιρίες και αγορές, στην αλλαγή των οικονομικών συνθηκών, και στην εμφάνιση ανταγωνιστικών προϊόντων και υπηρεσιών. Το λογισμικό αποτελεί τμήμα όλων σχεδόν των επιχειρηματικών λειτουργιών. Αυτό



έχει ως αποτέλεσμα οι νέες εφαρμογές είναι σημαντικό να αναπτύσσονται γρήγορα ώστε οι εταιρείες να εκμεταλλεύονται τις νέες ευκαιρίες και να ανταποκρίνονται στις πιέσεις του ανταγωνισμού. Σήμερα, λοιπόν, η γρήγορη ανάπτυξη και παράδοση αποτελεί συχνά την πλέον κρίσιμη απαίτηση για τα συστήματα λογισμικού.

Επειδή αυτές οι επιχειρήσεις λειτουργούν σε ένα μεταβαλλόμενο περιβάλλον, συχνά είναι πρακτικά αδύνατο να παραχθεί ένα πλήρες σύνολο σταθερών απαιτήσεων λογισμικού. Αναπόφευκτα οι απαιτήσεις που προτείνονται αλλάζουν, επειδή είναι αδύνατο για τους πελάτες να προβλέψουν πώς θα επηρεάσει το σύστημα τις εργασιακές πρακτικές, πώς θα αλληλεπιδράσει με άλλα συστήματα, και ποιες λειτουργίες χρηστών θα πρέπει να αυτοματοποιηθούν. Σε κάποιες περιπτώσεις, οι πραγματικές απαιτήσεις ξεκαθαρίζουν μόνον αφού το σύστημα παραδοθεί και οι χρήστες αποκτήσουν εμπειρία με αυτό.

Οι διαδικασίες ανάπτυξης λογισμικού που βασίζονται στην ολοκληρωτική προδιαγραφή των απαιτήσεων και κατόπιν στο σχεδιασμό, την κατασκευή, και τον έλεγχο του συστήματος, δεν έχουν δημιουργηθεί με στόχο τη γρήγορη ανάπτυξη λογισμικού.

Καθώς οι απαιτήσεις αλλάζουν ή καθώς διαπιστώνονται προβλήματα σε αυτές, ο σχεδιασμός ή η υλοποίηση του συστήματος πρέπει να αναθεωρηθούν και να δοκιμαστούν εκ νέου. Έτσι, η διάρκεια της συμβατικής διαδικασίας του καταρράκτη ή της διαδικασίας βάσει προδιαγραφών μεγαλώνει, και το τελικό λογισμικό παραδίδεται στον πελάτη αρκετά μετά από την αρχικά καθορισθείσα ημερομηνία.

Σε ένα επιχειρηματικό περιβάλλον που κινείται με γρήγορους ρυθμούς, κάτι τέτοιο μπορεί να προκαλέσει σημαντικότερα προβλήματα. Όταν το λογισμικό είναι πλέον διαθέσιμο για χρήση, οι αρχικοί λόγοι για την προμήθεια του μπορεί να έχουν αλλάξει τόσο ριζικά που το λογισμικό να είναι ουσιαστικά πλέον άχρηστο. Έτσι, ιδιαίτερα για επιχειρηματικά συστήματα, οι διαδικασίες που εστιάζονται στη γρήγορη ανάπτυξη και παράδοση είναι ουσιώδεις.



Οι διαδικασίες γρήγορης ανάπτυξης λογισμικού έχουν στόχο την παραγωγή χρήσιμου λογισμικού σε μικρό χρονικό διάστημα. Γενικά, είναι επαναληπτικές διαδικασίες στις οποίες συνυφαίνονται η εξαγωγή προδιαγραφών, ο σχεδιασμός, η ανάπτυξη, και ο έλεγχος. Το λογισμικό δεν αναπτύσσεται και διανέμεται στην ολότητα του αλλά σε μια σειρά επαυξήσεων, με κάθε βήμα να περιλαμβάνει νέα λειτουργικότητα του συστήματος. Αν και υπάρχουν πολλές προσεγγίσεις στη γρήγορη ανάπτυξη λογισμικού, όλες μοιράζονται κάποια θεμελιώδη χαρακτηριστικά:

1. **Οι διαδικασίες της εξαγωγής προδιαγραφών, του σχεδιασμού, και της υλοποίησης λαμβάνουν χώρα παράλληλα.** Δεν υπάρχουν λεπτομερείς προδιαγραφές του συστήματος, και η τεκμηρίωση του σχεδιασμού είναι η ελάχιστη δυνατή ή παράγεται αυτόματα από το προγραμματιστικό περιβάλλον που χρησιμοποιείται για την υλοποίηση του συστήματος. Το έγγραφο των απαιτήσεων χρήστη καθορίζει μόνο τα πιο σημαντικά χαρακτηριστικά του συστήματος.
2. **Το σύστημα αναπτύσσεται σε μια σειρά επαυξήσεων.** Στον καθορισμό και την αξιολόγηση κάθε επαύξησης συμμετέχουν οι τελικοί χρήστες του συστήματος και οι άλλοι ενδιαφερόμενοι, οι οποίοι μπορεί να προτείνουν αλλαγές στο λογισμικό και νέες απαιτήσεις που θα πρέπει να υλοποιηθούν σε κάποια μεταγενέστερη επαύξηση.
3. Οι διασυνδέσεις χρήστη της εφαρμογής συχνά αναπτύσσονται με τη χρήση ενός αλληλεπιδραστικού συστήματος, το οποίο επιτρέπει τη γρήγορη δημιουργία τους μέσω της σχεδίασης και της τοποθέτησης εικονιδίων.

Τα κύρια πλεονεκτήματα της υιοθέτησης μιας αυξητικής προσέγγισης για την ανάπτυξη λογισμικού είναι τα εξής:

- § *Ταχύτερη παράδοση υπηρεσιών στον πελάτη.* Οι πρώτες επαυξήσεις του συστήματος μπορούν να παραδώσουν λειτουργικότητα μεγάλης προτεραιότητας, έτσι ώστε οι πελάτες να έχουν αποτελέσματα από την αρχή της ανάπτυξης του συστήματος. Οι πελάτες μπορούν να δουν στην πράξη τις απαιτήσεις τους και να προσδιορίσουν αλλαγές οι οποίες θα πρέπει να ενσωματωθούν σε μεταγενέστερες εκδόσεις του συστήματος.



§ Συμμετοχή των χρηστών στο σύστημα. Οι χρήστες του συστήματος πρέπει να συμμετέχουν στη διαδικασία της αυξητικής ανάπτυξης, επειδή είναι απαραίτητο να ενημερώνουν την ομάδα ανάπτυξης για τις εντυπώσεις τους σχετικά με τις επαυξήσεις που έχουν παραδοθεί. Αυτή η εμπλοκή των χρηστών δε σημαίνει απλώς ότι το σύστημα έχει περισσότερες πιθανότητες να ικανοποιεί τις ανάγκες τους· σημαίνει επίσης ότι οι τελικές χρήστες του συστήματος νιώθουν ότι έχουν συνεισφέρει και εκείνοι, και ενδιαφέρονται για την επιτυχία της διαδικασίας.

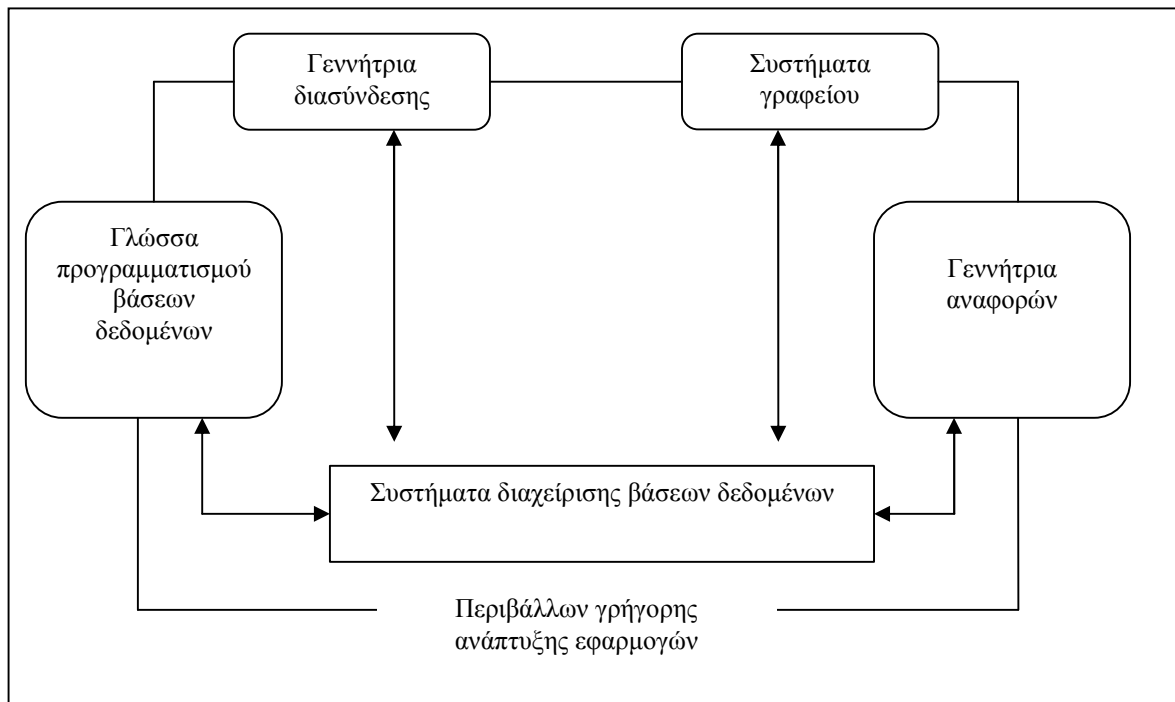
3.3.1 Γρήγορη Ανάπτυξη

Αν και τα τελευταία χρόνια η προσέγγιση επαναληπτικής ανάπτυξης που μονοπωλεί το ενδιαφέρον είναι οι ευέλικτες μέθοδοι, επιχειρηματικά συστήματα αναπτύσσονται με αυτόν τον τρόπο εδώ και πολύ καιρό με τη βοήθεια τεχνικών γρήγορης ανάπτυξης εφαρμογών. Οι τεχνικές αυτές προέκυψαν από τις επονομαζόμενες γλώσσες τέταρτης γενιάς της δεκαετίας του 1980, και χρησιμοποιούνται για την ανάπτυξη εφαρμογών οι οποίες διαχειρίζονται μεγάλο όγκο δεδομένων. Έτσι, συνήθως οργανώνονται ως ένα σύνολο εργαλείων τα οποία επιτρέπουν τη δημιουργία, την αναζήτηση, την εμφάνιση, και την παρουσίαση δεδομένων σε αναφορές. Τα εργαλεία που περιλαμβάνονται σε ένα περιβάλλον RAD είναι τα εξής:

1. **Μια γλώσσα προγραμματισμού βάσεων δεδομένων**, η οποία εμπεριέχει πληροφορίες για τις δομές των βάσεων δεδομένων και περιλαμβάνει θεμελιώδεις λειτουργίες χειρισμού μιας βάσης δεδομένων. Η καθιερωμένη γλώσσα προγραμματισμού βάσεων δεδομένων είναι η SQL (Groff κ.ά., 2002).
2. **Μια γεννήτρια διασύνδεσης**, η οποία χρησιμοποιείται για τη δημιουργία φορμών εισαγωγής και εμφάνισης των δεδομένων.
3. **Σύνδεσμοι σε εφαρμογές γραφείου**, όπως προγράμματα λογιστικών φύλλων για την ανάλυση και το χειρισμό αριθμητικών πληροφοριών, ή επεξεργαστές κειμένου για τη δημιουργία προτύπων αναφορών.
4. **Μια γεννήτρια αναφορών**, η οποία χρησιμοποιείται για τον ορισμό και τη δημιουργία αναφορών από πληροφορίες της βάσης δεδομένων.



Σχήμα 4: Περιβάλλον γρήγορης ανάπτυξης



Καθώς πολλές επιχειρηματικές εφαρμογές χρησιμοποιούν δομημένες φόρμες για την είσοδο και την έξοδο των δεδομένων, τα περιβάλλοντα RAD παρέχουν ισχυρές λειτουργίες για τον ορισμό οθονών και την παραγωγή αναφορών. Οι οθόνες συχνά ορίζονται ως ακολουθίες συνδεδεμένων φορμών οπότε το σύστημα παραγωγής οθονών πρέπει να παρέχει τα εξής:

1. **Αλληλεπιδραστικό ορισμό φορμών**, όπου ο προγραμματιστής ορίζει τα πεδία που θα εμφανιστούν και τον τρόπο με τον οποίο θα είναι οργανωμένα.
2. **Σύνδεση φορμών**, όπου ο προγραμματιστής μπορεί να καθορίσει ότι συγκεκριμένες είσοδοι οδηγούν σε περαιτέρω εμφάνιση φορμών.
3. **Επαλήθευση πεδίων**, όπου ο προγραμματιστής ορίζει τις επιτρεπτές τιμές εισόδου για τα πεδία της φόρμας.

3.3.2 Κατασκευή Προτύπων Λογισμικού

Υπάρχουν ορισμένες περιπτώσεις όπου πρακτικοί λόγοι ή εργασιακές συμβάσεις δεν επιτρέπουν τη χρήση της αυξητικής διαδικασίας στην παράδοση του λογισμικού. Σε αυτές τις περιπτώσεις δημιουργείται μια δήλωση των



απαιτήσεων συστήματος, η οποία χρησιμοποιείται από την ομάδα ανάπτυξης ως βάση για το λογισμικό. Ο δημιουργός του λογισμικού μπορεί να έχει κάποια από τα οφέλη της αυξητικής διαδικασίας ανάπτυξης μέσω της κατασκευής ενός πρωτοτύπου. Αυτή η προσέγγιση ονομάζεται μερικές φορές κατασκευή αναλώσιμων πρωτοτύπων, επειδή το πρωτότυπο δεν παραδίδεται στον πελάτη ούτε διατηρείται από τον προγραμματιστή.

Το πρωτότυπο αποτελεί μια αρχική εκδοχή ενός συστήματος λογισμικού, η οποία χρησιμοποιείται για την επίδειξη εννοιών, τη δοκιμή σχεδιαστικών επιλογών και, γενικά, τον εντοπισμό περισσότερων πληροφοριών σχετικά με το πρόβλημα και τις πιθανές λύσεις του. Απαραίτητο στοιχείο σε αυτή την προσέγγιση είναι η γρήγορη και επαναληπτική ανάπτυξη του πρωτοτύπου, ώστε το κόστος να παραμένει ελεγχόμενο και οι ενδιαφερόμενοι για το σύστημα να έχουν τη δυνατότητα να πειραματιστούν με το πρωτότυπο νωρίς στη διαδικασία της ανάπτυξης.

Το πρωτότυπο μπορεί να χρησιμοποιηθεί με διάφορους τρόπους στη διαδικασία ανάπτυξης του λογισμικού:

1. Κατά τη διαδικασία παραγωγής των απαιτήσεων, το πρωτότυπο μπορεί να βοηθήσει στην εξαγωγή και την επικύρωση των απαιτήσεων του συστήματος.
2. Κατά τη διαδικασία σχεδιασμού του συστήματος, το πρωτότυπο μπορεί να χρησιμοποιηθεί για την εξέταση συγκεκριμένων λύσεων στο λογισμικό, καθώς και για την υποστήριξη του σχεδιασμού της διασύνδεσης χρήστη.
3. Κατά τη διαδικασία των δοκιμών, το πρωτότυπο είναι χρήσιμο για την εκτέλεση συγκριτικών δοκιμών με το σύστημα που θα παραδοθεί στον πελάτη.

Τα πρωτότυπα συστημάτων επιτρέπουν στους χρήστες να δουν πόσο καλά υποστηρίζει τα σύστημα την εργασία τους. Με αυτόν τον τρόπο, οι χρήστες μπορεί να έχουν νέες ιδέες για τις απαιτήσεις και να εντοπίσουν ισχυρά και αδύνατα σημεία στο λογισμικό. Ίσως τότε να προτείνουν νέες απαιτήσεις συστήματος. Επιπλέον, καθώς αναπτύσσεται το πρωτότυπο, μπορεί να αποκαλυφθούν λάθη



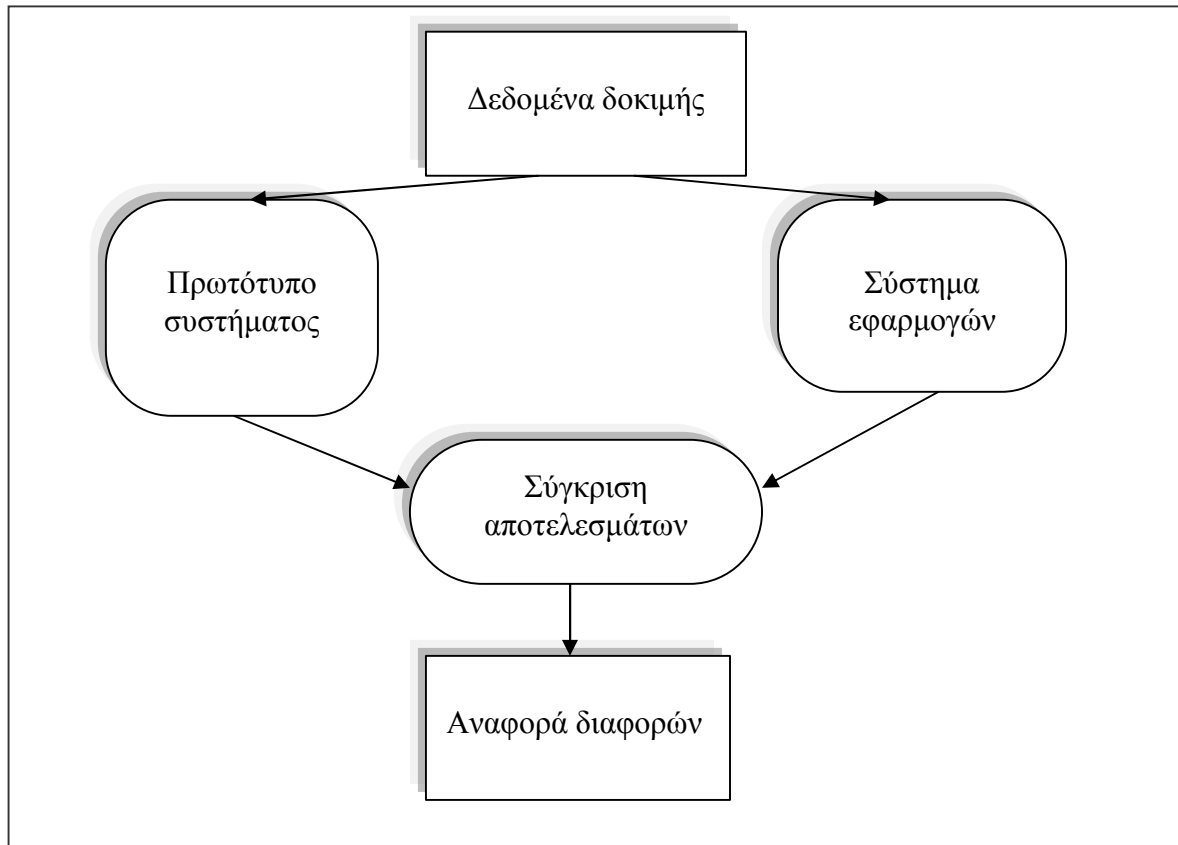
και παραλείψεις στις απαιτήσεις που έχουν προταθεί. Μια λειτουργία που περιγράφεται σε μια προδιαγραφή ίσως να φαίνεται χρήσιμη και ορισμένη με σαφήνεια. Όταν όμως η λειτουργία αυτή συνδυαστεί με άλλες λειτουργίες, οι χρήστες συχνά διαπιστώνουν ότι η αρχική τους άποψη ήταν λανθασμένη ή ελλιπής. Σε μια τέτοια περίπτωση ίσως χρειαστεί να τροποποιηθούν οι προδιαγραφές του συστήματος, ώστε να αντικατοπτρίζουν την καινούργια αντίληψη των πελατών για τις απαιτήσεις.

Το πρωτότυπο θα μπορούσε να χρησιμοποιηθεί κατά το σχεδιασμό του συστήματος για την εκτέλεση σχεδιαστικών πειραμάτων, τα οποία ελέγχουν την επιτευξιμότητα του προτεινόμενου σχεδιασμού. Η κατασκευή πρωτοτύπων αποτελεί επίσης ουσιαστικό κομμάτι της διαδικασίας σχεδιασμού διασύνδεσης χρήστη. Λόγω της δυναμικής φύσης των διασυνδέσεων χρήστη, οι περιγραφές με κείμενο και τα διαγράμματα δεν αρκούν για να εκφράσουν τις σχετικές απαιτήσεις. Έτσι, η γρήγορη κατασκευή πρωτοτύπων με συμμετοχή του τελικού χρήστη είναι ο μόνος πρακτικός τρόπος ανάπτυξης διασυνδέσεων χρήστη μέσω γραφικών για συστήματα λογισμικού.

Ένα μεγάλο πρόβλημα στις δοκιμές συστημάτων είναι ο έλεγχος της εγκυρότητας τους, όπου πρέπει να ελεγχθεί το αν τα αποτελέσματα μιας δοκιμής είναι τα αναμενόμενα. Όταν είναι διαθέσιμο κάποιο πρωτότυπο, ο κόπος που απαιτείται για τον έλεγχο των αποτελεσμάτων μπορεί να περιοριστεί με τη βοήθεια συγκριτικών δοκιμών (βλέπει σχήμα 5). Τόσο στο πρωτότυπο όσο και στο σύστημα που δοκιμάζεται υποβάλλονται οι ίδιες περιπτώσεις δοκιμής. Αν και τα δύο συστήματα δώσουν το ίδιο αποτέλεσμα, η περίπτωση δοκιμής πιθανότατα δεν έχει ανιχνεύσει κάποιο σφάλμα. Αν τα αποτελέσματα διαφέρουν, αυτό μπορεί να σημαίνει ότι υπάρχει κάποια ατέλεια στο σύστημα και θα πρέπει να ερευνηθούν οι λόγοι της διαφοράς.



Σχήμα 5: Συγκριτική Δοκιμή



Κεφάλαιο Τέταρτο

Επαλήθευση και Επικύρωση



Κεφάλαιο Τέταρτο

Επαλήθευση και Επικύρωση

4.1 Σχεδιασμός Επαλήθευσης & Επικύρωσης

Κατά τη διάρκεια και μετά τη διαδικασία υλοποίησης, το πρόγραμμα που αναπτύσσεται πρέπει να ελεγχθεί για να διασφαλιστεί ότι ικανοποιεί τις προδιαγραφές του καθώς παρέχει τη λειτουργικότητα που αναμένουν αυτοί που πλήρωσαν για το λογισμικό.

Επαλήθευση και επικύρωση (E & E) είναι το όνομα που δίνεται σε αυτές τις διαδικασίες ελέγχου και ανάλυσης. Οι δραστηριότητες επαλήθευσης και επικύρωσης λαμβάνουν χώρα σε κάθε στάδιο της διαδικασίας παραγωγής λογισμικού. Η E & E αρχίζει με αναθεωρήσεις των απαιτήσεων και φτάνει μέσω των αναθεωρήσεων των σχεδιασμών και των επιθεωρήσεων του κώδικα στις δοκιμές του προϊόντος.

Η επαλήθευση και η επικύρωση δεν είναι το ίδιο πράγμα, αν και συχνά μπερδεύονται. Ο Boehm (1979) εξέφρασε με λακωνικό τρόπο τις διαφορές μεταξύ τους

- "Επικύρωση: Κατασκευάζουμε **το** σωστό προϊόν;"
- "Επαλήθευση: Κατασκευάζουμε **σωστά** το προϊόν;"

Αυτοί οι ορισμοί μας λένε ότι ο ρόλος της επαλήθευσης αφορά τον έλεγχο ότι το η λογισμικό συμμορφώνεται με τις προδιαγραφές του. Θα πρέπει να ελέγξετε ότι ικανοποιεί τις καθορισμένες λειτουργικές και μη λειτουργικές απαιτήσεις. Η επικύρωση, είναι μια πιο γενική διαδικασία. Σκοπός της επικύρωσης είναι να εξασφαλίσει κ ότι το σύστημα λογισμικού ικανοποιεί τις προσδοκίες του πελάτη.

Ο τελικός στόχος της διαδικασίας επαλήθευσης και επικύρωσης είναι να εδραιώσει την πίστη ότι το σύστημα λογισμικού "είναι το κατάλληλο". Αυτό σημαίνει ότι το σύστημα πρέπει να είναι αρκετά καλό για τη χρήση που προορίζεται. Το επίπεδο της απαιτούμενης εμπιστοσύνης που πρέπει να υπάρχει



εξαρτάται από το σκοπό του συστήματος, τις προσδοκίες των χρηστών του συστήματος, και το τρέχον εμπορικό περιβάλλον για το σύστημα:

- § **Λειτουργία λογισμικού** Το επίπεδο της επιβεβλημένης εμπιστοσύνης εξαρτάται από το πόσο κρίσιμο είναι το λογισμικό για έναν οργανισμό. Για παράδειγμα, το επίπεδο της επιβεβλημένης εμπιστοσύνης για λογισμικό που χρησιμοποιείται σε ένα σύστημα κρίσιμης ασφαλούς λειτουργίας είναι κατά πολύ υψηλότερο από αυτό που απαιτείται για ένα πρωτότυπο σύστημα λογισμικού, το οποίο έχει αναπτυχθεί για να επιδείξει ορισμένες νέες ιδέες.
- § **Προσδοκίες χρηστών** Είναι μια λυπηρή εικόνα για τη βιομηχανία λογισμικού το γεγονός ότι πολλοί χρήστες έχουν χαμηλές προσδοκίες για το λογισμικό τους και δεν εκπλήσσονται όταν αστοχεί κατά τη χρήση του. Ωστόσο, η ανοχή των χρηστών στις αστοχίες συστημάτων έχει ελαττωθεί από τη δεκαετία του 1990. Σήμερα, είναι λιγότερο αποδεκτή η παράδοση αναξιόπιστων συστημάτων, οπότε οι εταιρείες λογισμικού πρέπει να αφιερώσουν μεγαλύτερη προσπάθεια στην επαλήθευση και στην επικύρωση.
- § **Εμπορικό περιβάλλον** Όταν ένα σύστημα βγει στην αγορά, οι πωλητές του συστήματος πρέπει να λάβουν υπόψη τους ανταγωνιστικά προγράμματα, την τιμή που είναι πρόθυμοι να πληρώσουν οι πελάτες για ένα σύστημα και το απαιτούμενο χρονοδιάγραμμα παράδοσης του συστήματος.

Όταν μια εταιρεία έχει λίγους ανταγωνιστές, μπορεί να αποφασίσει να αποδεσμεύσει ένα πρόγραμμα πριν δοκιμαστεί και αποσφαλμακοθεί πλήρως γιατί θέλει να είναι η πρώτη στην αγορά. Όπου οι πελάτες δεν είναι πρόθυμοι να πληρώσουν υψηλές τιμές για το λογισμικό, μπορεί να είναι πρόθυμοι να ανεχτούν περισσότερα σφάλματα λογισμικού. Θα πρέπει λοιπόν να εξεταστούν όλοι αυτοί οι παράγοντες όταν ληφθεί η απόφαση για το πόση προσπάθεια πρέπει να δαπανηθεί στη διαδικασία E & E. Στη διαδικασία E & E, υπάρχουν δύο συμπληρωματικές προσεγγίσεις για τον έλεγχο και την ανάλυση ενός συστήματος:



- § **Οι επιθεωρήσεις λογισμικού ή αναθεωρήσεις από ομότιμους** αναλύουν και ελέγχουν τις αναπαραστάσεις ενός συστήματος όπως το έγγραφο των απαιτήσεων, τα σχεδιαστικά διαγράμματα και τον πηγαίο κώδικα του προγράμματος. Μπορείτε να χρησιμοποιήσετε επιθεωρήσεις σε όλα τα στάδια της διαδικασίας. Οι επιθεωρήσεις μπορούν να συμπληρώνονται από κάποια αυτόματη ανάλυση του πηγαίου κειμένου ενός συστήματος ή αντίστοιχων εγγράφων. Οι επιθεωρήσεις λογισμικού και οι αυτοματοποιημένες αναλύσεις είναι στατικές τεχνικές E & E, καθώς δε χρειάζεται να εκτελέσετε το λογισμικό σε κάποιον υπολογιστή.
- § **Οι δοκιμές λογισμικού** αφορούν την εκτέλεση μιας υλοποίησης του λογισμικού, με δοκιμαστικά δεδομένα. Εξετάζετε τις εξόδους του λογισμικού και τη λειτουργική του συμπεριφορά για να ελέγξετε αν αποδίδει όπως πρέπει. Η εκτέλεση δοκιμών είναι μια δυναμική τεχνική επαλήθευσης και επικύρωσης.

4.2 Δοκιμή Λογισμικού

Η δοκιμή συστήματος (αφορά την ενοποίηση δύο ή περισσότερων συστατικών στοιχείων που υλοποιούν λειτουργίες ή χαρακτηριστικά του συστήματος και στη συνέχεια δοκιμή αυτού του ολοκληρωμένου συστήματος. Σε μια επαναληπτική διαδικασία, η δοκιμή συστήματος ασχολείται με τη δοκιμή μιας επαύξησης που θα παραδοθεί στον πελάτη σε μια διαδικασία καταρράκτη, η δοκιμή συστήματος ασχολείται με τη δοκιμή ολόκληρου του συστήματος. Για τα περισσότερα περίπλοκα συστήματα, υπάρχουν δύο διακριτές φάσεις της δοκιμής συστήματος:

- § **Δοκιμή ενοποίησης** όπου η ομάδα δοκιμών έχει πρόσβαση στον πηγαίο κώδικα του συστήματος. Όταν ανακαλυφθεί ένα πρόβλημα, η ομάδα ενοποίησης προσπαθεί να βρει την πηγή του προβλήματος και να προσδιορίσει τα συστατικά στοιχεία που πρέπει να αποσφαλματωθούν. Η δοκιμή ενοποίησης αφορά κυρίως την εύρεση ατελειών στο σύστημα.



§ **Δοκιμή τελικής έκδοσης** όπου δοκιμάζεται μια έκδοση του συστήματος που θα μπορούσε να παραδοθεί στους χρήστες. Εδώ, η ομάδα δοκιμών ενδιαφέρεται να επικυρώσει ότι το σύστημα ικανοποιεί τις απαιτήσεις του και να διασφαλίσει ότι το σύστημα είναι αξιόπιστο. Η δοκιμή τελικής έκδοσης συνήθως είναι δοκιμή "μαύρου κουτιού" όπου η ομάδα δοκιμών απλώς ενδιαφέρεται να επιδείξει ότι το σύστημα δουλεύει ή δε δουλεύει σωστά. Τα προβλήματα αναφέρονται στην ομάδα ανάπτυξης, της οποίας η δουλειά είναι να αποσφαλματώσει το πρόγραμμα. Όταν στις δοκιμές τελικής έκδοσης εμπλέκονται οι πελάτες τότε, μερικές φορές, ονομάζονται δοκιμές αποδοχής. Αν η τελική έκδοση είναι αρκετά καλή, ο πελάτης μπορεί να την αποδεχτεί για χρήση.

Βασικά, μπορεί κανείς να θεωρήσει τη δοκιμή ενοποίησης ως δοκιμή ημιτελών συστημάτων που συντίθενται από συστοιχίες ή ομαδοποιήσεις συστατικών στοιχείων ενός συστήματος. Η δοκιμή τελικής έκδοσης ασχολείται με τη δοκιμή της έκδοσης του συστήματος που προορίζεται για παράδοση στον πελάτη. Φυσικά, αυτές οι δοκιμές επικαλύπτονται, ειδικά όταν χρησιμοποιείται η αυξητική ανάπτυξη και το σύστημα που κυκλοφορεί είναι ημιτελές. Γενικά, προτεραιότητα στη δοκιμή ενοποίησης έχει η ανακάλυψη ατελειών στο σύστημα και προτεραιότητα στη δοκιμή συστήματος έχει η επικύρωση ότι το σύστημα ικανοποιεί τις απαιτήσεις του. Ωστόσο, στην πράξη, στη διάρκεια και των δύο αυτών διαδικασιών, πραγματοποιούνται κάποιες δοκιμές επικύρωσης και κάποιες δοκιμές για ατέλειες.

4.3 Επικύρωση Κρίσιμων Συστημάτων

Η επαλήθευση και επικύρωση ενός κρίσιμου συστήματος έχει, προφανώς, πολλά κοινά στοιχεία με την επικύρωση οποιουδήποτε άλλου συστήματος. Οι διαδικασίες E & E B (επαλήθευση και επικύρωση) θα πρέπει να αποδεικνύουν ότι το σύστημα ικανοποιεί την προδιαγραφή του, και ότι η συμπεριφορά και οι υπηρεσίες του συστήματος υποστηρίζουν τις απαιτήσεις του πελάτη. Για κρίσιμα συστήματα, όπου απαιτείται ένα υψηλό επίπεδο φερεγγυότητας, απαιτούνται



επιπλέον δοκιμές και ανάλυση για την παραγωγή τεκμηρίων ότι το σύστημα αξίζει της εμπιστοσύνης σας. Υπάρχουν δύο λόγοι γιατί θα πρέπει να κάνει κανείς:

1. **Κόστος αστοχιών:** Το κόστος και οι συνέπειες από αστοχίες κρίσιμων συστημάτων είναι ενδεχομένως πολύ μεγαλύτερες από ότι στα μη κρίσιμα συστήματα. Ελαττώνετε τους κινδύνους αστοχίας του συστήματος δαπανώντας περισσότερα στην επαλήθευση και επικύρωση του συστήματος
2. **Επικύρωση ιδιοτήτων φερεγγυότητας:** Μπορεί να χρειαστεί να γίνει δέσμευση απέναντι στους πελάτες ότι το σύστημα ικανοποιεί τις καθορισμένες απαιτήσεις φερεγγυότητας (διαθεσιμότητα, αξιοπιστία, ασφάλεια, και προστασία από εξωτερικούς κινδύνους). Η αξιολόγηση αυτών των χαρακτηριστικών φερεγγυότητας απαιτεί ειδικές δραστηριότητες E & E

Γι' αυτούς τους λόγους, τα κόστη E & E για κρίσιμα συστήματα είναι συνήθως πολύ υψηλότερα από αυτά για άλλες κατηγορίες συστημάτων. Είναι φυσιολογικό η E & E να αντιστοιχεί σε περισσότερο από το 50% του συνολικού κόστους ανάπτυξης για τα κρίσιμα συστήματα λογισμικού. Αυτό το κόστος φυσικά είναι δικαιολογημένο, αν πρόκειται να αποφευχθεί μια δαπανηρή αστοχία συστήματος.

Αν και οι διαδικασίες επικύρωσης κρίσιμων συστημάτων εστιάζουν κυρίως στην επικύρωση ενός συστήματος, σχετικές δραστηριότητες θα πρέπει να επαληθεύσουν ότι έχουν ακολουθηθεί καθορισμένες διαδικασίες ανάπτυξης του συστήματος.

Η διαδικασία μέτρησης της αξιοπιστίας ενός συστήματος φαίνεται στο σχήμα 6. Αυτή η διαδικασία περιλαμβάνει τέσσερα στάδια:

1. Ξεκινάτε με τη μελέτη υφιστάμενων συστημάτων του ίδιου τύπου για να εδραιώσετε ένα λειτουργικό προφίλ. Το λειτουργικό προφίλ προσδιορίζει κατηγορίες εισόδων στο σύστημα και την πιθανότητα να υπάρξουν αυτές οι εισοδοί στην κανονική χρήση.
2. Έπειτα κατασκευάζετε ένα σύνολο δοκιμαστικών δεδομένων που να αντικατοπτρίζουν το λειτουργικό προφίλ. Αυτό σημαίνει ότι

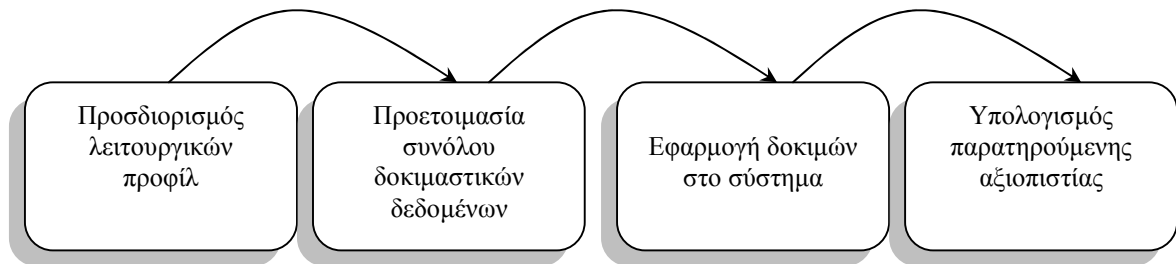


δημιουργείτε δοκιμαστικά δεδομένα με την ίδια πιθανότητα κατανομής των δοκιμαστικών δεδομένων για το σύστημα που έχετε μελετήσει.

3. Δοκιμάζετε το σύστημα χρησιμοποιώντας αυτά τα δεδομένα και τον αριθμό και τον τύπο των αστοχιών που συμβαίνουν.
4. Αφού παρατηρήσετε ένα στατιστικά σημαντικό αριθμό αστοχιών, μπορείτε να υπολογίσετε την αξιοπιστία του λογισμικού και να επεξεργαστείτε την κατάλληλη τιμή μέτρησης της αξιοπιστίας.

Αυτή η προσέγγιση ονομάζεται μερικές φορές στατιστική δοκιμή. Ο σκοπός των στατιστικών δοκιμών είναι να εκτιμηθεί η αξιοπιστία ενός συστήματος.

Σχήμα 6: Η διαδικασία μέτρησης της αξιοπιστίας



Κεφάλαιο Πέμπτο

Ο ρόλος της Διαχείρισης



Κεφάλαιο Πέμπτο

Ο ρόλος της Διαχείρισης

5.1 Διαχείριση Ανθρώπων

Οι άνθρωποι που εργάζονται σε έναν οργανισμό λογισμικού είναι το σημαντικότερο περιουσιακό του στοιχείο. Αντιπροσωπεύουν το πνευματικό κεφάλαιο, και εξαρτάται από τους διαχειριστές λογισμικού να διασφαλίσουν ότι ο οργανισμός έχει το μεγαλύτερο κέρδος από την επένδυση του στους ανθρώπους. Στις επιτυχημένες εταιρίες και οικονομίες, αυτό επιτυγχάνεται όταν οι άνθρωποι απολαμβάνουν το σεβασμό του οργανισμού. Θα πρέπει να έχουν ένα επίπεδο ευθυνών και ανταμοιβής που να αντικατοπτρίζει τα προσόντα τους.

Έτσι, η αποτελεσματική διαχείριση αφορά τη διαχείριση των ανθρώπων ενός οργανισμού. Οι διευθυντές έργων πρέπει να λύνουν τεχνικά και μη τεχνικά προβλήματα χρησιμοποιώντας τους ανθρώπους της ομάδας τους με τον πιο αποτελεσματικό τρόπο. Πρέπει να τους παροτρύνουν, να σχεδιάζουν και να οργανώνουν τη δουλειά τους και να διασφαλίζουν ότι η δουλειά γίνεται σωστά. Η κακή διαχείριση ανθρώπων είναι ένας από τους πιο σημαντικούς παράγοντες που συνεισφέρουν στην αποτυχία έργων.

Δυστυχώς, η κακή ηγεσία είναι κάτι πολύ συνηθισμένο στη βιομηχανία λογισμικού. Οι διαχειριστές παραλείπουν να λάβουν υπόψη τους περιορισμούς των ατόμων και επιβάλλουν εξωπραγματικές προθεσμίες στις ομάδες έργων. Εξισώνουν τη διαχείριση με τις συσκέψεις και δεν επιτρέπουν σε αυτούς που συμμετέχουν στις συναντήσεις να συνεισφέρουν στο έργο. Μπορεί να δέχονται νέες απαιτήσεις χωρίς κατάλληλη ανάλυση του τι σημαίνει αυτό για την ομάδα του έργου. Μερικές φορές βλέπουν το ρόλο τους σαν κάποιον που εκμεταλλεύεται το προσωπικό και όχι που θα συνεργαστεί μαζί τους για να προσδιορίσει πώς η δουλειά τους θα συνεισφέρει και στους προσωπικούς στόχους και στους στόχους του οργανισμού.



Σύμφωνα με τον Sommerville, (2007), υπάρχουν τέσσερις κρίσιμοι παράγοντες στη διαχείριση ανθρώπων:

1. **Συνέπεια:** Οι άνθρωποι σε μια ομάδα έργου θα πρέπει να αντιμετωπίζονται όλοι με παρόμοιο τρόπο. Αν και κανείς δεν περιμένει όλες οι ανταμοιβές να είναι ίδιες, δε θα πρέπει να αισθάνονται ότι υποτιμείται η προσφορά τους στον οργανισμό.
2. **Σεβασμός:** Διαφορετικοί άνθρωποι έχουν διαφορετικές ικανότητες και οι διαχειριστές θα πρέπει να σέβονται αυτές τις διαφορές. Σε όλα τα μέλη της ομάδας θα πρέπει να δίνεται η ευκαιρία να προσφέρουν. Φυσικά, σε ορισμένες περιπτώσεις, θα ανακαλύψετε ότι κάποιιοι δεν ταιριάζουν σε μια ομάδα και δεν μπορούν να συνεχίσουν, αλλά είναι σημαντικό να μη βγει βιαστικά ένα τέτοιο συμπέρασμα.
3. **Συμμετοχή:** Οι άνθρωποι προσφέρουν αποτελεσματικά όταν αισθάνονται ότι οι άλλοι τους ακούν και λαμβάνουν υπόψη τις προτάσεις τους. Είναι σημαντικό να αναπτυχθεί ένα εργασιακό περιβάλλον όπου εξετάζονται όλες οι απόψεις, ακόμη και του πιο κατώτερου ιεραρχικά προσωπικού.
4. **Ειλικρίνεια.** Σαν διαχειριστής, θα πρέπει πάντοτε να είστε ειλικρινείς για το τι πάει καλά και τι πάει στραβά στην ομάδα. Θα πρέπει επίσης να είστε ειλικρινείς για το επίπεδο των τεχνικών σας γνώσεων

Επιλογή Προσωπικού

Μία από τις πιο σημαντικές εργασίες διαχείρισης έργων είναι η επιλογή ομάδας. Σε εξαιρετικές περιπτώσεις, οι διευθυντές έργων μπορούν να διορίσουν ανθρώπους που είναι οι πιο κατάλληλοι για την εργασία, ανεξάρτητα από τις άλλες υποχρεώσεις τους ή από θέματα προϋπολογισμού. Πιο συχνά όμως, οι διευθυντές έργων δεν έχουν ελεύθερη επιλογή προσωπικού. Μπορεί να πρέπει να χρησιμοποιήσουν οποιοδήποτε είναι διαθέσιμος στον οργανισμό, να πρέπει να βρουν ανθρώπους πολύ γρήγορα και μπορεί να έχουν περιορισμένο προϋπολογισμό. Οι περιορισμοί στον προϋπολογισμό μπορεί να περιορίζουν τον αριθμό έμπειρων και ακριβών μηχανικών που θα είναι διαθέσιμοι να εργαστούν στο έργο.



Η απόφαση για το ποιος θα διοριστεί σε ένα έργο συνήθως λαμβάνεται με τη χρήση τριών τύπων πληροφοριών:

- § Πληροφορίες που παρέχουν οι υποψήφιοι για την κατάρτιση και την εμπειρία τους (το βιογραφικό τους).
- § Πληροφορίες που αποκτώνται με συνεντεύξεις των υποψηφίων. Οι συνεντεύξεις μπορούν να σας δώσουν μια καλή εντύπωση αν ένας υποψήφιος είναι καλός στην επικοινωνία και αν έχει καλές κοινωνικές σχέσεις.
- § Συστάσεις από ανθρώπους για τους οποίους έχουν εργαστεί οι υποψήφιοι. Αυτό μπορεί να είναι αποτελεσματικό όταν γνωρίζετε αυτούς που κάνουν τις συστάσεις.

5.2 Κόστος

Η εκτίμηση του κόστους διαχείρισης έργων πληροφορικής αφορά την απάντηση των ακόλουθων ερωτήσεων:

1. *Πόση προσπάθεια απαιτείται για την ολοκλήρωση κάθε δραστηριότητας;*
2. *Πόσος ημερολογιακός χρόνος χρειάζεται για την ολοκλήρωση κάθε δραστηριότητας;*
3. *Ποιο είναι το συνολικό κόστος κάθε δραστηριότητας;*

Κανονικά, η εκτίμηση του κόστους και ο χρονοπρογραμματισμός ενός έργου πραγματοποιούνται μαζί. Τα κόστη της ανάπτυξης είναι κυρίως τα κόστη της προσπάθειας που καταβάλλεται, οπότε ο υπολογισμός της προσπάθειας χρησιμοποιείται και για την εκτίμηση του κόστους και για την εκτίμηση του χρονοδιαγράμματος. Ωστόσο, μπορεί να χρειαστεί να κάνει κανείς κάποιους υπολογισμούς κόστους πριν καταλήξει σε λεπτομερή χρονοδιαγράμματα. Αυτές οι αρχικές εκτιμήσεις μπορεί να χρησιμοποιηθούν για να στοιχειοθετηθεί ένας προϋπολογισμός για το έργο ή για να καθοριστεί η τιμή του λογισμικού για κάποιον πελάτη.

Υπάρχουν τρεις παράμετροι που εμπλέκονται στον υπολογισμό του συνολικού κόστους ενός έργου ανάπτυξης λογισμικού:



- § Κόστη υλικού και λογισμικού, συμπεριλαμβανομένης της συντήρησης
- § Κόστη ταξιδιών και εκπαίδευσης
- § Κόστη προσπάθειας (τα κόστη για την πληρωμή των μηχανικών λογισμικού).

Για τα περισσότερα έργα, το κυρίαρχο κόστος είναι το κόστος της προσπάθειας. Υπολογιστές αρκετά ισχυροί για την ανάπτυξη λογισμικού, είναι σχετικά φθηνοί. Αν και μπορεί να χρειαστούν εκτεταμένα κόστη ταξιδιών για ένα έργο που αναπτύσσεται σε διαφορετικές τοποθεσίες, τα κόστη ταξιδιών συνήθως αποτελούν ένα μικρό μέρος του κόστους της προσπάθειας. Η χρήση ηλεκτρονικών συστημάτων επικοινωνίας όπως το ηλεκτρονικό ταχυδρομείο, οι κοινόχρηστοι ιστοτόποι και οι τηλεδιάσκεψεις μπορούν να μειώσουν σημαντικά τα αναγκαία ταξίδια.

Εικόνα 7: Τηλεδιάσκεψη



Πηγή: <http://img.youtube.com/vi/GWTtq4d1DWw/0.jpg>

Η χρήση ηλεκτρονικών συσκευέων σημαίνει επίσης ότι μειώνεται ο χρόνος σε ταξίδια και μπορεί να χρησιμοποιηθεί πιο παραγωγικά στην ανάπτυξη



λογισμικού, Σε ένα έργο που εργάσθηκα, η αλλαγή των συσκέψεων δια ζώσης με τηλεδιασκέψεις, μείωσε τα έξοδα και το χρόνο σε ταξίδια σχεδόν κατά 50%.

Τα κόστη προσπάθειας δεν είναι απλώς οι μισθοί των μηχανικών λογισμικού που εμπλέκονται στο έργο. Οι οργανισμοί υπολογίζουν τα κόστη προσπάθειας με βάση το κόστος επιβάρυνσης, όπου λαμβάνουν το συνολικό κόστος της λειτουργίας του οργανισμού και το διαιρούν με τον αριθμό των παραγωγικών υπαλλήλων. Έτσι, τα παρακάτω κόστη αποτελούν μέρος του συνολικού κόστους προσπάθειας:

1. Κόστη για τον εφοδιασμό, τη θέρμανση και το φωτισμό του χώρου των γραφείων
2. Κόστη για το προσωπικά υποστήριξης όπως λογιστές, διαχειριστές, διαχειριστές συστήματος, καθαρίστριες και τεχνικοί
3. Κόστη για τη δικτύωση και τις επικοινωνίες
4. Κόστη για κεντρικές λειτουργίες όπως η λειτουργία βιβλιοθήκης ή χώρων αναψυχής
5. Κόστη για την Κοινωνική Ασφάλιση και για ευεργετήματα προς τους υπαλλήλους όπως συντάξεις και ιατρική ασφάλεια.

Αυτός ο παράγοντας επιβάρυνσης είναι συνήθως τουλάχιστον διπλάσιος από τον μισθό του μηχανικού λογισμικού, ανάλογα με το μέγεθος του οργανισμού και τις αντίστοιχες επιβαρύνσεις. Έτσι, αν μια εταιρία αμείβει με 90.000\$ το χρόνο ένα μηχανικό λογισμικού, τα συνολικά κόστη είναι τουλάχιστον 180.000\$ το χρόνο ή 15.000\$ το μήνα.

Αφού ξεκινήσει ένα έργο, οι διαχειριστές του έργου θα πρέπει να ανανεώνουν τακτικά τις εκτιμήσεις τους για το κόστος και το χρονοδιάγραμμα. Αυτό βοηθά τη διαδικασία προγραμματισμού και την αποτελεσματική χρήση των πόρων. Αν οι πραγματικές δαπάνες είναι σημαντικά μεγαλύτερες από τις εκτιμήσεις, τότε ο διευθυντής έργου πρέπει να προβεί σε κάποιες ενέργειες. Αυτό



μπορεί να σημαίνει τη χρήση επιπλέον πόρων για το έργο ή την τροποποίηση της εργασίας που πρέπει να γίνει.

Η κοστολόγηση λογισμικού θα πρέπει να γίνεται αντικειμενικά με σκοπό την ακριβή πρόβλεψη του κόστους της ανάπτυξης του λογισμικού. Αν το κόστος ενός έργου έχει υπολογιστεί σαν μέρος μιας προσφοράς έργου προς κάποιον πελάτη, θα πρέπει να ληφθεί μία απόφαση για την τιμή που θα προσφερθεί στον πελάτη. Κλασικά, η τιμή είναι απλώς το κόστος συν το κέρδος. Ωστόσο, η σχέση ανάμεσα στο κόστος ενός έργου και στην τιμή για τον πελάτη δεν είναι συνήθως τόσο απλή.

Η τιμολόγηση λογισμικού πρέπει να λάβει υπόψη τους ευρύτερους οργανωτικούς, οικονομικούς, πολιτικούς και επιχειρηματικούς παράγοντες. Επομένως, μπορεί να μην υπάρχει μια απλή σχέση ανάμεσα στην τιμή για τον πελάτη του λογισμικού και στα κόστη ανάπτυξης. Λόγω των οργανωτικών παραγόντων που εμπλέκονται, η τιμολόγηση ενός έργου θα πρέπει να περιλαμβάνει τα ανώτερα διευθυντικά στελέχη δηλαδή, αυτούς που παίρνουν τις στρατηγικές αποφάσεις, καθώς και τους διευθυντές έργων λογισμικού.

5.3 Διαχείριση Ποιότητας

Βασικό αντικείμενο της Ερευνητικής Ομάδας Ποιότητας Λογισμικού θα μπορούσαμε να πούμε ότι είναι η αξιολόγηση και εφαρμογή των κατάλληλων τεχνικών για τη συλλογή της άποψης των χρηστών σχετικά με την ποιότητα προγραμμάτων λογισμικού. Ο χρήστης θα πρέπει να έχει γνώση του τι ζητάει από την ομάδα ανάπτυξης λογισμικού και να τους το μεταφέρει, ώστε οι προδιαγραφές να είναι όσο το δυνατόν πιο κοντά στο κατάλληλο αποτέλεσμα.

Το αντικείμενο εργασίας της Ερευνητικής Ομάδας Ποιότητας Λογισμικού είναι η διεξαγωγή μετρήσεων ποιότητας λογισμικού με τη χρήση μετρικών λογισμικού και η μελέτη και ανάλυση των αποτελεσμάτων. Η έρευνα της ομάδας στοχεύει όσο τον δυνατόν περισσότερο στην ανάπτυξη και εφαρμογή μετρικών λογισμικού και στη μελέτη της συσχέτισης εσωτερικών και εξωτερικών (user perceived) μετρήσεων.



Ένα άλλο σημείο που επικεντρώνεται η ομάδα ποιότητας λογισμικού είναι η μελέτη των εσωτερικών μετρικών ποιότητας λογισμικού με τη χρήση μεταμετρικών ποιότητας λογισμικού. Ακόμα, ερευνητικό αντικείμενο αποτελεί η μελέτη, η χρήση και –κυρίως - η ανάπτυξη εργαλείων για τη διεξαγωγή των μετρήσεων ποιότητας λογισμικού, καθώς και η ανάλυση των αποτελεσμάτων.

5.4 Διαχείριση Ασφάλειας

Η ευρεία χρήση του internet στη δεκαετία του 1990 παρουσίασε μια νέα πρόκληση για τους μηχανικούς λογισμικού—τη σχεδίαση και υλοποίηση συστημάτων που είναι ασφαλή. Καθώς όλο και περισσότερα συστήματα συνδέονταν με το internet, επινοήθηκε μια ποικιλία εξωτερικών επιθέσεων για την απειλή αυτών των συστημάτων. Τα προβλήματα της παραγωγής φερέγγυοι των συστημάτων αυξήθηκαν σε τεράστιο βαθμό.

Οι μηχανικοί συστημάτων έπρεπε να λάβουν υπόψη τους απειλές από κακόβουλους και τεχνικά εκπαιδευμένους επιτιθέμενους όπως και προβλήματα που προκύπτουν από ακούσια λάθη στη διαδικασία ανάπτυξης.

Σήμερα είναι απαραίτητη η σχεδίαση συστημάτων που αντέχουν σε εξωτερικές επιθέσεις και μπορούν να ανακάμψουν από τέτοιες επιθέσεις. Χωρίς προφυλάξεις ασφάλειας, είναι σχεδόν αναπόφευκτο ότι οι επιτιθέμενοι θα υπονομεύσουν ένα δικτυωμένο σύστημα. Μπορεί να κάνουν κακή χρήση του υλικού συστήματος, να υποκλέψουν εμπιστευτικά δεδομένα ή να διαταράξουν τις υπηρεσίες που παρέχονται από το σύστημα. Έτσι, η τεχνολογία ασφάλειας συστημάτων αποτελεί μια πλευρά του σχεδιασμού συστημάτων με όλο και πιο μεγάλη σπουδαιότητα.

Η τεχνολογία ασφάλειας ενδιαφέρεται για το πώς θα γίνει η ανάπτυξη και συντήρηση συστημάτων που θα μπορούν να αντισταθούν σε κακόβουλες επιθέσεις που έχουν στόχο να προκαλέσουν ζημιές σε ένα σύστημα υπολογιστών ή στα δεδομένα ενός συστήματος. Η τεχνολογία ασφάλειας είναι μέρος του πιο γενικού πεδίου του σχεδιασμού υπολογιστών. Ο τομέας αυτός έχει αποκτήσει την



πρώτη προτεραιότητα για επιχειρήσεις και μεμονωμένα άτομα καθώς όλο και περισσότεροι παράνομοι προσπαθούν να εκμεταλλευτούν δικτυωμένα συστήματα για παράνομους σκοπούς. Οι μηχανικοί λογισμικού θα πρέπει να είναι ενήμεροι για τις απειλές ασφάλειας που αντιμετωπίζουν τα συστήματα και τους τρόπους με τους οποίους αυτές οι απειλές μπορούν να εξουδετερωθούν.

Όταν κάποιος εξετάζει θέματα ασφάλειας, θα πρέπει να λάβει υπόψη του και το λογισμικό εφαρμογών (το σύστημα ελέγχου, το σύστημα πληροφοριών κ.λπ.) και την υποδομή στην οποία το σύστημα έχει δομηθεί. Η υποδομή για περίπλοκες εφαρμογές περιλαμβάνει την πλατφόρμα του λειτουργικού συστήματος, Στην πραγματικότητα, η πλειοψηφία των επιτιθέμενων εστιάζει στις υποδομές ενός συστήματος επειδή τα συστατικά στοιχεία (π.χ., φυλλομετρητές ιστού) είναι καλά γνωστά και ευρέως διαθέσιμα.

Τα επίπεδα συστήματος όπου μπορεί να υπονομευθεί η ασφάλεια

Εφαρμογή Επαναχρησιμοποιούμενα συστατικά στοιχεία και βιβλιοθήκες

Ενδιάμεσο λογισμικό

Διαχείριση βάσεων δεδομένων

Γενικές, κοινόχρηστες εφαρμογές (Φυλλομετρητές, ηλεκτρονικό ταχυδρομείο κ.λπ.)

Λειτουργικό σύστημα

Στην πράξη, υπάρχει μια σημαντική διαφορά μεταξύ της ασφάλειας εφαρμογών και της ασφάλειας υποδομών:

1. Η ασφάλεια εφαρμογών είναι ένα πρόβλημα της τεχνολογίας λογισμικού όπου οι μηχανικοί λογισμικού θα πρέπει να διασφαλίσουν ότι το σύστημα είναι σχεδιασμένο να ανθίσταται σε επιθέσεις.
2. Η ασφάλεια υποδομών είναι ένα πρόβλημα διαχείρισης συστημάτων όπου οι διαχειριστές συστημάτων θα πρέπει να διασφαλίσουν ότι το σύστημα είναι διευθετημένο να ανθίσταται σε επιθέσεις. Οι διαχειριστές συστημάτων πρέπει να διευθετήσουν την υποδομή ώστε να έχουν την πιο αποτελεσματική χρήση των όποιων χαρακτηριστικών ασφάλειας διαθέτει η υποδομή.



Η διαχείριση της ασφάλειας δεν είναι μία μόνο εργασία αλλά περιλαμβάνει ένα φάσμα δραστηριοτήτων όπως διαχείριση χρηστών και αδειών, διανομή και συντήρηση του συστήματος λογισμικού, και παρακολούθηση, ανίχνευση και ανάκαμψη από επιθέσεις:

1. Η διαχείριση χρηστών και αδειών περιλαμβάνει την προσθήκη και αφαίρεση χρηστών από το σύστημα, τη διασφάλιση ότι είναι ενεργοί οι κατάλληλοι μηχανισμοί πιστοποίησης της ταυτότητας των χρηστών και τη διευθέτηση των αδειών στο σύστημα έτσι ώστε οι χρήστες να έχουν πρόσβαση μόνο στους πόρους που χρειάζονται.
2. Η διανομή και συντήρηση ενός συστήματος λογισμικού περιλαμβάνει την εγκατάσταση του συστήματος λογισμικού και του ενδιάμεσου λογισμικού του συστήματος, και την κατάλληλη διευθέτηση τους ώστε να αποφευχθούν οι ευπάθειες ασφάλειας. Περιλαμβάνει επίσης την τακτική ενημέρωση αυτού του λογισμικού με νέες εκδόσεις ή επιδιορθώσεις που διορθώνουν προβλήματα ασφάλειας που έχουν ανακαλυφθεί.
3. Η παρακολούθηση, ανίχνευση και ανάκαμψη από επιθέσεις περιλαμβάνει δραστηριότητες που παρακολουθούν το σύστημα για μη εξουσιοδοτημένη πρόσβαση, ανιχνεύουν και θέτουν σε εφαρμογή στρατηγικές αντίστασης στις επιθέσεις και δραστηριότητες δημιουργίας εφεδρικών αντιγράφων έτσι ώστε να μπορεί να επαναληφθεί η κανονική λειτουργία μετά από μία εξωτερική επίθεση.



Κεφάλαιο Έκτο

Έρευνα – Συμπεράσματα



Κεφάλαιο Έκτο

Έρευνα – Συμπεράσματα

6.1 Έρευνα

ΜΕΡΟΣ 1^ο : ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΕΡΓΟΥ

Η Διοίκηση του της επιχείρησης «Χ» στα πλαίσια των ευρύτερων προσπάθειών της για την λειτουργική και θεσμική αναβάθμιση του Οργανισμού, αποφάσισε την εφαρμογή ενός ολοκληρωμένου πληροφοριακού συστήματος.

Ως πρώτο βήμα προς την κατεύθυνση αυτή η επιχείρηση «Χ» κατάρτισε το **Στρατηγικό Πρόγραμμα Πληροφορικής** (Information Systems Master Plan.) Σε συνέχεια του στρατηγικού Προγράμματος και εκπονήθηκαν οι ακόλουθες αναλυτικές τεχνικές μελέτες:

- 1. Μελέτη Αναλυτικών Προδιαγραφών Οικονομικού Συστήματος:** Για το οικονομικό σύστημα, η επιχείρηση «Χ» έλαβε την απόφαση προμήθειας έτοιμου πακέτου λογισμικού (software package) και προσαρμογής του με βάση τις ιδιαιτερότητες της
- 2. Μελέτη Αναλυτικών Προδιαγραφών Συστήματος Κατάρτισης Προϋπολο- γισμού:** Η μελέτη προδιαγράφει ένα σύστημα υποστήριξης αποφάσεων (DSS) για την κατάρτιση του προϋπολογισμού της επιχείρησης, το οποίο παρέχει εκτιμήσεις σχετικά με τις επιπτώσεις που θα έχουν μακρο-οικονομικές αλλαγές ή πολιτικές επιλογές στον προϋπολογισμό και τη διαχείριση διαθεσίμων της εν λόγω επιχείρησης.

Σκοπός του Έργου:

Ο βασικός στόχος της επιχείρησης που επιδιώκεται να επιτευχθεί μέσω της υλοποίησης του Ολοκληρωμένου Πληροφοριακού Συστήματος είναι η αναδιάρθρωση και αναβάθμιση της εταιρίας, πιο συγκεκριμένα:



- Η βελτίωση των οικονομικών μεγεθών
- Βελτίωση της διαχείρισης των πόρων του Ιδρύματος
- Μεγιστοποίηση Εσόδων
- Ελαχιστοποίηση Εξόδων
- Διασφάλιση της βιωσιμότητας της επιχείρησης
- Μείωση της γραφειοκρατίας
- Βελτίωση της πληροφόρησης των συναλλασσομένων και της ταχύτητας εξυπηρέτησής τους
- Ελαχιστοποίηση λαθών
- Μείωση κόστους
- Βελτίωση της ποιότητας πληροφόρησης
- Αναβάθμιση εσωτερικού εργασιακού περιβάλλοντος
- Εξασφάλιση αποτελεσματικότητας στην διεκπεραίωση καθημερινών εργασιών
- Διαχείριση και αξιοποίηση ανθρωπίνου δυναμικού
- Αξιοποίηση σύγχρονων τεχνολογιών πληροφορικής

ΤΡΟΠΟΙ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ ΟΛΟΚΛΗΡΩΜΕΝΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η επιχείρηση «Χ» υλοποίησε δύο διαγωνισμούς.

- Ο πρώτος από αυτούς συσχετίζεται με την παρούσα περιγραφή.
- Ο δεύτερος διαγωνισμός, ο οποίος θα προκηρυχθεί μετά την ολοκλήρωση της πιλοτικής φάσης λειτουργίας, θα αφορά μόνο την προμήθεια εξοπλισμού (hardware) ο οποίος θα εξοπλίσει όλες τις διοικητικές μονάδες της επιχείρησης και στον οποίο θα εγκατασταθεί και θα εκτελείται κατά περίπτωση το λογισμικό που θα έχει αναπτυχθεί/αγοραστεί με τον πρώτο διαγωνισμό.

ΣΥΜΜΕΤΟΧΗ ΤΗΣ ΙΔΙΑΣ ΤΗΣ ΕΠΙΧΕΙΡΗΣΗΣ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΕΡΓΟΥ

Για την αποτελεσματική καθοδήγηση, την παροχή των απαραίτητων στοιχείων και την παρακολούθηση της εξέλιξης της πορείας του Έργου, η



επιχείρηση έχει ήδη δημιουργήσει μια οργανωτική δομή από διάφορες Επιτροπές και Ομάδες, τα βασικά στοιχεία των οποίων εκτίθενται στη συνέχεια. Τονίζεται ότι η επιχείρηση «Χ» θεωρεί απολύτως απαραίτητη την ενεργό συμμετοχή των στελεχών σε όλες τις φάσεις του Έργου, και αυτό προκειμένου:

- να εξασφαλιστεί η ανταπόκριση του τελικού προϊόντος στις πραγματικές ανάγκες της επιχείρησης,
- να εξασφαλιστεί η αποδοχή του συστήματος από τους μελλοντικούς χρήστες του και
- να επιτευχθεί η απαραίτητη μεταφορά τεχνογνωσίας ώστε να είναι δυνατή στη συνέχεια η διαχείριση και λειτουργία του ΟΠΣ-Χ από το προσωπικό της Χ.

ΣΤΡΑΤΗΓΙΚΗ ΕΠΙΤΡΟΠΗ ΚΑΙ ΕΚΤΕΛΕΣΤΙΚΗ ΕΠΙΤΡΟΠΗ ΕΠΙΧΕΙΡΗΣΗΣ Χ

Η Στρατηγική Επιτροπή είναι αρμόδια για τη διοίκηση του συνολικού έργου σε στρατηγικό επίπεδο και οι αρμοδιότητες της Στρατηγικής Επιτροπής περιλαμβάνουν:

- Επικύρωση κατευθύνσεων του έργου και του πλαισίου εργασιών (χρόνος, προϋπολογισμός, βασικοί περιορισμοί του έργου κλπ.), παρέμβαση στα σημεία ελέγχου του έργου, επίλυση τυχόν αντιθέσεων κλπ.
- Επικύρωση των αρμοδιοτήτων και των υποχρεώσεων της Επιτροπής και της Ομάδας Διαχείρισης.
- Μελέτη και αξιολόγηση προτάσεων της Ομάδας Διαχείρισης οι οποίες διαβιβάζονται, όποτε απαιτείται, μέσω της Εκτελεστικής Επιτροπής και λήψη των απαραίτητων αποφάσεων.
- Προώθηση των απαραίτητων ενεργειών για τη στελέχωση των μονάδων της Χ μετά από μελέτη των προτάσεων του ΣΥΜΒΟΥΛΟΥ και λήψη των απαραίτητων αποφάσεων.
- Ευθύνη για την τελική αποδοχή του έργου του ΣΥΜΒΟΥΛΟΥ, μετά από μελέτη των εισηγήσεων της Ομάδας Διαχείρισης.
- Δυνατότητα μονομερούς διακοπής της συνεργασίας με το ΣΥΜΒΟΥΛΟ σε περίπτωση που αυτός αποδεδειγμένα δεν ανταποκρίνεται στις συμβατικές υποχρεώσεις του.



- Λήψη των αποφάσεων στρατηγικού χαρακτήρα για το έργο.

ΟΜΑΔΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧΕΙΡΗΣΗΣ Χ

Η **Επιτροπή Διαχείρισης** (η οποία είναι οργανωτική μονάδα υλοποίησης του έργου) έχει δραστηριότητες εποπτείας, παρακολούθησης, ελέγχου και αξιολόγησης της υλοποίησης του συνολικού έργου σύμφωνα με τις προκαθορισμένες απαιτήσεις και προδιαγραφές και οι δραστηριότητές της αφορούν τους ακόλουθους τομείς:

- Τακτική παρακολούθηση και συντονισμός του έργου σε Τεχνικό και Λειτουργικό επίπεδο,
- Συντονισμός των εμπλεκόμενων στο έργο
- Διαχείριση του έργου και Διασφάλιση Ποιότητας
- Επισήμανση θεμάτων που αφορούν στη διαχείριση του ανθρώπινου δυναμικού
- Επικοινωνία επιμέρους έργων

Η Επιτροπή Διαχείρισης συνεδριάζει τακτικά κάθε μήνα και όποτε κριθεί απαραίτητο έκτακτα μετά από εισήγηση της Ομάδας Διαχείρισης ή του ΣΥΜΒΟΥΛΟΥ. Στις συνεδριάσεις της καλείται ο Διευθυντής έργου και ο Project Manager εκ μέρους του ΣΥΜΒΟΥΛΟΥ. Ο Project Manager έχει την ευθύνη για την τήρηση των πρακτικών και την τελική αποδοχή τους από τους συμμετέχοντες.

Η Επιτροπή Διαχείρισης, με βάση τις αποφάσεις και τις προτάσεις της Ομάδας Διαχείρισης συμμετέχει στην οργάνωση και εκτέλεση των δραστηριοτήτων οι οποίες περιγράφηκαν παραπάνω σε συνεργασία με τον ΣΥΜΒΟΥΛΟ. Η **Ομάδα Διαχείρισης** (η οποία είναι οργανωτική μονάδα με επιτελικό, συντονιστικό και αποφασιστικό χαρακτήρα) αποτελείται από ορισμένα μέλη της Επιτροπής Διαχείρισης ώστε να είναι μία μικρότερη και ευέλικτη οργανωτική μονάδα με συχνότερες τακτικές συνεδριάσεις.



ΣΥΜΜΕΤΟΧΗ ΤΟΥ ΣΥΜΒΟΥΛΟΥ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΕΡΓΟΥ ΤΗΣ ΕΠΙΧΕΙΡΗΣΗΣ Χ

Το οργανωτικό σχήμα του Συμβούλου διακρίνεται στα εξής επίπεδα:

- (1) **ΟΜΑΔΑ ΣΥΝΤΟΝΙΣΜΟΥ** που περιλαμβάνει την Επιτροπή Διοίκησης Έργου, το Διευθυντή Έργου και τους Ειδικούς και Τεχνικούς Εμπειρογνώμονες.
- (2) **PROJECT MANAGER (ΥΠΕΥΘΥΝΟΣ ΕΡΓΟΥ)**
- (3) **ΟΜΑΔΑ ΣΥΜΒΟΥΛΩΝ**

Η **Επιτροπή Διοίκησης έργου** είναι τριμελής και έχει ως αντικείμενο την παρακολούθηση του συνολικού έργου του ΣΥΜΒΟΥΛΟΥ. Ο **Διευθυντής έργου** συμβάλλει στο συντονισμό και τη λειτουργία της ένωσης με ενιαία μορφή, τη μεσοπρόθεσμη και μακροπρόθεσμη οργάνωση του. Οι **Ειδικοί και Τεχνικοί Εμπειρογνώμονες** συνεισφέρουν στο επίπεδο σχεδιασμού της οργάνωσης του έργου και σε ειδικά θέματα σχεδιασμού και υλοποίησης. Ο **Project Manager** (Υπεύθυνος έργου) αφιερώνεται αποκλειστικά στην τεχνική διοίκηση του έργου και την αποτελεσματική οργάνωση και συντονισμό της Ομάδας.

ΜΕΡΟΣ 2^ο : ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΟΠΣ «Χ»

Το μοντέλο της Τεχνικής Αρχιτεκτονικής του ΟΠΣ της επιχείρησης «Χ», όπως αρχικά έχει παρουσιαστεί στο Master Plan, βασίζεται στην αρχή της αποκέντρωσης των λειτουργιών και των δεδομένων. Όσον αφορά τις λειτουργίες, οι περισσότερες εκτελούνται σε επίπεδο Περιφερειακών και Τοπικών Υποκαταστημάτων . Πιο συγκεκριμένα:

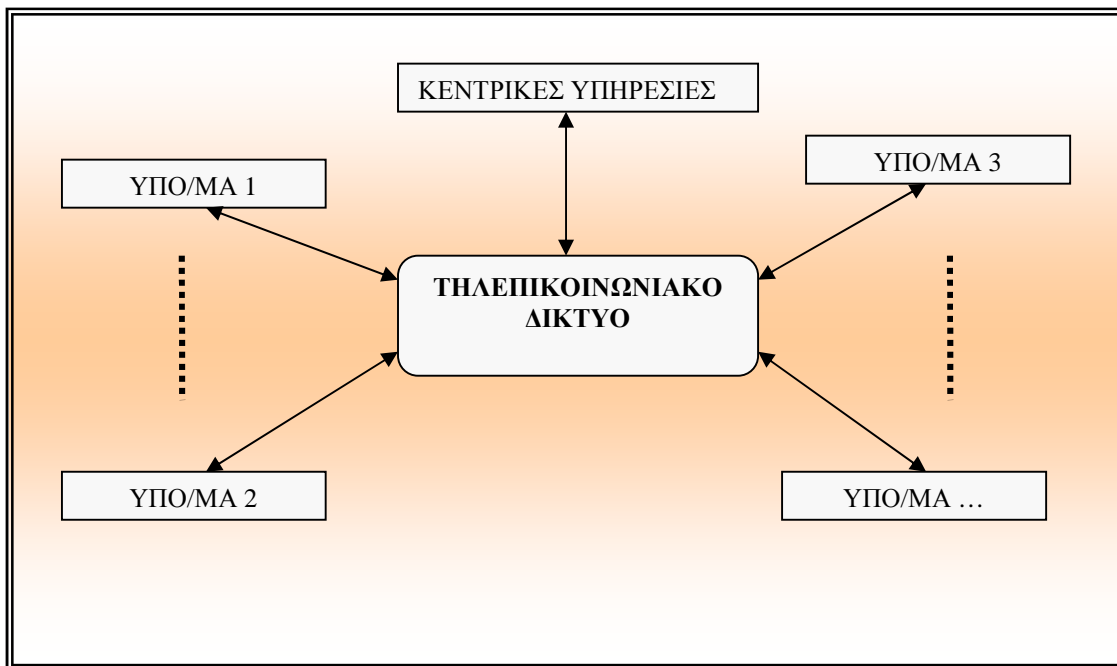
- Οι τρέχουσες λειτουργίες κάθε Συστήματος του ΟΠΣ με τις οποίες η «Χ» καλύπτει τις καθημερινές συναλλαγές του, εκτελούνται στα Τοπικά και Περιφερειακά Υποκαταστήματα ως on-line διαδικασίες.
- Ορισμένες λειτουργίες σημαντικού υπολογιστικού φορτίου (π.χ. οι λειτουργίες εισαγωγής δεδομένων), οι οποίες απαιτούν σημαντική



υπολογιστική δυνατότητα, εκτελούνται στα Περιφερειακά Υποκαταστήματα ως on-line ή batch διαδικασίες.

- Διαδικασίες batch (π.χ. μεγάλες εκτυπώσεις) εκτελούνται στα Περιφερειακά Υποκαταστήματα.

Στα Τοπικά Υποκ/ματα οι χρήστες είναι οργανωμένοι σε Τοπικά Δίκτυα τα οποία διασυνδέονται, με κατάλληλο τηλεπικοινωνιακό εξοπλισμό, με τα Περιφερειακά Υποκ/ματα.



Θα πρέπει να τονιστεί ότι η απευθείας επικοινωνία μεταξύ των επιμέρους οργανωτικών επιπέδων γίνεται μόνον από κάθε επίπεδο προς τα ανώτερα επίπεδα, δεν επιτρέπεται δηλαδή .οριζόντια επικοινωνία.. Έτσι, τα Τοπικά Υποκαταστήματα επικοινωνούν μόνο με το Περιφερειακό Υποκατάστημα στο οποίο υπάγονται (και μέσω αυτού με τις Κεντρικές Υπηρεσίες).

ΤΕΧΝΟΛΟΓΙΚΕΣ ΕΠΙΛΟΓΕΣ ΕΠΙΧΕΙΡΗΣΗΣ «Χ»

Η Τεχνική Αρχιτεκτονική του Ολοκληρωμένου Πληροφοριακού Συστήματος της «Χ ακολουθεί ορισμένες βασικές τεχνολογικές επιλογές, έτσι ώστε να ικανοποιούνται κατά τον καλύτερο τρόπο οι λειτουργικές απαιτήσεις της και να



εξασφαλίζεται η ανάπτυξη ενός σύγχρονου και ολοκληρωμένου περιβάλλοντος Πληροφορικής. Οι

σημαντικότερες από αυτές είναι:

- Ανοικτά Συστήματα
- Αρχιτεκτονική Client-Server
- Σχισιακές Βάσεις Δεδομένων (RDBMS)
- Graphical User Interfaces (GUI)
- Ενιαία τεχνολογική αρχιτεκτονική και πλατφόρμα ανάπτυξης/λειτουργίας
- Διεθνή τηλεπικοινωνιακά πρότυπα

ΜΕΡΟΣ 3^ο : ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Η ανάλυση, ο σχεδιασμός και η ανάπτυξη των εφαρμογών του πληροφοριακού συστήματος της επιχείρησης «Χ» θα βασιστούν στη χρήση εργαλείου CASE και της μεθοδολογίας Ανάλυσης και Σχεδίασης Πληροφοριακών Συστημάτων.

Για την ολοκληρωμένη διατύπωση των απαιτήσεων κρίνεται απαραίτητο να γίνει η ακόλουθη διάκριση: Μεμονωμένα εργαλεία CASE (CASE tools) υποστηρίζουν συνήθως μεμονωμένες δραστηριότητες της διαδικασίας Ανάπτυξης Λογισμικού ενώ ένα περιβάλλον CASE (CASE environment), αποτελούμενο από διάφορα CASE εργαλεία που αλληλεπιδρούν μεταξύ τους, υποστηρίζει όλες ή τις περισσότερες από τις διαδικασίες αυτές. Η διάκριση αυτή θα πρέπει να διατυπωθεί σαφώς εκ μέρους του αναδόχου, ο οποίος θα πρέπει να εντάξει το προτεινόμενο σύστημα σε μία από τις δύο κατηγορίες.

ΓΕΝΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ ΕΡΓΑΛΕΙΟΥ CASE

- A. Το προσφερόμενο εργαλείο ή περιβάλλον θα πρέπει:
- να είναι σύγχρονο από τεχνική άποψη (τελευταίο μοντέλο - version)
 - να προέρχεται από αξιόπιστη εταιρία ανάπτυξης λογισμικού



- να υποστηρίζει τα πρότυπα ανοικτής αρχιτεκτονικής UNIX, RDBMS, TCP/IP
- να έχει δοκιμαστεί στην αγορά σε αντίστοιχες περιπτώσεις
- να καλύπτεται από συνεχή τεχνική και εκπαιδευτική υποστήριξη

Β. Φιλικότητα (User Friendliness). Το σύστημα θα πρέπει να υποστηρίζει γραφικό περιβάλλον εργασίας (Graphical User Interface) και συμβολισμούς ώστε να είναι εύχρηστο τόσο από αναλυτές/ προγραμματιστές όσο και από μη εξειδικευμένους χρήστες.

Γ. Ευελιξία (Flexibility). Το σύστημα θα πρέπει να υποστηρίζει:

- Διάφορες δυνατότητες παρουσίασης με διαφορετικά επίπεδα λεπτομέρειας.
- Επιλογή στις μεθοδολογίες ανάλυσης (Object Oriented Analysis, Merise, κ.τ.λ.)

Δ. Πληρότητα και Συνέπεια (Completeness and Consistency). Το σύστημα θα πρέπει να υποστηρίζει έλεγχο λαθών (error-checking).

Ε. Υποστήριξη λειτουργικών διαδικασιών (security, backup, recovery και rollback)

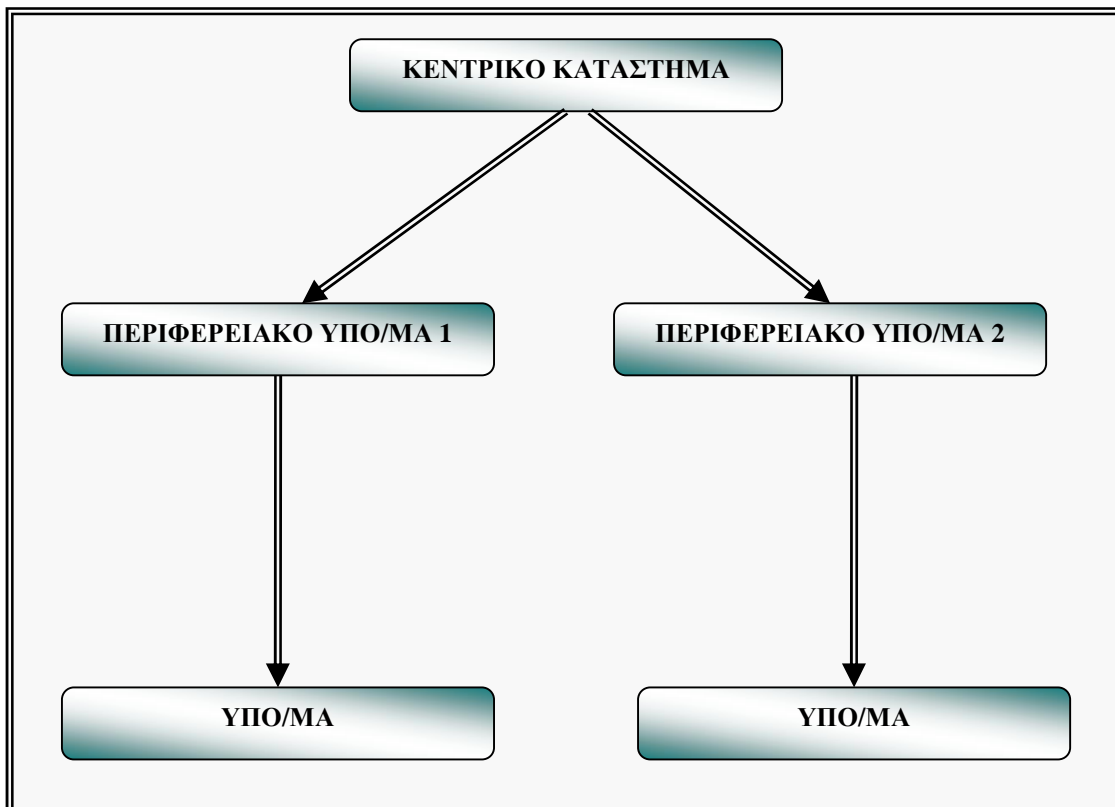
ΣΤ. Υποστήριξη της ελληνικής γλώσσας.

ΜΕΡΟΣ 4^ο : ΠΙΛΟΤΙΚΗ ΕΦΑΡΜΟΓΗ ΟΠΣ «Χ»

Σκοπός του πιλοτικού έργου είναι να δοκιμασθούν σε πραγματικές συνθήκες όλες οι λειτουργίες του ΟΠΣ, τόσο από πλευράς ικανοποίησης των απαιτήσεων που έχουν τεθεί, όσο και από πλευράς απόδοσής τους. Στα πλαίσια αυτά περιλαμβάνεται και η βαθμιαία εξοικείωση, τόσο των εργαζομένων της «Χ», όσο και των πελατών με αυτό, με τις νέες διαδικασίες. Ευνόητο είναι ότι το Πιλοτικό Έργο δεν αναιρεί, ούτε υποκαθιστά τη φάση δοκιμαστικής λειτουργίας του κάθε συστήματος, που πρέπει να έχει προηγηθεί κανονικά και περιλαμβάνει τον



έλεγχο ικανοποίησης των απαιτήσεων και τεχνικών προδιαγραφών που έχουν τεθεί από τη φάση του σχεδιασμού. Το Πιλοτικό Έργο τοποθετείται σε ανώτερο επίπεδο δοκιμαστικής λειτουργίας, αφού καλύπτει τη συνολική σύνθεση του ΟΠΣ, τον ολοκληρωμένο χαρακτήρα του (integration), τη συνεργασία των συστημάτων και υποσυστημάτων μεταξύ τους (interoperability), περιλαμβάνοντας και πολλά άλλα στοιχεία, πέραν των καθαρά τεχνικών. Οι πραγματικές συνθήκες κάτω από τις οποίες θα εκτελεστεί θα επιτρέψουν την εξαγωγή χρήσιμων συμπερασμάτων ως προς την επίτευξη των στρατηγικών στόχων που είχαν αρχικά τεθεί, την απόδοση και εγκυρότητα των νέων διαδικασιών, την αναβάθμιση της ποιότητας των υπηρεσιών προς τους πελάτες της «Χ» και τις νέες συνθήκες εργασίας του προσωπικού του.



Για την λειτουργία των Πληροφοριακών Συστημάτων του ΟΠΣ θα εγκατασταθούν τα εξής συστήματα:

- 1) Στις Κεντρικές Υπηρεσίες θα υπάρχει ο κεντρικός server, ο οποίος μαζί με τον Database server θα αποτελέσουν την .καρδιά. του ΟΠΣ. Στα



συστήματα αυτά θα βρίσκονται εγκατεστημένες οι κεντρικές εφαρμογές και τα master files των

- 2) Βάσεων Δεδομένων. Τα δύο συστήματα θα είναι συνδεδεμένα σε δίκτυο LAN υψηλής ταχύτητας (LAN του Κεντρικού Site), και θα μπορούν να μοιράζονται το σύστημα αποθήκευσης δεδομένων. Σε περίπτωση προβλήματος του ενός συστήματος, το άλλο σύστημα θα μπορεί να λειτουργήσει σαν backup server. Για την υποστήριξη του φορτίου κεντρικών εκτυπώσεων θα υπάρχει υποσύστημα εκτυπώσεων υψηλής ταχύτητας.
- 3) Η επικοινωνία των Κεντρικών servers με τις υπόλοιπες οργανωτικές μονάδες της «Χ» θα υλοποιείται με τη χρήση τηλεπικοινωνιακού εξοπλισμού που θα συνδέει το Κεντρικό Site με τα Περιφερειακά Υποκ/ματα μέσω μισθωμένων κυκλωμάτων (leased lines).
- 4) Σε κάθε ένα από τα Περιφερειακά Υποκαταστήματα θα εγκατασταθεί ένας Περιφερειακός server, δυναμικότητας ανάλογης του όγκου των συναλλαγών του Περιφερειακού, σύμφωνα με την τυπολογία που έχει ήδη παρουσιαστεί πιο πάνω.
- 5) Ο server θα στεγάζει τις εφαρμογές και τα δεδομένα που αφορούν την περιφέρεια, και θα υποστηρίζει το σύνολο του φορτίου online και batch συναλλαγών από όλα τα Τοπικά Υποκ/ματα και Παραρτήματα της περιφέρειας
- 6) Σε κάθε ένα από τα Τοπικά Υποκαταστήματα θα εγκατασταθεί ένα LAN, που θα περιλαμβάνει LAN server, σταθμούς εργασίας και τους ανάλογους client εκτυπωτές, και το οποίο θα συνδέεται με το Περιφερειακό Υποκ/μα με τον κατάλληλο τηλεπικοινωνιακό εξοπλισμό.

ΕΞΟΠΛΙΣΜΟΣ ΚΕΝΤΡΙΚΩΝ ΜΟΝΑΔΩΝ

Ο εξοπλισμός των Κεντρικών Υπηρεσιών περιλαμβάνει:

- LAN Κεντρικού Site το οποίο περιλαμβάνει:



- Κεντρικό Server
- DataBase Server
- Server Ανάπτυξης
- Υποσύστημα εκτυπώσεων υψηλής ταχύτητας
- Τηλεπικοινωνιακό εξοπλισμό (routers, modems, κλπ.) για τη διασύνδεση με τα LAN των Κεντρικών Υπηρεσιών και τα Περιφερειακά Υποκ/ματα

- LANs των sites των Κεντρικών Υπηρεσιών που περιλαμβάνουν:
 - LAN Server(s)
 - Σταθμούς Εργασίας
 - Εκτυπωτές

- Το Λογισμικό των Συστημάτων περιλαμβάνει:
 - Λειτουργικό Σύστημα UNIX
 - RDBMS
 - Transaction processing monitor
 - Λογισμικό διαχείρισης δικτύου
 - Λογισμικό clients
 - Εξοπλισμό, λογισμικό και τηλεπικοινωνιακή υποδομή για την υποστήριξη συστήματος X.400 (Message Transfer Agent και Message Store), X.500 (Directory services), EDI και X.25.

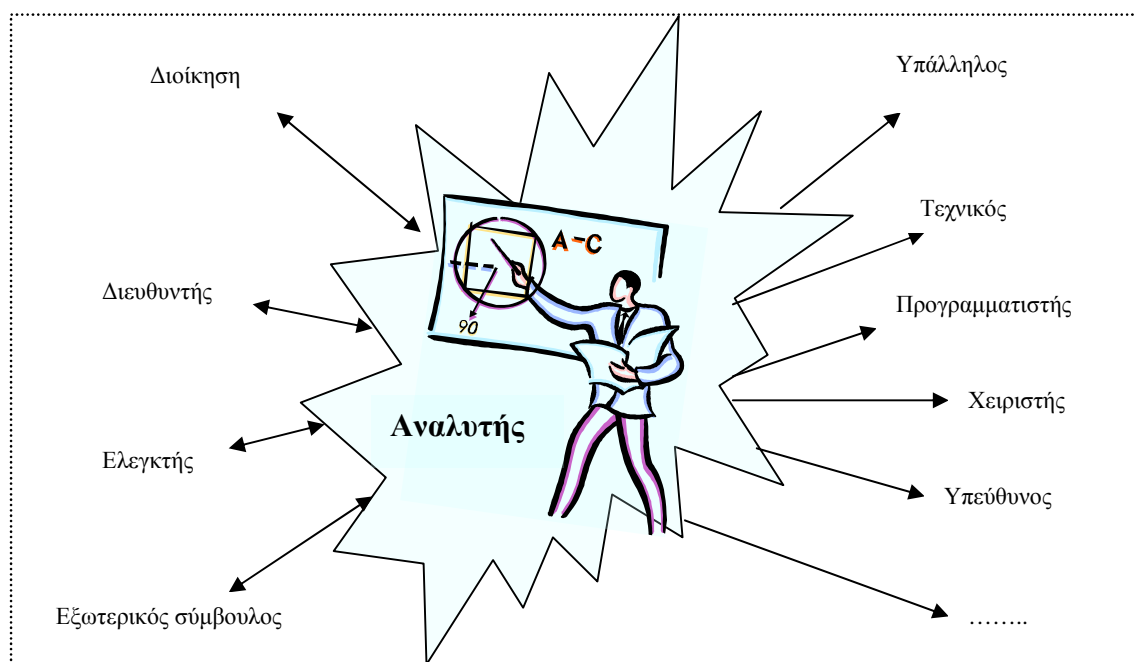
6.2 Συμπεράσματα

Όλες σχεδόν οι χώρες εξαρτώνται σήμερα από σύνθετα συστήματα που βασίζονται σε υπολογιστές. Η τεχνολογία λογισμικού είναι ένας τεχνικός κλάδος που εστιάζετε στην οικονομικά αποτελεσματική ανάπτυξη συστημάτων λογισμικού υψηλής ποιότητας. Το λογισμικό είναι αφηρημένο και άυλο. Δεν περιορίζεται από υλικά, ούτε διέπεται από φυσικούς νόμους ή διαδικασίες. Ο γεγονός αυτό απλουστεύει την τεχνολογία λογισμικού μιας και δεν υφίστανται φυσικοί περιορισμοί για τις δυνατότητες του λογισμικού.



Ο άνθρωπος κλειδί για την ανάπτυξη ενός πληροφοριακού συστήματος είναι ο αναλυτής. Ο ρόλος του αναλυτή είναι πολτός καθώς έχει επικοινωνία με χρηστές και ειδικούς, μπορεί να ικανοποιεί γνώση από διαφορετικές πηγές, στοχοθετεί οργανώνει και ελέγχει το έργο, γίνεται αποτελεσματικότερα ο εντοπισμός – ανάλυση και επίλυση του προβλήματος, έχει την δυνατότητα να καθορίζει τις προδιαγραφές του συστήματος και εκπαιδεύει τους μελλοντικούς χρηστές.

Τα προσόντα ενός αναλυτή έχουν άμεση σχέση με την τη γνώση τεχνικών επεξεργασίας πληροφοριών (π.χ. τεχνικές προγραμματισμού, εργαλεία ανάπτυξης λογισμικού, πακέτα εφαρμογών, εξοπλισμός πληροφορικής κ.τ.λ), γενικές γνώσεις διοίκησης και διαχείρισης επιχειρήσεων (επικοινωνία με ειδικούς – κατανόηση προβλημάτων και αναγκών κάθε λειτουργίας), ικανότητα στην επίλυση προβλημάτων, δημιουργικότητα, φαντασίας κ.τ.λ.



Οι γνώσεις και οι δεξιότητες που αναπτύσσονται για την ανάπτυξη ενός πληροφοριακού συστήματος έχουν να κάνουν με:

- § Τεχνικές γνώσεις και δεξιότητες.
- § Διαχείριση τεχνικών γνώσεων και δεξιοτήτων.
- § Γνώσεις και δεξιότητες σχετικά με τη λειτουργία οργανισμών.



§ Διοικητικές και διαπροσωπικές δεξιότητες.

ΣΤΙΣ ΤΕΧΝΙΚΕΣ ΓΝΩΣΕΙΣ ΚΑΙ ΔΕΞΙΟΤΗΤΕΣ ΠΕΡΙΛΑΜΒΑΝΟΝΤΑΙ ΓΝΩΣΤΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ
ΌΠΩΣ:

- § Άγλωσσε προγραμματισμού.
- § Λειτουργικά συστήματος.
- § Κατανεμημένη επεξεργασία.
- § Δίκτυα επικοινωνίας.
- § Εργαλεία CASE
- § Διαχείριση βάσεων δεδομένων.
- § Μεθοδολογίες αναπτυξης Π.Σ.

Η ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΤΕΧΝΙΚΩΝ ΓΝΩΣΕΩΝ ΚΑΛΥΨΤΕΙ ΤΙΣ ΔΥΝΑΤΟΤΗΤΕΣ:

- § Εκμάθηση νέων τεχνολογιών.
- § Εστίαση στην τεχνολογία ως μέσο και όχι ως σκοπό.
- § Αντίληψη των εκάστοτε τεχνολογικών τάσεων.

Η ΓΝΩΣΗ ΚΑΙ ΟΙ ΔΕΞΙΟΤΗΤΕΣ ΣΧΕΤΙΚΕΣ ΜΕ ΤΙΣ ΛΕΙΤΟΥΡΓΙΕΣ ΟΡΓΑΝΙΣΜΩΝ, ΜΕΤΑΞΥ
ΤΩΝ ΆΛΛΩΝ ΠΕΡΙΛΑΜΒΑΝΟΥΝ ΤΙΣ ΙΚΑΝΟΤΗΤΕΣ:

- § Εκμάθησης των λειτουργιών του οργανισμού
- § Αντίληψη του εργασιακού περιβάλλοντος του οργανισμού.
- § Αντίληψη των προβλημάτων του οργανισμού και εισήγησης καταλλήλων λύσεων.

ΣΤΙΣ ΔΙΑΠΡΟΣΩΠΙΚΕΣ ΚΑΙ ΔΙΟΙΚΗΤΙΚΕΣ ΔΕΞΙΟΤΗΤΕΣ ΑΝΗΚΟΥΝ ΟΙ ΙΚΑΝΟΤΗΤΕΣ ΝΑ:

- § Σχεδιάζει και υλοποιεί έργα μέσα σε ένα συνεργατικό περιβάλλον.
- § Αντιμετωπίζει αβεβαιότητα.
- § Διδάσκει – εκπαιδεύει του άλλους.
- § Δρα προληπτικά κλπ.



Business Process Re-Engineering :Θεμελιώδης ανάλυση και καινοτομικός επανασχεδιασμός των επιχειρηματικών διαδικασιών και των συστημάτων διοίκησης με στόχο τη δραματική άνοδο ή τη βελτίωση της απόδοσης.

Το BPR χρησιμοποιεί αντικειμενικές μεθόδους και εργαλεία που αναλύουν, επανασχεδιάζουν και μεταβάλλουν τις επιχειρηματικές διαδικασίες, συμπεριλαμβανομένων των δομών υποστήριξης, των πληροφορικών συστημάτων, της κατανομής αρμοδιοτήτων και του επιπέδου αποδοτικότητας.

Ο επανασχεδιασμός επιχειρεί την απεξάρτηση μιας επιχείρησης από τους παλιούς κανόνες οργάνωσης και λειτουργίας της και την υιοθέτηση νέων, σύγχρονων αντιλήψεων που θα οδηγήσουν σε θεαματικές βελτιώσεις στην απόδοση της.

Το υψηλό κόστος της εφαρμογής περιλαμβάνει μεγάλο ρίσκο και είναι δύσκολο όσον αφορά στην υλοποίηση και τη διαχείριση του. Για το λόγω αυτό πριν γίνει Πριν γίνει reengineering σε κάποια εφαρμογή πρέπει να προηγηθεί cost-benefit ανάλυση (η διαδικασία reengineering κοστίζει \hat{a} χρόνος + άτομα = κόστος). Πρέπει να ουσιαστικά να ορίσουμε δέκα παραμέτρους.

1. P1: τρέχον ετήσιο κόστος συντήρησης της εφαρμογής
2. P2: τρέχον ετήσιο κόστος λειτουργίας της εφαρμογής
3. P3: τρέχουσα ετήσια επιχειρηματική αξία της εφαρμογής
4. P4: προβλεπόμενο ετήσιο κόστος συντήρησης μετά το reengineering
5. P5: προβλεπόμενο ετήσιο λειτουργικό κόστος μετά το reengineering
6. P6: προβλεπόμενη ετήσια επιχειρηματική αξία μετά το reengineering
7. P7: εκτιμηθεί κόστος reengineering
8. P8: εκτιμηθείς ημερολογιακός χρόνος reengineering
9. P9: παράγοντας ρίσκου reengineering (συνήθως $P8=1$, ονομαστική αξία \hat{a} over, under)
- 10.L: αναμενόμενη διάρκεια ζωής της εφαρμογής



Το κόστος που σχετίζεται με συνεχή συντήρηση χωρίς reengineering :

$$C_{\text{maint}} = [P3 - (P1 + P2)] * L \text{ à επιθυμητό: μεγιστοποίηση}$$

Το κόστος που σχετίζεται με το reengineering:

$$C_{\text{reeng}} = [P6 - (P4 + P5) * (L - P8) - (P7 * P9)] \text{ à επιθυμητό: μεγιστοποίηση}$$

Το όφελος (συνολικά) του επανασχεδιασμού είναι:

$$\text{Cost Benefit} = C_{\text{reeng}} - C_{\text{maint}}$$

- Αν η ανάλυση δείξει **θετικό** Cost Benefit (σχετικά μεγάλο) τότε προχωράμε στον επανασχεδιασμό.
- Αν είναι **αρνητικό** (ή μικρό) τότε ο επανασχεδιασμός μπορεί να αναβληθεί για αργότερα.



Βιβλιογραφία

Ελληνική Βιβλιογραφία

1. Ε. Κιουντούζης, «Μεθοδολογίες Ανάλυσης και Σχεδιασμού Πληροφοριακών συστημάτων», Εκδόσεις Μενού, Αθήνα 1997.
2. Δημητριάδης Αντώνης, «Διοίκηση και διαχείριση Πληροφοριακών συστημάτων», Εκδόσεις Νέων Τεχνολογιών, Αθήνα 1998.
3. Οικονόμου Γεώργιος «Πληροφορικά Συστήματα και διοίκηση επιχειρήσεων», Τόμος α' β' Εκδόσεις Μπενου, Αθήνα 1995
4. Χάινας Κ., «Βασικά θέματα για τα πληροφοριακά συστήματα διαχείρισης επιχειρησιακών πόρων», Εκδόσεις Γκιούρδας, Αθήνα 2006.
5. ΠΟΛΛΑΛΗΣ Γ. ,ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ,ΠΑΠΟΥΤΣΗΣ, «Πληροφορικά συστήματα Ι, Εισαγωγή στην Τεχνολογία και Στρατηγική», Εκδόσεις Σταμούλη, Αθήνα 2004
6. ΠΟΛΛΑΛΗΣ Γ. ,ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ, ΠΑΠΟΥΤΣΗΣ, «Πληροφορικά συστήματα ΙΙ, Εισαγωγή στην Τεχνολογία και Στρατηγική», Εκδόσεις Σταμούλη, Αθήνα 2004
7. ΔΟΥΚΙΔΗΣ ΓΕΩΡΓΙΟΣ, «Διοίκηση επιχειρήσεων και Πληροφοριακά συστήματα», Εκδόσεις Σιδέρης, Αθήνα 2003
8. Υψηλάντης Παντελής, «Πληροφορικά συστήματα διοίκησης, Από τη θεωρία στην Πράξη», Εκδόσεις Πατάκη, Αθήνα 2001

Ξενόγλωσση Βιβλιογραφία

9. Bass, L., Clements, P., et al (2003), Software architecture in practica, 2nd edn. Boston: Addison – Wesley.



10. Boehm, B.W (1979). Software engineering; R&D trends and defense needs. In research directions in software technology.
11. Boehm, B.W (1988). A spiral model of software development an enhancement IEEE Computer.
12. Bosch, J., (2000), Design an use of software architectures. Harlow: Addison – Wesley.
13. Hall, E., (1988), Managing risk: Methods for software systems development. Reading MA: Addison – Wesley.
14. Hofmeister, C., Nord, R., et al (2000), Applied software architecture. Boston: Addison – Wesley.
15. Ould, M.A., (1999), Managing software quality and business risk. Chichester: John Wiley & Sons.

Πηγές Διαδικτύου

16. http://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%BF%CE%AF%CE%BA%CE%B7%CF%83%CE%B7_%CE%BA%CE%B1%CE%B9_%CE%94%CE%B9%CE%B1%CF%87%CE%B5%CE%AF%CF%81%CE%B9%CF%83%CE%B7_%CE%88%CF%81%CE%B3%CE%BF%CF%85
17. <http://www.pslint.com/images/Project%20Plan.jpg>
18. <http://www.engineering-intelligence.gr/admin/fckfiles/image/sem29.png>
19. http://techpubs.sgi.com/library/dynaweb_docs/0650/SGI_Developer/book/OrOn2_Theops/sgi_html/figures/LINK.block.diagram.gif
20. http://www.islab.demokritos.gr/gr/html/ptixiakes/ATM_IPv6_& SecurityConsiderations/images/image72.gif



21. http://el.wikipedia.org/wiki/%CE%9C%CE%BF%CE%BD%CF%84%CE%AD%CE%BB%CE%BF_%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CE%AC%CF%82_OSI
22. <http://img.youtube.com/vi/GWTtq4d1DWw/0.jpg>





