



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

Κύκλος ζωής και αξιολόγηση προϊόντων λογισμικού. Ενδεικτική μελέτη περίπτωσης για προϊόντα εκπαιδευτικού λογισμικού στη δημόσια εκπαίδευση.

Πτυχιακή εργασία της

Κολλίγρη Θεοδώρας, Α.Μ. 7169

Επιβλέπων καθηγητής: κ. Παναγιώτης Δρούζας

ΠΑΤΡΑ ΝΟΕΜΒΡΙΟΣ 2010

Περίληψη

Στην εργασία αυτή θα γνωρίσουμε τον κύκλο ζωής προϊόντων και θα εστιάσουμε στον κύκλο ζωής λογισμικών προϊόντων.

Στο πρώτο κεφάλαιο αναφέρονται τα στάδια του κύκλου ζωής και στη συνέχεια τα μειονεκτήματα της χρήσης μεθόδου κύκλου ζωής. Στη συνέχεια θα γνωρίσουμε τα πιο σημαντικά μοντέλα ανάπτυξης εκπαιδευτικού λογισμικού παρουσιάζοντας για κάθε μοντέλο τις εργασίες που περιλαμβάνει, την σειρά που θα πρέπει αυτές να γίνουν, υπό ποιες προϋποθέσεις καθώς και με ποια κριτήρια αυτές θα ολοκληρωθούν. Ακολουθεί σύγκριση μοντέλων, συμπεράσματα, καθώς και προσωπική άποψη σχετικά με το ποιο είναι το καταλληλότερο μοντέλο.

Και τέλος θα αναπτυχθεί συνοπτικά η αναγκαιότητα της ύπαρξης διοίκησης έργων ανάπτυξης εκπαιδευτικού λογισμικού και ο ρόλος που διαδραματίζει. Συνοπτικά παρουσιάζονται τα βασικά χαρακτηριστικά και οι απαιτήσεις αυτών που την στελεχώνουν.

Στο δεύτερο κεφάλαιο της εργασίας θα μελετηθεί η διαδικασία της αξιολόγησης, καθώς και η αναγκαιότητα και ο τρόπος, με τον οποίο αποπερατώνεται η αξιολόγηση λογισμικών συστημάτων. Παρουσιάζονται παράμετροι που συναινούν στην επιτυχία ενός εκπαιδευτικού λογισμικού καθώς και το «πρόβλημα» της αξιολόγησης. Στη συνέχεια αναφέρονται τρία είδη αξιολόγησης, αρκετές μέθοδοι καθώς και μοντέλα που χρησιμοποιούνται για αυτό το σκοπό.

Στο τρίτο κεφάλαιο θα αξιολογηθεί το εκπαιδευτικό λογισμικό, με τίτλο «ερευνώ το φυσικό κόσμο» Ε΄ και ΣΤ΄ Δημοτικού. Το λογισμικό αυτό συνοδεύει τα νέα διδακτικά βιβλία, που χρησιμοποιούνται στη δημόσια εκπαίδευση.

Περιεχόμενα

| | |
|------------------------------------------------------|--------|
| Εισαγωγή..... | σελ.8 |
| 1. Κύκλος Ζωής | |
| Βασικές έννοιες και ορισμοί..... | σελ.12 |
| 1.1 Τι είναι εκπαιδευτικό λογισμικό..... | σελ.14 |
| 1.1.1 Γιατί είναι απαραίτητο..... | σελ.14 |
| 1.1.2 Η εξέλιξη του εκπαιδευτικού λογισμικού..... | σελ.15 |
| 1.2 Κύκλος Ζωής προϊόντος..... | σελ.15 |
| 1.2.1 Στάδιο εισαγωγής του προϊόντος στην αγορά..... | σελ.16 |
| 1.2.2 Στάδιο αναπτύξεως των πωλήσεων..... | σελ.16 |
| 1.2.3 Στάδιο ωριμότητας..... | σελ.17 |
| 1.2.4 Στάδιο κορεσμού..... | σελ.18 |
| 1.2.5 Στάδιο παρακμής..... | σελ.18 |
| Σχήμα 1.1..... | σελ.19 |
| 1.3 Κύκλος Ζωής εκπαιδευτικού Λογισμικού..... | σελ.20 |
| Διαδρομές Κατασκευής Λογισμικού (σχήμα 1.2)..... | σελ.21 |
| 1.4 Στάδια κύκλου ζωής λογισμικού..... | σελ.20 |
| Μέση κατανομή κόστους (σχήμα 1.3)..... | σελ.21 |
| 1.4.1 Προδιαγραφή απαιτήσεων..... | σελ.21 |
| 1.4.2 Ανάλυση απαιτήσεων..... | σελ.22 |
| Α)Βήματα της ανάλυσης απαιτήσεων..... | σελ.22 |
| 1.4.3 Σχεδιασμός..... | σελ.24 |
| 1.4.4 Υλοποίηση..... | σελ.25 |
| 1.4.5 Τεκμηρίωση..... | σελ.26 |
| 1.4.6 Συντήρηση..... | σελ.27 |
| 1.4.7 Αποχώρηση..... | σελ.28 |

| | |
|-------------------------------------------------------------------|--------|
| 1.4.8 Μειονεκτήματα της μεθόδου κύκλου ζωής..... | σελ.29 |
| 1.5 Δημιουργία μοντέλων..... | σελ.50 |
| 1.5.1 Τα βήματα της δημιουργίας μοντέλου (σχήμα 1.4)..... | σελ.31 |
| 1.5.2 Μοντέλα κύκλου ζωής εκπαιδευτικού λογισμικού..... | σελ.32 |
| 1.5.3 Μοντέλο καταρράκτη (σχήμα 1.5)..... | σελ.33 |
| 1.5.4 Μοντέλο πρωτοτυποποίησης (σχήμα 1.6)..... | σελ.37 |
| 1.5.5 Μοντέλο λειτουργικής επαύξησης (σχήμα 1.7)..... | σελ.41 |
| 1.6 Εναλλακτικά Μοντέλα Ανάπτυξης Εκπαιδευτικού Λογισμικού..... | σελ.42 |
| 1.6.1 Λειτουργικό Μοντέλο (σχήμα 1.8)..... | σελ.43 |
| 1.6.2 Μοντέλο αυτόματου προγραμματισμού..... | σελ.44 |
| 1.6.3 Μοντέλο build 'n' fix (σχήμα 1.9)..... | σελ.44 |
| 1.6.4 Μοντέλο επαναχρησιμοποίησης λογισμικού..... | σελ.45 |
| 1.6.5 Αντικειμενοστραφές μοντέλο..... | σελ.47 |
| 1.6.6 Το μοντέλο 'V' (σχήμα 1.10)..... | σελ.47 |
| 1.6.7 Σπειροειδές μοντέλο (σχήμα 1.11)..... | σελ.48 |
| 1.6.8 Μοντέλο πίδακα (σχήμα 1.12)..... | σελ.50 |
| 1.6.9 Γενικό μοντέλο κύκλου ζωής (σχήμα 1.13)..... | σελ.52 |
| 1.6.10 Συγκριτικός πίνακας..... | σελ.53 |
| 1.6.11 Πλεονεκτήματα και μειονεκτήματα της χρήσης μοντέλων..... | σελ.53 |
| 1.7 Διοίκηση Έργων Εκπαιδευτικού Λογισμικού..... | σελ.54 |
| 1.7.1 Γιατί είναι απαραίτητη η διοίκηση έργων..... | σελ.55 |
| 1.7.2 Η ανάπτυξη λογισμικού στην πραγματικότητα (σχήμα 1.14)..... | σελ.56 |
| 1.8 Συμπεράσματα..... | σελ.57 |
| 2. Αξιολόγηση..... | σελ.59 |
| 2.1 Τί είναι αξιολόγηση..... | σελ.59 |
| 2.1.2 Σκοπός της αξιολόγησης..... | σελ.60 |
| 2.1.3 Γιατί είναι απαραίτητη..... | σελ.60 |
| 2.1.4 Παράμετροι αξιολόγησης..... | σελ.61 |

| | | |
|-------|---------------------------------------------------------------|--------|
| 2.1.5 | Τι πρέπει να προσέχουμε κατά τη διάρκεια της αξιολόγησης..... | σελ.62 |
| 2.1.6 | Το πρόβλημα της αξιολόγησης..... | σελ.62 |
| 2.1.7 | Τί αξιολογούμε σε ένα λογισμικό..... | σελ.64 |
| 2.1.8 | «Πώς» αξιολογούμε ένα λογισμικό..... | σελ.65 |
| 2.2 | Μέθοδοι αξιολόγησης..... | σελ.66 |
| 2.2.1 | Αθροιστική αξιολόγηση | σελ.66 |
| 2.2.2 | Εκπαιδευτική αξιολόγηση | σελ.67 |
| 2.2.3 | Εμπειρική αξιολόγηση | σελ.67 |
| 2.2.4 | Αντικειμενική αξιολόγηση | σελ.67 |
| 2.2.5 | Υποκειμενική αξιολόγηση..... | σελ.67 |
| 2.2.6 | Μη εμπειρική αξιολόγηση..... | σελ.67 |
| 2.2.7 | Αναλυτική μέθοδος | σελ.68 |
| 2.2.8 | Μέθοδοι ελέγχου | σελ.68 |
| 2.3 | Μοντέλα αξιολόγησης..... | σελ.70 |
| 2.3.1 | ISO 9241 (σχήμα 2.1)..... | σελ.70 |
| | Σχήμα 2.2 - Περιγραφή διεργασιών ISO 9241..... | σελ.71 |
| 2.3.2 | Το γενικό μοντέλο αξιολόγησης (σχήμα 2.3)..... | σελ.71 |
| 2.3.3 | Μοντέλο DeLone & McLean (σχήμα 2.4)..... | σελ.72 |
| 2.3.4 | Το Technology Acceptance Model (TAM)..... | σελ.75 |
| | Παράδειγμα ερωτηματολογίου βασισμένο στο TAM..... | σελ.76 |
| | . | |
| 3 | ΑΞΙΟΛΟΓΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ | |
| 3.1 | Γενικά στοιχεία..... | σελ.79 |
| | Εκπαιδευτική Βαθμίδα/Τάξη στην οποία απευθύνεται..... | σελ.79 |
| 3.1.1 | Γνωστικό αντικείμενο..... | σελ.79 |
| 3.1.2 | Γνωστικά αντικείμενα που πραγματεύεται το λογισμικό..... | σελ.79 |
| 3.1.3 | Γνωστικά αντικείμενα που προσεγγίζει το λογισμικό..... | σελ.79 |

| | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3.1.4 | Ανταπόκριση του λογισμικού.....σελ.79 |
| 3.1.5 | Χρησιμότητα του λογισμικού.....σελ.80 |
| 3.2 | Ποιότητα περιεχομένου.....σελ.81 |
| 3.2.1 | Καταλληλότητα περιεχομένου.....σελ.81 |
| 3.2.2 | Καταλληλότητα γλώσσα και λοιπών χαρακτηριστικών.....σελ.81 |
| 3.2.3 | Ορθότητα του περιεχομένου του λογισμικού.....σελ.81 |
| 3.2.4 | Υπάρχει συνέπεια στους όρους και στα σύμβολα που χρησιμοποιούνται;.....σελ.81 |
| 3.2.5 | Η δομή και η παρουσίαση των πληροφοριών.....σελ.81 |
| 3.2.6 | Συνέπεια στους όρους και στα σύμβολα.....σελ.81 |
| 3.2.7 | Καταλληλότητα του όγκου της πληροφορίας.....σελ.81 |
| 3.2.8 | Καταλληλότητα όγκου πληροφορίας σε επίπεδο μαθητών.....σελ.81 |
| 3.3 | Ποιότητα/δυνατότητα ένταξης στις σχολικές συνθήκες.....σελ.82 |
| 3.3.1 | Το λογισμικό μπορεί να ενταχθεί στο υπάρχον αναλυτικό πρόγραμμα.....σελ.82 |
| 3.3.2 | Το λογισμικό δημιουργεί νέες δυνατότητες στα πλαίσια του υπάρχοντος αναλυτικού προγράμματος;.....σελ.82 |
| 3.3.3 | Το λογισμικό είναι κατάλληλο για ανεξάρτητη ατομική χρήση από κάθε μαθητή;.....σελ.82 |
| 3.3.4 | Το λογισμικό είναι κατάλληλο για ολοκληρωμένη χρήση (καλύπτει πλήρως μια διδακτική ενότητα με θεωρία, παραδείγματα, ασκήσεις κλπ) στη σχολική τάξη από το σύνολο των μαθητών και με ενεργή παρουσία του διδάσκοντα;.....σελ.82 |
| 3.3.5 | Το λογισμικό επιτρέπει στο διδάσκοντα να επιλέξει τη σειρά διδασκαλίας;.....σελ.82 |

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3.3.6 | Το λογισμικό επιτρέπει στο διδάσκοντα να επέμβει για να προσαρμόσει ή να προσθέσει υλικό;.....σελ.82 |
| 3.3.7 | Το λογισμικό εξασφαλίζει τη δυνατότητα και τις προϋποθέσεις στους μαθητές να εμπλουτίσουν το διδακτικό υλικό;.....σελ.82 |
| 3.3.8 | Το λογισμικό επιτρέπει τη χάραξη εναλλακτικών διαδρομών από τον ίδιο τον μαθητή ανάλογα με τις ανάγκες του;.....σελ.83 |
| 3.4 | Διδακτική και παιδαγωγική προσέγγιση |
| 3.4.1 | Βασική οργάνωση.....σελ.83 |
| 3.4.2 | Προσέγγιση της μάθησης που ενθαρρύνεται.....σελ.84 |
| 3.4.3 | Τρόπος με τον οποίο σχηματίζονται ή αναπτύσσονται οι ιδέες καθώς ο μαθητής αλληλεπιδρά με το λογισμικό.....σελ.84 |
| 3.5 | Ειδικά κριτήρια ανά γνωστικό αντικείμενο.....σελ.85 |
| 3.5.1 | Μαθηματικά και Φυσικές Επιστήμες (Φυσική, Χημεία, Γεωγραφία, Βιολογία).....σελ.86 |
| 3.6 | Ποιότητα αλληλεπίδρασης.....σελ.87 |
| 3.6.1. | Το είδος της διαλογικότητας που επιτρέπει το λογισμικό ενθαρρύνει την εμπλοκή και την ενεργό συμμετοχή του μαθητή;.....σελ.87 |
| 3.6.2 | Επιτρέπεται στους μαθητές να πλοηγούνται ελεύθερα μέσα στο λογισμικό κάνοντας προσωπικές επιλογές;.....σελ.87 |
| 3.6.3. | Υποστηρίζονται επαρκώς οι διαδικασίες πλοήγησης μέσω του εκπαιδευτικού λογισμικού;.....σελ.87 |
| 3.6.4. | Στην περίπτωση που το λογισμικό χρησιμοποιεί εικονίδια επιλογών, οι συμβολικές τους αναπαραστάσεις είναι κατανοητές και κατάλληλες για τους μαθητές;.....σελ.87 |
| 3.6.5. | Το λογισμικό χρησιμοποιεί ικανοποιητικά τις τεχνικές άμεσου χειρισμού;.....σελ.87 |
| 3.6.6. | Ο σχεδιασμός και η υλοποίηση του λογισμικού κάνουν χρήση κάποιας σύμβασης ή μεταφοράς για την παρουσίαση της συνολικής δομής του λογισμικού;.....σελ.88 |
| 3.6.7. | Η ανατροφοδότηση που παρέχει το λογισμικό κρίνεται ουσιαστική;.....σελ.88 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------|---------|
| 3.6.8.Η αλληλεπίδραση που προσφέρει κατά τη δικτυακή επικοινωνία κρίνεται άμεση και ικανοποιητική;.....σελ.88 | σελ.88 |
| 3.7 βασικά χαρακτηριστικά συγκεκριμένων τύπων εκπαιδευτικού λογισμικού..σελ.88 | σελ.88 |
| 3.7.1 Λογισμικά που περιέχουν παρουσίαση περιεχομένου και Λογισμικά υπερκειμένων/εγκυκλιπαιδειών.....σελ.88 | σελ.88 |
| 3.7.2 Λογισμικά που περιέχουν κλειστές ερωτήσεις και ασκήσεις Ανατροφοδότηση και μαθησιακή αξιοποίηση του λάθους.....σελ.89 | σελ.89 |
| 3.7.3 Λογισμικά που εμπεριέχουν προσομοιώσεις.....σελ.90 | σελ.90 |
| 3.8 Συνοδευτικά εγχειρίδια.....σελ.91 | σελ.91 |
| 3.8.1Συνοδευτικό εγχειρίδιο διδάσκοντα.....σελ.91 | σελ.91 |
| Συμπεράσματα.....σελ.93 | σελ.93 |
| | |
| Βιβλιογραφία.....σελ.100 | σελ.100 |

Εισαγωγή

Στην εργασία αυτή θα γνωρίσουμε τον κύκλο ζωής ενός προϊόντος και θα εστιάσουμε στον κύκλο ζωής των λογισμικών προϊόντων.

Στον πρώτο κεφάλαιο θα μελετήσουμε αναλυτικά τα στάδια του κύκλου ζωής λογισμικού, από την σύλληψη της ιδέας έως και την απόσυρση του προϊόντος. Στη συνέχεια θα δούμε μία εναλλακτική μέθοδο ανάπτυξης και σχεδιασμού εκπαιδευτικών λογισμικών, τα μοντέλα.

Αρχικά παρουσιάζονται τα βήματα δημιουργίας μοντέλων και ακολουθεί η παρουσίαση των πιο σημαντικών μοντέλων ανάπτυξης εκπαιδευτικού λογισμικού. Στη συνέχεια αναλύονται για κάθε μοντέλο οι εργασίες που περιλαμβάνει, η σειρά που θα πρέπει αυτές να γίνουν, υπό ποιές προϋποθέσεις καθώς και με ποια κριτήρια αυτές θα ολοκληρωθούν.

Θα συνοπτικά τα συμβατικά μοντέλα ανάπτυξης εκπαιδευτικού λογισμικού. Το μοντέλο του Καταρράκτη ήταν το πρώτο που δημιουργήθηκε, ένα σειριακό μοντέλο με αρκετά μειονεκτήματα όπως το ότι γνωρίζουμε εάν θα ικανοποιηθεί ο χρήστης στο τελικό του στάδιο. Παρόλα αυτά ήταν πολύ δημοφιλές μοντέλο και παραμένει σε χρήση ακόμα και σήμερα λόγω της απλότητας του είτε όπως είναι είτε με κάποιες παραλλαγές του όπως το μοντέλο V το οποίο με την εφαρμογή του προσπαθεί να μειώσει τα μειονεκτήματα του αρχικού μοντέλου. Η παρουσίαση του μοντέλου της Πρωτοτυποποίησης ακολούθησε στο οποίο για την αποφυγή του κινδύνου δημιουργείται ένα πρωτότυπο το οποίο επαληθεύεται που είτε 'πετιέται' μετά την χρήση (throw-away prototype) και στην συνέχεια αναπτύσσεται το λογισμικό είτε βελτιώνεται συνεχώς (εξελικτική πρωτοτυποποίηση) ώστε να προκύψει μέσα από τις βελτιώσεις αυτές το τελικό σύστημα. Στην συνέχεια το μοντέλο της Λειτουργικής Επαύξησης παρουσιάστηκε κατά το οποίο αρχικά δημιουργείται μια έκδοση που περιέχει τις πιο σημαντικές και κρίσιμες λειτουργίες η οποία αφού επαληθευτεί προσαυξάνεται και δημιουργείται η επόμενη έκδοση έως ότου κατασκευαστεί το τελικό σύστημα.

Στη συνέχεια παρουσιάζονται κάποια εναλλακτικά μοντέλα όπως το Λειτουργικό μοντέλο που αφορά περισσότερο στην φάση των απαιτήσεων όπου δημιουργείται ένα είδος πρωτοτύπου που περιγράφει έμμεσα την λειτουργική συμπεριφορά του τελικού συστήματος το οποίο βελτιώνεται και έτσι ολοκληρώνεται η φάση των απαιτήσεων και με χρήση κάποιου άλλου μοντέλου

γίνεται η ανάπτυξη από την φάση της σχεδίασης και κάτω. Το μοντέλο Αυτόματου Προγραμματισμού παρουσιάστηκε στην συνέχεια που βασίζεται στην ιδέα της δημιουργίας ενός συστήματος που να μπορεί αυτόματα αφού του δοθούν οι προδιαγραφές ενός προβλήματος να δημιουργεί λογισμικό αυτόματα.

Στο μοντέλο Επαναχρησιμοποίησης Λογισμικού που παρουσιάστηκε στην συνέχεια, γίνεται χρήση ήδη υπάρχοντος ελεγμένου και δοκιμασμένου λογισμικού (ψηφίδες λογισμικού) το οποίο ενσωματώνεται σε νέα προϊόντα λογισμικού. Τα μεγάλα πλεονεκτήματα που προσφέρει είναι η μείωση του χρόνου για την ανάπτυξη του τελικού συστήματος και ο περιορισμός των τμημάτων που απαιτούν έλεγχο αξιοπιστίας σε αυτά που δημιουργούνται για πρώτη φορά. Μια παραλλαγή του μοντέλου του Καταρράκτη, το Αντικειμενοστραφές μοντέλο ακολούθησε το οποίο διαφοροποιείται στο ότι οι φάσεις ανάπτυξης αλληλοκαλύπτονται και όπου χρειαστεί οπισθοδρομεί στην προηγούμενη φάση.

Το Σπειροειδές μοντέλο που παρουσιάστηκε στο τέλος της ενότητας είναι ένα δημοφιλές μοντέλο γενίκευση των μοντέλων της λειτουργικής επαύξησης και της πρωτοτυποποίησης. Η διαδοχή των φάσεων δεν γίνεται ούτε γραμμικά αλλά ούτε και σταθερά αλλά σε μορφή σπείρας σε συνάρτηση με τον κίνδυνο που αναλαμβάνεται κάθε φορά για κάθε φάση.

Από τα μοντέλα που θα γνωρίσουμε, στην πράξη περισσότερο χρήσιμα είναι αυτά που αφήνουν ελευθερία εξειδίκευσης στις συνθήκες και στο πρόβλημα που επιλύουν και δεν περιορίζουν με αυστηρότητα τις ενέργειες και τις φάσεις που θα πρέπει να ακολουθηθούν. Έτσι, θεώρησα το σπειροειδές μοντέλο ως το μοντέλο το οποίο μπορεί να αποτελέσει βάση για την ανάπτυξη εκπαιδευτικού λογισμικού με ορισμένες παραλλαγές οι οποίες εξειδικεύονται κάθε φορά ανάλογα με το περιβάλλον ανάπτυξης, το συγκεκριμένο πρόβλημα κλπ..

Στο δεύτερο κεφάλαιο θα αναφερθούν κάποιες γενικές μέθοδοι αξιολόγησης και στην συνέχεια ακολουθεί η περιγραφή δύο από τα πιο ευρέως διαδεδομένα μοντέλα αξιολόγησης, το μοντέλο Delone-McLean και το Technology Acceptance Model.

Οι παράγοντες, οι οποίοι συναινούν στην επιτυχία και την αποτελεσματικότητα του πληροφοριακού συστήματος ενός οργανισμού, αποτελούν πάντοτε μία ουσιαστική και σημαντική πτυχή για τη δομή, την εύρωστη λειτουργία και την ανοδική πορεία των παραγόμενων υπηρεσιών και προϊόντων του οργανισμού στους τελικούς χρήστες του – εσωτερικούς (υπαλληλικό προσωπικό του οργανισμού) και εξωτερικούς (κοινό). Στις μέρες μας ολοένα και περισσότερο προκύπτει έντονα η

ανάγκη για τη δημιουργία κατάλληλων παραμέτρων, οι οποίες θα έχουν στόχο την εκ βάθους αξιολόγηση και μέτρηση της επιτυχούς λειτουργίας ενός ιδρυματικού πληροφοριακού συστήματος. Σκοπός της συγκεκριμένης εργασίας είναι, κατ' αρχήν να μας εισάγει στο περιβάλλον των ιδρυματικών πληροφοριακών συστημάτων, και συνάμα να μας προσφέρει μία όσο το δυνατόν πληρέστερη και εκτενέστερη εικόνα των λειτουργιών και διεργασιών, της υποδομής και των παραγόντων επιτυχίας τους. Μέσα από τη Θεωρία των Συστημάτων θα εξετάσουμε την ανάλυση και τον σχεδιασμό ενός πληροφοριακού συστήματος και θα γνωρίσουμε τις ποιοτικές παραμέτρους αξιολόγησης και επιτυχίας του.

Επιπροσθέτως, θα έρθουμε σε επαφή με το μοντέλο μέτρησης της ποιότητας ενός πληροφοριακού συστήματος, το λεγόμενο «Μοντέλο Επιτυχίας Πληροφοριακών Συστημάτων DeLone & McLean». Θα έχουμε τη δυνατότητα να δούμε τι ακριβώς εξετάζει η κάθε παράμετρος του συγκεκριμένου μοντέλου, διερευνώντας έννοιες, όπως η ποιότητα του συστήματος, η ποιότητα της πληροφορίας, η ικανοποίηση του χρήστη, η επίδραση της ποιότητας στο άτομο και στον οργανισμό. Παράλληλα, θα αναπτύξουμε μία εφαρμογή αξιολόγησης ποιότητας ενός πληροφοριακού συστήματος «Οργανωτικής Μνήμης». συγκεκριμένο μοντέλο αποτελεί επέκταση του μοντέλου επιτυχίας DeLone & McLean. Κύριο μέλημα της ακόλουθης εργασίας είναι να καταλήξουμε σε έναν γόνιμο προβληματισμό, όσον αφορά την ποιότητα στη λειτουργία, την υποδομή και τις προσφερόμενες υπηρεσίες των ιδρυματικών οργανισμών (Βιβλιοθηκών, Αρχείων και εν γένει Υπηρεσιών Πληροφόρησης), καθώς και να εστιάσουμε στη χρησιμότητα και τη σημαντικότητα των παραγόντων μέτρησης της αποτελεσματικότητας ενός πληροφοριακού συστήματος, έτσι όπως αυτή η διαδικασία εκτιμάται μέσα στους κόλπους της σύγχρονης Κοινωνίας της Πληροφορίας, η οποία μετασχηματίζεται σε Κοινωνία της Γνώσης.

Σκοπός της αξιολόγησης είναι να μετρήσει την ποιότητα του εκάστοτε λογισμικού. Σε όλο τον κόσμο τεράστια χρηματικά ποσά ξοδεύονται για την ανάπτυξη πληροφοριακών συστημάτων. Είναι επομένως σημαντικό να αξιολογηθούν τα αποτελέσματά τους. Επίσης απαραίτητη είναι η αξιολόγηση τους για την καλύτερη

κατανόηση της λειτουργίας των πληροφοριακών συστημάτων. Η αξιολόγηση δεν είναι ποτέ ένας εύκολος στόχος και συνεπώς υπάρχουν πολλές προτάσεις για το πώς να αξιολογήσει κανείς ένα πληροφοριακό σύστημα. Ένα μεγάλο μέρος της βιβλιογραφίας θεωρεί την αξιολόγηση κατά ένα μεγάλο μέρος ως ποσοτική διαδικασία με πιθανό κόστος/κέρδος βάσει των καθορισμένων κριτηρίων.

Συνήθως κατά την αξιολόγηση των πληροφοριακών συστημάτων γίνεται αξιολόγηση

των πληροφοριακών πόρων με κριτήρια όπως κάλυψη, επικάλυψη, επικαιρότητα, ποιότητα και καταλληλότητα των πληροφοριών. Παραδείγματα: εξέταση της κάλυψης

ενός θέματος από συγκεκριμένη βάση δεδομένων, ταχύτητα ενημέρωσης βάσεων δεδομένων και σύγκριση μεταξύ τους κ.α. Επίσης γίνεται αξιολόγηση των πληροφοριακών συστημάτων με κριτήρια την αποδοτικότητα και την αποτελεσματικότητα που οι ζητούμενες πληροφορίες φτάνουν στους χρήστες.

Τα σημαντικότερα στοιχεία που εξετάζονται κατά την αξιολόγηση είναι : Η απόδοση πώς δηλαδή εκτελούνται οι λειτουργίες με τους λιγότερους πόρους, η αποτελεσματικότητα δηλαδή πόσο καλά εκτελούνται οι λειτουργίες σε σχέση με τους στόχους, επιτυχία, ικανοποίηση, κόστος, όφελος, ποιότητα, συμπεριφορά, σχέση κόστους/οφέλους, χρηστικότητα.

Στην εργασία αυτή θα αναφερθούν οι σημαντικές θεμελιακές αντιλήψεις που διέπουν την επιλογή ή την ανάπτυξη ενός υποδείγματος αξιολόγησης εκπαιδευτικού λογισμικού και θα παρουσιαστεί ένα υπόδειγμα, χρησιμοποιώντας κριτήρια βασισμένα στις αρχές του TAM (Technology Acceptance Model) σύμφωνα με το οποίο είναι βασισμένη η μελέτη του προγράμματος «ερευνώ το φυσικό κόσμο». Το λογισμικό αυτό συνοδεύει τα νέα διδακτικά βιβλία φυσικής, χημείας, περιβαλλοντικής εκπαίδευσης και βιολογία. Τέλος γίνεται μια εφαρμογή του TAM για την αξιολόγηση του συστήματος «ερευνώ το φυσικό κόσμο» της τάξης Ε΄ και ΣΤ΄ Δημοτικού στο μάθημα της Φυσικής.

1.Κύκλος Ζωής

Βασικές έννοιες και ορισμοί

Πληροφοριακό σύστημα: Είναι εκείνος ο μηχανισμός, ο οποίος παρέχει τα μέσα για τη συλλογή, την αποθήκευση, την παραγωγή και τη διανομή των πληροφοριών, που εξυπηρετούν τις πληροφοριακές ανάγκες ενός οργανισμού και υποστηρίζουν τις δραστηριότητές του, τόσο σε επίπεδο διαχείρισης και λειτουργίας, όσο και σε επίπεδο σχεδιασμού και λήψης αποφάσεων για τον οργανισμό.

Δραστηριότητα ή διαδικασία ανάπτυξης λογισμικού διαδικασία ανάπτυξης λογισμικού καθορίζει ποιες ενέργειες πρέπει να γίνουν για να επιτευχθεί ένα επιθυμητό αποτέλεσμα σε κάποια από τις φάσεις του κύκλου ζωής. Μία δραστηριότητα μπορεί να αναλύεται σε περισσότερες από μία επιμέρους φάσεις.

Μεθοδολογία ανάπτυξης λογισμικού: Καθορίζει το πώς θα πρέπει να εκτελούνται οι δραστηριότητες ανάπτυξης, δηλαδή ποιες επιμέρους ενέργειες περιλαμβάνουν, ποια βήματα γίνονται σε καθεμιά, ποια προϊόντα παράγονται, καθώς και πότε αυτές θεωρούνται περατωθείσες.

Εργαλείο λογισμικού: το εργαλείο λογισμικού είναι ένα σύστημα (συνήθως είναι και το ίδιο εφαρμογή λογισμικού) το οποίο υποστηρίζει τη μερική ή (σπάνια) την ολική αυτοματοποίηση των εργασιών που λαμβάνουν χώρα κατά την εφαρμογή των μεθοδολογιών ανάπτυξης λογισμικού.

Μοντέλο Κύκλου Ζωής Λογισμικού: το μοντέλο λογισμικού είναι μία περιγραφή των δραστηριοτήτων και των επιμέρους φάσεων από τις οποίες διέρχεται μία εφαρμογή λογισμικού από τη σύλληψη μέχρι την απόσυρσή της, καθώς και των εργασιών που λαμβάνουν χώρα σε καθεμιά από αυτές τις φάσεις.

Κίνδυνος: Ο Κίνδυνος ή πρόβλημα, είναι δύσκολο να καθοριστεί πλήρως. Μια απλή εξήγηση που θα μπορούσαμε να δώσουμε για το κίνδυνο είναι ότι είναι

οτιδήποτε θεωρείται ότι μπορεί να πάει στραβά. Για παράδειγμα, αν χρησιμοποιηθεί ένα καινούριο πακέτο για τρισδιάστατα γραφικά, κίνδυνο θα αποτελέσει κατά πόσο είναι δύσκολο να το μάθουν γρήγορα οι εμπλεκόμενοι στην ομάδα ανάπτυξης. Άλλος σημαντικός κίνδυνος στην ανάπτυξη λογισμικού είναι τα λάθη και περισσότερο οι παραλείψεις των απαιτήσεων του συστήματος. Οι κίνδυνοι είναι συνέπεια ελλιπούς πληροφόρησης. Για την μείωση ή εξάλειψη τους χρειαζόμαστε περισσότερες πληροφορίες. Για το παραπάνω παράδειγμα, ο κίνδυνος μπορεί να μειωθεί κάνοντας στην ομάδα ανάπτυξης ένα ταχύρυθμο πρόγραμμα εκμάθησης, ή προσλαμβάνοντας νέα ομάδα ανάπτυξης με γνώση του συγκεκριμένου πακέτου.

Παράλληλα, ένα **λογισμικό** διακρίνεται συνήθως από κάποια **συστατικά μέρη**, τα οποία αποτελούν διακριτά χαρακτηριστικά του συστήματος, και είναι

- Εισροές όπως υλικά, πληροφορίες, ανθρώπινο δυναμικό, πόροι κ.λ.π.
- Διαδικασίες μετατροπής οι οποίες είναι οι μηχανισμοί σχεδιασμού, οργάνωσης και ελέγχου, οι δραστηριότητες παραγωγής, η έρευνα και ανάπτυξη, κ.λ.π.
- Εκροές, όπως υπηρεσίες, προϊόντα, ιδέες, και άλλα.

Επίσης, το πληροφοριακό σύστημα ενός οργανισμού χωρίζεται σε υποσυστήματα, από τα οποία το καθένα μπορούμε να το εξετάσουμε ως ένα ξεχωριστό και ανεξάρτητο σύστημα. Κάθε υποσύστημα διαιρείται σε άλλα μικρότερα, και έτσι αναπτύσσεται μία ιεραρχική δομή, όπου σε κάθε επίπεδο ανήκουν υποσυστήματα, τα οποία είναι ανεξάρτητα και λειτουργικά και βρίσκονται σε στενή σχέση και συνεργασία με τα αντίστοιχα υποσυστήματα του παραπάνω και παρακάτω επιπέδου. Οι τρόποι ανάλυσης και σχεδιασμού ενός πληροφοριακού συστήματος αποτελούν τα πιο βασικά στάδια ανάπτυξης και λειτουργίας του.

1.1 Τι είναι εκπαιδευτικό λογισμικό

Ο όρος "εκπαιδευτικό λογισμικό" αφορά σε εφαρμογές ηλεκτρονικού υπολογιστή που χρησιμοποιούνται για την εξυπηρέτηση εκπαιδευτικών αναγκών και την επίτευξη παιδαγωγικών και εκπαιδευτικών στόχων. Αρκετές εφαρμογές γενικής χρήσης (κειμενογράφος, λογιστικά φύλλα, ζωγραφική, κ.α.) μπορούν να εξυπηρετήσουν (και εξυπηρετούν) εκπαιδευτικούς σκοπούς. Ωστόσο, υπάρχει και μια μεγάλη ποικιλία τίτλων εκπαιδευτικού λογισμικού και ηλεκτρονικών εκπαιδευτικών υπηρεσιών που έχει σχεδιαστεί αποκλειστικά για εκπαιδευτική χρήση.

Η χρήση της ψηφιακής τεχνολογίας στην εκπαίδευση ξεκίνησε στις προηγμένες χώρες από τη δεκαετία του '60. Στη χώρα μας, όμως, μόλις τη δεκαετία του '90 εμφανίστηκαν οι πρώτοι ηλεκτρονικοί υπολογιστές στα σχολεία, ενώ την ίδια, περίπου, περίοδο οι οικιακοί ηλεκτρονικοί υπολογιστές έκαναν την εμφάνισή τους, σποραδικά στην αρχή, αλλά με αυξανόμενους ρυθμούς. Έτσι, στα τέλη της δεκαετίας του '90 οι κάτοχοι ηλεκτρονικού υπολογιστή μπορούσαν να βρουν αρκετούς τίτλους εκπαιδευτικού λογισμικού στα ράφια των καταστημάτων.

Μπορούμε να χωρίσουμε την ιστορία του εκπαιδευτικού λογισμικού σε τρεις περιόδους. Τα CD-ROM της δεκαετίας του '90 χαρακτηρίζουν την πρώτη εποχή του εκπαιδευτικού λογισμικού. Ακολούθησε η δεύτερη εποχή, με λογισμικό που συνεργάζεται με κατάλληλες συσκευές και είναι ανοιχτό ως προς το περιεχόμενο. Στην τρίτη εποχή, που ζούμε τώρα, το λογισμικό προσφέρεται περισσότερο στη μορφή της διαδικτυακής υπηρεσίας και έχει περισσότερα επικοινωνιακά χαρακτηριστικά.

1.1.1 Γιατί είναι απαραίτητο

Το λογισμικό στην εκπαίδευση δίνει πολλές δυνατότητες στον εκπαιδευτικό. Η χρήση λογισμικού στην εκπαίδευση μπορεί να γίνει η βάση για την παραγωγή εκπαιδευτικών θαυμάτων, αρκεί να υπάρχει όρεξη και δημιουργικότητα. Επιπλέον, ο εκπαιδευτικός έχει την ευκαιρία να απελευθερωθεί από την τυποποίηση της μίας

περιορισμένης ύλης, και να συντονιστεί με τις σύγχρονες παιδαγωγικές αντιλήψεις που θέλουν το μαθητή να ανακαλύπτει και να χτίζει μόνος του τη γνώση (εποικοδομισμός). Τέλος, οι νέοι τίτλοι λογισμικού αυξάνουν το βαθμό αλληλεπίδρασης και ενθαρρύνουν τη δημιουργική συμμετοχή των μαθητών, συμβαδίζοντας έτσι, με τις νεότερες παιδαγωγικές θεωρίες.

1.1.2 Η εξέλιξη του εκπαιδευτικού λογισμικού

Η Πληροφορική στην ελληνική εκπαίδευση αποτελεί μέχρι σήμερα μόνο γνωστικό αντικείμενο, παρά την ύπαρξη εξοπλισμού στα περισσότερα σχολεία της χώρας. Από το σχολικό έτος 1998-99 έχει εισαχθεί το μάθημα στο Ενιαίο Λύκειο, ενώ παράλληλα οργανώνεται η εφαρμογή της Πληροφορικής και των Νέων Τεχνολογιών ως εκπαιδευτικό εργαλείο για την υποστήριξη της διδασκαλίας των άλλων μαθημάτων. Το Παιδαγωγικό Ινστιτούτο θέσπισε προδιαγραφές και προκήρυξε διαγωνισμούς για την ανάπτυξη εκπαιδευτικού λογισμικού για διάφορα γνωστικά αντικείμενα. Τα λογισμικά προϊόντα είναι βασικά εργαλεία για την εκπαιδευτική διαδικασία.

Τα τελευταία χρόνια αναπτύσσεται σημαντικά ο ρόλος της πληροφορικής και των νέων τεχνολογιών (πολυμέσα, εικονική πραγματικότητα, Διαδίκτυο κλπ.) με εφαρμογές που προέρχονται από ερευνητικά-ακαδημαϊκά ιδρύματα και από ιδιωτικούς φορείς. Όμως τα προϊόντα λογισμικού που προέρχονται από τον ιδιωτικό τομέα δεν περιέχουν συνήθως το σημαντικό στοιχείο της περιβαλλοντικής εκπαίδευσης ή στερούνται παιδαγωγικού υποβάθρου.

1.2 Κύκλος Ζωής προϊόντος

Σαν κύκλος ζωής ενός προϊόντος μπορεί να ορισθεί η χρονική περίοδος από την εισαγωγή του προϊόντος στην αγορά ως τη στιγμή κατά την οποία αποσύρεται από αυτή. Συνήθως ο κύκλος ζωής των προϊόντων έχει μια αρχή, ένα σημείο ακμής και ένα τέλος. Υπάρχουν προϊόντα τα οποία διατηρούν τη θέση τους στην αγορά για

πολλές δεκαετίες και άλλα των οποίων ο κύκλος ζωής διαρκεί για μερικούς μήνες ή εβδομάδες. Ο κύκλος ζωής των προϊόντων στις περισσότερες περιπτώσεις, περιλαμβάνει πέντε στάδια. Τα στάδια αυτά διαγράφονται στην καμπύλη των πωλήσεων.

1.2.1 Στάδιο εισαγωγής του προϊόντος στην αγορά.

Η εισαγωγή συμβαίνει όταν το προϊόν εισέρχεται για πρώτη φορά στην αγορά. Κατά το στάδιο αυτό, το προϊόν συνοδεύεται με υψηλές επενδύσεις με την προώθηση του στην αγορά από λεπτομερώς ελεγμένα προγράμματα, καθώς και με προσαρμογές στη στρατηγική του marketing. Άλλα χαρακτηριστικά αυτού του σταδίου είναι το υψηλό κόστος παραγωγής, το χαμηλό ύψος των πωλήσεων, η περιορισμένη διανομή, όπως και η έλλειψη άμεσου ανταγωνισμού.

1.2.2 Στάδιο αναπτύξεως των πωλήσεων.

Το κυριότερο χαρακτηριστικό αυτού του σταδίου είναι η ταχεία αύξηση των πωλήσεων για εκείνα τα προϊόντα που επέζησαν κατά το στάδιο της εισαγωγής τους. Σ' αυτή τη φάση η επιχείρηση θα πρέπει να αρχίσει να σκέφτεται για νέες χρήσεις και νέες αγορές για το προϊόν της. Η αύξηση των πωλήσεων έχει σαν αποτέλεσμα τη μείωση του κόστους παραγωγής του προϊόντος και ως εκ τούτου την αύξηση των κερδών. Τα κέρδη αυτά δημιουργούν τον ανταγωνισμό. Όσο δε περισσότερο αυξάνουν οι πωλήσεις και τα κέρδη, τόσο περισσότερο αυξάνει και ο ανταγωνισμός.

1.2.3 Στάδιο ωριμότητας.

Η ωριμότητα ακολουθεί την ταχεία ανάπτυξη των πωλήσεων του δεύτερου σταδίου αυτού το προϊόν έχει φθάσει στην αποκορύφωση της αποδοχής του από μέρος των καταναλωτών του τμήματος της αγοράς, στο οποίο απευθύνεται. Οι πωλήσεις και η χρησιμότητα έχουν αρχίσει να πλησιάζουν στο μέγιστο της αποδοτικότητας. Επίσης

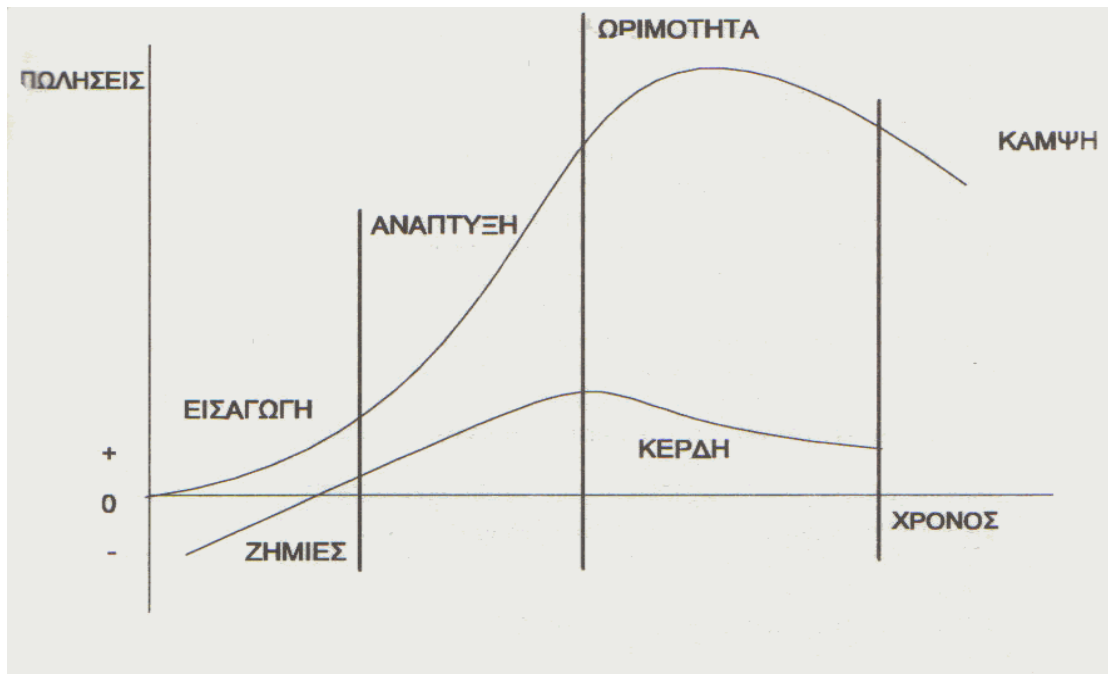
κατά το στάδιο αυτό εισάγονται νέα ανταγωνιστικά προϊόντα στην αγορά και έτσι αρχίζει ο ανταγωνισμός «σημάτων και τιμών».

1.2.4 Στάδιο κορεσμού.

Ο κορεσμός συμβαίνει όταν, οι ανταγωνιστές αντιγράφουν, τελειοποιούν ή διαφορετικά εξασθενούν τη δημοτικότητα ενός προϊόντος. Κατά το τέλος αυτού του σταδίου, οι πωλήσεις και τα κέρδη αρχίζουν να μειώνονται, λόγω του ότι η αγορά στρέφεται προς άλλα ανταγωνιστικά προϊόντα.

1.2.5 Στάδιο παρακμής.

Η παρακμή χαρακτηρίζεται από τη συνεχή μείωση των πωλήσεων του προϊόντος, το οποίο πιθανός πρέπει να αντικατασταθεί από νέα προϊόντα και αυτό γιατί έχασε την προηγούμενη του θέση στην αγορά. Εδώ τελειώνει ο κύκλος ζωής του προϊόντος. Εν τούτης είναι δυνατόν πολλές φορές μια επιχείρηση να διατηρήσει ένα προϊόν στη ζωή, παρά το γεγονός ότι βρίσκεται στη ζημιογόνο περιοχή, αν δηλαδή αυτό το επιβάλλουν τα μακροπρόθεσμα σχέδια της επιχείρησης.



Σχήμα 1.1

1.3 Κύκλος Ζωής εκπαιδευτικού Λογισμικού

Όπως όλα τα προϊόντα, έτσι και τα εκπαιδευτικά λογισμικά έχουν κύκλο ζωής. Ο κύκλος ζωής λογισμικού είναι η πιο παλιά μέθοδος ανάπτυξης συστημάτων πληροφοριών και χρησιμοποιείται ακόμη και σήμερα σε μεγάλα ή μεσαία έργα σύνθετων συστημάτων. Σε αυτή τη μεθοδολογία γίνεται η υπόθεση ότι ένα σύστημα πληροφοριών έχει μια διάρκεια ζωής παρόμοια με κάθε ζωντανό οργανισμό με αρχή, μέση και τέλος.

Η σωστή ανάπτυξη και η εύρυθμη λειτουργία του πληροφοριακού συστήματος εξαρτώνται σε σημαντικό βαθμό από ορισμένα συγκεκριμένα στάδια, τα οποία ακολουθούνται, προκειμένου να πραγματοποιηθεί η ανάλυση και ο σχεδιασμός του. Πρωτίστως, τίθεται η αρχή με την οποία προσδιορίζονται οι στόχοι του συστήματος, που σχετίζονται με τα σημεία αναφοράς του, τις αρχικές ανάγκες ανάλυσής του, την αξιολόγηση πιθανών λύσεων και την ανάλυση πιθανών υπάρχοντων συστημάτων. Εν κατακλείδι, πραγματοποιείται και μία πιλοτική έρευνα. Στη συνέχεια, προσδιορίζονται οι ανάγκες του συστήματος και δηλώνονται οι προδιαγραφές του. Το σύστημα σχεδιάζεται, κάνοντας το μοντέλο του λογικού και του φυσικού, ενώ παράλληλα επιλέγονται ανάλογα προγράμματα και υπολογιστικά συστήματα. Ακολουθεί η εφαρμογή των προαναφερθέντων, ξεκινώντας από την εκπαίδευση και την εξάσκηση των εργαζομένων, τη δημιουργία βάσεως δεδομένων και την εγκατάσταση του συστήματος μετάπτωσης από το ένα στο άλλο. Στο τελικό στάδιο, πραγματοποιούνται διαδικασίες όπως, η αξιολόγηση του συστήματος, καθώς και η παρακολούθηση, η συντήρηση και η εξέλιξή του μέσα στον ιδρυματικό οργανισμό.

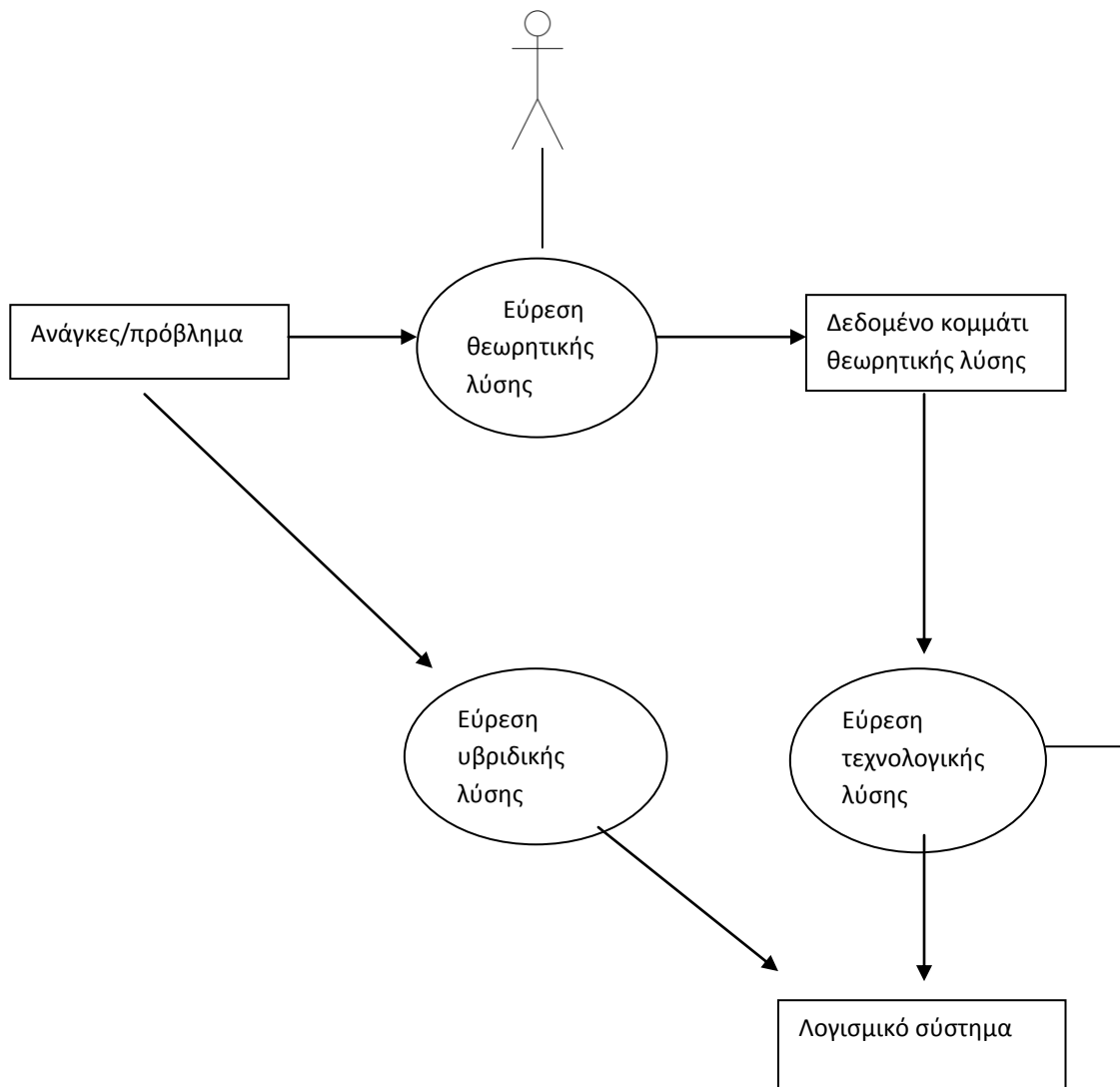
Η μεθοδολογία ανάπτυξης συστήματος πληροφοριών, χωρίζει τη διεργασία ανάπτυξης των συστημάτων σε τυποποιημένα στάδια/φάσεις, τα οποία πρέπει να ολοκληρώνονται με τη σειρά και προβλέπει μια πολύ τυπική κατανομή εργασίας μεταξύ των χρηστών και των ειδικών στα συστήματα πληροφοριών.

Οι τεχνικοί, όπως οι αναλυτές συστημάτων και οι προγραμματιστές είναι υπεύθυνοι για το μεγαλύτερο μέρος των εργασιών ανάλυσης συστήματος, σχεδιασμού και υλοποίησης. Οι τελικοί χρήστες περιορίζονται στο να περιγράφουν τις ανάγκες τους σε πληροφορίες και να εξετάζουν το αποτέλεσμα της δουλειάς του τεχνικού

προσωπικού. Το τέλος κάθε σταδίου πιστοποιείται με επίσημη συμφωνία μεταξύ των χρηστών και των ειδικών.

Ο κύκλος ζωής ενός συστήματος πληροφοριών έχει τα έξι στάδια. Κάθε στάδιο αποτελείται από βασικές δραστηριότητες που πρέπει να ολοκληρωθούν πριν αρχίσει το επόμενο.

Διαδρομές Κατασκευής Λογισμικού



Σχήμα 1.2

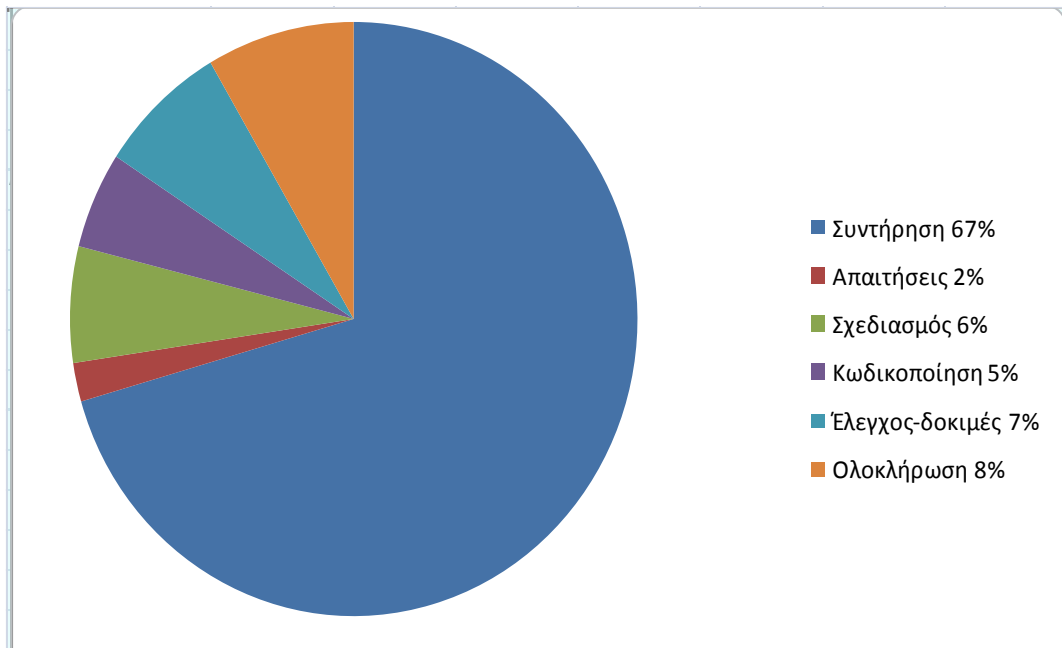
1.4 Στάδια κύκλου ζωής λογισμικού

Τα στάδια του κύκλου ζωής λογισμικού είναι διαδικασίες οι οποίες εφαρμόζονται είτε παράλληλα, είτε με κάποια συγκεκριμένη σειρά. Η εφαρμογή αυτών των διαδικασιών ορίζεται από κάποιο συγκεκριμένο μοντέλο και από το πλαίσιο του έργου. Μία εφαρμογή λογισμικού διέρχεται από τις φάσεις του κύκλου ζωής της από τη στιγμή της σύληψής της, κατά την κατασκευή, χρήση και συντήρησή της, μέχρι και την απόσυρσή της.

Δεν υπάρχει ένας και μοναδικός τρόπος, ο οποίος ακολουθείται. Ο τρόπος που θα εφαρμοστούν τα στάδια εξαρτάται από τις ιδιαίτερες συνθήκες που επικρατούν κάθε φορά, από την εμπειρία και την τεχνογνωσία του κάθε κατασκευαστή και από το μέγεθος και το θεματικό πεδίο εφαρμογής του λογισμικού. Οι φάσεις του κύκλου ζωής εξαρτώνται επίσης από το εκάστοτε περιβάλλον ανάπτυξης και από τις πιθανότητες να αλλάξουν οι απαιτήσεις από την εφαρμογή

- Προδιαγραφή απαιτήσεων
- Ανάλυση απαιτήσεων
- Σχεδιασμός
- Υλοποίηση
- Τεκμηρίωση
- Συντήρηση
- Αποχώρηση

Μέση κατανομή κόστους



σχήμα 1.3

1.4.1 Προδιαγραφή απαιτήσεων

Σε αυτήν την φάση, η ομάδα ανάπτυξης επισκέπτεται τον πελάτη και μελετά το σύστημά του. Ερευνά την ανάγκη για την πιθανή αυτοματοποίηση λογισμικού στο δεδομένο σύστημα. Το σημαντικό έργο για τη δημιουργία ενός προϊόντος λογισμικού είναι απόσπασμα των απαιτήσεων ή ανάλυση απαιτήσεων. Οι πελάτες συνήθως έχουν μια αφηρημένη ιδέα για το τι θέλουν ως τελικό αποτέλεσμα. Ελλιπείς, ασαφείς, ή ακόμη και αντιφατικές απαιτήσεις έχουν αναγνωριστεί από ειδικευμένους και έμπειρους μηχανικούς λογισμικού σε αυτό το σημείο. Η συλλογή πληροφοριών για επεξεργασία γίνεται με μεθόδους όπως ανάλυση σεναρίων, σύνθεση ομάδων για ειδικά θέματα απαιτήσεων, κατανόηση στρατηγικών της εταιρείας, μελέτη της οργάνωσης της ιεραρχίας, περιγραφή των καθηκόντων των χρηστών, παρακολούθηση του χρήστη όσο εργάζεται, ερωτηματολόγια και συνεντεύξεις.

Μέχρι το τέλος της ανάλυσης απαιτήσεων, η ομάδα εφοδιάζει ένα έγγραφο που κρατά τις διαφορετικές και συγκεκριμένες συστάσεις για το σύστημα υποψηφίων.

Περιλαμβάνει επίσης τις αναθέσεις προσωπικού, τις δαπάνες, το πρόγραμμα, τις προβλεπόμενες ημερομηνίες κ.λπ.

Ο ουσιαστικός σκοπός αυτής της φάσης είναι να εντοπιστεί η ανάγκη και να καθοριστεί το πρόβλημα που πρέπει να λυθεί. Στη συνέχεια, ακολουθεί η διαδικασία αναλυτικής διερεύνησης του προβλήματος που εντοπίστηκε. Πρέπει η ερευνητική ομάδα να κατανοήσει πλήρως το τί χρειάζεται και όχι τί θέλει ο πελάτης και πραγματοποιείται η καταγραφή περιορισμών.

1.4.2 Ανάλυση απαιτήσεων

Το επόμενο βήμα είναι η περαιτέρω ανάλυση και εκλέπτυνση του προβλήματος. Σε αυτό το βήμα καλούμαστε να απαντήσουμε στα ερωτήματα: Είναι το προϊόν τεχνικά εφικτό; Είναι απαραίτητη προϋπόθεση, η διαδικασία να θεωρείται εύλογη οικονομικά. Αφού έχουν συγκεντρωθεί οι απαιτήσεις και υπάρχει το βασικό σχέδιο, ακολουθεί ο καθορισμός προδιαγραφών. Σκοπός αυτής της ενέργειας είναι να δώσει μια συγκεκριμένη άποψη σχετικά με το κόστος και το χρόνο παράδοσης και να οργανώσει την περαιτέρω εργασία, αλλά κυρίως να σχεδιάσει το πρόγραμμα διαχείρισης του προϊόντος. Εδώ θα πρέπει τα σχετικά έγγραφα, που έχουν νομική υπόσταση να συμπληρωθούν, έτσι ώστε να καθορίσουν επακριβώς το πρόβλημα, με σαφήνεια και υπευθυνότητα. Μετά τη συμπλήρωση των σχετικών εγγράφων ακολουθεί αναλυτικός σχεδιασμός της παραγωγής και υπολογισμός του κόστους. Πριν τη μετάβαση στο επόμενο στάδιο είναι απαραίτητος ο έλεγχος της φάσης ορισμού και απαιτήσεων, αλλά και σε ολόκληρο το πρόγραμμα διαχείρισης του λογισμικού υποβάλλεται σε έλεγχο όσον αφορά την δόμηση του και γίνεται ανασκόπηση.

A) Βήματα της ανάλυσης απαιτήσεων:

Η ανάλυση απαιτήσεων γίνεται σε 3 βήματα:

- Συλλογή απαιτήσεων (γνώση του πεδίου εφαρμογής και απαιτήσεων χρήστη).

Η ομάδα ανθρώπων που είναι υπεύθυνοι για τη συλλογή των απαιτήσεων παίρνουν συνεντεύξεις από τους χρήστες .Από την συνέντευξη με τον αγοραστή του συστήματος, παίρνουν μια γενική ιδέα, η οποία αποσαφηνίζεται αργότερα, για το πεδίο εφαρμογής του συστήματος , δηλ. σε τι θα χρησιμεύσει και που. Επίσης πληροφορούνται για την οργάνωση της δουλειάς και πιθανές προεκτάσεις του συστήματος. Από την συνέντευξή τους με τους τελικούς χρήστες πληροφορούνται για τις απαιτήσεις που έχουν από το σύστημα, για παράδειγμα, πώς θα ήθελαν να τους οργανώνει τη δουλειά τους και σε ποια σημεία θέλουν να τους διευκολύνει. Πληροφορούνται ποια ακριβώς είναι τα καθήκοντα του κάθε χρήστη και ποια τα σημεία επαφής του με τους συναδέλφους του,δηλ, τι δεδομένα δέχονται και τι αποτελέσματα δίνουν. Υπάρχει ακόμα η δυνατότητα να παρακολουθούν τους χρήστες ,την ώρα που δουλεύουν και να κρατούν σημειώσεις . Γίνεται ακόμα και χρήση σεναρίων, τι θα κάνει για να αντιμετωπίσει μια συγκεκριμένη κατάσταση. Επίσης παίρνουν δείγματα από τα έγγραφα που κινούνται μέσα στη εταιρεία.

- Διατύπωση απαιτήσεων και σύνταξη προδιαγραφών.

Χρησιμοποιώντας τις παραπάνω πληροφορίες, γίνεται η διατύπωση των απαιτήσεων σε μια αυστηρή γλώσσα, η οποία δεν επιδέχεται παρερμηνεία και παρέχει μια πολύ σαφή και πλήρη περιγραφή των απαιτήσεων.

- Επικύρωση απαιτήσεων ή προδιαγραφών.

Εφόσον έχει ετοιμαστεί το έγγραφο που περιέχει την παραπάνω σύνταξη προδιαγραφών, καλείται μια ομάδα από χρήστες να το επικυρώσει. Στόχος είναι να έρθουν οι δύο πλευρές σε μια συμφωνία, οποία θα τηρηθεί σε όλη την υπόλοιπη διάρκεια της σχεδίασης . Είναι καλό, ό,τι υποσχεθούμε στην καταγραφή των απαιτήσεων να υλοποιηθεί έστω και αν βγούμε εκτός χρονοδιαγράμματος και προϋπολογισμού. Με αυτό τον τρόπο θα επιδείξουμε συνέπεια , θα κερδίσουμε την εμπιστοσύνη των χρηστών και θα αποδείξουμε την αξιοπιστία των υπηρεσιών μας.

Οι κύριοι στόχοι της ανάλυσης απαιτήσεων είναι:

1. Η εμπλοκή των εκπαιδευτικών και, κυρίως, των μαθητών σε θέματα οικολογίας, περιβάλλοντος και περιβαλλοντικής εκπαίδευσης.

2. Η διασύνδεση και παιδαγωγική αλληλεπίδραση μεταξύ φορέων δευτεροβάθμιας και τριτοβάθμιας εκπαίδευσης σε θέματα

- έρευνας, αναζήτησης και συλλογής του πρωτογενούς υλικού της εφαρμογής
- διεπιστημονικής και εκπαιδευτικής προσέγγισης του θέματος
- αξιολόγησης του τελικού προϊόντος.

3. Η μεταφορά εμπειριών και τεχνογνωσίας από το χώρο της τριτοβάθμιας εκπαίδευσης και η διάχυση στους εκπαιδευτικούς και τα σχολεία της δευτεροβάθμιας εκπαίδευσης καινοτόμων εργαλείων και τεχνολογιών, όπως είναι τα πολυμέσα.

4. Η παραγωγή μιας εκπαιδευτικής εφαρμογής η οποία

- να χαρακτηρίζεται από ένα εύχρηστο και φιλικό περιβάλλον διεπαφής
- να διαθέτει όλα τα στοιχεία αξιοπιστίας ενός σύγχρονου εκπαιδευτικού λογισμικού
- να μπορεί να αξιοποιηθεί άμεσα στην εκπαιδευτική διαδικασία (σε δημοτικά, γυμνάσια και λύκεια).

1.4.3 Σχεδιασμός

Σε αυτήν την φάση, καθορίζεται η γενική δομή του λογισμικού και οι αποχρώσεις της. Οι δοκιμές αποτελούν αναπόσπαστο και σημαντικό μέρος της διαδικασίας σχεδιασμού λογισμικού. Αυτό το μέρος της διαδικασίας εξασφαλίζει ότι οι ατέλειες έχουν αναγνωριστεί όσο το δυνατόν νωρίτερα. Εδώ δημιουργείται ένα πρότυπο ανάπτυξης λογισμικού. Η ανάλυση και το σχέδιο είναι πολύ κρίσιμες σε ολόκληρο τον κύκλο ανάπτυξης. Οποιαδήποτε δυσλειτουργία στη φάση σχεδίου, θα μπορούσε να είναι πολύ ακριβή για να λύσει στο προχωρημένο στάδιο της ανάπτυξης λογισμικού. Πολλή προσοχή θα πρέπει λαμβάνεται κατά τη διάρκεια αυτής της φάσης. Το λογικό σύστημα του προϊόντος αναπτύσσεται σε αυτήν την φάση.

Στον σχεδιασμό μελετώνται επίσης οι προδιαγραφές και το πώς θα σχεδιαστεί το προϊόν. Καθορίζονται, οι δομικές ενότητες του συστήματος και οι επικοινωνία

μεταξύ αυτών των ενοτήτων. Γίνεται επίσης η επιλογή αλγορίθμων των δομών δεδομένων και των ροών δεδομένων. Ακολουθεί η καταγραφή των σχετικών αποφάσεων και τέλος ο έλεγχος της φάσης του σχεδιασμού. Όταν οι γενικές απαιτήσεις του πελάτη έχουν συγκεντρωθεί θα πρέπει να καταγραφούν αναλυτικά. Αυτό καλείται συχνά ένα έγγραφο πεδίο ή αρχειοθέτηση.

Μέθοδος σχεδιασμού:

Οι ομάδες των χρηστών είναι πολλές και διαφορετικές με δικές τους ανάγκες και χαρακτηριστικά γι' αυτό και ένα καλό λογισμικό πρέπει να απευθύνεται σε όσο περισσότερες ομάδες χρηστών γίνεται. Η τρέχουσα προσέγγιση της εξυπηρέτησης των ατομικών διαφορών των χρηστών είναι η ανθρωποκεντρική σχεδίαση.

Η ανθρωποκεντρική σχεδίαση ως μία προσέγγιση του interactive system development είναι μια επαναληπτική διαδικασία που μέσω της ενεργής ανάμιξης των πιθανών χρηστών κατά τη διάρκεια της σχεδίασης έχει ως σκοπό την ανάπτυξη συστημάτων χρησιμοποιήσιμων και ασφαλών για τους χρήστες τους μειώνοντας έτσι το ρίσκο να δημιουργηθούν αποτυχημένα συστήματα. Τα ανθρωποκεντρικά συστήματα εξουσιοδοτούν και παρακινούν τους χρήστες να εξετάσουν και να μάθουν νέες τεχνικές.

Ακολουθεί ο έλεγχος της φάσης του σχεδιασμού. Στο σημείο του ελέγχου γίνεται αντιστοίχιση με τις αρχικές προδιαγραφές και αναζήτηση λαθών μεταξύ των ενοτήτων.

Πρακτικά, για να μπει σε λειτουργία το το λογισμικό πρέπει να γίνει υλοποίηση του αναλυτικού σχεδιασμού, να γραφτεί δηλαδή ο κώδικας. Στη συνέχεια ακολουθεί η υλοποίηση.

1.4.4 Υλοποίηση

Η υλοποίηση ή αλλιώς κωδικοποίηση αποτελεί το μέρος της διαδικασίας, όπου οι μηχανικοί λογισμικού προγράμματος γράφουν τον κώδικα. Το σχέδιο πρέπει να μεταφραστεί σε μια μορφή αναγνωρίσιμη από μηχανή. Το βήμα παραγωγής κώδικα εκτελεί αυτόν τον στόχο. Εάν το σχέδιο εκτελείται με έναν λεπτομερή τρόπο, η

παραγωγή κώδικα μπορεί να ολοκληρωθεί χωρίς πολλή περιπλοκή. Τα εργαλεία προγραμματισμού όπως οι μεταγλωττιστές, οι διερμηνείς, οι διορθωτές κλπ. χρησιμοποιούνται για να παραγάγουν τον κώδικα. Οι διαφορετικές γλώσσες προγραμματισμού όπως το PASCAL και java, χρησιμοποιούνται για την κωδικοποίηση και επιλέγεται η σωστή γλώσσα προγραμματισμού.

Στην φάση αυτή πραγματοποιείται η συνένωση των δομικών ενοτήτων του λογισμικού καθώς επίσης και ο έλεγχος της σωστής λειτουργίας του προϊόντος, ως μία ενιαία οντότητα. Η υλοποίηση και η συνένωση του κώδικα πρέπει να γίνονται παράλληλα.

Σε αυτό το σημείο τεκμηριώνεται η εσωτερική σχεδίαση του λογισμικού με σκοπό την μελλοντική συντήρηση και αναβάθμιση που θα γίνεται σε όλη την διάρκεια της ανάπτυξης. Είναι πολύ σημαντικό να τεκμηριωθούν τα πάντα στο έργο.

Οι δοκιμές του λογισμικού αποτελούν αναπόσπαστο και σημαντικό μέρος της διαδικασίας ανάπτυξης λογισμικού. Αυτό το μέρος της διαδικασίας εξασφαλίζει ότι οι ατέλειες έχουν αναγνωριστεί όσο το δυνατόν νωρίτερα.

Ακολουθεί ο έλεγχος της φάσης υλοποίησης. Εξετάζεται δηλαδή η λειτουργικότητα με βάση τις προδιαγραφές, οι πιθανοί περιορισμοί, ορθότητα, αξιοπιστία και συμβατότητα με άλλα λογισμικά. Και τέλος πραγματοποιείται έλεγχος αποδοχής από τον πελάτη με πραγματικά, πλέον, δεδομένα.

1.4.5 Τεκμηρίωση

Στη συνέχεια ακολουθεί η τεκμηρίωση, δηλαδή επιπλέον ανάλυση των περιπτώσεων και αποτελεσμάτων. Αυτή η φάση είναι ένα από τα τελικά βήματα το οποίο είναι απαραίτητο, ώστε το νέο ή τροποποιημένο σύστημα να μπει σε λειτουργία. Σε αυτό το στάδιο γίνονται δοκιμές, εκπαίδευση και μετατροπή. Το λογισμικό δοκιμάζεται για να διαπιστωθεί αν λειτουργεί ικανοποιητικά τόσο από την τεχνική, όσο και από την επιχειρηματική και λειτουργική άποψη. Το προσωπικό της επιχείρησης και οι ειδικοί τεχνικοί εκπαιδεύονται στη χρήση του νέου συστήματος. Το επίσημο πρόγραμμα μετατροπής περιέχει λεπτομερές

χρονοδιάγραμμα όλων των δραστηριοτήτων εγκατάστασης του νέου συστήματος και αντικατάστασης του παλιού συστήματος από το νέο.

Γίνεται ανασκόπηση του κώδικα, έλεγχος περιπτώσεων, άτυπος και τυπικός έλεγχος από το τμήμα διασφάλισης ποιότητας λογισμικού.

Η εγκατάσταση ξεκινά εφόσον ο κώδικας έχει δοκιμαστεί καταλλήλως, έχει εγκριθεί για έκδοση ή πώληση, ή διανέμεται σε ένα περιβάλλον παραγωγής.

Η εκπαίδευση και υποστήριξη του λογισμικού είναι σημαντική και πολλοί προγραμματιστές αποτυγχάνουν να το συνειδητοποιήσουν αυτό. Δεν θα έχει σημασία πόσος χρόνος χρειάστηκε από τον σχεδιασμό, την ανάπτυξη μέχρι και την λειτουργία του λογισμικού, αν κανένας δεν καταφέρει να το χρησιμοποιήσει. Οι άνθρωποι είναι συχνά αρνητικοί στις αλλαγές, οπότε θα πρέπει να αποφεύγεται η διανομή εγχειριδίων, έτσι ως μέρος της φάσης ανάπτυξης, είναι πολύ σημαντικό να πραγματοποιούνται μαθήματα κατάρτισης για τους νέους χρήστες του λογισμικού.

1.4.6 Συντήρηση

Το στάδιο μετά την ανάπτυξη αποτελείται από τη χρήση και αξιολόγηση του συστήματος, αφού αυτό εγκατασταθεί και μπει σε κανονική λειτουργία. Οι χρήστες και οι ειδικοί τεχνικοί κάνουν έναν έλεγχο μετά την εγκατάσταση, με σκοπό να διαπιστωθεί σε ποιο βαθμό το νέο σύστημα ικανοποιεί τους αρχικούς στόχους του και κατά πόσο χρειάζονται αναθεωρήσεις ή μετατροπές. Αφού το σύστημα ρυθμιστεί με ακρίβεια, θα πρέπει να συντηρείται κατά τη διάρκεια της παραγωγής του για να διορθωθούν λάθη, να ικανοποιηθούν απαιτήσεις, ή να βελτιωθεί η απόδοση επεξεργασίας. Το στάδιο της συντήρησης περιλαμβάνει οποιαδήποτε αλλαγή αφότου ο πελάτης έχει παραλάβει το προϊόν λογισμικού και πρέπει πάντα να θεωρείται ως αναπόσπαστο τμήμα της διαδικασίας ανάπτυξης του προϊόντος. Βασικό και συνηθισμένο πρόβλημα είναι η έλλειψη καλής τεκμηρίωσης. Στη συνέχεια πραγματοποιείται ο έλεγχος, όχι μόνο στο αν έγιναν σωστά οι αλλαγές, αλλά και αν αυτές επέφεραν κάποιες άλλες ανεπιθύμητες αλλαγές. Σε αυτό το στάδιο το σύστημα χρησιμοποιείται και αξιολογείται ενώ βρίσκεται σε λειτουργία, και τροποποιείται για να γίνουν βελτιώσεις ή να ικανοποιηθούν νέες απαιτήσεις. Το

στάδιο της συντήρησης είναι το πιο ακριβό στάδιο, κοστίζει περισσότερο απ' ότι όλες οι άλλες φάσεις μαζί.

Σχετικά με τη διατήρηση και ενίσχυση του λογισμικού για την αντιμετώπιση των προβλημάτων που ανακαλύφθηκαν πρόσφατα ή νέες απαιτήσεις, μπορεί να πάρει πολύ περισσότερο χρόνο από την αρχική ανάπτυξη του λογισμικού. Είναι πιθανό να χρειαστεί να προστεθεί κώδικας που δεν χωράει το αρχικό σχέδιο για να διορθώσει κάποιο απρόβλεπτο πρόβλημα ή μπορεί το πρόβλημα να είναι, ότι ο πελάτης ζητά περισσότερη λειτουργικότητα και ο υπάρχων κώδικας μπορεί να μην μπορεί να ανταποκριθεί στα αιτήματά τους. Αν το κόστος εργασίας από τη φάση συντήρησης υπερβαίνει το 25% του κόστους εργασίας των προτέρων φάσεων, τότε είναι πιθανό η συνολική ποιότητα να είναι κακή. Στην περίπτωση αυτή, η διαχείριση πρέπει να εξετάσει τη δυνατότητα για ανοικοδόμηση του συστήματος (ή τμημάτων του συστήματος) πριν το κόστος συντήρησης να φτάσει εκτός ελέγχου. Με την πάροδο του χρόνου το σύστημα θα χρειάζεται όλο και περισσότερη συντήρηση για να παραμένει αποδοτικό και να ικανοποιεί τις ανάγκες των χρηστών, μέχρι τη στιγμή που θα κριθεί ότι έφτασε στο τέλος της ωφέλιμης ζωής του. Όταν ολοκληρωθεί ο κύκλος ζωής του συστήματος, τότε θα απαιτηθεί ένα εντελώς νέο σύστημα, οπότε το λογισμικό αποσύρεται και ο κύκλος μπορεί να αρχίσει ξανά.

1.4.7 Αποχώρηση

Το στάδιο της απόσυρσης είναι η διαδικασία στην οποία το λογισμικό γράφεται από την αρχή. Δεν φτάνουν όλα τα λογισμικά προϊόντα στο στάδιο της απόσυρσης. Συμβαίνει όταν το προϊόν είναι ελαττωματικό. Ένα λογισμικό θα οδηγηθεί στην απόσυρση αν απαιτούνται δραστικές αλλαγές στο σχεδιασμό του όσον αφορά την τελική του μορφή, εάν η συντήρηση του καταντά αδύνατη ή ασύμφορη, εάν αντικατασταθεί το hardware ή το λειτουργικό σύστημα και σε περίπτωση που η τεκμηρίωση είναι ανύπαρκτη ελλιπής ή ανακριβής. Η πραγματική (ολική) απόσυρση ενός προϊόντος είναι σπάνιο φαινόμενο. Το «καλό» λογισμικό συντηρείται.

1.4.8 Μειονεκτήματα της μεθόδου κύκλου ζωής:

Η μέθοδος του κύκλου ζωής συστημάτων εκτός από τη διαδικασία ανάπτυξης λογισμικών, χρησιμοποιείται και για τη δημιουργία μεγάλων συστημάτων επεξεργασίας συναλλαγών και συστημάτων πληροφοριών διοίκησης, στα οποία οι απαιτήσεις είναι πολύ καλά δομημένες και προσδιορισμένες. Θα συνεχίσει επίσης να είναι κατάλληλη για περίπλοκα τεχνικά συστήματα, όπως για εκτοξεύσεις στο διάστημα, για έλεγχο εναέριας κυκλοφορίας, και για λειτουργίες διυλιστηρίων. Τέτοιες εφαρμογές χρειάζονται επακριβή και τυποποιημένη ανάλυση απαιτήσεων, προκαθορισμένες προδιαγραφές, και αυστηρούς ελέγχους της διαδικασίας ανάπτυξης συστημάτων.

Η μέθοδος του κύκλου ζωής συστημάτων έχει όμως μεγάλο κόστος και είναι χρονοβόρα και ανελαστική. Πρέπει να παράγονται μεγάλες ποσότητες νέων εγγράφων τεκμηρίωσης και τα βήματα να επαναλαμβάνονται σε περίπτωση που χρειαστεί να γίνει αναθεώρηση των απαιτήσεων και των προδιαγραφών. Λόγω του χρόνου και του κόστους επανάληψης των φάσεων του κύκλου ζωής, αυτή η μεθοδολογία παρακινεί την οριστικοποίηση των προδιαγραφών σε πρώιμο στάδιο της διεργασίας ανάπτυξης και αποθαρρύνει τις αλλαγές. Η συγκεκριμένη μέθοδος δεν είναι τόσο κατάλληλη για εφαρμογές υποστήριξης αποφάσεων, όπου αυτοί που παίρνουν αποφάσεις μπορεί να χρειάζονται να πειραματιστούν με συγκεκριμένα συστήματα για να ξεκαθαρίσουν το είδος των αποφάσεων που θέλουν να πάρουν. Η ανάγκη για επίσημη προδιαγραφή των απαιτήσεων είναι πιθανό να παρεμποδίζει τους

δημιουργούς συστημάτων στην εξερεύνηση και ανακάλυψη της δομής του προβλήματος. Επίσης, η μέθοδος του κύκλου ζωής δεν είναι κατάλληλη για πολλά μικρά συστήματα επιτραπέζιων υπολογιστών, τα οποία τείνουν να είναι λιγότερο δομημένα και εξατομικευμένα.

1.5 Δημιουργία μοντέλων

Η δημιουργία μοντέλων είναι εναλλακτική μέθοδος ανάπτυξης συστημάτων και μπορεί να αντιμετωπίσει μερικούς από τους περιορισμούς του παραδοσιακού κύκλου ζωής συστημάτων. Σε αυτές τις μεθόδους περιλαμβάνεται η δημιουργία μοντέλων. Η δημιουργία μοντέλου είναι η διεργασία ανάπτυξης ενός πειραματικού συστήματος γρήγορα και με μικρό κόστος για παρουσίαση στους χρήστες και αξιολόγηση, ώστε αυτοί να μπορέσουν να καθορίσουν καλύτερα τις ανάγκες τους σε πληροφορίες.

Σκοπός της δημιουργίας μοντέλων είναι η γρήγορη ανάπτυξη ενός πειραματικού συστήματος με μικρό κόστος. Σε συνδυασμό με την αλληλεπίδρασή τους με το πρωτότυπο, οι χρήστες μπορούν να αποκτήσουν μια πιο ολοκληρωμένη άποψη των αναγκών τους σε πληροφορίες. Το πρωτότυπο που επικυρώνεται από τους χρήστες μπορεί κατόπιν να χρησιμοποιηθεί ως πρότυπο για τη δημιουργία του τελικού λογισμικού.

Τα μοντέλα κύκλου ζωής λογισμικού είναι μια λειτουργική έκδοση ενός συστήματος πληροφοριών, ή ενός μέρους του, αλλά θεωρείται απλώς προκαταρκτικό μοντέλο. Αφού γίνει λειτουργικό, το μοντέλο θα συνεχίσει να βελτιώνεται μέχρι να ικανοποιήσει απόλυτα τις απαιτήσεις των πελατών. Όταν ολοκληρωθεί ο σχεδιασμός, το πρωτότυπο μπορεί να μετατραπεί σε ένα πλήρες σύστημα παραγωγής.

Η διαδικασία ανάπτυξης ενός προκαταρκτικού σχεδιασμού, δοκιμής, βελτίωσής του, και νέας δοκιμής είναι μια επαναληπτική διαδικασία ανάπτυξης συστημάτων, επειδή τα βήματα για την ανάπτυξη του συστήματος μπορεί να επαναλαμβάνονται ξανά και ξανά. Η δημιουργία ενός μοντέλου είναι σαφώς πιο επαναληπτική διαδικασία από τη συμβατική μέθοδο του κύκλου ζωής και ενθαρρύνει ενεργά τις αλλαγές στο σχεδιασμό του συστήματος. Θεωρείται ότι η δημιουργία πρωτοτύπου αντικαθιστά την απρογραμμάτιστη επανάληψη εργασίας με προγραμματισμένη επαναληπτική εργασία, στην οποία κάθε έκδοση αντανάκλα με μεγαλύτερη ακρίβεια τις απαιτήσεις των χρηστών.

1.5.1 Τα βήματα της δημιουργίας μοντέλου

Βήμα 1. Καθορισμός των βασικών αναγκών του χρήστη: εδώ ο σχεδιαστής του συστήματος, ο οποίος είναι συνήθως ένας ειδικός στα συστήματα πληροφοριών, δουλεύει μαζί με το χρήστη μόνο στο βαθμό που χρειάζεται για να καταλάβει τις βασικές του ανάγκες πληροφοριών.

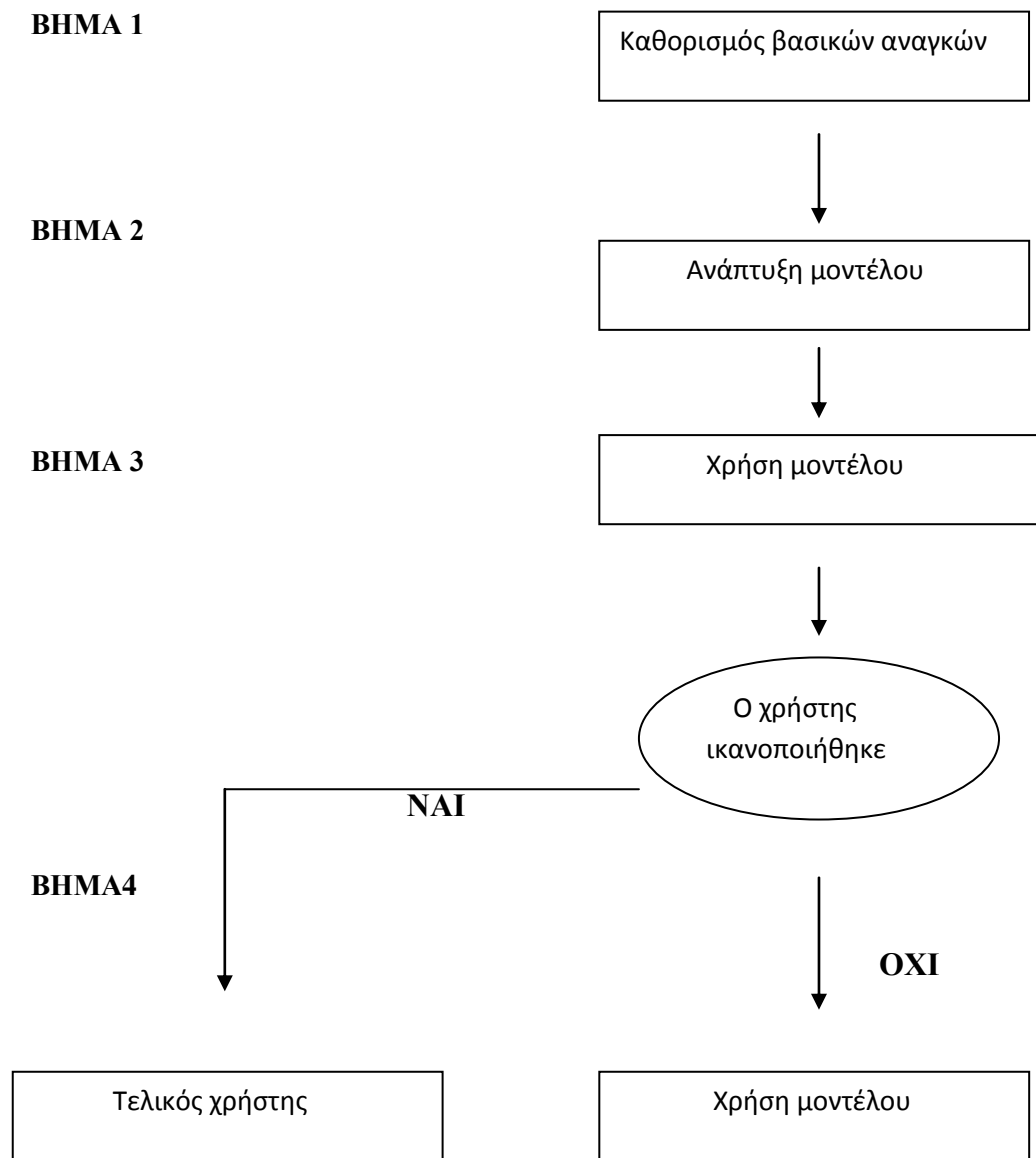
Βήμα 2. Ανάπτυξη αρχικού πρωτοτύπου: Σε αυτό το σημείο ο σχεδιαστής του συστήματος κατασκευάζει γρήγορα ένα λειτουργικό πρωτότυπο, με τη χρήση μιας γλώσσας τέταρτης γενιάς, αλληλεπιδραστικών πολυμέσων, ή με εργαλεία παραγωγής λογισμικού με τη βοήθεια υπολογιστή.

Βήμα 3. Χρήση του μοντέλου: Εδώ ο χρήστης παρακινείται να δουλέψει με το σύστημα προκειμένου να καταλάβει πόσο καλά το μοντέλο ικανοποιεί τις ανάγκες του και για να κάνει υποδείξεις για την βελτίωσή του.

Βήμα 4. Διόρθωση και Βελτίωση τον πρωτοτύπου: σε αυτό το σημείο ο προγραμματιστής σημειώνει όλες τις αλλαγές που ζητάει ο χρήστης και βελτιώνει ανάλογα το μοντέλο. Μετά τη διόρθωση του πρωτοτύπου, ο κύκλος επιστρέφει στο τρίτο βήμα και επαναλαμβάνεται μέχρι να ικανοποιηθεί ο χρήστης.

Όταν δε χρειάζονται πια άλλες επαναλήψεις, το εγκεκριμένο μοντέλο γίνεται τελικό. Αυτό σημαίνει ότι περιέχει όλες τις προδιαγραφές της εφαρμογής και το μοντέλο είναι έτοιμο για χρήση.

Μερικές φορές το ίδιο το μοντέλο υιοθετείται ως λειτουργική έκδοση του συστήματος.



Σχήμα 1.4

1.5.2 Μοντέλα κύκλου ζωής εκπαιδευτικού λογισμικού

Όπως προαναφέραμε οι φάσεις του κύκλου ζωής είναι διαδικασίες που εφαρμόζονται είτε παράλληλα, είτε με κάποια συγκεκριμένη σειρά. Η εφαρμογή αυτών των διαδικασιών ορίζεται από κάποιο συγκεκριμένο μοντέλο και από το πλαίσιο του έργου. (από πολυτεχνία) Το μοντέλο κύκλου ζωής περιγράφει τις φάσεις και τις εργασίες που γίνονται, τα προϊόντα που παράγονται σε κάθε φάση και τη σειρά διαδοχής. Τα μοντέλα κύκλου ζωής αφορούν τη σειρά των βημάτων

μέσω των οποίων αναπτύσσεται ένα προϊόν λογισμικού. Παρέχουν οδηγίες για τις εργασίες που πρέπει να γίνουν, τη χρονική τους σειρά καθώς και τα κριτήρια μετάβασης από τη μία φάση στην άλλη. Υπάρχει ποικιλία μοντέλων και όλα έχουν δυνατά σημεία και αδύνατα σημεία.

Τα μοντέλα κύκλου ζωής διαφοροποιούνται ως προς: τη σύλληψη ιδέας τρόπου κατασκευής, επιμέρους φάσεις, επαναληπτικότητα εργασιών, εμβέλεια εργασιών καθώς και έκταση στην οποία εφαρμόζονται οι διαδικασίες, ενδιάμεσα προϊόντα και περιγραφή, ενδιάμεσες αποτιμήσεις από τον πελάτη ή τον κατασκευαστή ή οικονομικές και επιχειρηματικές πλευρές.

Είναι πολύ σημαντικό για την Επιστήμη της Μηχανικής, η εύρεση και θεμελίωση μεθόδων για να περιγράφεται, να κατασκευάζεται και να συντηρείται λογισμικό καλής ποιότητας, παρέχοντας οδηγίες που αφορούν στις ενέργειες που θα πρέπει να πραγματοποιηθούν. Ακόμα, χρειάζεται να κατανοηθεί η σειρά που θα πρέπει οι ενέργειες αυτές να γίνουν, οι προϋποθέσεις καθώς και τα κριτήρια με τα οποία θα ολοκληρωθεί μια ενέργεια για να γίνει μετάβαση στην επόμενη.

Δεν υπάρχει μοναδιαίος τρόπος για τον προσδιορισμό των ενεργειών ανάπτυξης, εκπαιδευτικού λογισμικού αν και σε γενικές γραμμές θα πρέπει να περιλαμβάνονται οι διαδικασίες της προδιαγραφής, της ανάπτυξης, της επαλήθευσης και της εξέλιξης.

Ακολουθεί η παρουσίαση των πιο σημαντικών μοντέλων ανάπτυξης εκπαιδευτικού λογισμικού. Πιο συγκεκριμένα για κάθε μοντέλο παρουσιάζονται οι εργασίες που περιλαμβάνει, η σειρά που θα πρέπει αυτές να γίνουν, οι προϋποθέσεις καθώς και τα κριτήρια με τα οποία θα ολοκληρωθούν.

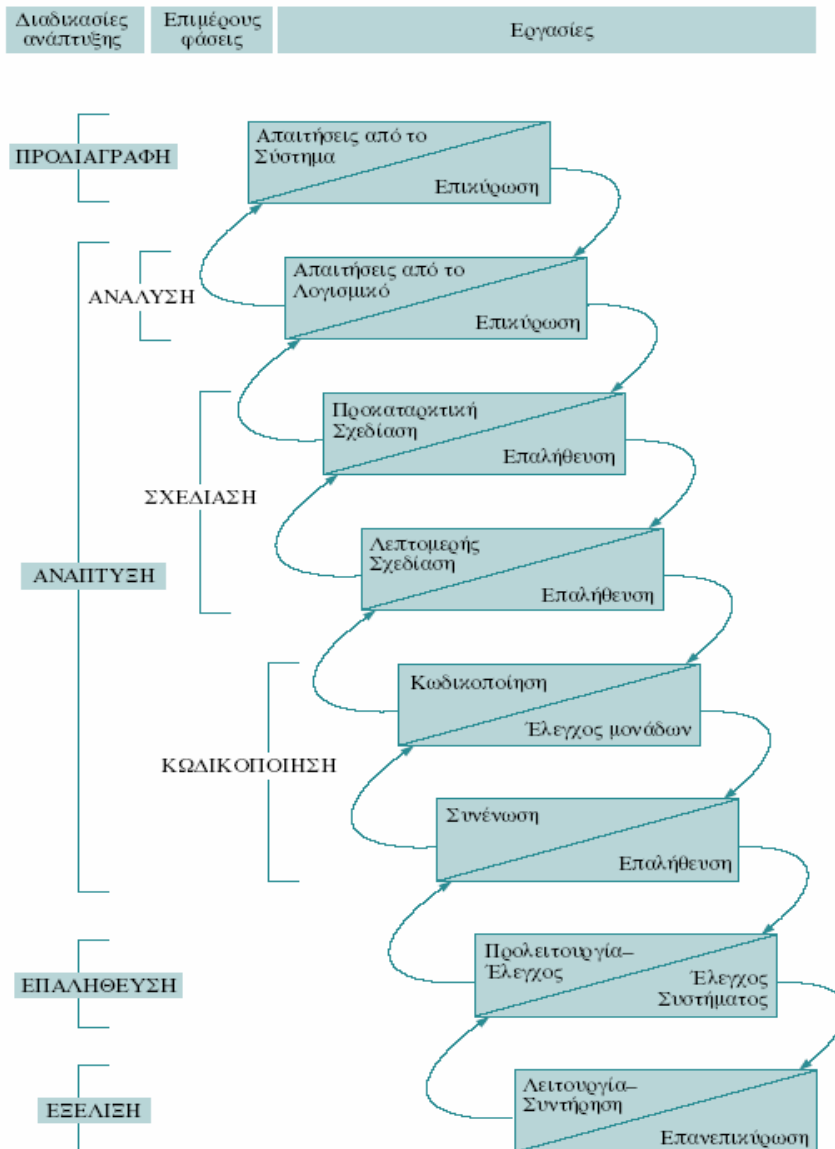
Χαρακτηριστικά μοντέλα κύκλου ζωής λογισμικού είναι: Το Μοντέλο καταρράκτη, το μοντέλο πρωτοτυποποίησης, μοντέλο λειτουργικής επαύξησης, το σπειροειδές μοντέλο, το μοντέλο πίδακα, το γενικό μοντέλο κύκλου ζωής και άλλα.

1.5.3 Μοντέλο καταρράκτη

Το μοντέλο του Καταρράκτη αναπτύχθηκε από τον Royce, το 1970 και περιλαμβάνει 8 διακριτές φάσεις. Ήταν το πρώτο μοντέλο που δημιουργήθηκε και έγινε ευρέως αποδεκτό, ενώ ακόμα παραμένει δημοφιλές ιδιαίτερα για μικρά ή

μεσαία μεγέθη εφαρμογών αφού συμβάλλει στην επιτυχή κατασκευή αξιόπιστων προϊόντων σε μικρό χρονικό διάστημα.

Στο μοντέλο αυτό οι διάφορες φάσεις διαχωρίζονται και ακολουθούνται σειριακά. Η κάθε φάση παράγει ενδιάμεσα αποτελέσματα τα οποία χρησιμοποιούνται από τις επόμενες φάσεις και κορυφώνεται από μια διαδικασία επικύρωσης ή επαλήθευσης των προϊόντων που παράγονται, με σκοπό να απαλειφθούν τυχόν σφάλματα.



Σχήμα 1.5

Το μοντέλο του καταρράκτη χαρακτηρίζεται από:

Επικύρωση. Αυτό σημαίνει πως επαληθεύει ότι έχουν υλοποιηθεί όλες οι προδιαγραφές και το λογισμικό ανταποκρίνεται στις απαιτήσεις από το σύστημα, το λογισμικό κάνει αυτό που ήθελε ο πελάτης να κάνει.

Επαλήθευση. Επαληθεύει δηλαδή ότι κάθε λειτουργία/μονάδα του συστήματος εκτελείται σωστά από προγραμματιστικής άποψης, το λογισμικό κάνει σωστά αυτό που ήθελε ο πελάτης να κάνει.

Κάθε επιμέρους φάση ολοκληρώνεται με μία εργασία επαλήθευσης ή επικύρωσης των προϊόντων της κατά την οποία αποφασίζεται η μετάβαση ή όχι στην επόμενη φάση.

Το μοντέλο του καταρράκτη έχει ακολουθιακή ανάπτυξη. Αυτό σημαίνει ότι οι επιμέρους φάσεις είναι διακριτές και ακολουθούν η μία την άλλη. Για να ξεκινήσει μία φάση πρέπει να έχει ολοκληρωθεί πλήρως η προηγούμενη

Το μοντέλο του καταρράκτη χαρακτηρίζεται επίσης από: σειριακά βήματα, ανάδραση ανάμεσα σε δύο γειτονικά βήματα και βασίζεται στην δημιουργία προδιαγραφών σε κάθε βήμα.

Στάδια

- Καθορισμός απαιτήσεων από το σύστημα και από το λογισμικό.
- Προκαταρκτική σχεδίαση λογισμικού εδώ καθορίζονται οι μονάδες που θα αποτελούν το λογισμικό και οι μεταξύ τους συσχετίσεις, σε περισσότερα από ένα επίπεδα λεπτομέρειας, ανάλογα με το μέγεθος και την πολυπλοκότητα.
- Λεπτομερή σχεδίαση λογισμικού σε αυτό το στάδιο καθορίζεται η εσωτερική δομή της κάθε μονάδας λογισμικού (αλγόριθμοι, δομές δεδομένων κτλ)
- Συγγραφή πηγαίου κώδικα
- Συνένωση μονάδων σε σύστημα και έλεγχος συστήματος

Στη συνέχεια το προϊόν παραδίδεται στον πελάτη και περνάει στη φάση λειτουργίας και συντήρησης.

Πλεονεκτήματα

Το μοντέλο του καταρράκτη είναι το πιο διαδομένο μοντέλο κύκλου ζωής Λογισμικού και είναι πολύ χρήσιμο όταν οι απαιτήσεις από το λογισμικό είναι γνωστές από την αρχή και δεν μεταβάλλονται κατά την ανάπτυξη του λογισμικού.

Επίσης το μοντέλο του καταρράκτη διευκολύνει την συντήρηση επισή υπάρχουν προδιαγραφές.

Μειονεκτήματα

Τα μειονεκτήματα που παρουσιάζει το μοντέλο του καταρράκτη εντοπίζονται στα εξής: στο το ότι γνωρίζουμε αν θα είναι ικανοποιημένος ο χρήστης μόνο στο τελικό στάδιο (ουσιαστικά αργά), στο ότι δεν προβλέπει επαναχρησιμοποίηση του λογισμικού που πιθανά υπάρχει, και στο ότι είναι απαιτητικό τόσο σε χρόνο όσο και σε κόστος και αρκετά δύσκαμπτο μοντέλο. Δεν είναι δυνατό να υπάρχει μια πρώτη εικόνα του συστήματος λογισμικού που κατασκευάζεται, παρά μόνο σε προχωρημένη φάση της ανάπτυξης. Όσο μεγαλώνει η έκταση της εφαρμογής λογισμικού, τόσο δυσκολότερη η μετάβαση από τη μία φάση στην επόμενη και η αποφυγή σφαλμάτων τα οποία εντοπίζονται πολύ αργότερα. Όσο αργότερα στην ανάπτυξη εντοπίζεται ένα σφάλμα, τόσο μεγαλύτερες οι επιπτώσεις που η διόρθωσή του μπορεί να έχει σε κόστος οπισθοδρόμησης, παρενέργειες, καθυστερήσεις, δημιουργία νέων σφαλμάτων.

Τα προβλήματα αυτά προσπαθεί να τα απαλείψει το μοντέλο V, που αποτελεί παραλλαγή του μοντέλου του καταρράκτη. Το μοντέλο αυτό διαφοροποιείται περισσότερο στον τρόπο που γίνεται ο έλεγχος, η διόρθωση, η επαλήθευση, η επικύρωση καθώς και ο τρόπος που θα πραγματοποιηθούν αυτές οι φάσεις. Δηλαδή σε ποια φάση θα πρέπει να επιστρέψουμε αν χρειαστεί ενώ παράλληλα οι φάσεις που περιλαμβάνει χαρακτηρίζονται είτε από την σύμπτυξη είτε από τον επιμέρους διαχωρισμό των αντίστοιχων φάσεων του μοντέλου του καταρράκτη.

Συμπέρασμα

Το μοντέλο του Καταρράκτη θεωρεί ότι η ανάπτυξη του εκπαιδευτικού λογισμικού θα πρέπει να γίνει ακολουθιακά και σε όλα τα στάδια. Η προσέγγιση αυτή είναι άλλοτε λογική και άλλοτε, όχι τόσο απαραίτητη.

Όσον αφορά την αναγκαιότητα όλων των φάσεων είναι ολοφάνερο ότι τόσο η φάση 'Κωδικοποίηση και Εκκαθάριση Λαθών' όσο και όλες οι επόμενες δηλαδή 'Συνένωση', 'Έλεγχος και Προλειτουργία και 'Λειτουργία και Συντήρηση' είναι

αναγκαίες για όλα τα προϊόντα λογισμικού. Το ζητούμενο είναι αν είναι απαραίτητες και οι προηγούμενες φάσεις.

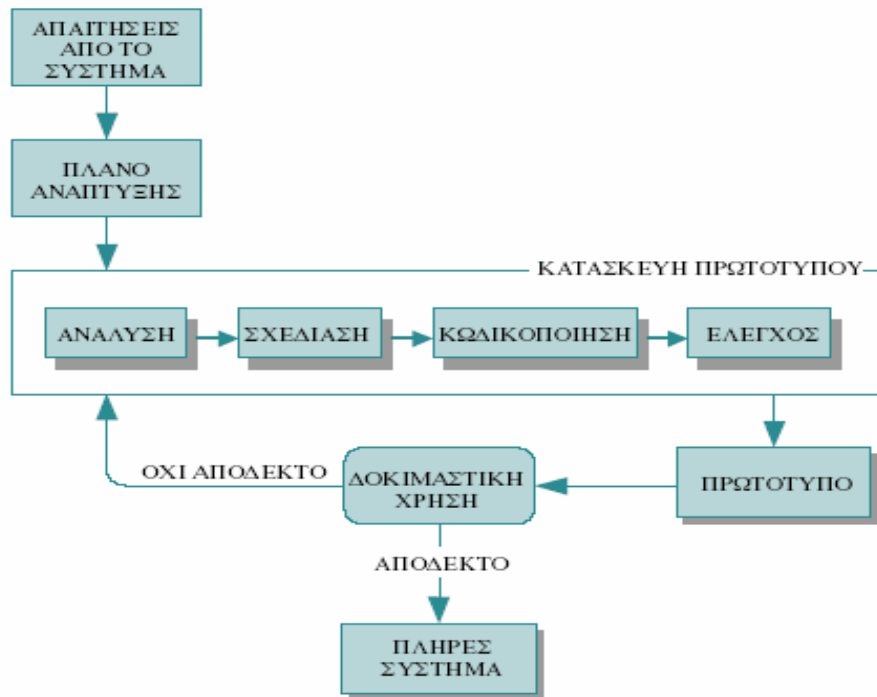
Είναι δύσκολο να καθοριστεί αν οι προηγούμενες φάσεις είναι απαραίτητες σε μικρά προϊόντα λογισμικού παρόλα αυτά καλόν είναι να μην αγνοούνται ούτε και εκεί. Σε μεγάλα προϊόντα οι φάσεις αυτές είναι απαραίτητες όπως έδειξε η εμπειρία αφού υπήρξαν περιπτώσεις που ο κώδικας ξαναγράφηκε αφού υπήρξαν παρανοήσεις στις απαιτήσεις του συστήματος. Επίσης η ανάπτυξη λογισμικού χρειάστηκε να σταματήσει και να ακυρωθεί επειδή υπήρξαν ελλείψεις στις προδιαγραφές των απαιτήσεων του λογισμικού και έλλειψη μελέτης εφικτότητας του συστήματος.

Όσον αφορά στην ακολουθιακή εκτέλεση των διαφόρων φάσεων αυτή είναι υποχρεωτική αφού κάθε επόμενη φάση χρησιμοποιεί τα προϊόντα της προηγούμενης φάσης. Έτσι η ακολουθιακή σχεδίαση με την προϋπόθεση ότι τα ενδιάμεσα προϊόντα της προηγούμενης φάσης είναι ορθά, επιτρέπει την ανάπτυξη λογισμικού σε σύντομο χρονικό διάστημα άρα οικονομικό και το οποίο είναι σωστό.

1.5.4 Μοντέλο πρωτοτυποποίησης

Ένας τρόπος να δούμε την πρωτοτυποποίηση είναι ως μια τεχνική για μείωση του κινδύνου. Ο πιο σημαντικός κίνδυνος στην ανάπτυξη λογισμικού είναι τα λάθη και περισσότερο οι παραλείψεις που προκύπτουν από μη σαφείς απαιτήσεις των χρηστών για το τελικό σύστημα. Το κόστος της διόρθωσης αυτών των λαθών και παραλείψεων σε επόμενα στάδια μπορεί να είναι πολύ υψηλό. Είναι προφανές ότι η δημιουργία ενός πρωτοτύπου μπορεί να μειώσει τον αριθμό των προβλημάτων των απαιτήσεων και ως εκ τούτου να μειώσει το συνολικό κόστος ανάπτυξης.

Μια σχηματική αναπαράσταση της διαδικασίας ανάπτυξης ενός πρωτοτύπου φαίνεται στο παρακάτω σχήμα.



Σχήμα 1.6

- Σχεδίαση πρωτότυπου ακολουθούμενου από το μοντέλο Καταρράκτη
 - Το πρωτότυπο δεν είναι το προϊόν
 - Η Πρωτοτυποποίηση δεν πρέπει να είναι μέρος της σχεδίασης αλλά μόνο της συλλογής απαιτήσεων
 - Προτερήματα
 - Καλύτερη προδιαγραφή απαιτήσεων
 - Καλύτερη μελέτη σκοπιμότητας
 - Ο χρήστης συμμετέχει ενεργά στη διαδικασία συλλογής / μοντελοποίησης απαιτήσεων
 - Μειονεκτήματα
 - Περισσότερη εργασία απαιτείται για την παραγωγή του πρωτότυπου
 - Λόγω χρονικών περιορισμών το πρωτότυπο γίνεται μέρος του συστήματος

Αρχικά θα πρέπει να καθοριστούν επακριβώς οι στόχοι του πρωτότυπου. Το πρωτότυπο σύστημα μπορεί να αφορά τη διεπιφάνεια χρήστη ή να περιέχει τις λειτουργίες εκείνες που θεωρούνται περισσότερο κρίσιμες. Είναι προφανές ότι ένα πρωτότυπο δεν μπορεί να καλύπτει όλες τις απαιτήσεις του συστήματος. Για τον

λόγο αυτό κάθε φορά θα πρέπει να ορίζονται πλήρως οι απαιτήσεις που αυτό θα καλύπτει, αλλιώς τελικά μπορεί να μην λάβουμε τα πλεονεκτήματα που μας προσφέρει η μέθοδος αυτή.

Το επόμενο στάδιο αφορά στην επιλογή των λειτουργιών εκείνων από το σύνολο των λειτουργιών του τελικού συστήματος, που θα συμπεριληφθούν στο προς δημιουργία πρωτότυπο (άρα και αυτών που δεν θα συμπεριληφθούν). Ο λόγος για την επιλογή αυτή είναι, γιατί προφανώς έχει πολύ υψηλό κόστος εάν το πρωτότυπο δημιουργηθεί με όλες τις λειτουργίες του τελικού συστήματος. Βέβαια θα μπορούσε να αποφασιστεί να περιλαμβάνονται όλες οι λειτουργίες που έχει αποφασιστεί αλλά σε μειωμένο επίπεδο (πχ. χωρίς διαχείριση λαθών).

Η τελευταία φάση, μετά την ανάπτυξη του πρωτοτύπου, είναι η επαλήθευση του πρωτοτύπου και είναι ίσως η πιο σημαντική φάση. Θα πρέπει να καταγραφούν συμπεράσματα για το πως νιώθουν οι χρήστες με το σύστημα, αν γίνεται κατανοητό το περιβάλλον και η λειτουργία του και να βρεθούν τυχόν λάθη και προβλήματα.

Πλεονεκτήματα

Τα πλεονεκτήματα της χρήσης πρωτοτύπου είναι ότι ανακαλύπτονται και διορθώνονται:

- Παρεξηγήσεις μεταξύ των χρηστών και των δημιουργών.
- Παραλειπόμενες υπηρεσίες στο σύστημα
- Δυσκολίες στη χρήση
- Ασυνέχειες και κενά στις προδιαγραφές

Η πρωτοτυποποίηση μπορεί να χρησιμοποιηθεί και για άλλους σκοπούς, όπως στην εκπαίδευση των χρηστών, δηλαδή το πρωτότυπο να χρησιμοποιηθεί ως εκπαιδευτικό εργαλείο για την εκμάθηση του τελικού συστήματος, αλλά ταυτόχρονα είναι και ένας τρόπος μείωσης του κινδύνου, αφού περιορίζονται τα λάθη και οι παραλείψεις. Αν τα λάθη αφεθούν για διόρθωση στις τελευταίες φάσεις του κύκλου ζωής το κόστος αυξάνεται κατακόρυφα.

Μειονεκτήματα

Ένα βασικό μειονέκτημα του μοντέλου της πρωτοτυποποίησης είναι ότι το κόστος ανάπτυξής του αποτελεί ένα μεγάλο μέρος του συνολικού κόστους του συστήματος

που αναπτύσσεται. Πολλές φορές είναι οικονομικά πιο συμφέρον να μεταβληθεί το τελικό προϊόν από το να δημιουργηθεί ένα πρωτότυπο.

Είναι προφανές, ότι είναι πολύ δύσκολο να προβλεφτεί ποιες ακριβώς δυσκολίες θα αντιμετωπίσει ο τελικός χρήστης από την καθημερινή χρήση ενός νέου συστήματος λογισμικού. Ιδιαίτερα εάν αναφερόμαστε σε μεγάλα συστήματα λογισμικού η δυσκολία αυτή μπορεί να καταφανεί μόνο όταν το ολοκληρωμένο σύστημα αναπτυχθεί και τεθεί σε λειτουργία. Για να αντιμετωπιστεί αυτή η δυσκολία μπορεί να χρησιμοποιηθεί η εξελικτική πρωτοτυποποίηση. Κατά τη προσέγγιση αυτή δημιουργείται μια περιορισμένη και ατελής έκδοση του συστήματος πάνω στην οποία γίνονται οι διορθώσεις και οι προσθήκες, καθώς απαιτήσεις είτε διευκρινίζονται είτε ανακαλύπτονται είτε βελτιώνονται μέχρι να καταλήξουμε σε μια έκδοση που να είναι επαρκής και ικανοποιητική.

Ένα βασικό πρόβλημα του μοντέλου της εξελικτικής πρωτοτυποποίησης είναι ότι με τις συνεχείς διορθωτικές και προσθετικές παρεμβολές παράγεται ‘μπλεγμένος’ κώδικας που είναι πολύ δύσκολα συντηρήσιμος ιδιαίτερα όταν δεν τηρούνται κανόνες ποιότητας λογισμικού.

Εναλλακτικά και προκειμένου να αποφύγουμε το φαινόμενο του «μπλεγμένου κώδικα», θα μπορούσε να δημιουργηθεί ένα πρωτότυπο το οποίο θα είχε αποκλειστικό σκοπό την αποσαφήνιση των απαιτήσεων και την παροχή πληροφοριών για την εκτίμηση του κινδύνου του τελικού συστήματος. Μετά την αξιολόγηση το πρωτότυπο σύστημα πετιέται και δεν χρησιμοποιείται για την ανάπτυξη του συστήματος.

Ο χρόνος που απαιτείται για την ανάπτυξη ενός συστήματος μπορεί να μειωθεί ακόμα περισσότερο εάν κάποια τμήματα του συστήματος μπορούν να επαναχρησιμοποιηθούν. Έτσι τα πρωτότυπα μπορούν να κατασκευαστούν ακόμα πιο γρήγορα εάν υπάρχει μια βιβλιοθήκη με επαναχρησιμοποιήσιμες ψηφίδες και φυσικά κάποιος τρόπος σύνθεσης των ψηφίδων αυτών. Η επαναχρησιμοποίηση ψηφίδων αρμόζει περισσότερο στην throw – away προσέγγιση της πρωτοτυποποίησης.

Συμπέρασμα

Οι προσεγγίσεις της εξελικτικής και της throw away πρωτοτυποποίησης παρουσιάζουν μειονεκτήματα και προβλήματα.

Στην εξελικτική πρωτοτυποποίηση πρέπει να αντιμετωπισθεί το πρόβλημα ότι συνεχείς αλλαγές καταστρέφουν την δομή του πρωτότυπου συστήματος. Η συντήρηση γίνεται δύσκολη και κοστίζει πολύ, ενώ ο κώδικας τείνει να λάβει πολύπλοκη και κακή δομή, γεγονός που επιδρά και στην συντηρησιμότητα του συστήματος καθώς και στον κύκλο ζωής του.

Ένα σημαντικό πρόβλημα στην throw-away πρωτοτυποποίηση, είναι ότι η χρήση του πρωτότυπου μπορεί να μην είναι ίδια με τον τρόπο που χρησιμοποιείται το τελικό σύστημα. Για παράδειγμα, αν το πρωτότυπο χρησιμοποιηθεί για την εκπαίδευση των χρηστών και αυτό έχει αργούς χρόνους στα αποτελέσματά του, παρουσιάζονται προβλήματα αν το τελικό σύστημα λειτουργεί σε πιο σύντομα χρονικά διαστήματα αφού οι χρήστες θα πρέπει να προσαρμόσουν τις εργασίες τους σε διαφορετικούς χρόνους απ' ότι είχαν προβλέψει.

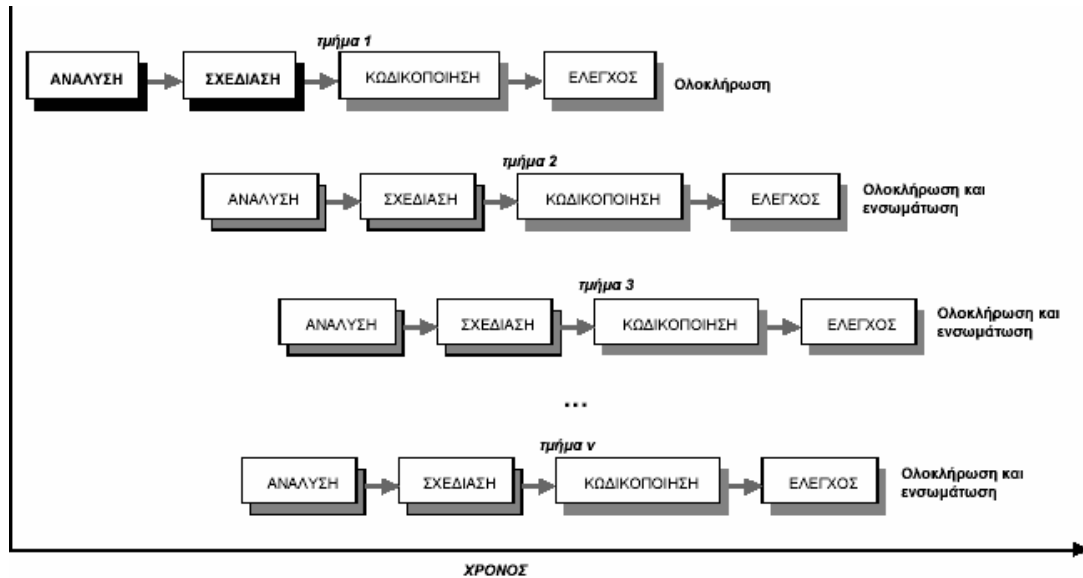
Η throw away προσέγγιση όμως είναι η πιο κατάλληλη για να υποστηρίξει ένα λογισμικό στα αρχικά στάδια του σχεδιασμού του.

Το σημαντικό της δραστηριότητας αυτής είναι η αναφορά στο 'κατασκευάσετε για πρώτη φορά' άρα το πλέον σημαντικό είναι η ασάφεια (σε μικρότερο ή μεγαλύτερο βαθμό) των απαιτήσεων του συστήματος που θα αναπτυχθεί άρα απαιτείται μοντέλο που να μπορεί εύκολα να υποστηρίξει συχνές μεταβολές, προσθήκες και διορθώσεις στις απαιτήσεις του συστήματος, ιδιαίτερα βέβαια όταν η ανάπτυξη αυτή γίνεται για πρώτη φορά και δεν υπάρχει σχετική πρότερη εμπειρία. Για τους λόγους αυτούς προτείνεται η χρήση του μοντέλου της throw – away πρωτοτυποποίησης και στην συνέχεια θα μπορούσε να χρησιμοποιηθεί μια οποιαδήποτε συμβατική μέθοδος (ακόμα και καταρράκτης ή κάποια παραλλαγή της).

1.5.5 Μοντέλο λειτουργικής επαύξησης

Μια εναλλακτική διαδικασία που συνδυάζει τα πλεονεκτήματα της εξελικτικής προσέγγισης με τον έλεγχο που απαιτείται για μεγάλα συστήματα είναι η λειτουργική επαύξηση. Σύμφωνα με το μοντέλο αυτό αρχικά αναπτύσσεται μια έκδοση του συστήματος που περιέχει τις περισσότερο σημαντικές και κρίσιμες λειτουργίες. Από την χρησιμοποίηση αυτής της έκδοσης κερδίζεται εμπειρία η οποία χρησιμοποιείται για την βελτίωσή της. Στην συνέχεια γίνεται μια

προσαύξηση που επεκτείνει την προηγούμενη έκδοση και περιέχει και άλλες λειτουργίες. Η νέα έκδοση εκλεπτύνεται και προσαυξάνεται με την σειρά της με τον ίδιο τρόπο έως ότου κατασκευαστεί η τελική έκδοση.



Σχήμα 1.7

Το μοντέλο ενδείκνυται στις περιπτώσεις που υπάρχει σαφής γνώση και πολύ μικρή ή καθόλου μεταβλητότητα των απαιτήσεων του υπό ανάπτυξη εκπαιδευτικού λογισμικού. Άρα πρόκειται για μοντέλο που χρησιμοποιείται σε λίγες περιπτώσεις μια και το βασικό πρόβλημα της ανάπτυξης εκπαιδευτικού λογισμικού είναι η ασάφεια (σε μικρότερο ή μεγαλύτερο βαθμό) των απαιτήσεων του συστήματος.

Με αυτό το μοντέλο αποφεύγονται προβλήματα που προκύπτουν από τις συνεχείς αλλαγές, όπως στην εξελικτική πρωτοτυποποίηση. Η αρχιτεκτονική του συστήματος καθορίζεται σχετικά νωρίς και λειτουργεί ως πλαίσιο. Τα μέρη που αποτελούν το σύστημα αναπτύσσονται με επαυξήσεις και παραδίδονται με αυτό τον τρόπο.

1.6 Εναλλακτικά Μοντέλα Ανάπτυξης Εκπαιδευτικού Λογισμικού

Τα εναλλακτικά μοντέλα έχουν προταθεί ως μια εναλλακτική λύση στα συμβατικά με σκοπό να ελαττώσουν τα προβλήματα και τις αδυναμίες που έχουν τα συμβατικά μοντέλα.

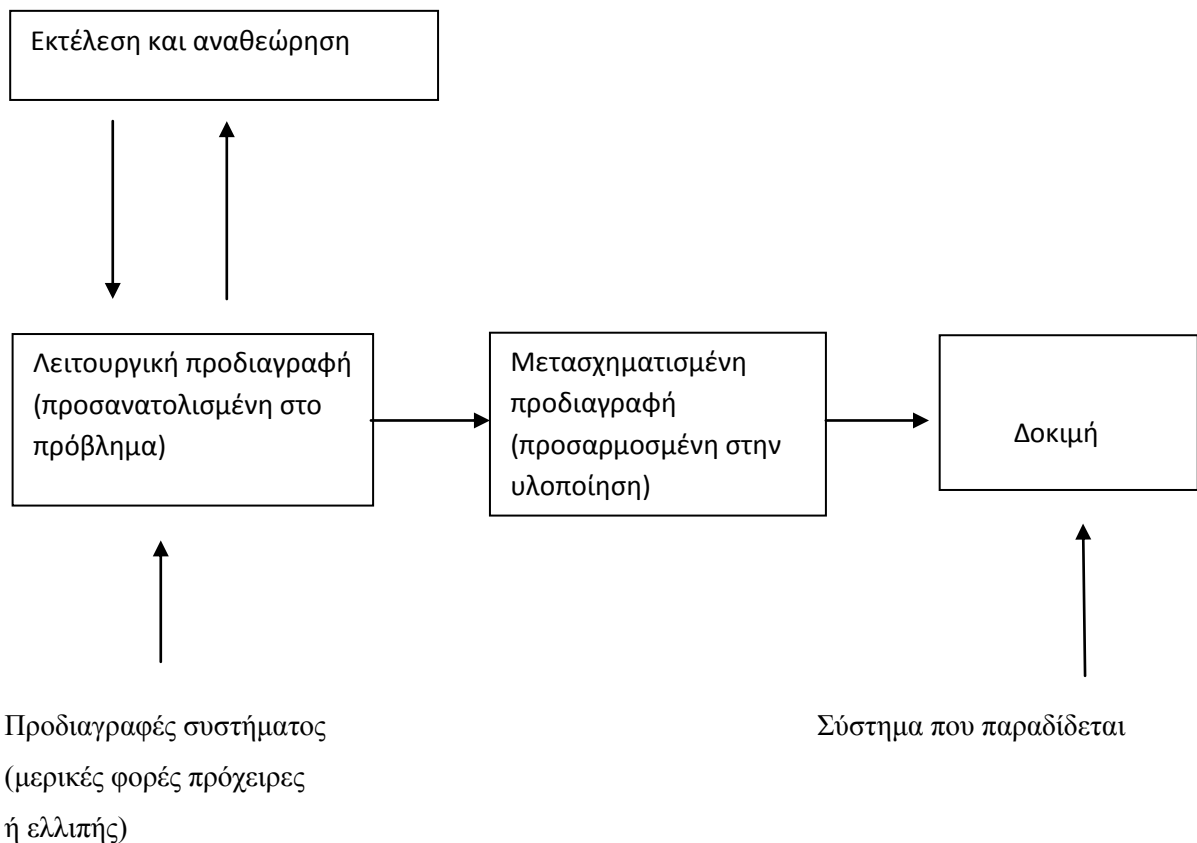
1.6.1 Λειτουργικό Μοντέλο

Το Λειτουργικό μοντέλο χρησιμοποιεί αποκλειστικά τις προδιαγραφές που είναι λειτουργικές και περιγράφουν τι θα κάνει το σύστημα έμμεσα. Έτσι, δημιουργείται μια περιγραφή η οποία δείχνει πως αυτό θα λειτουργεί η οποία υλοποιείται σε κάποια γλώσσα ώστε να μπορεί να εκτελεστεί. Έτσι μπορεί να γίνει φανερή η συμπεριφορά του τελικού συστήματος και ως εκ τούτου να αξιολογηθεί και να βελτιωθεί εάν απαιτείται.

Οι λειτουργικές προδιαγραφές που χρησιμοποιούνται σύμφωνα με το μοντέλο αυτό είναι ένα είδος πρωτότυπου στο οποίο είναι εμφανής όλη η λειτουργική συμπεριφορά του συστήματος, χρησιμοποιώντας διαφορετικά μέσα από αυτά που θα χρησιμοποιούσε το τελικό σύστημα.

Αξιολογώντας τη συμπεριφορά του συστήματος, οι χρήστες μπορούν να κάνουν παρατηρήσεις και αλλαγές στις λειτουργικές προδιαγραφές. Ο κύκλος αυτός αξιολόγησης – αλλαγών επαναλαμβάνεται έως ότου θεωρηθεί ότι το σύστημα έχει την επιθυμητή λειτουργικότητα. Έτσι ολοκληρώνεται η φάση των απαιτήσεων και στην συνέχεια μπορεί να χρησιμοποιηθεί για παράδειγμα το μοντέλο του καταρράκτη από την φάση της σχεδίασης και κάτω.

Βασικό πρόβλημα του μοντέλου αποτελεί το γεγονός ότι χρησιμοποιεί εκτελέσιμες γλώσσες προδιαγραφών που είναι αυστηρά τυπικές και απαιτούν ιδιαίτερες γνώσεις από την ομάδα ανάπτυξης αλλά και εργαλεία υλοποίησης.



Σχήμα 1.8

1.6.2 Μοντέλο αυτόματου προγραμματισμού

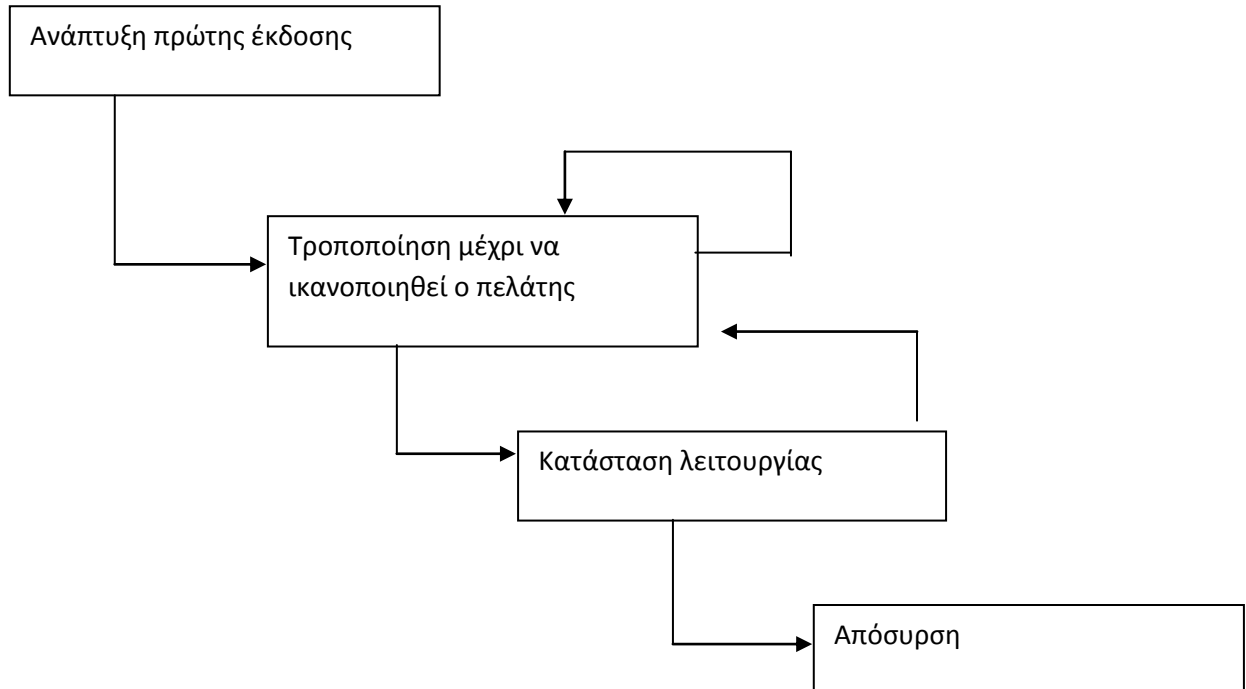
Το μοντέλο αυτόματου προγραμματισμού βασίζεται στην ιδέα της δημιουργίας ενός συστήματος που να μπορεί να δημιουργήσει λογισμικό αυτόματα αφού πρώτα του δοθούν οι προδιαγραφές του προβλήματος.

Η αυτοματοποιημένη δημιουργία λογισμικού είναι μια παλιά ιδέα που χρησιμοποιήθηκε και για την δημιουργία μεταφραστών γλωσσών προγραμματισμού.

Το μοντέλο αυτόματου προγραμματισμού ενδείκνυται για τις περιπτώσεις που οι απαιτήσεις του συστήματος είναι σαφώς καθορισμένες ώστε να μπορούν να περιγραφούν με ένα πολύ τυπικό τρόπο όπως είναι οι γραμματικές χωρίς συμφραζόμενα κλπ.

Τα εργαλεία που χρησιμοποιούνται (μετα-μεταφραστές και οι αντίστοιχες γραμματικές) απαιτούν υψηλές γνώσεις πληροφορικής και επομένως πολύ εξειδικευμένη και έμπειρη ομάδα ανάπτυξης.

1.6.3 Μοντέλο build 'n' fix



Σχήμα 1.9

- Δεν γίνεται καθορισμός προδιαγραφών.
- Δεν γίνεται σχεδιασμός.
- Δουλεύει καλά μόνο σε περιπτώσεις που το λογισμικό δεν ξεπερνά τις 100 – 200 σειρές κώδικα.

1.6.4 Μοντέλο επαναχρησιμοποίησης λογισμικού

Με το μοντέλο επαναχρησιμοποίησης λογισμικού γίνεται χρήση ήδη υπάρχοντος και δοκιμασμένου λογισμικού, σχεδίων και κώδικα. Οι υπάρχουσες ψηφίδες λογισμικού, με ελεγμένη ορθότητα, ενσωματώνονται σε νέα προϊόντα λογισμικού.

Η διαδικασία αυτή δεν είναι εύκολη, αφού παρουσιάζονται δυσκολίες, λόγω της ανυπαρξίας εργαλείων και τεχνικών καταλλήλων για αυτή τη δουλειά, αλλά και της έλλειψης προτύπων κατασκευής ψηφίδων λογισμικού που να μπορούν να επαναχρησιμοποιηθούν.

Τα βασικά πλεονεκτήματα του μοντέλου είναι η συντόμευση του χρονικού διαστήματος κατασκευής λογισμικού αλλά και η βελτίωση της αξιοπιστίας του αφού στηρίζεται σε έτοιμα, δοκιμασμένα και άρα αξιόπιστα τμήματα λογισμικού.

Τα συστατικά λογισμικού που θα επαναχρησιμοποιηθούν μπορεί να είναι πολλών και διαφόρων μεγεθών, όπως για παράδειγμα :

- Επαναχρησιμοποίηση ολόκληρων συστημάτων εφαρμογών τα οποία είτε ενσωματώνονται στο καινούριο σύστημα χωρίς αλλαγή είτε δημιουργούνται ολόκληρες οικογένειες εφαρμογών* που μπορούν να τρέξουν σε διαφορετικές πλατφόρμες ώστε να ικανοποιήσουν συγκεκριμένες ανάγκες.
- Επαναχρησιμοποίηση ψηφίδων όπου διάφορα «συστατικά» μιας εφαρμογής από ένα υποσύστημα μέχρι ένα μεμονωμένο αντικείμενο μπορούν να επαναχρησιμοποιηθούν. Για παράδειγμα αντικείμενο τύπου «χρονόμετρο» μπορεί να επαναχρησιμοποιηθεί πολλές φορές στην ανάπτυξη ενός εκπαιδευτικού λογισμικού φυσικής, χημείας κλπ.
- Επαναχρησιμοποίηση κάποιας λειτουργίας ή συνάρτησης όπου συστατικά (συναρτήσεις, διαδικασίες) που υλοποιούν μια απλή λειτουργία όπως μια μαθηματική συνάρτηση επαναχρησιμοποιούνται σε άλλες εφαρμογές. Αυτού του είδους η επαναχρησιμοποίηση που βασίζεται στις βιβλιοθήκες είναι και η πιο διαδεδομένη.

Συμπέρασμα

Το προφανές κέρδος από την επαναχρησιμοποίηση είναι η μείωση του κόστους ανάπτυξης δεδομένου ότι λιγότερα συστατικά του συστήματος χρειάζεται να προσδιοριστούν, να σχεδιαστούν, να υλοποιηθούν και να αξιολογηθούν.

Εκτός όμως από την μείωση του κόστους ανάπτυξης υπάρχει και ένα σύνολο άλλων πλεονεκτημάτων όπως τα:

Αύξηση της αξιοπιστίας του συστήματος

Οι ψηφίδες που χρησιμοποιούνται έχουν ήδη δοκιμαστεί σε μια ποικιλία από περιβάλλοντα. Τα λάθη στο σχεδιασμό και την υλοποίηση έχουν ήδη εντοπιστεί και διορθωθεί και έτσι μειώνεται και η πιθανότητα να 'αστοχήσει' η συγκεκριμένη ψηφίδα καθώς επαναχρησιμοποιείται.

Μείωση του επεξεργαστικού κινδύνου

Το κόστος επαναχρησιμοποίησης είναι περισσότερο εύκολο να προβλεφθεί με ακρίβεια από το κόστος ανάπτυξης. Έτσι η διεύθυνση έχει λιγότερη ανασφάλεια ως προς την εκτίμηση του κόστους ανάπτυξης.

Χρησιμοποίηση των ειδικών πιο αποτελεσματικά

Αντί να χρειαστεί να σχεδιάζουν τα ίδια πράγματα για διαφορετικά projects αναπτύσσουν επαναχρησιμοποιήσιμες ψηφίδες στις οποίες ενσωματώνουν τις γνώσεις τους.

Ομογενοποίηση των standards

Κάποια τυποποιημένα χαρακτηριστικά, όπως τα χαρακτηριστικά του περιβάλλοντος διεπαφής, μπορούν να υλοποιηθούν ως ένα σύνολο από ψηφίδες. Η χρήση ενός τυποποιημένου περιβάλλοντος διεπαφής αυξάνει την αξιοπιστία του συστήματος αφού οι χρήστες είναι απίθανο να κάνουν λάθη με μια διεπαφή που τους είναι γνωστή.

Επιτάχυνση της διαδικασίας ανάπτυξης

Το προϊόν φτάνει στην αγορά νωρίτερα γιατί η διάρκεια της ανάπτυξης και του ελέγχου αξιοπιστίας (validation) μειώνεται.

1.6.5 Αντικειμενοστραφές μοντέλο

Το αντικειμενοστραφές μοντέλο βασίζεται στον αντικειμενοστραφή προγραμματισμό. Αναπτύσσεται με τρόπο παρόμοιο με το μοντέλο του καταρράκτη, αλλά διαφέρει σε δύο βασικά σημεία:

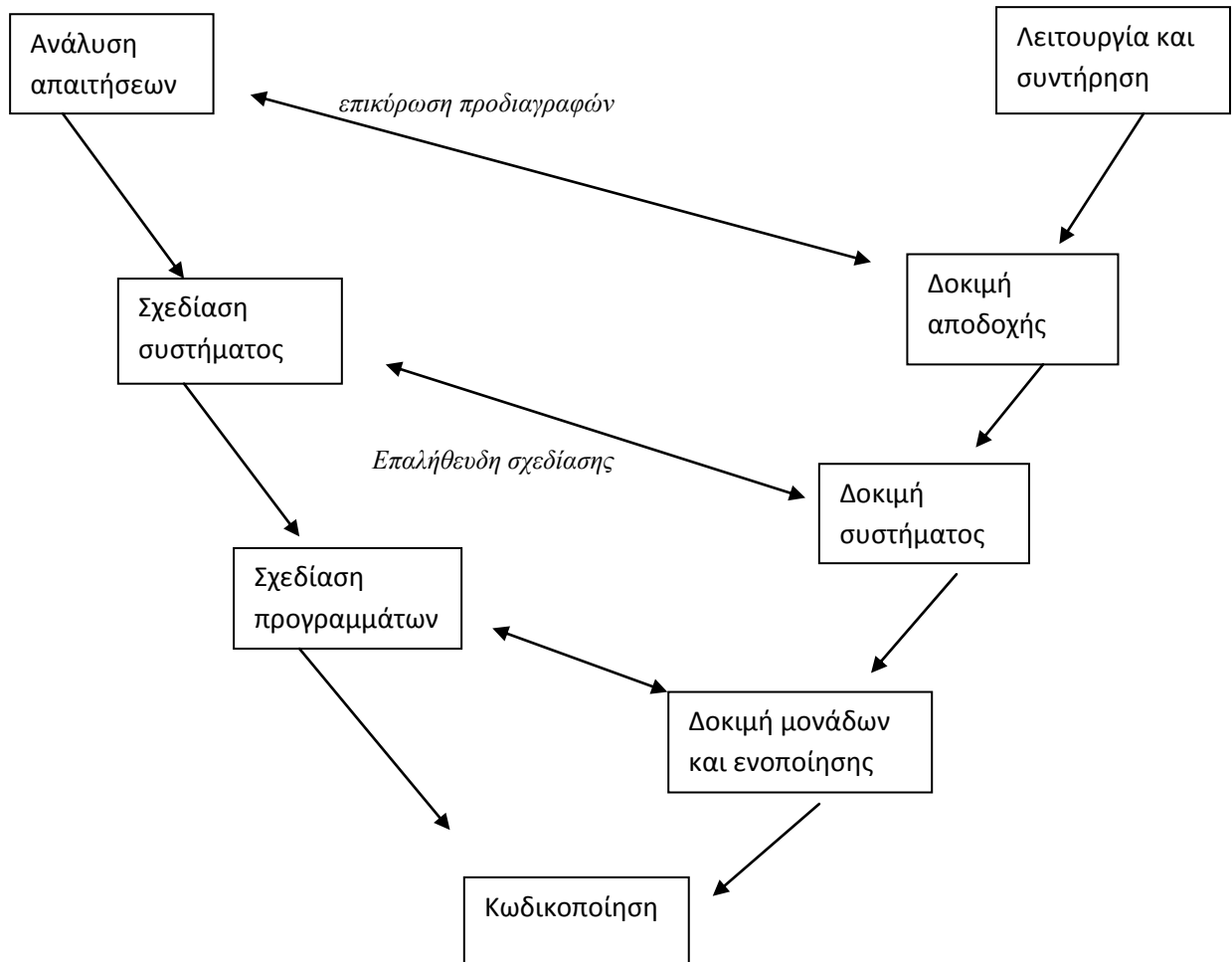
- Οι διάφορες φάσεις υπερκαλύπτονται μεταξύ τους.
- Η ανάπτυξη του, αν χρειαστεί οπισθοδρομεί στην προηγούμενη φάση, εκτός από την τελευταία που οπισθοδρομεί στην αρχή.

Το κύριο **πλεονέκτημα** του μοντέλου είναι ότι κάνει χρήση επαναχρησιμοποιήσιμων μονάδων και με αυτό τον τρόπο συντομεύεται τόσο η φάση της ανάπτυξης όσο και η φάση της συντήρησης.

1.6.6 Το μοντέλο 'V'

Παραλλαγή γραμμικού μοντέλου

- Αναδεικνύει συσχετίσεις δραστηριοτήτων δοκιμών με ανάλυση και σχεδίαση.
- Προτείνει τη χρήση δοκιμών δομικών μονάδων και ενοποίησης για την επαλήθευση των σχεδίων
- Έλεγχος αποδοχής από πελάτη -> επικύρωση
- Επανάληψη σε περίπτωση προβλημάτων
- Ορίζει επαναληπτικές και αναθεωρητικές εργασίες
- Εστιάζει στις δραστηριότητες και στην ορθότητα



Σχήμα 1.10

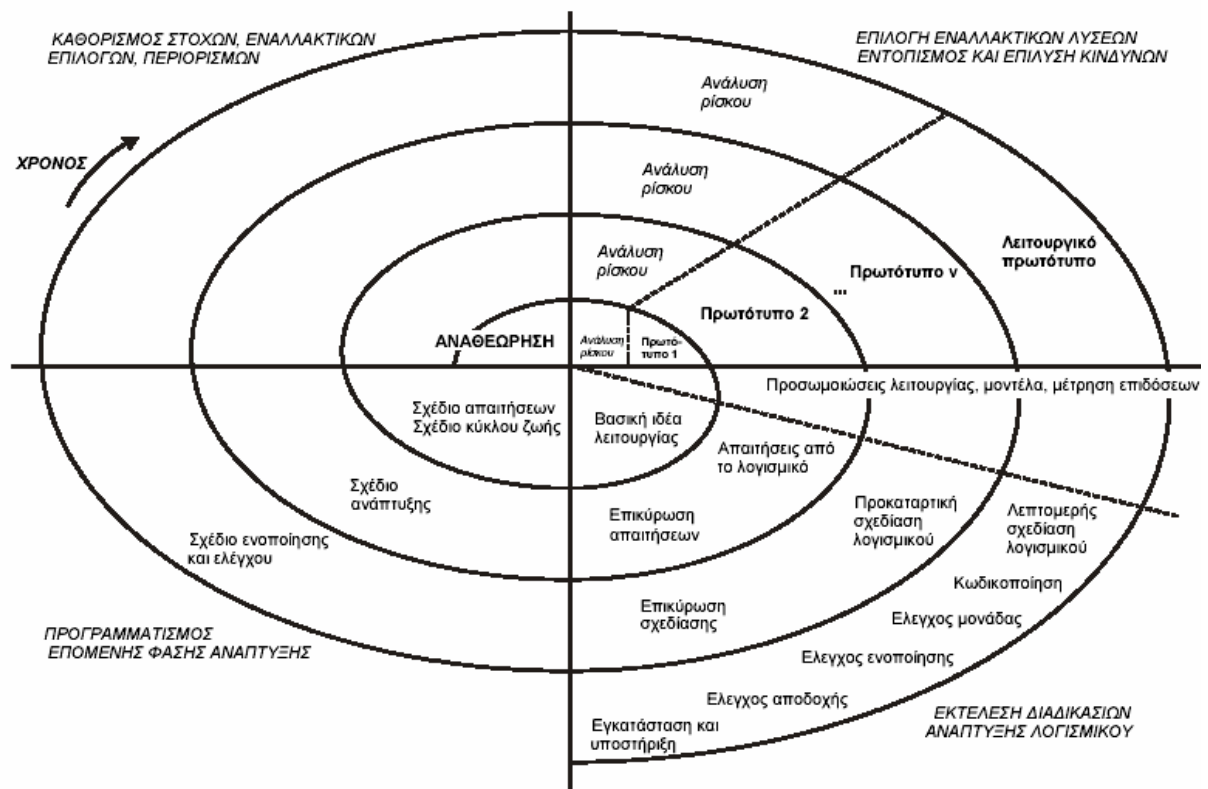
1.6.7 Σπειροειδές μοντέλο

Το σπειροειδές μοντέλο ή αλλιώς μοντέλο του Boehm είναι ένα δημοφιλές μοντέλο που ουσιαστικά είναι γενίκευση των μοντέλων της λειτουργικής επαύξησης και της πρωτοτυποποίησης και διαφέρει από όλα τα άλλα μοντέλα που βασίζονται στο ακολουθιακό μοντέλο του καταρράκτη. Το μοντέλο του Boehm έχει τη μορφή μιας σπείρας όπου κάθε γύρος στην σπείρα αναπαριστά και μια φάση έτσι όπως αυτή καθορίζεται κάθε φορά από την διοίκηση. Η ανάπτυξη ολόκληρου του συστήματος χωρίζεται σε πολλούς κύκλους, όπου σε καθέναν προστίθενται νέα λειτουργικά χαρακτηριστικά. Το περιεχόμενο κάθε κύκλου εξαρτάται από την πορεία του προηγούμενου. Κάθε κύκλος επιχειρεί να προσδιορίσει καλύτερα το τελικό προϊόν μειώνοντας ταυτόχρονα τους κινδύνους του έργου. Το σπειροειδές μοντέλο είναι ευέλικτο, αλλά η εφαρμογή του δεν είναι πάντα εύκολη γιατί

είναι αρκετά πολύπλοκο.

Το σπειροειδές μοντέλο δεν καθορίζει εκ των προτέρων: ποιες ακριβώς οι εργασίες ανάπτυξης λογισμικού που πρέπει να γίνουν, ούτε σε ποια έκταση του συστήματος θα εφαρμοστούν. Χρησιμοποιούνται διαφορετικές διαδικασίες ανάπτυξης για διαφορετικά τμήματα λογισμικού. Ο καθορισμός λεπτομερειών υλοποίησης να γίνεται συνεχώς κατά την ανάπτυξη με ευθύνη και τεκμηρίωση από τον κατασκευαστή. Η εφαρμογή του σπειροειδές μοντέλου δεν είναι πάντα εύκολη υπόθεση. Πραγματοποιείται εισαγωγή νέων εργασιών που αφορούν τεκμηρίωση σκοπιμότητας και τμηματικός προγραμματισμό ανάπτυξης. Το μοντέλο του Boehm απαιτεί μεγάλο χρηματικό κόστος, όμως μπορεί να αποσβεστεί από έγκαιρο εντοπισμό προβλημάτων και αποφυγή πιθανής αποτυχίας.

Το σπειροειδές μοντέλο δεν είναι κατάλληλο για την υλοποίηση μικρών έργων ανάπτυξης λογισμικού διότι: έχει μη αμελητέο κόστος για την εκτέλεση ενεργειών προγραμματισμού, εκτίμησης ρίσκου κλπ. Σε μικρές εφαρμογές λογισμικού αυτό το κόστος είναι δυσανάλογο. Το σπειροειδές μοντέλο δεν είναι η καλύτερη επιλογή από οικονομική άποψη. Υπάρχουν άλλα μοντέλα κύκλου ζωής καλύτερα.



Σχήμα 1.11

Στάδια

Η διαδοχή των σταδίων δεν γίνεται ούτε σταθερά ούτε γραμμικά, ενώ η εκτέλεσή τους μπορεί να γίνει είτε με τη φορά της σπείρας, είτε με την αντίθετη φορά, ανάλογα με το κίνδυνο που λαμβάνεται και το οποίο αποτελεί θεμελιώδη έννοια στο σπειροειδές μοντέλο.

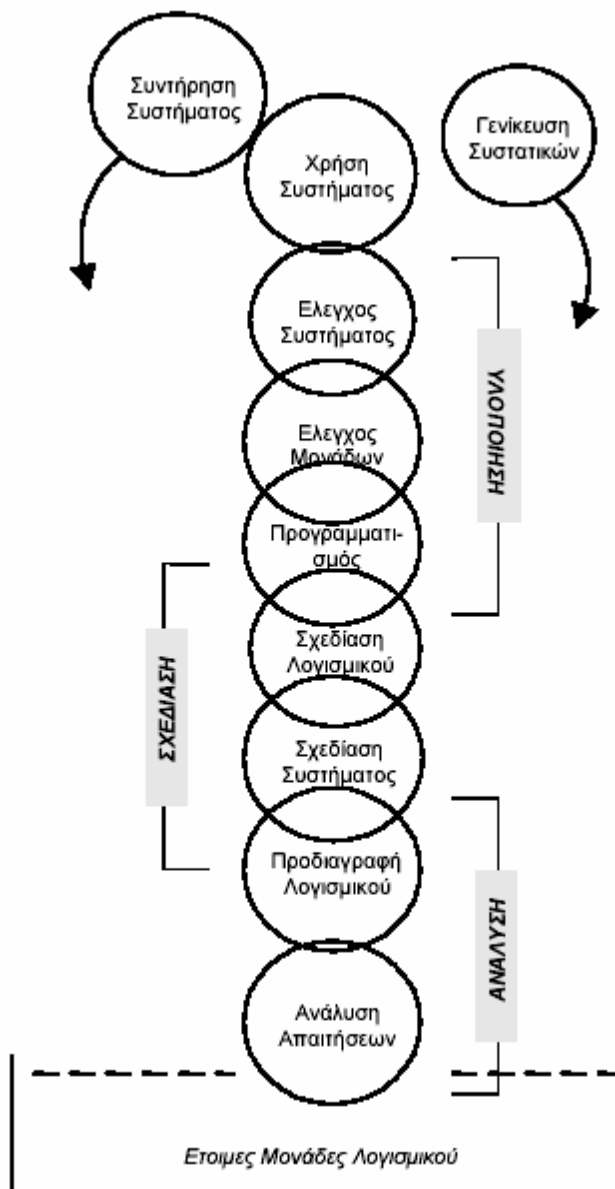
Σε κάθε γύρο που διανύεται υπάρχουν τα εξής στάδια:

- Καθορισμός στόχων, εναλλακτικών λύσεων και υπολογισμός περιορισμών.
- Ανάλυση και υπολογισμός του κινδύνου και προσπάθεια μείωσης του.
- Ανάπτυξη και επαλήθευση ενδιάμεσου προϊόντος – εφόσον η προηγούμενη φάση δεν έδειξε κάποιο σοβαρό κίνδυνο – πρόσθεση νέων λειτουργικών προδιαγραφών.
- Σχεδιασμός των επόμενων βημάτων.

Η βασική διαφορά του παραπάνω μοντέλου από τα υπόλοιπα είναι ότι σε αυτό υπολογίζεται πριν την έναρξη κάθε φάσης ο κίνδυνος, γεγονός που ουσιαστικά αποτελεί και το βασικό του πλεονέκτημα, αν και, από πρακτική άποψη, ο υπολογισμός και η ανάλυση του κινδύνου δεν είναι εύκολη υπόθεση.

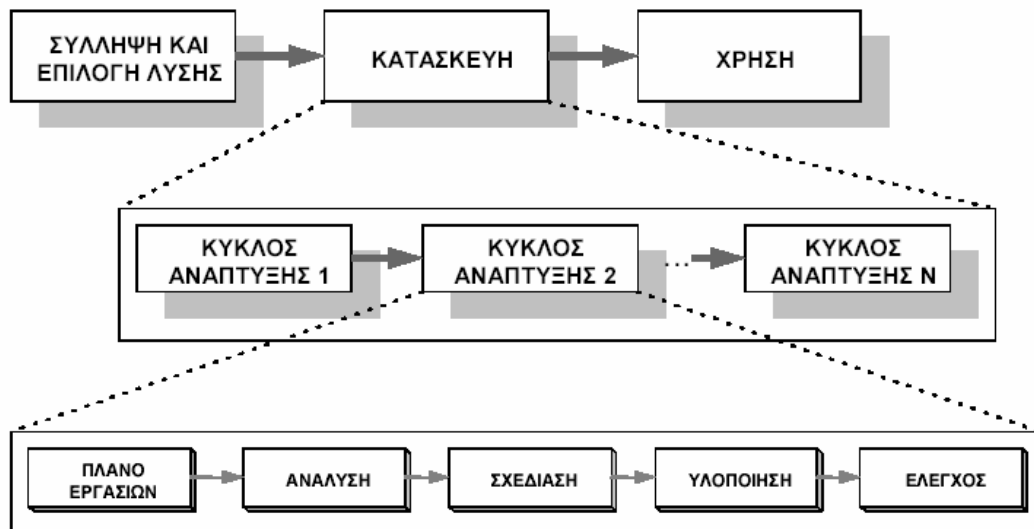
1.6.8 Μοντέλο πίδακα

- Αντικειμενοστραφής φιλοσοφία ανάπτυξη λογισμικού
- Επαναχρησιμοποίηση έτοιμων συστατικών
- Αντικειμενοστραφής τεχνολογία
- Οι έννοιες «ανάλυση – σχεδίαση - κωδικοποίηση» έρχονται στο αντικειμενοστραφές παράδειγμα πολύ πιο κοντά
- Το αποτέλεσμα κάθε διαδικασίας κατασκευής λογισμικού είναι όχι μόνο ένα σύστημα αλλά και **επαναχρησιμοποιήσιμες** μονάδες
- Επικαλύψεις φάσεων «ανάλυση – σχεδίαση - κωδικοποίηση»
- Ορισμένα συστατικά λογισμικού ενσωματώνονται σε δεξαμενή συστατικών και διατίθενται για χρήση στην ανάπτυξη νέων συστημάτων



Σχήμα 1.12

1.6.9 Γενικό μοντέλο κύκλου ζωής



Σχήμα 1.13

- Περιέχει γενικές κατευθύνσεις
 - Εξειδικεύονται στο εκάστοτε περιβάλλον ανάπτυξης, πρόβλημα κτλ.
 - Αφήνει σημαντικούς βαθμούς ελευθερίας στον κατασκευαστή
 - Περιλαμβάνει φάσεις σύλληψης, κατασκευής και λειτουργίας
 - Ανάλυση σε επιμέρους εργασίες
 - Ανάλυση φάσης κατασκευής σε «κύκλους ανάπτυξης» - καθένας προσθέτει νέα χαρακτηριστικά και λειτουργίες
-
- Ενσωματώνει χαρακτηριστικά πολλών μοντέλων
 - Ανάπτυξη σε κύκλους σύμφωνα με τα χαρακτηριστικά και τις δυνατότητες του κατασκευαστή
 - Όλα τα μοντέλα κύκλου ζωής που αναφέρθηκαν μπορούν να θεωρηθούν ειδικές εκδοχές του γενικού μοντέλου κύκλου ζωής
 - Κάθε κύκλος ανάπτυξης:
 - Επανάληψη
 - Βήμα επαύξησης
 - Παράλληλη εκτέλεση τμήματος έργου
 - Πλάνο εργασιών:

- Εκτίμηση ρίσκου συνέχισης ανάπτυξης

1.6.10 Συγκριτικός πίνακας

| Μοντέλο | Μέγεθος εφαρμογών | Μεταβολές στις απαιτήσεις | Προσαρμοστικότητα στον κατασκευαστή | Διάδοση |
|------------------------|-------------------|---------------------------|-------------------------------------|---------------------------------|
| Καταρράκτη | Μικρό έως μεσαίο | Ανεπιθύμητες | Καμία | Μεγάλη με τάση μείωσης |
| Πρωτοτυποποίησης | Μικρό έως μεσαίο | Δεκτές | Μικρή | Μικρή με τάση αύξησης |
| Λειτουργικής επαύξησης | Μεσαίο ως μεγάλο | Ανεπιθύμητες | Καμία | Μικρή με τάση μείωσης |
| Σπειροειδές | Μεσαίο ως μεγάλο | Δεκτές | Αρκετή | Μικρή με τάση μείωσης |
| Πίδακα | Οποιοδήποτε | Δεκτές | Αρκετή | Μικρή |
| Γενικό | Οποιοδήποτε | Δεκτές | Μεγάλη | Μικρή με ισχυρές τάσεις αύξησης |

1.6.11 Πλεονεκτήματα και μειονεκτήματα της χρήσης μοντέλων

Η μέθοδος δημιουργίας μοντέλων είναι πολύ χρήσιμη στις περιπτώσεις που υπάρχει κάποια αβεβαιότητα ως προς τις ανάγκες ή την ενδεδειγμένη λύση σχεδιασμού. Για παράδειγμα, μια μεγάλη χρηματιστηριακή εταιρεία ζητάει συγχωνευμένες πληροφορίες για να αναλύσει την απόδοση των υπευθύνων των λογαριασμών πελατών. Ποιο θα είναι όμως το μέτρο της απόδοσης; Οι πληροφορίες θα εξαχθούν μόνο από το αρχείο προσωπικού, ή θα πρέπει να συμπεριληφθούν και δεδομένα από τις χρεώσεις των πελατών; Ποια πράγματα θα συγκρίνονται στις αναφορές; Σε πρώτο στάδιο, οι χρήστες μπορεί να μην είναι σε θέση να ξεκαθαρίσουν πώς θα δουλεύει το σύστημα. Η δημιουργία πρωτοτύπων είναι ιδιαίτερα πολύτιμη για το σχεδιασμό σε ένα σύστημα πληροφοριών μιας διασύνδεσης τελικού χρήστη. (το τμήμα του συστήματος με το οποίο αλληλεπιδρά ο χρήστης, όπως μια οθόνη του δικτύου, οθόνες εισαγωγής δεδομένων, ή ιστοσελίδες). υποσημείωση Το πρωτότυπο

δίνει στους χρήστες τη δυνατότητα να ενεργήσουν αμέσως στα τμήματα του συστήματος τα οποία θα χειρίζονται.

Η μέθοδος δημιουργίας πρωτοτύπων υποκινεί την εντατική συμμετοχή των τελικών χρηστών σε όλη τη διάρκεια του κύκλου ανάπτυξης του λογισμικού, οπότε είναι πιο πιθανή η κατασκευή ενός λογισμικού που θα εκπληρώνει τις απαιτήσεις τους. Ωστόσο, η γρήγορη δημιουργία πρωτοτύπων μπορεί να μειώσει τη σημασία άλλων ουσιαστικών βημάτων στην ανάπτυξη συστημάτων. Όταν το πρωτότυπο τελειώσει και

λειτουργεί ικανοποιητικά, τα στελέχη μπορεί να μη θεωρήσουν αναγκαίο τον επαναπρογραμματισμό, την επανασχεδίαση, ή την πλήρη τεκμηρίωση και τις δοκιμές. Μερικά από αυτά τα βιαστικά συστήματα μπορεί να μην είναι σε θέση να εξυπηρετήσουν μεγάλες ποσότητες δεδομένων, ή μεγάλο αριθμό χρηστών σε περιβάλλον παραγωγής. Η επιτυχημένη δημιουργία πρωτοτύπων προϋποθέτει την ύπαρξη διαδικασιών και μηχανισμών για τον καθορισμό των προσδοκιών, την εκχώρηση πόρων, τον εντοπισμό προβλημάτων, και την αξιολόγηση της προόδου.

1.7 Διοίκηση Έργων Εκπαιδευτικού Λογισμικού

Λόγω των πολλαπλών απαιτήσεων που υπάρχουν στην ανάπτυξη εκπαιδευτικού λογισμικού καθίσταται ακόμα πιο απαραίτητη η διοίκηση του έργου η οποία εκτός του γενικότερου συντονισμού και οργάνωσης των (ανθρώπινων) πόρων που χρησιμοποιούνται σε ένα έργο ανάπτυξης εκπαιδευτικού λογισμικού, θα πρέπει να λαμβάνει μέτρα ώστε το έργο που θα αναληφθεί θα ολοκληρωθεί εντός της προθεσμίας, εντός του προβλεφθέντος προϋπολογισμού και με την ορισθείσα ποιότητα με δεδομένο ότι τόσο η χρονική προθεσμία όσο και ο προϋπολογισμός θα πρέπει να υπολογιστούν από αυτή.

Η διοίκηση έργων εκπαιδευτικού λογισμικού εκτελείται παράλληλα με τις εργασίες ανάπτυξης επιτελώντας έργο οργανωτικό και διαχειριστικό χωρίς να ασχολείται με αυτή καθεαυτή την ανάπτυξη.

Η διοίκηση ενός έργου εκπαιδευτικού λογισμικού υπολογίζει αλλά και οργανώνει σωστά τους (ανθρώπινους) πόρους που απαιτούνται και βρίσκει τρόπους επικοινωνίας μεταξύ των διαφόρων ομάδων που μετέχουν στην ανάπτυξη και

επιπλέον πρέπει να έχει την ικανότητα να αναγνωρίσει έγκαιρα και να κάνει σωστές επιλογές όταν βρίσκεται μπροστά σε αβεβαιότητα.

Θα πρέπει ακόμα να τονιστεί το γεγονός ότι στην ανάπτυξη έργων εκπαιδευτικού λογισμικού είναι απαραίτητη η συνεργασία επιστημόνων (παιδαγωγοί της θεματικής περιοχής που πραγματεύεται το λογισμικό, ειδικοί της διδακτικής, γνωστικοί ψυχολόγοι, ειδικοί γραφίστες και ειδικοί στον σχεδιασμό του περιβάλλοντος διεπαφής και τέλος προγραμματιστές) οι οποίοι δεν έχουν, στο μεγαλύτερο ποσοστό τους, γνώσεις λογισμικού, δεν έχουν μεγάλη επιστημονική συνοχή και συνήθως δεν βρίσκονται ούτε στον ίδιο χώρο. Πρόκειται για δεδομένα που δυσκολεύουν ακόμα περισσότερο τον συντονισμό των ανθρώπινων πόρων των έργων αυτών.

Έτσι, οι άνθρωποι που στελεχώνουν την διοίκηση ενός έργου θα πρέπει να διαθέτουν εκτός των γενικών προσόντων που απαιτούνται όπως γνώσεις διοίκησης, γνώσεις λογισμικού, γνώσεις θεωριών αποφάσεων και γνώσεις επικοινωνιακών ζητημάτων και κάποιου είδους εξοικείωση με παιδαγωγικά μοντέλα και εκπαιδευτικά συστήματα και ζητήματα.

1.7.1 Γιατί είναι απαραίτητη η διοίκηση έργων:

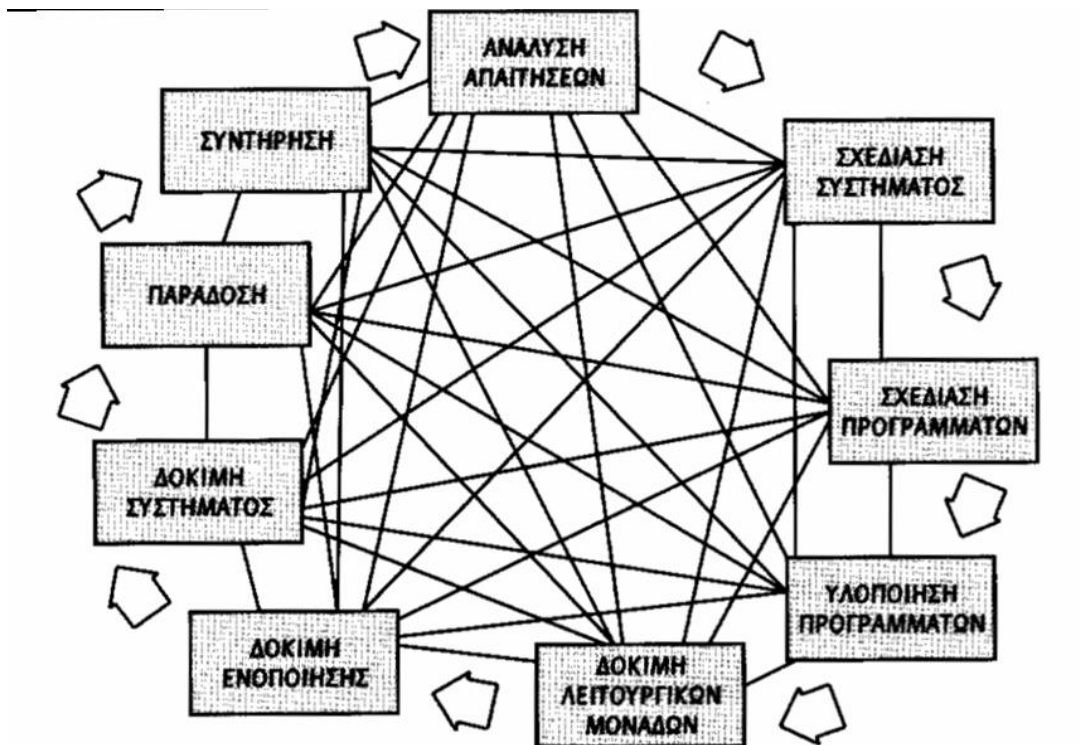
Οι άνθρωποι που ασκούν την διοίκηση ενός έργου λογισμικού δεν απαιτείται να έχουν γνώση των διεργασιών που θα πραγματοποιηθούν ώστε να επιτύχουν στον ρόλο τους.

Οι άνθρωποι που ασκούν την διοίκηση ενός έργου λογισμικού απαιτείται να έχουν γνώση των εργασιών που θα γίνουν, της χρονικής διάρκειας κάθε μιας από αυτές, της χρονικής αλληλουχίας αυτών αλλά και των πιθανών αλληλοεξαρτήσεών τους, καθώς και των απαιτούμενων πόρων για την εκτέλεση κάθε μιας από αυτές.

Έτσι ο ρόλος της διοίκησης είναι να εκτιμήσει σωστά τον χρόνο και τους πόρους που απαιτούνται (πρόβλεψη), και μετά να χρησιμοποιήσει σωστά αυτούς τους πόρους (οργάνωση) ώστε να ολοκληρωθεί το έργο (επιτήρηση) επιτυχώς, δηλαδή εντός της προθεσμίας, εντός του προβλεφθέντος προϋπολογισμού και με την ορισθείσα ποιότητα. Βέβαια στην εκτίμηση του χρόνου και των πόρων θα πρέπει να λαμβάνεται υπόψη και ο κίνδυνος που λαμβάνεται αλλά επίσης θα πρέπει να υπάρχει και η ικανότητα να ξεπεραστούν τυχόν απρόβλεπτες καταστάσεις (λήψη

αποφάσεων) οι οποίες φυσικά όσο πιο γρήγορα αναγνωριστούν τόσο πιο εύκολα μπορούν να ληφθούν τα απαραίτητα μέτρα ώστε να αντιμετωπιστούν επιτυχώς.

1.7.2 Η ανάπτυξη λογισμικού στην πραγματικότητα



Σχήμα 1.13

1.8 Συμπεράσματα

1.8.1 Ποιο είναι το καταλληλότερο μοντέλο:

Από όσα αναφέρθηκαν, η επιλογή του κατάλληλου μοντέλου ανάπτυξης εκπαιδευτικού λογισμικού είναι λογικό να μην περιλαμβάνει τα μοντέλα αυτόματου προγραμματισμού. Και αυτό γιατί η ανάπτυξη του εκπαιδευτικού λογισμικού είναι μια δημιουργική διαδικασία που εμπλέκει πολλών περιοχών ειδικούς και όχι μια αυτόματη διαδικασία. Πολλά, επίσης, μοντέλα όπως το λειτουργικό χρησιμοποιούνται περισσότερο για την ανάπτυξη ειδικών περιπτώσεων λογισμικού. Το μοντέλο του καταρράκτη, και όλα τα μοντέλα αυτής της μορφής, αν και είναι δημοφιλής επιλογή, είναι επίσης απορριπτέα γιατί όπως έχει αναφερθεί, ξέρουμε εάν έχουμε κατασκευάσει αυτό που θέλαμε, μόνο όταν η διαδικασία ανάπτυξης έχει ολοκληρωθεί. Επίσης, δεδομένου ότι το προϊόν κάθε φάσης θεωρείται τελικό είναι σχεδόν αδύνατο να παρέμβουμε σε αυτό όταν βρισκόμαστε σε αρκετά μεταγενέστερο στάδιο ανάπτυξης στο οποίο θα παρουσιαστεί κάποιο πρόβλημα ή λάθος. Και αυτό γιατί αυξάνεται γεωμετρικά το κόστος της διόρθωσης.

Το μοντέλο επαναχρησιμοποίησης λογισμικού δεν θα μπορούσε να οδηγήσει σε ολοκληρωμένη ανάπτυξη εκπαιδευτικού λογισμικού παρόλα αυτά θα μπορούσε να βοηθήσει ιδιαίτερα ως τμήμα ενός άλλου μοντέλου.

Το μοντέλο πρωτοτυποποίησης θα μπορούσε να δώσει έναν τρόπο ανάπτυξης δεδομένου ότι ειδικά στην εξελικτική του μορφή μπορεί να δώσει πληροφορίες που βοηθούν στην ανάπτυξη πολύ πιο γρήγορα από το μοντέλο του καταρράκτη, τόσο ως προς την εφικτότητα κατασκευής του τελικού συστήματος όσο και ως προς την ακρίβεια των προδιαγραφών των απαιτήσεων από το λογισμικό.

Επίσης, το μοντέλο της throw-away πρωτοτυποποίησης είναι μια σοβαρή υποψηφιότητα για την ανάπτυξη εκπαιδευτικού λογισμικού διότι αφενός δίνει πληροφορίες για την εφικτότητα του τελικού συστήματος αλλά και γιατί αποσαφηνίζει τις απαιτήσεις του συστήματος. Αυτό συμβαίνει διότι το τελικό σύστημα όπως προκύπτει με την χρήση αυτού του μοντέλου είναι πιο συντηρήσιμο, τελικά για ολόκληρη την διάρκεια ζωής του λογισμικού προκύπτει σημαντική μείωση του συνολικού κόστους αν και όπως έχει ήδη αναφερθεί επειδή το αρχικό

πρωτότυπο θα 'πεταχτεί', αυξάνεται ως ένα βαθμό το κόστος ανάπτυξης από το γεγονός αυτό.

Το σπειροειδές μοντέλο ανάπτυξης διαφέρει από όλα τα προηγούμενα κυρίως στο ότι σε κάθε φάση ανάπτυξης υπάρχει μια άμεση εκτίμηση του κινδύνου. Καταρχήν σε αυτό καθορίζονται οι στόχοι, οι εναλλακτικές λύσεις και οι περιορισμοί τους και στη συνέχεια αξιολογούνται αυτές οι λύσεις και καθορίζονται στρατηγικές επίλυσης του κινδύνου εάν αυτό είναι μεγάλο. Με αυτό τον τρόπο είναι δυνατό να αναπτυχθεί το λογισμικό χωρίς να υπάρχει ο κίνδυνος της μη ολοκλήρωσης μιας φάσης ανάπτυξης του και συγχρόνως μετά από κάθε φάση έχουμε ένα ενδιάμεσο πρωτότυπο που μπορεί σταδιακά να αξιολογείται.

Λαμβάνοντας υπόψη ότι η σύγχρονη τάση στην ανάπτυξη εκπαιδευτικού λογισμικού ουσιαστικά χρησιμοποιεί κάποιες γενικές κατευθύνσεις από τα υπάρχοντα μοντέλα, αλλά αφήνει αρκετούς βαθμούς ελευθερίας στον κατασκευαστή, μπορούμε να συμπεράνουμε ότι το σπειροειδές μοντέλο μπορεί να αποτελέσει βάση για την ανάπτυξη εκπαιδευτικού λογισμικού με ορισμένες παραλλαγές οι οποίες εξειδικεύονται κάθε φορά ανάλογα με το περιβάλλον ανάπτυξης, το συγκεκριμένο πρόβλημα κλπ.

2.Αξιολόγηση

Λέξεις Κλειδιά: Αξιολόγηση και Ποιότητα Πληροφοριακών Συστημάτων, Ικανοποίηση Χρηστών, Μέτρηση Αποτελεσματικότητας Πληροφοριακών Συστημάτων

2.1 Τί είναι αξιολόγηση

Όταν αξιολογούμε ένα πληροφοριακό σύστημα, ασχολούμαστε κατά κύριο λόγο με την οργάνωση και τη διοίκηση ενός οργανισμού, καθώς και με τους χρήστες του. Χρησιμοποιώντας τον όρο «χρήστες» εννοούμε το προσωπικό, τους ερευνητές, παραγωγούς, προμηθευτές και διαθέτες του συστήματος, δηλαδή, όλους τους εσωτερικούς και εξωτερικούς χρήστες του οργανισμού.

Οι παράγοντες, που καλούμαστε να χρησιμοποιούμε για να αξιολογήσουμε ένα πληροφοριακό σύστημα, είναι: τα συστατικά στοιχεία του συστήματος, οι πληροφοριακές διεργασίες, οι υπηρεσίες και τα προϊόντα, οι πληροφοριακές λειτουργίες, ολόκληρο το πληροφοριακό σύστημα, καθώς και το περιβάλλον του.

Οι παράγοντες, οι οποίοι συναινούν στην επιτυχία και την αποτελεσματικότητα του πληροφοριακού συστήματος ενός οργανισμού, αποτελούν πάντοτε μία ουσιαστική και σημαντική πτυχή για τη δομή, την εύρωστη λειτουργία και την ανοδική πορεία των παραγόμενων υπηρεσιών και προϊόντων του οργανισμού στους τελικούς χρήστες του, τους εσωτερικούς, δηλαδή το υπαλληλικό προσωπικό του οργανισμού και εξωτερικούς δηλαδή το κοινό. Στις μέρες μας ολοένα και περισσότερο προκύπτει έντονα η ανάγκη για τη δημιουργία κατάλληλων παραμέτρων, οι οποίες θα έχουν στόχο την εκ βάθους αξιολόγηση και μέτρηση της επιτυχούς λειτουργίας ενός ιδρυματικού πληροφοριακού συστήματος

2.1.2 Σκοπός της αξιολόγησης

Η χρησιμότητα της αξιολόγησης είναι ιδιαίτερος σημαντική, γιατί με την εφαρμογή της το σύστημα παρέχει καλύτερα προϊόντα και υπηρεσίες, δικαιολογείται η υπάρχουσα κατάστασή του και γίνεται κατανοητή η λειτουργία του. Επιπροσθέτως, επιβεβαιώνεται το γεγονός ότι οι πόροι του συστήματος χρησιμοποιούνται αποδοτικά και αποτελεσματικά, ενώ συγχρόνως πείθονται, τόσο οι αρμόδιοι φορείς, που το εποπτεύουν, όσο και οι χρήστες, που το χρησιμοποιούν, ότι τα οφέλη που προσφέρονται από το σύστημα είναι αυτά που πρέπει να αποδίδονται στον ιδρυματικό οργανισμό. Ολόκληρη η διαδικασία της αξιολόγησης συμβάλλει στο να λαμβάνονται σωστές αποφάσεις, να διασφαλίζεται η ποιότητα του συστήματος και να εκτιμάται η έκταση στην οποία μπορούν να λυθούν τα προβλήματα, που τυχόν το σύστημα αντιμετωπίζει. Επίσης, προσδιορίζονται οι ανάγκες των διαφορετικών ομάδων χρηστών, σχεδιάζονται οι δημόσιες σχέσεις και οι υπηρεσίες διάχυσης των πληροφοριών του οργανισμού και γίνεται αντιληπτή η συμμετοχή των χρηστών στην γενικότερη οργάνωση και λειτουργία του πληροφοριακού συστήματος.

Η αξιολόγηση αποσκοπεί στην τελειοποίηση της σχεδίασης της διεπαφής χρήστη. Αρχικά επικυρώνονται ή απορρίπτονται οι αρχικές αποφάσεις και προτείνονται καινούργιες προτάσεις για τα προβλήματα που προκύπτουν. Επιπλέον, η αξιολόγηση διαβεβαιώνει ότι το σύστημα συμπεριφέρεται όπως ακριβώς αναμενόταν και δίνει ένα τέλος στη διαδικασία της επαναληπτικής σχεδίασης. Όπως προκύπτει η αξιολόγηση είναι πολύ σημαντική για την σωστή λειτουργία ενός συστήματος.

2.1.3 Γιατί είναι απαραίτητη:

Η αξιολόγηση είναι βασικό κομμάτι της ανθρωποκεντρικής σχεδίασης. Δεν πρέπει να θεωρείται απλή φάση της σχεδίασης αλλά πρέπει να συμμετέχει ενεργά σε όλες τις φάσεις της σχεδίασης και υλοποίησης και κυρίως στην αξιολόγηση της ευχρηστίας. Η βασική διαφορά με την παραδοσιακή αξιολόγηση είναι ότι προσδίδει μεγάλη σημασία στους χρήστες και την διεπαφή χρήστη την εργονομία και την ευχρηστία.

Με την εφαρμογή της αξιολόγησης δημιουργούνται πιο εύχρηστα συστήματα και οι αλλαγές που γίνονται στα πρώιμα στάδια της παραγωγής γίνονται πιο εύκολα και εξοικονομούνται χρήματα. Αντιθέτως αν έχει προχωρήσει η διαδικασία παραγωγής και οι σχεδιαστικές δεσμεύσεις έχουν παγιωθεί τότε πολύ δύσκολα γίνονται οι απαραίτητες διορθώσεις. Όταν η δημιουργία του συστήματος έχει σχεδόν τελειώσει η αξιολόγηση επιβεβαιώνει ότι οι στόχοι έχουν επιτευχθεί και ότι το σύστημα ανταποκρίνεται στις απαιτήσεις που έχουν διατυπωθεί.

Μία ακόμη χρήση της αξιολόγησης είναι η σύγκριση μεταξύ των προϊόντων για να αποφασίσει ο αγοραστής ποιο θα αγοράσει ή για να τεθούν οι στόχοι ευχρηστίας όταν πρόκειται να υλοποιηθεί ένα ανταγωνιστικό προϊόν.

2.1.4 Παράμετροι αξιολόγησης

Οι παράμετροι με τις οποίες εξετάζουμε την επιτυχία και την αποτελεσματικότητα ενός πληροφοριακού συστήματος μπορούν να συνεκτιμηθούν σε τρία επίπεδα, τα οποία είναι ακολούθως:

1. **Το Επίπεδο οργανισμού:** στρατηγικοί στόχοι, λειτουργικό κόστος, διαθεσιμότητα συστήματος, χρόνοι απόκρισης, έσοδα και κέρδη από νέα προϊόντα και υπηρεσίες.

2. **Το Επίπεδο διεργασιών και λειτουργιών:** μείωση κόστους σε ειδικές λειτουργίες, μείωση χρόνου σε επιμέρους διεργασίες, ολοκλήρωση διεργασιών, χαμηλότερο κόστος και χρόνοι σε σχέση με την γενική πρακτική, και

3. **Το Επίπεδο ατόμου:** ικανοποίηση χρηστών, χρησιμότητα και λειτουργικότητα του συστήματος.

Ανάλογα με το μέγεθος του έργου του εκπαιδευτικού λογισμικού που αναλαμβάνεται να διεκπεραιωθεί, καθώς και από το μέγεθος της ομάδας ανάπτυξης, τα μοντέλα στην πραγματικότητα εφαρμόζονται με μεγαλύτερη

ευελιξία και με μεγαλύτερη παραμετρικοποίηση. Ο λόγος είναι προφανής, αφού επιδιώκεται να γίνει η μεγαλύτερη δυνατή προσαρμογή του εφαρμοζόμενου μοντέλου στην συγκεκριμένη υπό ανάπτυξη εφαρμογή ώστε να υπάρξει η μεγαλύτερη δυνατή αποδοτικότητα.

Έτσι είναι φανερό ότι δεν είναι και τόσο σημαντική η αυστηρή τήρηση της δομής των μοντέλων και η χρονική αλληλουχία του συνόλου των ενδιάμεσων προϊόντων που παράγονται. Περισσότερο σημαντικό είναι να υπάρχει δυνατότητα προσαρμογής των μοντέλων στις εκάστοτε συνθήκες ανάπτυξης και στα χρησιμοποιούμενα εργαλεία. Αυτό γίνεται περισσότερο κατανοητό αν λάβει κάποιος υπόψη του την συνθετότητα και ποικιλότητα των εργαλείων ανάπτυξης όπως οι σύγχρονες γλώσσες προγραμματισμού, οι ψηφίδες λογισμικού, τα περιβάλλοντα συγγραφής κώδικα κ.α. .

2.1.5 Τι πρέπει να προσέχουμε κατά τη διάρκεια της αξιολόγησης:

- Σχεδίαση βάση της απλής λογικής. Πράγματα που θεωρούνται αυτονόητα δυστυχώς δεν είναι.
- Υποθέτουμε ότι η προσωπική άποψη είναι αντιπροσωπευτική. Οι χρήστες ενός συστήματος έχουν διαφορετικές ικανότητες και γνώσεις από τους σχεδιαστές-προγραμματιστές.
- Αποδοχή μίας συνήθειας ή παράδοσης στην σχεδίαση.
- Καθυστερήση της αξιολόγησης
- Πειράματα που δεν αναλύονται για να κατανοηθούν τα βαθύτερα νοήματα.

2.1.6 Το πρόβλημα της αξιολόγησης

Η διαδικασία της αξιολόγησης εκπαιδευτικού λογισμικού μας επιτρέπει να αποφανθούμε αν ένα συγκεκριμένο λογισμικό αξίζει να επιλεγεί, να τροποποιηθεί ή ακόμα και να απορριφθεί. Με το σκεπτικό αυτό, η αξιολόγηση αποτελεί αναμφίβολα ένα από τα σημαντικότερα ζητήματα που αντιμετωπίζει η εκπαιδευτική

και ερευνητική κοινότητα καθώς και οι φορείς/ομάδες ανάπτυξης εκπαιδευτικού λογισμικού. Η σπουδαιότητα της αξιολόγησης απορρέει τόσο από το γεγονός ότι η αξιολόγηση αυτή καθεαυτή είναι μια διαδικασία με την οποία αποτιμούμε την αποτελεσματικότητα και αποδοτικότητα ενός εκπαιδευτικού λογισμικού, όσο και από την κατάλληλη αξιοποίηση των πορισμάτων της αξιολόγησης.

Είναι γεγονός, ότι παρά τις σημαντικές εξελίξεις στο χώρο της ανάπτυξης εκπαιδευτικού λογισμικού, υπάρχει έλλειψη ολοκληρωμένων και δυναμικών υποδειγμάτων ή σχημάτων αξιολόγησης. Η ανάπτυξη, όμως, ενός γενικού υποδείγματος αξιολόγησης εκπαιδευτικού λογισμικού είναι δύσκολη, κυρίως λόγω των πολλαπλών παραγόντων, προσεγγίσεων και ιδιαιτεροτήτων κάθε τύπου εκπαιδευτικού λογισμικού καθώς και της φύσης αυτής καθεαυτής της αξιολόγησης. Παρόλα αυτά, θεωρείται σημαντική η προσπάθεια ανάπτυξης υποδειγμάτων αξιολόγησης εκπαιδευτικού λογισμικού, ιδιαίτερα όταν αυτά: α) αξιοποιούν τα θετικά στοιχεία των ποικίλων προσεγγίσεων και τύπων αξιολόγησης και β) συνδέονται με τη θεωρία και την εκπαιδευτική πράξη. Συχνά, όμως, υποβαθμίζεται η σύνδεση αυτή με το αιτιολογικό ότι η αξιολόγηση εκπαιδευτικού λογισμικού έχει μόνο πρακτική, σημασία. Η τάση να αντιμετωπίζεται η εκπαιδευτική αξιολόγηση γενικότερα σαν κάτι το διαφορετικό από την εκπαιδευτική θεωρία και πράξη δεν κρύβει, βέβαια, τίποτε άλλο από την αντίληψη να χαράζονται διαχωριστικές γραμμές ανάμεσα στα θεωρητικά ζητήματα, σχετιζόμενα με την ουσία των εκπαιδευτικών πραγμάτων, και στα πρακτικά ζητήματα, που αφορούν αυτό που πρέπει να γίνει. Όπως έχει σημειωθεί, η συνήθεια να σκέφτεται και να δρα κανείς σύμφωνα με τους όρους που υπαγορεύει αυτή η διχοτομία δεν αναπτύχθηκε σε κοινωνικό και ιστορικό κενό. Για το λόγο αυτό θα ήταν λάθος να πιστεύουμε ότι μπορούμε να αποκτήσουμε μια σωστή αντίληψη της αξιολόγησης εκπαιδευτικού λογισμικού, αν θεωρήσουμε τις θεμελιώδεις επιδράσεις από το χώρο της φιλοσοφίας της εκπαίδευσης ως ζήτημα δευτερεύουσας σημασίας. Θεωρούμε ότι ο τρόπος με τον οποίο συγκεντρώνουμε, αναλύουμε και ερμηνεύουμε τα δεδομένα μιας αξιολόγησης, είτε είναι ποιοτικά ή/και ποσοτικά, εξαρτάται από το αντίστοιχο επιστημονικό υπόδειγμα μέσα στο οποίο λειτουργούμε. Το υπόδειγμα στηρίζεται σε ένα πλαίσιο αναφοράς το οποίο προσδιορίζει τις οντολογικές, επιστημολογικές, μεθοδολογικές και αξιολογικές μας αντιλήψεις, θέσεις και συμβάσεις. Η αξιολόγηση εκπαιδευτικού λογισμικού πρέπει να γίνεται σε σχέση με κάποιο υπάρχον ή στοχευόμενο εκπαιδευτικό περιβάλλον μέσα στο οποίο θα λειτουργεί. Η

αξιολόγηση εκπαιδευτικού λογισμικού, όπως και κάθε μορφή εκπαιδευτικής αξιολόγησης, δεν θα πρέπει να αποτελεί μια διαδικασία αξιολόγησης του μέσου με βάση το μέσο ή του μέσου για το μέσο, όπως συχνά παρατηρείται. Αντίθετα, θα πρέπει να είναι μια διαδικασία η οποία εξαρτάται κυρίως από το τί θεωρεί ο αξιολογητής ως προσδοκώμενα μαθησιακά αποτελέσματα, χωρίς να αποκλείεται ότι αυτά μπορούν να συμπίπτουν με εκείνα πάνω στα οποία πιθανόν στηρίχθηκε ο σχεδιασμός και η ανάπτυξη του υπό αξιολόγηση εκπαιδευτικού λογισμικού.

2.1.7 Τί αξιολογούμε σε ένα λογισμικό

Όταν αξιολογούμε ένα πληροφοριακό σύστημα είναι σημαντικό να αποφασίσουμε «τι»

αξιολογούμε. Μπορούμε να θεωρήσουμε τουλάχιστον δύο διαφορετικές καταστάσεις που μπορούν να αξιολογηθούν:

α) Το πληροφοριακό σύστημα όπως είναι: που σημαίνει ότι αξιολογούμε το πληροφοριακό σύστημα χωρίς καμία συμμετοχή από τους χρήστες. Τα αποτελέσματα της αξιολόγησης βασίζονται στην εκτίμηση του αξιολογητή για το πώς το πληροφοριακό σύστημα υποστηρίζει την εκάστοτε οργάνωση. Αυτή η στρατηγική είναι απαλλαγμένη από τις εκτιμήσεις των χρηστών για το πώς το πληροφοριακό σύστημα ωφελεί την εργασία τους. Το αντικείμενο της αξιολόγησης είναι το πληροφοριακό σύστημα αυτό καθ' εαυτό. Δεν υπάρχει καμία μελέτη πραγματικής κατάστασης χρήσης του συστήματος. Ο αξιολογητής εξερευνά τι είναι δυνατό να κάνει με το σύστημα.

β) Το πληροφοριακό σύστημα σε χρήση: που σημαίνει ότι μελετάμε μία κατάσταση χρήσης όπου ένας χρήστης αλληλεπιδρά με το σύστημα. Αυτή η περίπτωση είναι πιο πολύπλοκη από την άλλη γιατί συμπεριλαμβάνει και έναν χρήστη και δίνει μια πιο ολοκληρωμένη εικόνα. Τα δεδομένα για αυτή την αξιολόγηση μπορούν να

προκύψουν από συνεντεύξεις των χρηστών και τις εκτιμήσεις τους για την ποιότητα του συστήματος, από παρατηρήσεις της αλληλεπίδρασης των χρηστών με το πληροφοριακό σύστημα και από το ίδιο το πληροφοριακό σύστημα.

2.1.8 «Πώς» αξιολογούμε ένα λογισμικό;

Μπορούμε να διακρίνουμε τρεις διαφορετικές στρατηγικές αξιολόγησης λογισμικού. Η στοχευμένη αξιολόγηση, η μη στοχευμένη αξιολόγηση και η αξιολόγηση βάση κριτηρίων.

α) **Στοχευμένη αξιολόγηση** (*Goal-based evaluation*) όπου ρητοί στόχοι από το οργανωτικό πλαίσιο οδηγούν την αξιολόγηση. Η εστίαση γίνεται πάνω στα επιθυμητά αποτελέσματα του συστήματος:

τους στόχους. Οι στόχοι που χρησιμοποιούνται για την αξιολόγηση προέρχονται από ένα

συγκεκριμένο οργανωτικό πλαίσιο. Αυτό σημαίνει ότι ισχύουν περιστασιακά. Η βασική

στρατηγική αυτής της προσέγγισης είναι να μετρήσουμε εάν συγκεκριμένοι στόχοι εκπληρώνονται ή όχι, σε πιο βαθμό και με πιο τρόπο. Η προσέγγιση είναι συμπερασματική. Τι είναι αυτό που μετριέται εξαρτάται από τον χαρακτήρα των στόχων

και γι' αυτό μπορούν να χρησιμοποιηθούν και ποιοτικές και ποσοτικές μέθοδοι.

Στη στοχευμένη αξιολόγηση εξετάζουμε την κατάσταση με κύριο γνώμονα τους ρητούς στόχους από το αρχικό οργανωτικό διάγραμμα, ακολουθούμε την προκαθορισμένη σειρά και στη συνέχεια οδηγούμαστε στην αξιολόγηση.

β) **Μη στοχευμένη αξιολόγηση** (*Goal-free evaluation*) όπου κανένας ρητός στόχος δεν χρησιμοποιείται, είναι μία επαγωγική και κατά περίπτωση οδηγημένη στρατηγική. Είναι μια πιο ερμηνευτική προσέγγιση που βλέπει το πληροφοριακό σύστημα σαν κοινωνικό σύστημα που σε αυτό έχει εισχωρήσει η τεχνολογία. Η μη στοχευμένη αξιολόγηση γίνεται συλλέγοντας στοιχεία όσον αφορά μια ευρεία

περιοχή πραγματικών αποτελεσμάτων και αξιολογώντας τη σημασία αυτών των αποτελεσμάτων. Μόνο τα αποτελέσματα του συστήματος μετρούνται. Η βασική στρατηγική αυτής της προσέγγισης είναι η επαγωγική αξιολόγηση. Η προσέγγιση έχει σκοπό να ανακαλύψει ποιότητες του αντικειμένου της μελέτης. Κάποιος μπορεί να πει ότι ο αξιολογητής ψάχνει για πιθανά προβλήματα και ότι η γνώση του αντικειμένου της μελέτης προκύπτει κατά την διάρκεια της αξιολόγησης.

Στην μη στοχευμένη αξιολόγηση δεν χρησιμοποιείται κανένας ρητός στόχος, είναι μια επαγωγική και κατά περίπτωση οδηγούμενη στρατηγική, η οποία πρέπει να έχει ευέλικτο χαρακτήρα.

γ) **Αξιολόγηση βάση κριτηρίων** (*Criteria-based evaluation*) όπου μερικά ρητά γενικά κριτήρια χρησιμοποιούνται ως κριτήρια αξιολόγησης – η διαφορά με την στοχευμένη αξιολόγηση είναι ότι τα κριτήρια είναι γενικά και μη περιορισμένα σε ένα συγκεκριμένο οργανωτικό πλαίσιο. Υπάρχουν πολλές αξιολογήσεις βάση κριτηρίων προσεγγίσεις όπως οι πίνακες ελέγχου, αρχές ή ποιοτικά ιδανικά. Αυτό που είναι χαρακτηριστικό για αυτές τις προσεγγίσεις είναι ότι το πληροφοριακό σύστημα και/ή η αλληλεπίδραση μεταξύ των χρηστών και του πληροφοριακού συστήματος λειτουργούν σαν την βάση για την αξιολόγηση μαζί με ένα σύνολο προκαθορισμένων κριτηρίων. Τα κριτήρια που χρησιμοποιούνται σε αντίθεση με την στοχευμένη δεν προέρχονται από ένα συγκεκριμένο οργανωτικό πλαίσιο. Στην αξιολόγηση βάση κριτηρίων μερικοί από τους ρητούς στόχους χρησιμοποιούνται ως κριτήρια αξιολόγησης. Η διαφορά με την στοχευμένη αξιολόγηση είναι ότι τα κριτήρια είναι γενικά και μη περιορισμένα σε ένα συγκεκριμένο οργανωτικό πλαίσιο. Δεν ακολουθείται αυστηρά η προκαθορισμένη σειρά των ενεργειών και έχει πιο ευέλικτο χαρακτήρα.

2.2 Μέθοδοι αξιολόγησης

2.2.1 Αθροιστική αξιολόγηση (Summative evaluation)

Λαμβάνει χώρα όταν η παραγωγή ενός προϊόντος έχει τελειώσει. Εφαρμόζεται σε γραμμικές διαδικασίες παραγωγής όπως το waterfall model και για το debugging των beta version. Το μεγαλύτερο μειονέκτημα είναι ότι τα προβλήματα

ανακαλύπτονται μετά από πολλές επαναλήψεις της σχεδίασης, που δεν το υποστηρίζει η summative evaluation συνεπώς δεν υποστηρίζει την ανθρωποκεντρική σχεδίαση.

2.2.2 Εκπαιδευτική αξιολόγηση (Formative evaluation)

Λαμβάνει χώρα καθ' όλη την διάρκεια του κύκλου παραγωγής του συστήματος και αποτελεί feedback για την μετέπειτα πορεία της παραγωγής.

2.2.3 Εμπειρική αξιολόγηση (Empirical evaluation)

Η αξιολόγηση της διεπαφής χρήστη γίνεται με πραγματικούς χρήστες και απαιτεί προσομοιώσεις, prototypes ή ολόκληρη την υλοποίηση του συστήματος. Τα αντικείμενα πρέπει να είναι αντιπροσωπευτικά και χρησιμοποιούνται για να προσδιοριστούν τα προβλήματα ευχρηστίας. Εκτιμάται η γνώμη των χρηστών, παρατηρείται και αποτιμάται η συμπεριφορά του χρήστη απέναντι στο σύστημα. Οι μέθοδοι αυτοί είναι ευρέως γνωστοί και συνιστούν τους πιο διαδεδομένες μεθόδους αξιολόγησης. Το μεγαλύτερο μειονέκτημα της μεθόδου είναι ότι απαιτεί ειδικούς να σχεδιάσουν και να διευθύνουν τα πειράματα προϋποθέτει χρόνο και κοστίζει αρκετά. Έτσι, λειτουργεί σαν συμπληρωματική σε non-empirical techniques και συνήθως στο τέλος του κύκλου παραγωγής.

Η εμπειρική αξιολόγηση χωρίζεται σε:

2.2.4 Αντικειμενική αξιολόγηση

Τα πειράματα γίνονται σε εργαστήριο με πραγματικούς χρήστες που εκφράζουν τις σκέψεις τους, την γνώμη τους και τα συναισθήματά τους καθ' όλη την διάρκεια της χρήσης του συστήματος.

2.2.5 Υποκειμενική αξιολόγηση

Συνήθως γίνεται με ερωτήσεις και συνεντεύξεις που αποσκοπούν στη γνώμη του χρήστη για συγκεκριμένα θέματα ή για όλο το σύστημα.

2.2.6 Μη εμπειρική αξιολόγηση

Λέγεται και predictive evaluation. Σχετίζεται με την λεπτομερή αξιολόγηση από ειδικούς της διεπαφής χρήστη, των προδιαγραφών, του αρχικού μοντέλου,

prototype και του τελικού συστήματος. Έχει ως σκοπό τον προσδιορισμό των προβλημάτων από τα πρώτα βήματα προκειμένου να ελαττωθούν τα έξοδα ο χρόνος παραγωγής.

Χωρίζεται σε:

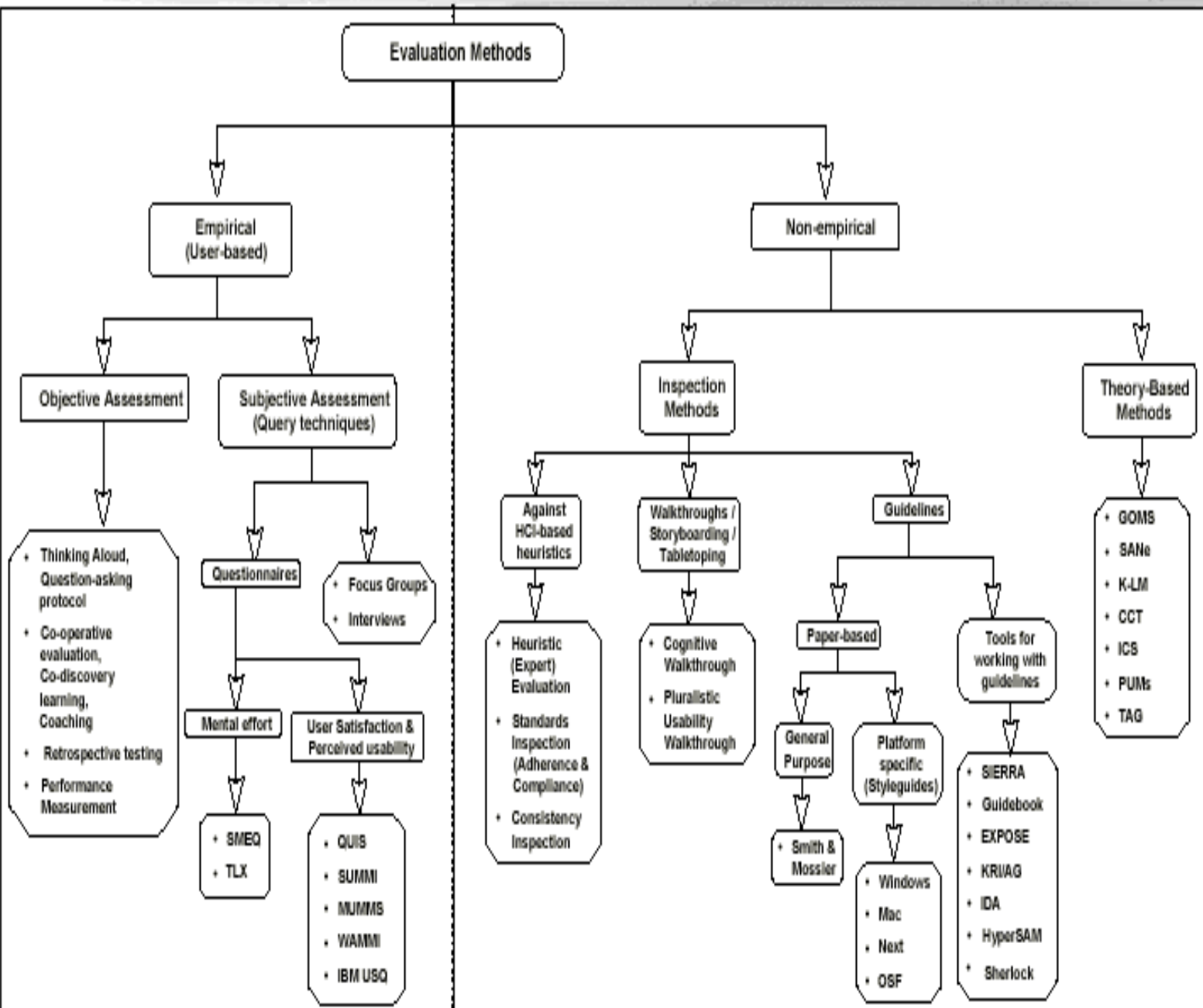
2.2.7 Αναλυτική μέθοδος

Βασίζεται σε μοντέλα που προβλέπουν την συμπεριφορά των χρηστών. Το βασικό μειονέκτημα είναι ότι είναι δύσκολο να εφαρμοστούν, απαιτούν πολύ έμπειρους ανθρώπους, κοστίζουν πάρα πολύ σε χρόνο και χρήμα. Έτσι δεν είναι δημοφιλή στην παραγωγή του software. Χρησιμοποιούνται κυρίως σε μέρη της αξιολόγησης που μπορούν να αυτοματοποιηθούν. Η πιο δημοφιλής μέθοδος είναι η GOMS.

2.2.8 Μέθοδοι ελέγχου

Αυτοί οι μέθοδοι βοηθούν στην επιθεώρηση της διεπαφής χρήστη από ειδικούς και καμιά φορά με τη συμμετοχή των χρηστών. Το αποτέλεσμα αυτής της αξιολόγησης είναι συστάσεις για την διόρθωση των προβλημάτων. Χρησιμοποιείται κυρίως για το debugging του κώδικα. Το κύριο πλεονέκτημα αυτής της μεθόδου είναι ευκολία της εφαρμογής και ότι αποτελεί φτηνή και γρήγορη λύση. Το μειονέκτημα της μεθόδου είναι ότι το αποτέλεσμά της εξαρτάται από τις ικανότητες, τη γνώση και την εμπειρία των αξιολογητών.

Στο παρακάτω σχήμα φαίνονται σχηματικά οι μέθοδοι της αξιολόγησης.

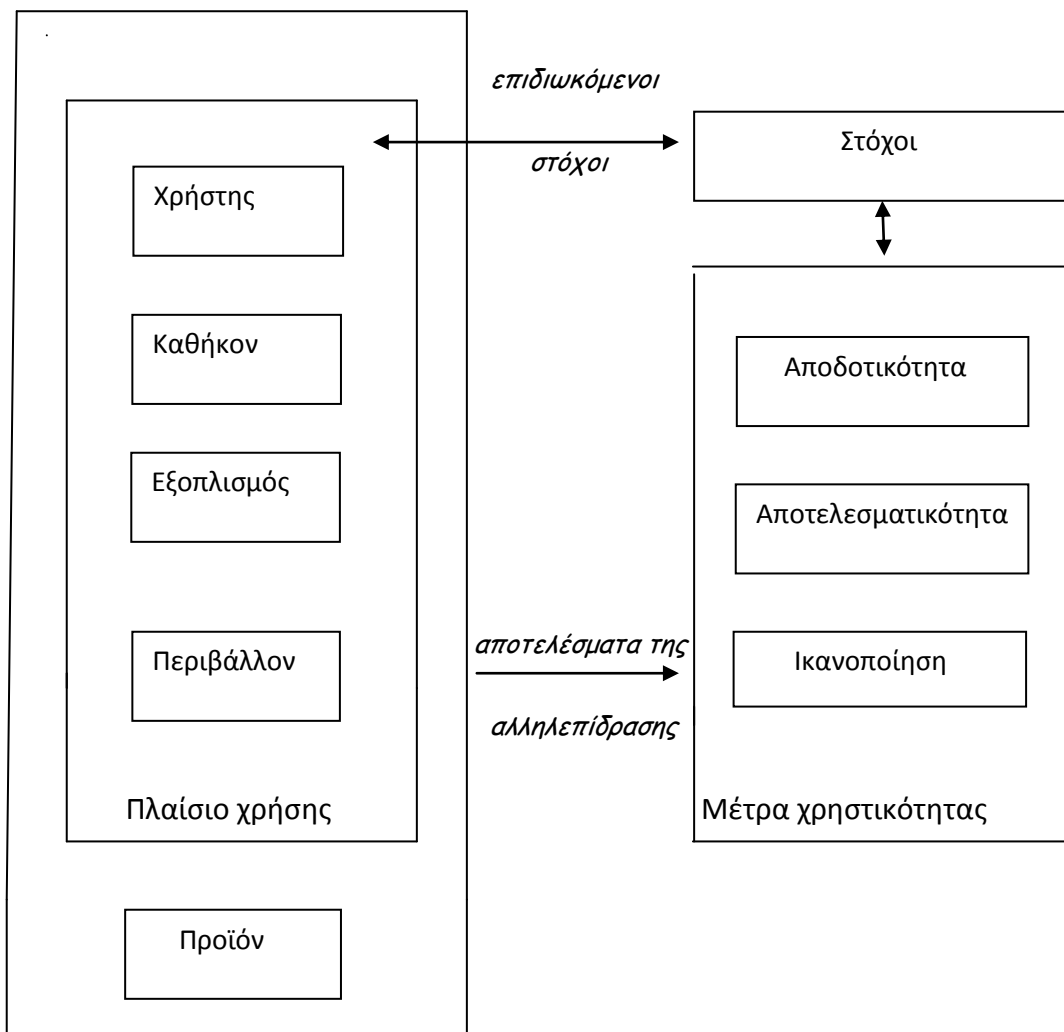


2.3 Μοντέλα αξιολόγησης

2.3.1 ISO 9241 - Ευχρηστία

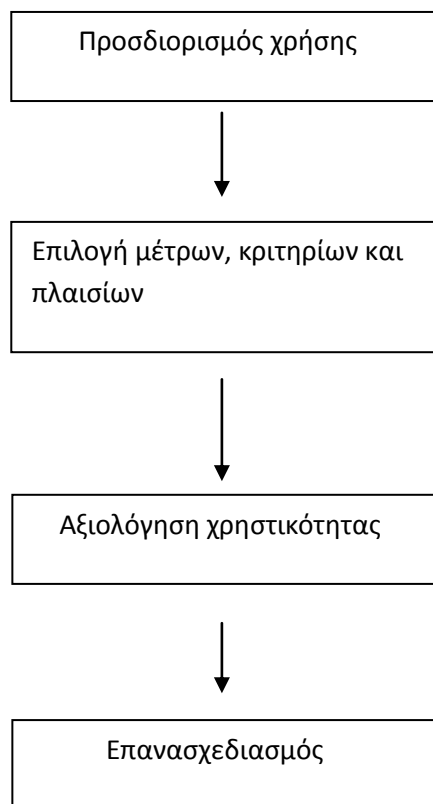
Το μοντέλο αξιολόγησης Part 11 του ISO 9241 αξιολογεί την ευχρηστία του συστήματος αλλά δεν εμπεριέχει απαιτήσεις ή συστάσεις.

- Η ευχρηστία αξιολογείται από την αποδοτικότητα, αποτελεσματικότητα και από το πόσο είναι ικανοποιημένοι οι χρήστες από το σύστημα.
- Δίνεται έμφαση στο context of use και στο γεγονός ότι το επίπεδο ευχρηστίας εξαρτάται από τις συγκεκριμένες περιστάσεις όπου χρησιμοποιείται το σύστημα.
- Πλαίσιο χρήσης: χρήστες, εξοπλισμός, περιβάλλον, στόχοι και διεργασίες. βαθμός στον οποίο οι στόχοι επιτυγχάνονται με την αποτελεσματικότητα



Σχήμα 2.1

Το παρακάτω σχεδιάγραμμα περιγράφει τις διεργασίες:



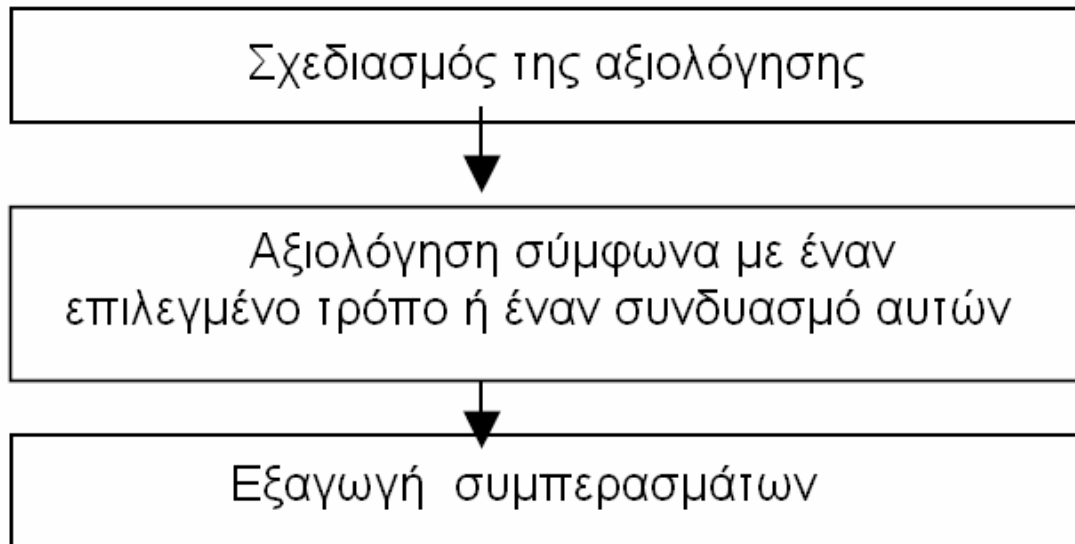
Σχήμα 2.2

2.3.2 Το γενικό μοντέλο αξιολόγησης

Το πρώτο πράγμα που πρέπει να γίνει είναι ο σχεδιασμός της αξιολόγησης. Αυτό σημαίνει να αποφασιστούν τα ακόλουθα ζητήματα: ο σκοπός, το επίπεδο λεπτομερειών

της αξιολόγησης, ο χρόνος, οι πόροι, ποιος πρέπει να κάνει την αξιολόγηση και διάφορα

άλλα.. Το σχέδιο αξιολόγησης λειτουργεί ως βάση για την επιλογή ενός τύπου αξιολόγησης ή ενός συνδυασμού διαφορετικών τύπων αξιολόγησης. Η διαδικασία της αξιολόγησης πρέπει να τελειώσει με τα συμπεράσματα, τα οποία πρέπει να αποτελούνται από μια συνοψισμένη ανάλυση του προβλήματος.



Σχήμα 2.3

Επιτυχία ενός πληροφοριακού συστήματος

Η επιτυχία συνδέεται με το νέο που πρόκειται να υλοποιηθεί, με την καινοτομία, άρα

συνδέεται άμεσα με τους στόχους αυτής της καινοτομίας. Μπορεί να εκτιμηθεί σε πολλά

επίπεδα:

- **επίπεδο οργανισμού** (συμφωνία με στρατηγικούς στόχους, λειτουργικό κόστος, διαθεσιμότητα συστήματος, χρόνοι απόκρισης, έσοδα/κέρδη κλπ)
- **επίπεδο διεργασιών ή λειτουργιών** (μείωση κόστους σε ειδικές λειτουργίες, μείωση χρόνων σε επί μέρους διεργασίες, ολοκλήρωση διεργασιών κλπ)
- **επίπεδο ατόμου** (ικανοποίηση χρηστών, χρησιμότητα του συστήματος)

2.3.3. Μοντέλο DeLone & McLean

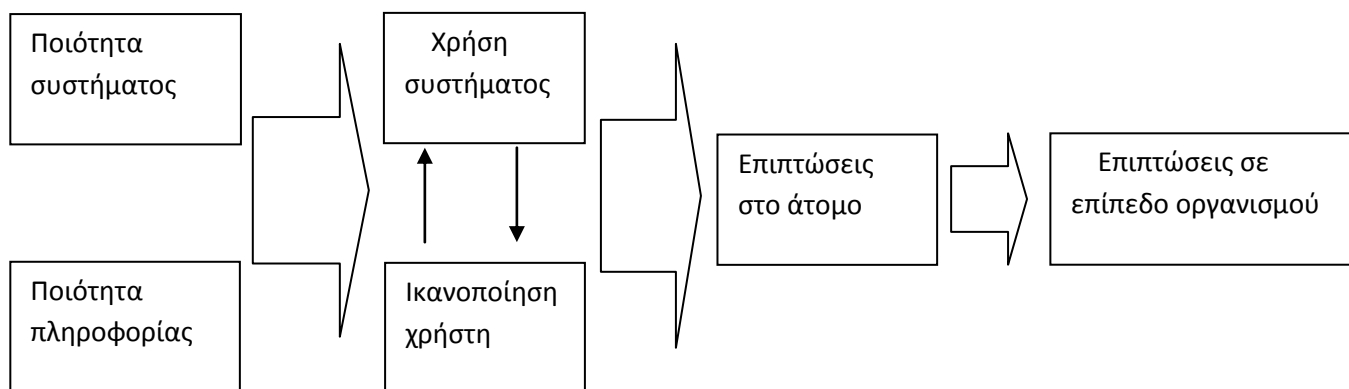
Οι DeLone και McLean χαρακτηρίζουν την ποιότητα συστήματος σαν τα επιθυμητά χαρακτηριστικά του ίδιου του συστήματος, και σαν ποιότητα πληροφοριών τα επιθυμητά χαρακτηριστικά των παραγόμενων πληροφοριών. Πιο συγκεκριμένα υιοθετούν μια τετραπλή κλίμακα αξιολόγησης για την ποιότητα του συστήματος

και μια εννιαπλή κλίμακα για την ποιότητα.

Η ικανοποίηση χρηστών στο συγκεκριμένο μοντέλο αναφέρεται γενικά στην ικανοποίηση χρηστών και μετρείται ανεξάρτητα από την ποιότητα του συστήματος και την ποιότητα πληροφοριών. Τέλος, οι DeLone και McLean χαρακτηρίζουν τις μεμονωμένες επιπτώσεις σαν την ένδειξη ότι ένα πληροφοριακό σύστημα δίνει στον χρήστη μια καλύτερη κατανόηση του πλαισίου αποφάσεων.

Το μοντέλο αξιολόγησης των DeLone & McLean αποτελεί ένα ευρέως αναγνωρισμένο και διαδεδομένο μοντέλο πάνω στο οποίο έχουν βασιστεί και επεκταθεί διάφορα άλλα μοντέλα επιτυχίας πληροφοριακών συστημάτων. Ένα από τα περισσότερο επιτυχημένα είναι και το λεγόμενο μοντέλο πληροφοριακού «οργανωτικής μνήμης». Για να το γνωρίσουμε καλύτερα και να εμβαθύνουμε σε αυτό, θα ξεκινήσουμε ευθύς αρχής δίνοντας τον ορισμό του. Είναι από τα πιο αποτελεσματικά και περισσότερο εφαρμοζόμενα και δοκιμασμένα εργαλεία μέτρησης της ποιότητας ενός πληροφοριακού συστήματος αποτελεί το μοντέλο αξιολόγησης συστημάτων των DeLone & McLean. Στο μοντέλο αυτό, στην ποιότητα του συστήματος λαμβάνονται υπόψη στοιχεία που έχουν σχέση με την παραγωγή, όπως χρόνοι απόκρισης, αξιοποίηση των πόρων, αξιοποίηση των επενδύσεων. Για την ποιότητα πληροφοριών χρησιμοποιούνται όροι προϊόντων, όπως ακρίβεια, πληρότητα κλπ.

Μέσα από το συγκεκριμένο μοντέλο εξετάζονται ουσιαστικοί παράγοντες, οι οποίοι αλληλεξαρτώνται και αλληλεπιδρούνται. Οι παράγοντες αυτοί διερευνούν έννοιες, όπως η ποιότητα του συστήματος, η ποιότητα της πληροφορίας, η ικανοποίηση του χρήστη, καθώς και η επίδραση της ποιότητας, τόσο στο άτομο, όσο και στον οργανισμό. Σχηματικά μπορεί να αποδοθεί ως εξής:



Σχήμα 2.4

Το μοντέλο αποτελείται από έξι αλληλένδετες μεταβλητές: ποιότητα συστήματος, ποιότητα πληροφοριών, χρήση του συστήματος, ικανοποίηση χρηστών, επίδραση στο άτομο, και επίδραση στον οργανισμό. Όπως φαίνεται και στο σχήμα το μοντέλο υποστηρίζει ότι η ποιότητα του συστήματος και η ποιότητα πληροφοριών επηρεάζουν την ικανοποίηση των χρηστών και την χρήση, τα οποία με την σειρά τους επηρεάζουν την επίδραση στο άτομο και στην συνέχεια την επίδραση στον οργανισμό.

Από μία έρευνα που έγινε το φθινόπωρο του 2005 προκύπτει ότι έχουν γίνει γύρω στις 150 αναφορές του συγκεκριμένου μοντέλου σε άρθρα περιοδικών κατά την διάρκεια από το 1993 μέχρι το 1999

α) Η **Ποιότητα Συστήματος** περιγράφει το πόσο «καλό» είναι το πληροφοριακό σύστημα, όσον αφορά τα λειτουργικά του χαρακτηριστικά.

β) Η **Ποιότητα Πληροφορίας** ορίζει το πόσο «καλό» είναι το πληροφοριακό σύστημα, όσον αφορά τις εκροές του. Ορισμένοι παράγοντες οι οποίοι έχουν σχέση με την ποιότητα της πληροφορίας ενδεικτικά είναι:

- η σημαντικότητα
- η σχετικότητα
- η χρησιμότητα
- η ακρίβεια

- η πληρότητα, και
- το περιεχόμενο της πληροφορίας

Παράλληλα, η ποιότητα της πληροφορίας έχει σημαντικό αντίκτυπο και στη χρήση του πληροφοριακού συστήματος.

α) Η **Χρήση του Συστήματος** αναφέρεται στην χρησιμοποίηση και αξιοποίηση των εκροών από το ίδιο το πληροφοριακό σύστημα

β) Η **Ικανοποίηση του Χρήστη** μετρά το πώς αντιλαμβάνονται οι χρήστες το ίδιο το σύστημα κατά τη χρησιμοποίησή του, και θεωρείται ως μία σημαντική παράμετρος για τη μέτρηση της επιτυχίας ενός πληροφοριακού συστήματος. Το αποτέλεσμα του συνόλου της χρήσης είναι ισοδύναμο και σημαντικό, ανεξάρτητα από την αποτελεσματικότητα του καθαυτού συστήματος.

γ) Η **Επίδραση στο Άτομο** αφορά το πώς επιδρά η χρήση ενός πληροφοριακού συστήματος στην εκτέλεση των καθηκόντων του ατόμου μέσα στον χώρο της εργασίας του. Σύμφωνα με τους DeLone & McLean, η επίδραση θα μπορούσε να αποτελεί μία ισχυρή ένδειξη ότι το πληροφοριακό σύστημα προσφέρει στον χρήστη καλύτερη κατανόηση του περιεχομένου των αποφάσεων του, βελτίωση σχετικά με τις αποφάσεις του για την παραγωγικότητα, αλλαγή στις δραστηριότητές του, καθώς και αλλαγή στην κατανόηση της σημαντικότητας και παράλληλα της χρησιμότητας του συστήματος του οργανισμού.

δ) Τέλος, η **Επίδραση στον Οργανισμό** εξετάζει κατά πόσο τα αποτελέσματα της επίδρασης του ατόμου επηρεάζουν την λειτουργία του οργανισμού, και επιπροσθέτως μετράει την αποτελεσματικότητα του οργανισμού ως ένα ολόκληρο σύνολο, ως μία πλήρη οντότητα.

2.3.4 To Technology Acceptance Model

Το Μοντέλο της Αποδοχής της Τεχνολογίας (TAM) αναπτύχθηκε προκειμένου να εξηγήσει και να προβλέψει την αποδοχή της Τεχνολογίας της Πληροφορικής από άτομα. Αναπτύχθηκε από τους Fred Davis και Richard Bagozzi .

Το TAM βασίστηκε στην Θεωρία της Δικαιολογημένης Δράσης (Theory of Reasoned Action -TRA), η οποία διατυπώθηκε από τους Ajzen και Fishbein και ισχυρίζεται ότι η κοινωνική συμπεριφορά παρακινείται από τη στάση ενός ατόμου απέναντι σε αυτή τη συμπεριφορά, από τις πεποιθήσεις του ατόμου σχετικά με το

αποτέλεσμα που θα έχει η υιοθέτηση της συγκεκριμένης συμπεριφοράς και από την αξιολόγηση της αξίας που θα έχει καθένα από αυτά τα αποτελέσματα.

Το TAM υιοθετεί τις αιτιακές σχέσεις της TRA προκειμένου να εξηγήσει τις συμπεριφορές αποδοχής της Τεχνολογίας που εκδηλώνουν τα άτομα. Σύμφωνα με αυτό το μοντέλο, η *Αντιλαμβανόμενη Ευκολία Χρήσης* και η *Αντιλαμβανόμενη Χρησιμότητα* από τη χρήση μιας συγκεκριμένης τεχνολογίας είναι οι δύο καθοριστικότεροι παράγοντες υιοθέτησής της.

Ο Davis ορίζει την αντιλαμβανόμενη χρησιμότητα σαν «το βαθμό, στον οποίο ένα άτομο πιστεύει ότι χρησιμοποιώντας ένα συγκεκριμένο σύστημα θα αυξήσει την απόδοσή του στην εργασία του» και την αντιλαμβανόμενη ευκολία χρήσης σαν «το βαθμό, στον οποίο ένα άτομο πιστεύει ότι η χρησιμοποίηση ενός συγκεκριμένου συστήματος δεν θα απαιτεί προσπάθεια».

Τα υπόλοιπα δύο μέρη του TAM είναι η Στάση προς τη Χρήση και η Συμπεριφορική Πρόθεση για Χρήση. Η Στάση προς τη Χρήση είναι η αξιολόγηση του χρήστη όσον αφορά την τοποθέτηση μιας συγκεκριμένης εφαρμογής πληροφοριακών συστημάτων. Η Συμπεριφορική Πρόθεση για χρήση είναι ένα μέτρο της πιθανότητας ότι ένα άτομο θα χρησιμοποιήσει μια συγκεκριμένη εφαρμογή. Η εξαρτημένη τέλος μεταβλητή του TAM είναι η Πραγματική Χρήση. Συνήθως μετράται με τη χρονική διάρκεια ή τη συχνότητα χρήσης μια συγκεκριμένης εφαρμογής. Η μέχρι σήμερα έρευνα έχει αποδείξει την ισχύ του συγκεκριμένου μοντέλου, το οποίο είναι πια ευρέως αποδεκτό.

Παράδειγμα ερωτηματολογίου βασισμένο στο TAM

Το παράδειγμα ερωτηματολογίου βασισμένο στο TAM χρησιμοποιήθηκε για την αξιολόγηση του συστήματος «ερευνώ το φυσικό κόσμο».

Μετά την εμπειρία που απέκτησαν οι χρήστες του συστήματος «ερευνώ το φυσικό κόσμο» τους δόθηκε ένα ερωτηματολόγιο το οποίο αναπτύχθηκε με βάση τα κύρια μέτρα αποδοχής της τεχνολογίας που προτείνει το μοντέλο TAM. Τα αποτελέσματα του ερωτηματολογίου συνοψίζονται στον παρακάτω πίνακα:

| Χαρακτηριστικά | Συμφωνώ | | | Διαφωνώ | |
|---------------------------------------------------------------------|---------|---------|-------------|---------|--------|
| | απόλυτα | Συμφωνώ | Ουδέτερος/η | Διαφωνώ | έντονα |
| Οργανώνει τη συζήτηση αποδοτικά | 1 | 9 | 3 | 0 | 0 |
| Υποκινεί τη συζήτηση | 2 | 8 | 2 | 1 | 0 |
| Ενισχύεται η απόκτηση γνώσεων μέσα από την επιχειρηματολογία | 3 | 6 | 4 | 0 | 0 |
| Διευκολύνεται η απόκτηση γνώσεων μέσα από την επιχειρηματολογία | 2 | 7 | 3 | 1 | 0 |
| Είναι εύκολη η εκμάθησή του | 9 | 3 | 1 | 0 | 0 |
| Είναι εύκολη η χρήση του | 9 | 2 | 2 | 0 | 0 |
| Είναι ευχάριστο ως προς τη χρήση του | 2 | 6 | 5 | 0 | 0 |
| Το συνολικό περιβάλλον εργασίας (δομή και εμφάνιση) είναι ελκυστικό | 1 | 2 | 8 | 2 | 0 |
| Θα το χρησιμοποιούσατε ξανά | 2 | 8 | 2 | 1 | 0 |
| Είναι εύκολο να βρεθούν οι διαθέσιμες επιλογές | 1 | 10 | 1 | 1 | 0 |
| Οι λειτουργίες, τα μενού και οι εικόνες είναι εύκολο να κατανοηθούν | 4 | 7 | 2 | 0 | 0 |
| Οι δυνατότητες πλοήγησης είναι εύκολο να κατανοηθούν | 1 | 9 | 2 | 1 | 0 |
| Το συνολικό περιβάλλον εργασίας είναι εύχρηστο | 2 | 8 | 3 | 0 | 0 |

| | | | | | |
|--------------------------------------------------------------------------|---|---|---|---|---|
| Το περιεχόμενο της επιχειρηματολογίας είναι καλά οργανωμένο και δομημένο | 1 | 8 | 4 | 0 | 0 |
| Αυτό που είχατε να κάνετε ήταν ξεκάθαρο για σας | 7 | 6 | 0 | 0 | 0 |

Πίνακας: Αποτελέσματα αξιολόγησης

Όπως φαίνεται οι περισσότεροι συμμετέχοντες ήταν θετικοί όσο αφορά τα κύρια μέτρα αποδοχής της τεχνολογίας που προτείνει το μοντέλο TAM. Ιδιαίτερα θετικοί ήταν στις ερωτήσεις 5,6 που αφορούν την Αντιλαμβανόμενη Ευκολία Χρήσης (Είναι εύκολη η εκμάθηση του; Είναι εύκολη η χρήση του;) όπου απάντησαν ότι συμφωνούν απόλυτα σε ποσοστό 70% . Ουδέτεροι ήταν σχετικά με την Στάση προς την Χρήση, ερωτήσεις 7,8 (Είναι ευχάριστο ως προς την χρήση του; Το συνολικό περιβάλλον εργασίας (δομή και εμφάνιση interfaces) είναι ελκυστικό;) σε ποσοστό 39% και 61,5%. Στις υπόλοιπες ερωτήσεις ήταν στο μεγαλύτερο ποσοστό θετικοί (συμφωνώ) και αυτό δείχνει την αποδοχή του συστήματος.

Σε γενικές γραμμές οι συμμετέχοντες δεν είχαν δυσκολίες στον πώς να μάθουν να χρησιμοποιούν το σύστημα, ήταν θετικοί αλλά κάπως πιο συντηρητικοί σχετικά με την οργάνωση της επιχειρηματολογίας και την απόκτηση γνώσεων από αυτό, και τέλος, ήταν σχετικά ουδέτεροι με την αισθητική πλευρά του συστήματος.

3. ΑΞΙΟΛΟΓΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ

Αξιολογούμενο εκπαιδευτικό λογισμικό: «Ερευνώ το φυσικό κόσμο»

(Φυσικά Ε΄ – ΣΤ΄ Δημοτικού)

Λογισμικά που συνοδεύουν τα νέα διδακτικά βιβλία.

3.1 ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ

Εκπαιδευτική Βαθμίδα/Τάξη στην οποία απευθύνεται:

Το λογισμικό απευθύνεται σε μαθητές της Πέμπτης και Έκτης Δημοτικού.

3.1.1 Γνωστικό αντικείμενο:

«Ερευνώ το φυσικό κόσμο»

(Φυσικά Ε΄ – ΣΤ΄ Δημοτικού)

3.1.2 Αν το λογισμικό πραγματεύεται περισσότερα από ένα γνωστικά αντικείμενα, σημειώστε ποια είναι αυτά:

Το λογισμικό πραγματεύεται τα γνωστικά αντικείμενα: Φυσική, Χημεία, Περιβαλλοντική εκπαίδευση, Βιολογία.

3.1.3 Αν το εκπαιδευτικό λογισμικό πραγματεύεται περισσότερα από ένα γνωστικά αντικείμενα, αυτά προσεγγίζονται:

Ολιστικά (δίχως εμφανή διάκριση αντικειμένων) και διεπιστημονικά

3.1.4 Ποια/ποιες από τις περιγραφές ανταποκρίνεται καλύτερα στο είδος του εκπαιδευτικού λογισμικού;

Το λογισμικό είναι μια πολυμεσική παρουσίαση (εικόνα, κείμενο, ήχος, βίντεο, animation) με εκπαιδευτικά παιχνίδια, βιογραφίες, φύλλα εργασίας, κριτήρια αξιολόγησης, προτεινόμενα βιβλία για περαιτέρω ανάγνωση και συνδέσμους σε ενδιαφέροντες ιστοσελίδες.

3.1.5 Το λογισμικό αυτό μπορεί να χρησιμοποιηθεί για:

- Παρουσίαση περιεχομένου
- Επανάληψη, υποστηρικτική διδασκαλία
- Εξάσκηση και πρακτική
- Αξιολόγηση μαθητή/αυτοαξιολόγηση
- Παιχνίδι
- Διερεύνηση Προσομοιώσεων - φαινομένων
- Εμβάθυνση
- Συνεργατικές Εργασίες
- Δραστηριότητες εργαστηρίου βασισμένες σε υπολογιστή

3.2 Ποιότητα περιεχομένου

3.2.1 Το περιεχόμενο είναι κατάλληλο για την ηλικία και τις γνώσεις των μαθητών;

Είναι απόλυτα κατάλληλο

3.2.2 Η γλώσσα και το ύφος είναι κατάλληλα για την ηλικία των μαθητών;

Ναι

3.2.3 Το περιεχόμενο είναι επιστημονικά σωστό (στα πλαίσια του διδακτικού μετασχηματισμού της επιστημονικής γνώσης);

Είναι σωστό επιστημονικά.

3.2.4 Υπάρχει συνέπεια στους όρους και στα σύμβολα που χρησιμοποιούνται;

Και οι όροι και τα σύμβολά που χρησιμοποιούνται είναι τα ίδια σε όλο το λογισμικό.

3.2.5 Η δομή και η παρουσίαση των πληροφοριών ακολουθούν ενιαίους κανόνες σε όλα τα τμήματα του λογισμικού;

Η δομή και η παρουσίαση των πληροφοριών είναι ενιαία για όλα τα τμήματα.

3.2.6 Είναι απαλλαγμένο από γραμματικά και συντακτικά λάθη;

Δεν εντοπίσαμε γραμματικά ή συντακτικά λάθη.

3.2.7 Το περιεχόμενο είναι απαλλαγμένο από εθνικά, φυλετικά ή άλλα στερεότυπα;

Δεν διακρίναμε πουθενά τέτοια στερεότυπα.

3.2.8 Ο όγκος της πληροφορίας είναι κατάλληλος για το συγκεκριμένο περιεχόμενο και το γνωστικό επίπεδο των μαθητών;

Ο όγκος της πληροφορίας είναι ο πλέον κατάλληλος για το επίπεδο των μαθητών της Ε΄ και Στ΄ τάξης Δημοτικού.

3.3 ΔΥΝΑΤΟΤΗΤΑ/ΠΟΙΟΤΗΤΑ ΕΝΤΑΞΗΣ ΣΤΙΣ ΣΧΟΛΙΚΕΣ ΣΥΝΘΗΚΕΣ

3.3.1 Το λογισμικό μπορεί να ενταχθεί στο υπάρχον αναλυτικό πρόγραμμα;

Μπορεί να ενταχθεί και μάλιστα απολύτως ομαλά, μιας και συμβαδίζει με το υπάρχον Αναλυτικό πρόγραμμα.

3.3.2 Το λογισμικό δημιουργεί νέες δυνατότητες στα πλαίσια του υπάρχοντος αναλυτικού προγράμματος;

Οι νέες δυνατότητες που δημιουργούνται είναι πολλαπλές, όπως: πειράματα που δεν μπορούν να εκτελεστούν στην τάξη (λιώσιμο σιδήρου, καύση μεθανίου κ.ά.), ομαδοσυνεργατικές δραστηριότητες, διαθεματικές εργασίες κ.ά.

3.3.3 Το λογισμικό είναι κατάλληλο για ανεξάρτητη ατομική χρήση από κάθε μαθητή;

Μπορεί το ίδιο εύκολα να χρησιμοποιηθεί και από κάθε μαθητή ξεχωριστά.

3.3.4 Το λογισμικό είναι κατάλληλο για ολοκληρωμένη χρήση (καλύπτει πλήρως μια διδακτική ενότητα με θεωρία, παραδείγματα, ασκήσεις κλπ) στη σχολική τάξη από το σύνολο των μαθητών και με ενεργή παρουσία του διδάσκοντα;

Όχι. Το λογισμικό λειτουργεί ως συμπλήρωμα του σχολικού βιβλίου και δεν μπορεί να καλύψει μόνο του μια διδακτική ενότητα.

3.3.5 Το λογισμικό επιτρέπει στο διδάσκοντα να επιλέξει τη σειρά διδασκαλίας;

Ναι, γιατί δεν είναι γραμμικά σχεδιασμένο.

3.3.6 Το λογισμικό επιτρέπει στο διδάσκοντα να επέμβει για να προσαρμόσει ή να προσθέσει υλικό;

Όχι, δεν παρέχει τέτοια δυνατότητα.

3.3.7 Το λογισμικό εξασφαλίζει τη δυνατότητα και τις προϋποθέσεις στους μαθητές να εμπλουτίσουν το διδακτικό υλικό;

Όχι.

3.3.8 Το λογισμικό επιτρέπει τη χάραξη εναλλακτικών διαδρομών από τον ίδιο τον μαθητή ανάλογα με τις ανάγκες του;

Ναι. Η μη γραμμικότητα της παρουσίασης το επιτρέπει.

3.4 ΔΙΔΑΚΤΙΚΗ ΚΑΙ ΠΑΙΔΑΓΩΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

3.4.1 Βασική οργάνωση

α) Οι διδακτικοί στόχοι χρήσης του λογισμικού γίνονται φανεροί (μέσα από την αλληλεπίδραση με το λογισμικό);

Σε αρκετά ικανοποιητικό βαθμό.

β) Το διδακτικό υλικό είναι οργανωμένο και δομημένο σε ενότητες;

Ναι. Υπάρχουν τέσσερεις μεγάλες ενότητες (Φυσική – Χημεία, το σώμα μας, οικολογία – περιβάλλον, διαθεματικά σενάρια) που χωρίζονται σε πολλές υποενότητες ή κεφάλαια.

γ) Η οργάνωση του υλικού είναι ευέλικτη έτσι ώστε να δίνει τη δυνατότητα στο διδάσκοντα να προσαρμόζει το περιεχόμενό του στις ανάγκες του μαθητή;

Δεν υπάρχει αυτή η δυνατότητα.

δ) Το λογισμικό στην παρούσα μορφή του προσφέρει στο διδάσκοντα τη δυνατότητα άμεσης επέκτασης του διδακτικού υλικού;

Ναι. Σε κάθε κεφάλαιο η επέκταση του διδακτικού υλικού μπορεί να επιτευχθεί:

- με την παράθεση πολλών ιστοσελίδων ανάλογου περιεχομένου που μπορούν να τον οδηγήσουν σε πρόσθετο διδακτικό υλικό και δραστηριότητες ή σχέδια εργασίας.

- με τα βιβλία που προτείνει, για περισσότερο διάβασμα
- με τις βιογραφίες των επιστημόνων που παραθέτει
- με την ιστορική αναδρομή σε θεωρίες και επιστημονικές εξελίξεις

ε) Είναι δυνατή η παρακολούθηση των δραστηριοτήτων των μαθητών με το λογισμικό από το διδάσκοντα;

Μόνο έμμεσα, από τα φύλλα δραστηριοτήτων και τα κριτήρια αξιολόγησης.

3.4.2 Προσέγγιση της μάθησης που ενθαρρύνεται

α) Το λογισμικό ενθαρρύνει την ενεργητική προσέγγιση της μάθησης, ενεργοποιώντας και διατηρώντας το ενδιαφέρον του μαθητή;

Σε πολύ μεγάλο βαθμό, γιατί δεν είναι ένα λογισμικό επίδειξης, αλλά απαιτεί την ενεργό συμμετοχή του μαθητή για την επίτευξη των μαθησιακών στόχων του, μέσα από τα παιχνίδια, τις ερωτήσεις, τις δραστηριότητες και τα τεστ.

β) Το λογισμικό υποστηρίζει ουσιαστικά την επικοινωνία μεταξύ των μαθητών μέσω τοπικού δικτύου;

Όχι.

γ) Το λογισμικό υποστηρίζει ουσιαστικά την επικοινωνία μεταξύ των μαθητών μέσω του διαδικτύου;

Ούτε αυτό το υποστηρίζει.

δ) Το λογισμικό ενθαρρύνει τη μάθηση μέσω διερεύνησης;

Σε πολύ μεγάλο βαθμό.

ε) Το λογισμικό ενθαρρύνει και υποστηρίζει τη συνεργατική μάθηση;

Ναι. Αν και αυτό εξαρτάται και από τον τρόπο που θα το χειριστεί ο δάσκαλος.

3.4.3 Τρόπος με τον οποίο σχηματίζονται ή αναπτύσσονται οι ιδέες καθώς ο μαθητής αλληλεπιδρά με το λογισμικό

α) Σε ποιο βαθμό το λογισμικό δίνει έμφαση σε κεντρικές έννοιες και αρχές του γνωστικού αντικειμένου, όπως αυτές προδιαγράφονται από το Πρόγραμμα Σπουδών; Σε μεγάλο βαθμό.

β) Σε ποιο βαθμό το λογισμικό οδηγεί σε σημαντικές συνδέσεις μεταξύ των εννοιών και αρχών εντός ή/και εκτός γνωστικού αντικειμένου; Σε ικανοποιητικό βαθμό.

γ) Σε ποιο βαθμό το λογισμικό ευνοεί ή/και καλλιεργεί τη χρήση κριτικών μεθόδων σκέψης για να βελτιώσει το βαθμό κατανόησης; Σε ικανοποιητικό βαθμό, με την επίλυση γρίφων, τα παιχνίδια και τις δραστηριότητες.

δ) Πόσο καλά το λογισμικό εμπλέκει τους μαθητές σε διαπραγμάτευση νοήματος και οικοδόμηση της γνώσης, μέσω έρευνας ή επίλυσης προβλήματος; Σε σχετικά ικανοποιητικό βαθμό.

ε) Σε ποιο βαθμό το λογισμικό προτείνει στους μαθητές αυθεντικές, αξιόπιστες εργασίες (εφαρμογή δεξιοτήτων σε καταστάσεις πραγματικές, ερωτήσεις, θέματα, αντιθέσεις, διλήμματα και προβλήματα);

Όλες οι εργασίες, τα προβλήματα και οι καταστάσεις που εμπλέκονται οι μαθητές αναφέρονται σε σύγχρονα και υπαρκτά προβλήματα όπως: η διατροφή, το οικολογικό, η εξοικονόμηση ενέργειας, τα μεταλλεία κ.ά.

στ) Σε ποιο βαθμό το λογισμικό εμπλέκει τους μαθητές στη χρησιμοποίηση ποικίλων μορφών παρουσίασης και έκφρασης;

Σε ικανοποιητικό βαθμό όπως: να κατασκευάσει κινούμενα σχέδια, η δημιουργία καταλόγου υλικών (αγωγών και μονωτών), η κατασκευή απλών ηλεκτρικών κυκλωμάτων, η μέτρηση του αριθμού των συγκρούσεων των μορίων ενός αερίου με τα τοιχώματα του δοχείου σε συνάρτηση με το χρόνο κλπ

3.5 ΕΙΔΙΚΑ ΚΡΙΤΗΡΙΑ ΑΝΑ ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ

3.5.1 Μαθηματικά και Φυσικές Επιστήμες (Φυσική, Χημεία, Γεωγραφία, Βιολογία)

α) Η πληροφορία που παρουσιάζεται είναι ακριβής και σύγχρονη;

Ναι.

β) Το λογισμικό χρησιμοποιεί μεθόδους επίλυσης προβλημάτων για να βοηθήσει τους μαθητές να οικοδομήσουν τη γνώση και να κατανοήσουν τις έννοιες των μαθηματικών ή/και των φυσικών επιστημών;

Σε αρκετές περιπτώσεις το επιτυγχάνει.

γ) Το λογισμικό παρουσιάζει πραγματικά-αυθεντικά προβλήματα για επίλυση ή αρκείται σε τυπικές και κλειστές ασκήσεις και ερωτήματα;

Όλα τα προβλήματα που θέτει το λογισμικό είναι πραγματικά, χωρίς τυπικές απαντήσεις και λύσεις.

δ) Το λογισμικό εμπλέκει τους μαθητές σε διερεύνηση γύρω από ανοικτά προβλήματα;

Όχι ιδιαίτερα.

ε) Το λογισμικό παρέχει εργαλεία, που να επιτρέπουν στους μαθητές να συλλέξουν, να αναλύσουν και να χειριστούν δεδομένα;

Όχι.

στ) Το λογισμικό προωθεί διεπιστημονικές συνδέσεις μεταξύ μαθηματικών και φυσικών επιστημών, μεταξύ φυσικής και χημείας, κ.ά.;

Πολλές φορές και με τρόπο ιδιαίτερα απλό και εύστοχο.

ζ) Το λογισμικό και τα προτεινόμενα σενάρια μελέτης λαμβάνουν υπόψη τα σύγχρονα πορίσματα της διδακτικής των επιστημών αναφορικά με τις δυσκολίες μάθησης συγκεκριμένων βασικών εννοιών;

Πάντοτε. Το λογισμικό ξεκινάει με τις ιδέες του μαθητή και μέσα από συγκεκριμένες δραστηριότητες επιδιώκει την αναδόμηση αυτών των ιδεών προς την επιστημονική ορθότητα.

η) Το λογισμικό, τα προτεινόμενα σενάρια μελέτης, και οι προτεινόμενες διδακτικές προσεγγίσεις φαίνεται να λαμβάνουν υπόψη τα σύγχρονα πορίσματα της διδακτικής των επιστημών αναφορικά με τις κατάλληλες και μαθησιακά αποτελεσματικές στρατηγικές διδασκαλίας;

Ναι. Αυτό λαμβάνει χώρα σε όλο το λογισμικό ξεκινώντας από το εισαγωγικό ερέθισμα, προχωρώντας στην ανάδειξη των ιδεών του μαθητή, συνεχίζοντας με την αναδόμηση των ιδεών αυτών και τελειώνοντας με το έλεγχο και την ανασκόπηση.

3.6 ΠΟΙΟΤΗΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ

3.6.1. Το είδος της διαλογικότητας που επιτρέπει το λογισμικό ενθαρρύνει την εμπλοκή και την ενεργό συμμετοχή του μαθητή;

Σχεδόν πάντοτε.

3.6.2 Επιτρέπεται στους μαθητές να πλοηγούνται ελεύθερα μέσα στο λογισμικό κάνοντας προσωπικές επιλογές;

Πάντοτε.

3.6.3.Υποστηρίζονται επαρκώς οι διαδικασίες πλοήγησης μέσω του εκπαιδευτικού λογισμικού;

Ναι. Η πλοήγηση στο λογισμικό υποστηρίζεται πλήρως.

3.6.4.Στην περίπτωση που το λογισμικό χρησιμοποιεί εικονίδια επιλογών, οι συμβολικές τους αναπαραστάσεις είναι κατανοητές και κατάλληλες για τους μαθητές;

Αρκετά κατανοητές.

3.6.5.Το λογισμικό χρησιμοποιεί ικανοποιητικά τις τεχνικές άμεσου χειρισμού;

Αρκετά ικανοποιητικά.

3.6.6.Ο σχεδιασμός και η υλοποίηση του λογισμικού κάνουν χρήση κάποιας σύμβασης ή μεταφοράς για την παρουσίαση της συνολικής δομής του λογισμικού;

Όχι ιδιαίτερα.

3.6.7.Η ανατροφοδότηση που παρέχει το λογισμικό κρίνεται ουσιαστική;

Δεν παρέχει ουσιαστική ανατροφοδότηση.

3.6.8.Η αλληλεπίδραση που προσφέρει κατά τη δικτυακή επικοινωνία κρίνεται άμεση και ικανοποιητική;

Ναι. Η αλληλεπίδραση είναι άμεση και ικανοποιητική.

3.7 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΣΥΓΚΕΚΡΙΜΕΝΩΝ ΤΥΠΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ

3.7.1 Λογισμικά που περιέχουν παρουσίαση περιεχομένου και Λογισμικά υπερκειμένων/εγκυκλοπαιδειών

α)Η χρήση των πολυμέσων γίνεται με ένα συμπληρωματικό και κατάλληλο τρόπο;

Ναι, όπου είναι απαραίτητο.

β) Περιλαμβάνει επαρκή και κατάλληλα βοηθήματα για το μαθητή (όπως δυνατότητες αυτόματων αναφορών, αποκοπής πληροφοριών, ιστορικό της πλοήγησης, δομημένη παρουσίαση των βασικών εννοιών, σημειωματάρια για σελίδες μαθητών, ή άλλα);

Όχι. Κανένα από τα παραπάνω.

γ) Ο συνδυασμός της χρήσης των πολυμέσων (κείμενο με ήχο, ήχος και εικόνα) είναι κατάλληλος και ουσιαστικός;

Είναι.

δ) Οι λειτουργίες του λογισμικού για αναζήτηση πληροφοριών είναι εύχρηστες;

Ναι, γιατί περιέχει χάρτη πλοήγησης.

ε) Ο χαρακτήρας των υπερ-συνδέσμων είναι σαφής (γίνεται διάκριση ανάμεσα σε συνδέσμους επεξηγηματικούς, ορισμών, εμβάθυνσης, κλπ.);

Όχι, δεν υπάρχει τέτοια διάκριση.

Στ) Υπάρχει δυνατότητα εκτύπωσης και μεταφοράς του υλικού;

Όχι.

3.7.2 Λογισμικά που περιέχουν κλειστές ερωτήσεις και ασκήσεις Ανατροφοδότηση και μαθησιακή αξιοποίηση του λάθους

α) Το λογισμικό, κατά την επίλυση των ασκήσεων ή την απάντηση των ερωτήσεων, επιτρέπει στο μαθητή πάνω από μία προσπάθειες για να βρει τη σωστή λύση/απάντηση;

Επιτρέπει όσες προσπάθειες θέλει ο μαθητής.

β) Σε περίπτωση λάθους, το λογισμικό παρουσιάζει αμέσως τη σωστή απάντηση, ή δίνει νύξεις, ή/και υποδείξεις στο μαθητή για να οδηγηθεί μόνος του στη σωστή απάντηση;

Δεν παρουσιάζει τη σωστή απάντηση ποτέ, αλλά ούτε δίνει και υποδείξεις στο μαθητή για να απαντήσει σωστά. Ο μαθητής πρέπει να επαναλάβει τη δραστηριότητα

πιο προσεχτικά για να βρει την σωστή απάντηση στην ερώτηση που είναι συνήθως πολλαπλών επιλογών.

γ) Αναλύει και σχολιάζει το λάθος του μαθητή (εξηγεί γιατί η απάντηση δεν είναι κατάλληλη και πού βρίσκεται πιθανόν το λάθος του);

Όχι, ποτέ.

δ) Σε περίπτωση που το λογισμικό δίνει στο μαθητή τη σωστή απάντηση, η απάντηση αυτή συνοδεύεται από κατάλληλη εξήγηση;

Δεν δίνεται ποτέ η σωστή απάντηση.

δ) Στην περίπτωση των ανοικτών ερωτήσεων, επιτρέπει ποικιλία στους τρόπους απάντησης, κάνοντας ένα ευρύ φάσμα απαντήσεων αποδεκτό;

Δεν περιέχει ανοιχτές απαντήσεις.

ε) Εκτιμάτε ότι οι ανοιχτές απαντήσεις θα είναι χρήσιμες, στις ερωτήσεις/προβλήματα που επεξεργάζεται το παρόν λογισμικό;

Όχι. Οι ανοιχτές απαντήσεις δεν είναι χρήσιμες σε ένα λογισμικό που διαπραγματεύεται έννοιες της φυσικής, εφόσον αυτές έχουν ένα συγκεκριμένο περιεχόμενο.

3.7.3 Λογισμικά που εμπεριέχουν προσομοιώσεις

α) Οι προσομοιώσεις των φαινομένων/καταστάσεων/ μηχανημάτων που παρουσιάζονται έχουν σημαντική εκπαιδευτική αξία;

Ναι, και σε ένα λογισμικό που ασχολείται με τη φυσική αυτό είναι απαραίτητο.

β) Οι παράγοντες (αντικείμενα ή μεταβλητές) που μπορεί να επηρεάσει ή να μεταβάλλει ο χρήστης είναι αρκετοί;

Για το γνωστικό επίπεδο στο οποίο αναφέρεται είναι αρκετό.

γ) Η μεταβολή των τιμών των παραγόντων είναι ανοικτή ή βασίζεται σε λίγες προκαθορισμένες τιμές;

Οι τιμές σε μερικές μεταβλητές είναι προκαθορισμένες και σχετικά αρκετές και σε άλλες η μεταβολή των τιμών είναι ανοικτή.

δ) Η παραγόμενη προσομοίωση της εξέλιξης του φαινομένου είναι ικανοποιητική; Στις περισσότερες προσομοιώσεις είναι αρκετά ικανοποιητική.

ε) Το λογισμικό επιτρέπει στο μαθητή το χειρισμό ακραίων/οριακών καταστάσεων; Όχι.

στ) Υπάρχει υποστήριξη του μαθητή στο χειρισμό ακραίων/οριακών καταστάσεων; Όχι.

ζ) Το λογισμικό προσφέρει μια ικανοποιητική ποικιλία τρόπων αναπαράστασης της μεταβολής των μεγεθών (γραφικές παραστάσεις, πίνακες τιμών, κ.ά.);

Οι πίνακες και οι γραφικές παραστάσεις είναι ελάχιστες, αλλά δεν είναι και ιδιαίτερα απαραίτητες για το συγκεκριμένο γνωστικό επίπεδο των μαθητών.

η) Οι τρόποι αναπαράστασης της μεταβολής των μεγεθών (μέσω γραφικών παραστάσεων, πινάκων τιμών, κ.ά.) είναι ικανοποιητικοί;

Όπου υπάρχουν, η αναπαράστασή τους είναι ικανοποιητική.

θ) Το λογισμικό επιτρέπει στο μαθητή την ελεύθερη διερεύνηση του φαινομένου που προσομοιώνεται;

Όχι, είναι κυρίως καθοδηγούμενη η διερεύνηση της προσομοίωσης.

3.8 ΣΥΝΟΔΕΥΤΙΚΑ ΕΓΧΕΙΡΙΔΙΑ

3.8.1 Συνοδευτικό εγχειρίδιο διδάσκοντα

Α) Οδηγίες χρήσης του λογισμικού

- Περιλαμβάνει εισαγωγικό κεφάλαιο με συνοπτική περιγραφή του λογισμικού; Σαφές και περιεκτικό.

- Δηλώνει τις απαιτήσεις του συστήματος;

Η μόνη απαίτηση είναι ο υπολογιστής να έχει πλοηγό διαδικτύου (πρόγραμμα που υπάρχει σε κάθε σημερινό υπολογιστή) και αυτό είναι ένα επί πλέον πλεονέκτημα του λογισμικού που του παρέχει τη δυνατότητα της άμεσης αναβάθμισής του μέσω διαδικτύου.

- Οι οδηγίες εγκατάστασης και χρήσης του λογισμικού είναι επαρκείς και σαφείς; Ναι.

- Δίδονται ικανοποιητικές οδηγίες για προσαρμογές του λογισμικού και επεμβάσεις από το διδάσκοντα;

Δεν υπάρχει αυτή η δυνατότητα.

- Το ύφος του περιεχομένου είναι κατάλληλο για τους διδάσκοντες;

Είναι σαφές και κατανοητό.

B)Οδηγίες διδακτικής αξιοποίησης του λογισμικού

- Παρουσιάζεται με σαφήνεια το απαιτούμενο αρχικό επίπεδο γνώσεων των μαθητών και η αντιστοίχιση με τις ηλικίες και τις σχολικές τάξεις;

Παρουσιάζονται οι εναλλακτικές ιδέες των παιδιών οι οποίες πρέπει να αναδομηθούν μέσω των διδακτικών δραστηριοτήτων.

- Παρουσιάζονται με σαφήνεια οι διδακτικοί στόχοι;

Ναι.

- Παρουσιάζονται αναλυτικές προτάσεις διδακτικής αξιοποίησης του λογισμικού σε συγκεκριμένες ώρες διδασκαλίας;

Ναι.

- Τα σενάρια διδακτικής αξιοποίησης του λογισμικού είναι σε εύρος και στρατηγικές κατάλληλα και πλούσια στα πλαίσια των υπαρχουσών εκπαιδευτικών συνθηκών;

Απόλυτα. Ειδικά τα διαθεματικά διδακτικά σενάρια είναι πολύ πλούσια σε προτεινόμενες δράσεις και δραστηριότητες ομαδοσυνεργατικές, χωρισμένες σε αποστολές με ευρύτερους στόχους για τα παιδιά και το δάσκαλο. Και ενώ δεν είναι πολύ πλούσια σε υλικό δίνουν οδηγίες για την εξεύρεση και την επεξεργασία και παρουσίαση ανάλογου διαθεματικού υλικού.

- Δίδονται υποδείξεις για εναλλακτικά σενάρια διδακτικής αξιοποίησης του λογισμικού;

Όχι.

- Παρουσιάζεται βιβλιογραφία ή/και συμπληρωματικό υλικό για υποστήριξη του διδάσκοντα;

Ικανοποιητική βιβλιογραφία και αρκετές παραπομπές σε δικτυακό υποστηρικτικό υλικό.

- Συνοδευτικό εγχειρίδιο μαθητή

Δεν υπάρχει συνοδευτικό εγχειρίδιο μαθητή.

Συμπεράσματα

Οι περισσότεροι από εμάς, είτε έχουμε εξοικείωση με τους υπολογιστές είτε ακόμη περισσότερο το θαυμαστό εργαλείο της νέας εποχής μάς γεμίζει με αμηχανία και ίσως λίγο φόβο, δεν μπορεί παρά να νιώθουμε δέος απέναντι στην ευκολία με την οποία τα μικρά παιδιά, ακόμη και προσχολικής ηλικίας, συμφιλιώνονται με τη φωτεινή οθόνη, το ποντίκι και το πληκτρολόγιο. Λίγα λεπτά της ώρας αρκούν, συνήθως, και το παιδί κινεί τον δείκτη πάνω στην οθόνη με αυτοπεποίθηση, ανοίγει

μενού, κάνει κλικ σε ζωγραφιστά, εικονικά κουμπιά στην οθόνη, ζωγραφίζει, παίζει. Τα παιδιά δεν έχουν προκαταλήψεις απέναντι στον υπολογιστή ούτε γνωρίζουν τι είναι τεχνοφοβία, ενώ συντονίζονται πολύ πιο εύκολα, σε σύγκριση με τους ενηλίκους, με αυτά που δείχνει να ζητά το μηχάνημα από αυτά.

Αναγνωρίζοντας έμπρακτα το γεγονός ότι οι υπολογιστές δεν πρόκειται να φύγουν από δίπλα μας πολύ σύντομα, το αντίθετο μάλιστα, οι περισσότερες χώρες του δυτικού κόσμου έχουν προσθέσει μαθήματα πληροφορικής στα εκπαιδευτικά προγράμματα. Σε κάποιες χώρες τα πράγματα έχουν προχωρήσει ακόμη περισσότερο· εκτός των μαθημάτων για την εκμάθηση του υπολογιστή, ο υπολογιστής χρησιμοποιείται ως βοηθητικό εργαλείο και για τα υπόλοιπα μαθήματα. Τα όμορφα γραφικά, τα χρώματα, η μουσική, τα ηχητικά εφέ, όλα αυτά κεντρίζουν το ενδιαφέρον του παιδιού και επιτυγχάνουν αυτό που λίγοι μόνο δάσκαλοι κατάφεραν παλαιότερα: να διατηρήσουν αμείωτη την προσοχή του για περισσότερο από λίγη ώρα, καθώς μαθαίνει παίζοντας. Τα αποτελέσματα είναι μάλιστα πιο εντυπωσιακά στα πολύ μικρά παιδιά στο νηπιαγωγείο ή στα πρώτα χρόνια του δημοτικού.

Είναι όμως τα πράγματα τόσο ρόδινα; Όχι, απαντά η Jane Healy, συγγραφέας του βιβλίου *Failure to connect: How computers affect our children's minds for better and worse* (Αδυναμία σύνδεσης: Πώς επηρεάζουν οι υπολογιστές το μυαλό των παιδιών μας θέλουμε, δεν θέλουμε), το οποίο κυκλοφόρησε πρόσφατα στις ΗΠΑ και ήδη προκαλεί μεγάλες συζητήσεις. Στην Αμερική οι υπολογιστές έχουν ενσωματωθεί στην εκπαιδευτική διαδικασία εδώ και αρκετά χρόνια, ήδη από τα νηπιαγωγεία, και πια υπάρχει η συσσωρευμένη πείρα ώστε να μπορούν να βγουν μερικά πρώτα συμπεράσματα. Και η Healy το αποτολμά.

Η Jane Healy είναι εκπαιδευτικός με δεκαετίες εμπειρίας και οι απόψεις της στηρίζονται σε εξαντλητική μελέτη και αξιολόγηση στοιχείων από ανεξάρτητες έρευνες και προσωπικές παρατηρήσεις. Η εικόνα που ζωγραφίζει με το βιβλίο της δεν συμβαδίζει καθόλου με τις τρέχουσες απόψεις, είναι μάλιστα αρκετά δυσοίωνα, ώστε να ξυπνήσει από τον εφησυχασμό τους όλους εκείνους που θεωρούν ότι ένας υπολογιστής εξοπλισμένος με τα κατάλληλα εκπαιδευτικά - ψυχαγωγικά προγράμματα είναι ό,τι καλύτερο μπορούμε να προσφέρουμε στα παιδιά μας σήμερα,

για να τα προετοιμάσουμε για το μέλλον και να τα απομακρύνουμε από την τηλεόραση.

Δεν είναι έτσι, λέει η Healy· όχι μόνο δεν υπάρχουν στοιχεία που να τεκμηριώνουν την άποψη ότι οι υπολογιστές αυξάνουν τη μαθησιακή ικανότητα των μικρών παιδιών, αλλά υπάρχουν βάσιμοι λόγοι να πιστέψουμε ότι η υπερβολική ενασχόληση του παιδιού με το μηχάνημα μπορεί στην πραγματικότητα να ανακόψει τη φυσιολογική ανάπτυξη του μυαλού του. «Κάποια από τα πιο δημοφιλή (στις ΗΠΑ) εκπαιδευτικά προγράμματα υπολογιστή ενδέχεται ακόμη και να βλάπτουν τη δημιουργικότητα, την ικανότητα συγκέντρωσης και την ικανότητα δραστηριοποίησης». Παράλληλα, εξωπολογιστικές δραστηριότητες που έχει αποδειχθεί ότι βοηθούν, όπως η μουσική, η χειροτεχνία και η ζωγραφική με κανονικά κραγιόνια και μπογιές, περνούν σε δεύτερη μοίρα ή σταματούν εντελώς.

Και όμως, τα λόγια της Healy δεν είναι νεο-λουδιτική φλυαρία, γράφει ο Andrew Leonard παρουσιάζοντας το βιβλίο από τις σελίδες του δικτυακού περιοδικού «Salon». Τα στοιχεία στα οποία στηρίζεται είναι πολύ ανησυχητικά. Οι πιο πρόσφατες μελέτες στον τομέα της νευροφυσιολογίας, στις οποίες παραπέμπει, έχουν δείξει ότι η δομή του ανθρώπινου εγκεφάλου αμέσως μετά τη γέννηση και κατά τα πρώτα χρόνια της ζωής είναι εξαιρετικά εύπλαστη. Τα παιδιά μαθαίνουν για τον κόσμο με τη βοήθεια του σώματός τους: αγγίζοντας, ψηλαφώντας, σπάζοντας, σκουντουφλώντας. Ο,τι μαθαίνουν χαράσσεται ανεξίτηλα στον εγκέφαλο ως νέες συνάψεις μεταξύ νευρώνων. Το να υποκαταστήσουμε αυτή τη φυσική αλληλεπίδραση, με το ποντίκι και τον δισδιάστατο κόσμο που προσφέρει ο υπολογιστής, είναι κάτι που μπορεί να βλάψει ανεπανόρθωτα την ίδια την ανάπτυξη του εγκεφάλου, επισημαίνει η Healy.

Βέβαια, τέτοιος κίνδυνος υπάρχει μόνο για παιδιά πολύ μικρής ηλικίας και μόνο αν γίνεται κατάχρηση του υπολογιστή. Παρ' όλα αυτά δεν υπάρχει κανένας λόγος να φέρνουμε τα παιδιά μας σε επαφή με τους υπολογιστές πριν από την ηλικία των 10 - 11 ετών. Η άποψη, γράφει η Healy, ότι η εξοικείωση (έστω, όχι κατάχρηση) σε πολύ μικρή ηλικία μπορεί να τα βοηθήσει αργότερα είναι λανθασμένη. Στην πραγματικότητα φαίνεται να συμβαίνει ακριβώς το αντίθετο. Παιδιά που είχαν πάρε-δώσε με υπολογιστές από την ηλικία των τεσσάρων-πέντε ετών αντιμετώπιζαν πιο

μεγάλες δυσκολίες να προσαρμοστούν στις εξελίξεις της τεχνολογίας φθάνοντας στην ηλικία των 10-12, από τα παιδιά που μόλις τότε πρωτοέρχονταν σε επαφή.

Μα καλά, θα αναρωτηθούν πολλοί, κάνουν τόσο κακό οι υπολογιστές; Δηλαδή, τι; Να αφήσουμε το παιδί να βλέπει τηλεόραση;

Κι όμως, δεν είναι απαραίτητα χειρότερη η τηλεόραση, γράφει η Healy. Εκτός των ενοχλήσεων στα μάτια και στα χέρια που μπορεί να προκαλέσει η παρατεταμένη παραμονή μπροστά σε υπολογιστή, τα «οπτικά εφέ» στα περισσότερα εκπαιδευτικά προγράμματα ίσως αποδειχθούν περισσότερο αποβλακωτικά ακόμη και από την τηλεόραση. «Με τον υπολογιστή, η προσοχή εστιάζεται εύκολα σε ανάξιο για τόση προσοχή υλικό».

Η ενσωμάτωση των τελευταίων επινοήσεων της τεχνολογίας στην εκπαιδευτική διαδικασία δεν είναι πρόσφατη έμπνευση. Στις αρχές του αιώνα, και ενώ ο κινηματογράφος δεν μετρούσε παρά μερικά χρόνια ζωής, πολλοί μιλούσαν για την επανάσταση που πρόκειται να φέρει στην εκπαιδευτική διαδικασία. Οραματίζονταν ένα σχολείο γεμάτο κινηματογραφικές μηχανές προβολής να παίζουν ταινίες που θα συνοδεύουν την προφορική παράδοση και ίσως τελικά καταφέρουν να αντικαταστήσουν τον δάσκαλο. Το σχέδιο δεν απέδωσε παρά σε περιορισμένη, πειραματική κλίμακα. Όταν κατέφθασε η τηλεόραση, η ιδέα επανήλθε με τη μορφή των εκπαιδευτικών εκπομπών που θα παρακολουθούσαν τα παιδιά την ώρα του μαθήματος. Και αυτή η εκδοχή δεν διαδόθηκε ιδιαίτερα, κυρίως λόγω προβλημάτων προγραμματισμού. Κατά τις δεκαετίες του '60, του '70 και του '80, τα «προχωρημένα» σχολεία προσέφεραν οπτικοακουστική διδασκαλία με τη βοήθεια μαγνητοφώνων, διαφανειών και βίντεο, αλλά έπρεπε να φθάσουμε στην εποχή του υπολογιστή για να μπορέσει να γίνει πράξη η πλήρως πολυμεσική, αλληλεπιδραστική διδασκαλία και το άνοιγμα του σχολείου στον κόσμο μέσω του Internet.

Στην Ελλάδα σε όσα σχολεία έχουν εργαστήρι πληροφορικής, οι υπολογιστές χρησιμοποιούνται περιθωριακά. Τα παιδιά μαθαίνουν να χρησιμοποιούν κάποια προγράμματα και να κάνουν μικρές βόλτες στο Internet. Μερικά σχολεία έχουν δική τους παρουσία στο Internet και τα παιδιά μπορούν να έρθουν σε επαφή με άλλα παιδιά μέσω του Δικτύου, όμως κάποια άλλα σχολεία δεν έχουν καν υπολογιστή.

Σε άλλες χώρες της Ευρώπης, ωστόσο, και κυρίως στις ΗΠΑ, οι υπολογιστές έχουν μπει σε κάθε σχολική τάξη και σχεδόν σε κάθε σπίτι. Η βιομηχανία εκπαιδευτικού λογισμικού αυξάνεται με ρυθμούς της τάξης του 20% ετησίως, ενώ κανένας δεν διανοείται πια να ξεκινήσει τη διδασκαλία ενός μαθήματος αν δεν έχει εφοδιαστεί με τα σχετικά προγράμματα υπολογιστή. Χωρίς να μπορούμε να διαγράψουμε με μία μονοκοντυλιά τα καλά που προσφέρουν υπολογιστές και Internet, δεν μπορούμε παρά να επισημάνουμε ότι σήμερα, για πρώτη φορά, ο δάσκαλος φαίνεται να έρχεται σε δεύτερη μοίρα σε σχέση με ένα εργαλείο που αρχικά επρόκειτο να παίξει τον ρόλο του εκπαιδευτικού βοηθήματος. Και ο σχετικός προβληματισμός ακούγεται τελευταίως στις ΗΠΑ ολοένα και πιο συχνά.

Η συζήτηση για την Πληροφορική, τους υπολογιστές, το Internet και για το πώς αυτά μπορούν να ενσωματωθούν στην εκπαιδευτική διαδικασία έχει σήμερα πάρει μεγάλες διαστάσεις. Το υπουργείο Παιδείας έχει βεβαίως εξοπλίσει τα σχολεία με εργαστήρια Πληροφορικής, αλλά αυτό είναι ένα πολύ μικρό πρώτο βήμα. Πρέπει να εξοικειωθούν οι μαθητές και κυρίως οι εκπαιδευτικοί στη χρήση των υπολογιστών για να μπορέσουν να ενσωματωθούν (οι υπολογιστές) στην εκπαιδευτική διαδικασία. Αλλά και αυτά μόνα τους δεν αρκούν. Χρειάζεται και το κατάλληλο εκπαιδευτικό λογισμικό.

Σήμερα η παραγωγή εκπαιδευτικού λογισμικού στην Ελλάδα είναι πολύ μικρή, ενώ η ποιότητα του διεθνούς εκπαιδευτικού λογισμικού σύμφωνα με την άποψη του Παιδαγωγικού Ινστιτούτου είναι η εξής:

"...Δυστυχώς, τα παραδείγματα εκπαιδευτικού λογισμικού καλής ποιότητας, είναι διεθνώς πολύ λίγα. Σχετικές μελέτες που διεξάγονται τα δεκαπέντε τελευταία χρόνια στην Ευρώπη και στις ΗΠΑ, συγκλίνουν στο συμπέρασμα ότι τα τρία βασικά αίτια που προκαλούν την έλλειψη εκπαιδευτικού λογισμικού καλής ποιότητας, είναι τα εξής:

α. Το εκπαιδευτικό λογισμικό παράγεται ως ανεξάρτητο προϊόν και όχι ως μέρος ενός πακέτου διδακτικού υλικού που εξυπηρετεί συγκεκριμένο Πρόγραμμα Σπουδών.

β. Οι μηχανισμοί για την αξιολόγηση του εκπαιδευτικού λογισμικού είναι ανεπαρκείς.

γ. Το κόστος παραγωγής είναι μεγάλο."

Αν και τα παραπάνω δεν αναφέρονται ευθέως στο υπάρχον Ελληνικό εκπαιδευτικό λογισμικό, η κατάσταση δεν είναι διαφορετική. Η εξέταση αρκετών (από τα λίγα) Ελληνικών προϊόντων εκπαιδευτικού λογισμικού που αναφέρονται στη Φυσική και τη Χημεία μου δημιούργησε τις παρακάτω εντυπώσεις:

α. Τα θέματα που πραγματεύονται τα διάφορα προϊόντα, αναφέρονται σε γενικά στοιχεία που υπάρχουν στο αναλυτικό πρόγραμμα και είναι ασυμβίβαστα με αυτό και τις κατευθύνσεις του. Αυτό καθιστά δύσκολη έως αδύνατη την ένταξη του λογισμικού στην διδασκαλία, αλλά και πολυτέλεια την ενασχόληση του μαθητή με αυτό στο σπίτι.

β. Πολλές φορές τα προϊόντα εκπαιδευτικού λογισμικού λειτουργούν σαν βιβλίο, με αποτέλεσμα να μην είναι προτιμητέα η μελέτη με αυτά, επειδή πιο εύκολα μελετά κανείς από ένα βιβλίο, παρά από την οθόνη του υπολογιστή.

γ. Λείπουν οι προσομοιώσεις πραγματικών καταστάσεων (στα θετικά μαθήματα, αλλά μπορεί να γίνει και σε άλλα π.χ. Ιστορία, Κοινωνιολογία), με αποτέλεσμα να μη μπορεί ο μαθητής να πειραματιστεί με μεταβολές μεγεθών, ώστε να οδηγηθεί σε σχέσεις και νόμους.

δ. Δεν λειτουργούν σαν άμεσα βοηθήματα στα μαθήματα των μαθητών και έτσι οι μαθητές δεν έχουν άμεσο ενδιαφέρον να ασχοληθούν με αυτά.

ε. Δεν μπορούν να χρησιμοποιηθούν στην τάξη εξ αιτίας αυτών που αναφέρονται παραπάνω, αλλά και από έλλειψη των κατάλληλων οπτικοακουστικών μέσων των σχολικών μονάδων.

Σήμερα βρίσκονται στη φάση ανάπτυξης στα πλαίσια του προγράμματος ΟΔΥΣΣΕΙΑ (4) αρκετά προγράμματα, που στοχεύουν στην παραγωγή εκπαιδευτικού

λογισμικού για τη δευτεροβάθμια εκπαίδευση. Αν και τα προγράμματα αυτά δεν έχουν ακόμα δείξει δείγματα έργου, είναι φανερό από την περιήγηση στις ηλεκτρονικές σελίδες τους, ότι προσανατολίζονται στον εξελληνισμό του υπάρχοντος ξένου εκπαιδευτικού λογισμικού και όχι στην παραγωγή Ελληνικού.

Οι εκπαιδευτικοί και το εκπαιδευτικό λογισμικό

Ο μεγάλος όγκος των εκπαιδευτικών σήμερα απέχει πολύ από το να θεωρείται οικείος με την Πληροφορική. Όμως η Πληροφορική έχει εισχωρήσει σε όλους τους τομείς της ζωής μας, άρα η χρήση της και η γενίκευσή της στην εκπαιδευτική διαδικασία θεωρείται απαραίτητη. Σε λίγο καιρό το εκπαιδευτικό λογισμικό θα έχει καθιερωθεί σαν ένα επί πλέον εκπαιδευτικό εργαλείο. Όμως σ' αυτό το σημείο εμφανίζεται ένα παράδοξο. Παραδοσιακά οι εκπαιδευτικοί κατασκευάζουν οι ίδιοι κυρίως τα εκπαιδευτικά εργαλεία τους. Οι σημειώσεις, οι διαφάνειες, τα αποσπάσματα video και ηχογραφήσεων, τα πειράματα κ.α. που χρησιμοποιεί ο εκπαιδευτικός στην τάξη γίνονται από τον ίδιο. Ακόμα και τα βιβλία καθώς και τα σχολικά βοηθήματα γράφονται από εκπαιδευτικούς. Αλλά στην παραγωγή του εκπαιδευτικού λογισμικού τα πράγματα είναι διαφορετικά. Λόγω της ειδικής τεχνικής φύσης αυτής της εργασίας η παραγωγή γίνεται από άτομα ξένα με την εκπαιδευτική πράξη. Είναι πιθανό σε ορισμένα στάδια της παραγωγής να συμμετέχουν και εκπαιδευτικοί, αλλά αυτό δεν αλλάζει το γενικό συμπέρασμα. Βλέπουμε λοιπόν ότι η κατασκευή ενός εργαλείου που αποσκοπεί στη διδασκαλία, ξεφεύγει από τους φυσικούς φορείς της. Αυτό μπορεί να έχει σαν συνέπειες, αφ' ενός οι εκπαιδευτικοί να μην αγκαλιάσουν ένα τόσο ισχυρό εκπαιδευτικό εργαλείο, αφ' ετέρου το παραγόμενο εκπαιδευτικό λογισμικό να μην έχει τα στοιχεία, που μόνο οι μάχιμοι εκπαιδευτικοί κατακτούν με την εμπειρία τους και με τα οποία καταφέρουν να προσαρμόζουν το μάθημά τους στις εκάστοτε συνθήκες (5).

Προτάσεις

Έχοντας υπόψη τα παραπάνω, η προτάσεις απευθύνονται και στους

εκπαιδευτικούς και στην πολιτεία.

α. Οι εκπαιδευτικοί οφείλουν να μάθουν τη χρήση των ηλεκτρονικών υπολογιστών και των δημοφιλών προγραμμάτων. Με αυτά θα μπορέσουν να ανταποκριθούν στοιχειωδώς στη χρήση εκπαιδευτικού λογισμικού στη διδασκαλία. Εύκολη είναι και η εκμάθηση προϊόντων δημιουργίας παρουσιάσεων, ώστε να είναι σε θέση να κάνουν οι ίδιοι εκπαιδευτικές παρουσιάσεις με όλη τη δύναμη και τα μέσα που προσφέρουν σήμερα οι υπολογιστές (ήχος, κίνηση, αλληλεπίδραση, διασύνδεση). Ακόμα και οι γλώσσες προγραμματισμού έχουν γίνει σήμερα αρκετά φιλικές, ώστε να είναι σχετικά εύκολη υπόθεση η δημιουργία ενός απλού προγράμματος.

β. Η πολιτεία οφείλει να δημιουργήσει το κατάλληλο περιβάλλον, ώστε να μπορέσουν οι εκπαιδευτικοί να ασχοληθούν με τη χρήση και την κατασκευή εκπαιδευτικού λογισμικού. Πρέπει να γίνει κατάλληλη επιμόρφωση των εκπαιδευτικών στη χρήση ηλεκτρονικών υπολογιστών και ακόμα να ενθαρρύνει αυτούς που είναι διατεθειμένοι να ασχοληθούν με την κατασκευή εκπαιδευτικού λογισμικού. Δεν είναι αρκετό να εξοπλιστούν τα σχολεία με εργαστήρια πληροφορικής και προγράμματα εκπαιδευτικού λογισμικού. Πρέπει να είναι δυνατή η χρήση του εκπαιδευτικού λογισμικού στην τάξη. Επειδή σήμερα υπάρχουν σε αρκετές σχολικές μονάδες εκπαιδευτικοί με αρκετά καλές γνώσεις Πληροφορικής, αυτοί μπορούν να χρησιμοποιηθούν σαν πυρήνες σε προγράμματα ενδοσχολικής επιμόρφωσης σχετικά με την παραγωγή εκπαιδευτικού λογισμικού με απλά μέσα. Τα παραπάνω κοστίζουν βεβαίως αρκετά, αλλά είναι απαραίτητα για να μην αποξενωθεί το εκπαιδευτικό λογισμικό από τους εκπαιδευτικούς.

Βιβλιογραφία

Σ. Μπακογιάννης, Καθ. Πληροφορικής, αποσ. στο Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών, sbakogia@atlas.uoa.gr

Γ. Φιλοκύπρου, Καθηγητής, Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών, gphilo@di.uoa.gr

Π. Γεωργιάδης, Αν. Καθηγητής, Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών,
pgeor@di.uoa.gr

Μ. Γρηγοριάδου, Επίκ. Καθηγήτρια, Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών,
gregor@di.uoa.gr

Παναγιώτης Θεοδωρόπουλος: Σχολικός Σύμβουλος ΠΕ03 www.ptheodoropoulos.gr
, Τμήμα Μηχανολόγων και Αεροναυπηγών Μηχανικών
Πανεπιστήμιο Πατρών

Αξιολόγηση και Ποιότητα Πληροφοριακών Συστημάτων Ιδρυματικών Οργανισμών
Ελένη Μαμμά *Γενικά Αρχεία του Κράτους ΥΠ.Ε.Π.Θ. Δάφνης 61, 154 52, Αθήνα*
(mamma.archives.gak@gmail.com)

Αξιολόγηση Εκπαιδευτικού Λογισμικού με checklist: Εφαρμογές και Προβλήματα

Μαρκάδας Σωτήρης

Εκπαιδευτικός – Μεταπτυχιακός φοιτητής Π.Τ.Δ.Ε. Φλώρινας, Α.Π.Θ.

3^ο *Ειδικό Σχολείο Θεσσαλονίκη - Παιδαγωγικό Τμήμα Δημοτικής Εκπαίδευσης*
Φλώρινας, sotiris@eled-fl.auth.gr

Βιβλιοθήκη ΑΤΕΙ Θεσσαλονίκης, Τμήμα Πληροφορικής:

Προγραμματισμός Η/Υ Οδηγίες Συγγραφής και Αξιολόγησης Εργασιών

Μοντέλο αξιολόγησης εκπαιδευτικού λογισμικού:

Σ. Μπακογιάννης, Καθ. Πληροφορικής, αποσ. στο Τμήμα Πληροφορικής
Πανεπιστημίου Αθηνών, sbakogia@atlas.uoa.gr

Γ. Φιλοκύπρου, Καθηγητής, Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών,
gphilo@di.uoa.gr

Π. Γεωργιάδης, Αν. Καθηγητής, Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών,
pgeor@di.uoa.gr

Μ. Γρηγοριάδου, Επίκ. Καθηγήτρια, Τμήμα Πληροφορικής Πανεπιστημίου Αθηνών,
gregor@di.uoa.gr

Το εκπαιδευτικό λογισμικό στη δευτεροβάθμια εκπαίδευση – σκέψεις και προτάσεις για την παραγωγή και χρήση του: Παντελής Μπαζάνος χημικός - καθηγητής στο Γυμνάσιο Γαργαλιάνων Email: pbazanos@otenet.gr Web: http://www.geocities.com/pbazanos_gr

Βιβλιοθήκη Πανεπιστημίου Πειραιά, Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων τεχνολογία λογισμικού, Οκτώβρης 2007, Δρ. Ανδριάννα Πρέντζα, aprentza@biomed.ntua.gr

Standards - Αξιολόγηση

Καλαϊτζή Έφη, Χαδήρογλου Μαρία, Μπάρκη Αγγελική, Ιούνιος 2001

HMMY

Τεχνολογία Λογισμικού - Διδάσκων: Κώστας Κοντογιάννης, Αναπλ. Καθηγητής, Ε.Μ.Π

Το λήμμα της Wikipedia για το εκπαιδευτικό λογισμικό http://en.wikipedia.org/wiki/Educational_software

Εκπαιδευτική πύλη του ΥΠΕΠΘ με αρκετούς προτεινόμενους τίτλους λογισμικού και υπηρεσίες. <http://www.e-yliko.gr>

Το Παιδαγωγικό Ινστιτούτο παρέχει ελεύθερα μία συλλογή από λογισμικό κλειστού ως προς το περιεχόμενο. <http://www.pi-schools.gr/>

Τμήμα Μηχανολόγων και Αεροναυπηγών Μηχανικών - Πανεπιστήμιο Πατρών: Μέθοδοι Αξιολόγησης Πληροφοριακών Συστημάτων: Φουντουλάκη Αικατερίνη Α.Μ. 441

Μεταπτυχιακό Μάθημα: «Τεχνολογίες Υποστήριξης Συνεργασίας»

Διδάσκων: Νίκος Καρακαπιλίδης Πάτρα, Γενάρης 2006

Ανάλυση Συστημάτων (2002-03 site) <http://www.di.uoa.gr/~afrodite/SA2003.html>
Υπευθ. Καθηγ.: Τσαλγατίδου Αφροδίτη

Πληροφοριακό σύστημα αξιολόγησης και πρόκρισης επενδύσεων: Βασίλειος Κώστογλου, επίκουρος καθηγητής Τ.Ε.Ι. Θεσσαλονίκης, Φώτιος Αποστολίδης, πληροφορικός, Τ.Ε.Ι. Θεσσαλονίκης, Τμήμα Πληροφορικής bas@it.teithe.gr

Σχεδίαση και ανάπτυξη εκπαιδευτικού λογισμικού υπερμέσων: Α. Τζιμογιάννης, Α. Κατσίκης, Α. Τσιμάκης, Ε. Νικολού, Α. Γιούνης, Τ. Α. Μικρόπουλος

Μοντέλα Κύκλου-Ζωής Λογισμικού: SOFTWARE LIFE-CYCLE MODELS
Δρ. Μαρία Ι. Ανδρέου

Παγκόσμιο χωριό γνώσης θεωρία μάρκετινγκ κύκλος ζωής προϊόντος, Φιλιατρά,
Απρίλιος-2000

<http://odysseia.cti.gr/actions.htm#category-2>

<http://hdtc.pi-schools.gr/material/software.htm>

Γ. Παπαδόπουλος: "Έλεγχος Ποιότητας Εκπαιδευτικού Λογισμικού: Ο σχεδιασμός και το έργο του Παιδαγωγικού Ινστιτούτου", <http://hdtc.pi-schools.gr/material/ict.htm>

D. Fontana: Ο εκπαιδευτικός στην τάξη, σελ. 221-222, εκδ. ΣΑΒΒΑΛΑ, 1996.
http://www.pedia.gr/corner/ekp_logismiko.htm

Ψηφιακή βιβλιοθήκη Νημερτής, Μακρής Χρήστος, Δεκέμβρης 2009