ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΤΗΣ ΔΙΟΙΚΗΣΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΟΙΚΟΝΟΜΙΚΗ ΚΑΙ ΔΙΟΙΚΗΣΗ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ
(ΠΜΣ-Ο.ΔΙ.Μ.)

Εφαρμογή Εκπαιδευτικού Λογισμικού για την Εκπαίδευση στη Διοίκηση Έργων

Επώνυμο : Γρηγορίου
Όνομα : Γεώργιος
Αρ. Μητρώου : 233Μ/07/005
fmem07005@fme.aegean.gr

Επιβλέπων Καθηγητής : Κηρυττόπουλος Κωνσταντίνος

Χίος, 2009

Ευχαριστίες

Ευχαριστώ την οικογένεια μου, την αρραβωνιαστικιά μου Κωνσταντίνα, τον επιβλέπων καθηγητή μου κ. Κηρυττόπουλο Κωνσταντίνο, τον καθηγητή κ. Τσάκωνα Αθανάσιο, τον Κυρικλίδη Χρήστο, και πάνω απ’ όλα το Θεό, για τη πολύτιμη βοήθεια και καθοδήγηση τους.

Δήλωση κυριότητας

Το αποτέλεσμα της παρούσας Διπλωματικής εργασίας, λογισμικό **Trainer©,** είναι διαθέσιμο προς ελεύθερη χρήση για εκπαιδευτικούς – ακαδημαϊκούς σκοπούς. Για οποιαδήποτε εμπορική εκμετάλλευση του λογισμικού θα πρέπει να υπάρχει έγγραφη συναίνεση του δημιουργού του κ. Γεωργίου Γρηγορίου και του Τμήματος Μηχανικών Οικονομίας & Διοίκησης, Παν. Αιγαίου.

Πίνακας Περιεχομένων:

Εικόνες – Σχήματα:

Πίνακες:

0 Έποψη

Ο βασικός στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη εκπαιδευτικού λογισμικού για την εκπαίδευση στη Διοίκηση Έργων. Υπάρχουν βεβαίως και επιμέρους στόχοι, οι οποίοι παρουσιάζονται στη συνέχεια, και αφορούν το περιεχόμενο, τα παιδαγωγικά χαρακτηριστικά, την ευκολία χρήσης, πλοήγησης και παρουσίασης και τα λειτουργικά χαρακτηριστικά του εκπαιδευτικού λογισμικού.

Οι πληροφορίες και η γνώση που προσφέρει το εκπαιδευτικό λογισμικό, λήφθηκε πρόνοια ώστε να καλύπτουν την εκπαίδευση αρχάριων χρηστών και είναι κατάλληλο για εκπαίδευση φοιτητών.

Σε ότι αφορά τα παιδαγωγικά χαρακτηριστικά το εκπαιδευτικό λογισμικό παρέχει δυνατότητες αλληλεπίδρασης και ανατροφοδότησης στους χρήστες. Δίνει τη δυνατότητα προσαρμογής των χρηστών στο ρυθμό εκπαίδευσης που επιθυμούν.

Σε ότι αφορά την ευκολία χρήσης, πλοήγησης και παρουσίασης το εκπαιδευτικό λογισμικό παρέχει το απαραίτητο γραφικό περιβάλλον στους χρήστες και τα γραφικά που χρησιμοποιούνται έχουν λειτουργικό χαρακτήρα. Το εκπαιδευτικό λογισμικό περιλαμβάνει μενού και παρέχει βοήθεια στους χρήστες, βοηθώντας έτσι στην ευκολία χρήσης και πλοήγησης.

Τέλος, σε ότι αφορά τα λειτουργικά χαρακτηριστικά το εκπαιδευτικό λογισμικό παρέχει στους χρήστες δυνατότητες αντιγραφής, επικόλλησης και αποθήκευσης. Επίσης η διαδικασία εγκατάστασης του εκπαιδευτικού λογισμικού γίνεται με απλό και εύκολο τρόπο. Τελευταίο αλλά και πολύ σημαντικό είναι ότι απουσιάζουν τα τεχνικά σφάλματα από τον κώδικα, είναι αξιόπιστο και λειτουργεί χωρίς προβλήματα.

Με την ανάπτυξη του εκπαιδευτικού λογισμικού επιτεύχθηκε η δημιουργία του εκπαιδευτικού λογισμικού. Σαν κομμάτια ενός πάζλ, καταφέραμε μέσα από την επιτυχία των επιμέρους στόχων –που αναφέρονται στο διδακτικό περιεχόμενο, στα παιδαγωγικά χαρακτηριστικά, στην ευκολία χρήσης, στα λειτουργικά χαρακτηριστικά– να συνενώσουμε τα κομμάτια αυτά μεταξύ τους, λαμβάνοντας στο τέλος τη συνολική εικόνα.

Εικόνα 1 Η επιτυχία των επιμέρους στόχων οδηγεί στην επιτυχία του βασικού στόχου[1].

[1] Το φόντο είναι από «HQ Wallpapers».

1 Εισαγωγή

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η δημιουργία εκπαιδευτικού λογισμικού κατάλληλου για την εκπαίδευση στη Διοίκηση Έργων.

Η Διοίκηση Έργων αναπτύχθηκε ως ξεχωριστό γνωστικό πεδίο από την εφαρμογή των αρχών της διοίκησης αλλά και της επιχειρησιακής έρευνας σε διάφορους τομείς εφαρμογής, όπως αυτός των κατασκευών, της μηχανολογίας και των μεγάλων στρατιωτικών προγραμμάτων (Kloppenborg, 2008). Το PMBoK Guide (PMI, 2004) καθορίζει ως έργο «*μια προσωρινή προσπάθεια που αναλαμβάνεται για να δημιουργήσει ένα μοναδικό προϊόν ή υπηρεσία*». Έτσι κι εμείς βάσει του ορισμού του έργου θέλουμε να αναπτύξουμε ένα εκπαιδευτικό λογισμικό το οποίο θα έχει ως βασικό στόχο την εκπαίδευση των χρηστών στη Διοίκηση Έργων.

Υπάρχουν διάφορα εργαλεία που χρησιμοποιούνται στη Διοίκηση Έργων όπως η μέθοδος κρισίμου δρόμου, η τεχνική αξιολόγησης και αναθεώρησης προγράμματος για τη διαχείριση χρόνου, η δομή ανάλυσης εργασιών για τη διαχείριση εύρους, ο πίνακας αρμοδιοτήτων για τη διαχείριση ανθρώπινου δυναμικού. Τα αναφερθέντα εργαλεία Διοίκησης Έργου συμπεριλαμβάνονται στο εκπαιδευτικό λογισμικό.

Σύμφωνα με τους Γαρυφαλλίδου και λοιποί, (2003), «*Το Εκπαιδευτικό λογισμικό είναι ένα υπολογιστικό πρόγραμμα το οποίο έχει κύριο στόχο τη διδασκαλία και τη μάθηση του αντικειμένου που πραγματεύεται στους χρήστες*». Και όπως μας λέει ο Johnson (2003), ο συνδυασμός καινοτομίας και εκπαιδευτικού λογισμικού δίνει το μεγαλύτερο όφελος στους χρήστες του. Σκοπός μας είναι η δημιουργία ενός πρωτότυπου εκπαιδευτικού λογισμικού για τη διδασκαλία των μεθόδων και τεχνικών Διοίκησης Έργου.

2 Μέθοδος Υλοποίησης Διπλωματικής Εργασίας

Η μεθοδολογία ανάπτυξης του εκπαιδευτικού λογισμικού που χρησιμοποιήθηκε ακολουθεί τα παρακάτω βήματα (Γαρυφαλλίδου, Ιωαννίδης, Σκέλλας, & Τσιτσίρης, 2003):

1. ***Καθορισμός του στόχου.*** *Ο στόχος πρέπει να είναι συγκεκριμένος, ξεκάθαρος και σαφής. Καθορίζουμε το κόστος του σχεδιαζόμενου εκπαιδευτικού λογισμικού, το χρόνο συγγραφής του και το χρόνο που θα θέλαμε να παραμείνει σε χρήση.* Επίσης προκύπτουν ορισμένα ερωτήματα όπως:

    i. *Πόσο χρήσιμο είναι το εκπαιδευτικό λογισμικό;*

    ii. *Υπάρχει άλλο παρόμοιο εκπαιδευτικό λογισμικό στην αγορά;*

    iii. *Για πόσο καιρό πρόκειται να χρησιμοποιηθεί;*

    iv. *Πως μπορεί το εκπαιδευτικό λογισμικό να καλύψει πιο γενικούς σκοπούς;*

    v. *Ποιοι είναι οι χρήστες που θα το χρησιμοποιήσουν (σε ποιους απευθύνεται);*

2. *Αφορά τη **συγκέντρωση της ύλης** που σχετίζεται με το αντικείμενο και αν το πρόγραμμα περιορίζεται στην ύλη του μαθήματος ή καλύπτει μια ευρύτερη περιοχή.*

3. ***Εκπαιδευτικός σχεδιασμός του προγράμματος.*** *Συνεργασία των προγραμματιστών (ειδικών αναλυτών) με τον/την καθηγητή/τρια («ειδικό» στο αντικείμενο). Η συνεργασία αυτή είναι πολύπλευρη και συγχρόνως καθοριστική για την αρτιότητα του σχεδιασμού του προγράμματος. Οι προγραμματιστές καθορίζουν ποια στοιχεία εντάσσονται στο πρόγραμμα ενώ ο/η «ειδικός» καθορίζει τον τρόπο με τον οποίο το αντικείμενο γίνεται κατανοητό στο χρήστη. Ο τελικός χρήστης είναι ο/η φοιτητής/τρια ο οποίος αποφασίζει αν του είναι χρήσιμο ή όχι κι έτσι θα το δεχτεί ή θα το απορρίψει. Επίσης η συμμετοχή του χρήστη είναι πολύ σημαντική διότι έτσι μπορούμε να δούμε αν μπορεί να χειριστεί το πρόγραμμα του εκπαιδευτικού λογισμικού, αν το βρίσκει χρήσιμο, αν ασχολείται με ενθουσιασμό με αυτό. Εδώ θα αποφασιστεί η μορφή του προγράμματος. Κακή συνεργασία μπορεί να οδηγήσει σε κακά αποτελέσματα όπως:*

i. Απλή αντιγραφή ενός βιβλίου με σκοπό τον έλεγχο «ξερών» γνώσεων.

ii. Το πρόγραμμα που θα αναπτυχθεί θα περιέχει λάθη.

iii. Απλή προβολή του αντικειμένου χωρίς να υπάρχει δυνατότητα ουσιαστικής αλληλεπίδρασης του χρήστη με το λογισμικό, κι έτσι το αντικείμενο να προβάλλεται ως μια βιντεοταινία χωρίς να μπορεί να προσφέρει οποιοδήποτε είδος μάθησης.

Αν το εκπαιδευτικό λογισμικό πρόκειται να χρησιμοποιηθεί για κατ' οίκον διδασκαλία πρέπει να δίνεται ιδιαίτερη σημασία στην αυτενέργεια του/της φοιτητή/τριας - διδασκόμενου, στην πληρότητα της προσφερόμενης πληροφορίας, στην απλότητα στη χρήση και τελικά λόγω του περιβάλλοντος που θα χρησιμοποιηθεί το πρόγραμμα στο να είναι το εκπαιδευτικό λογισμικό αρεστό και ελκυστικό.

Η μορφή που αποφασίζεται δεν περιγράφεται μόνο με λόγια που μπορεί να προκαλέσουν ασάφειες και παρερμηνείες αλλά και με τη μορφή DFD (data flow diagrams) όπου φαίνεται πώς θα πρέπει να αντιδρά κάθε φορά το πρόγραμμα.

4. ***Υλοποίηση του εκπαιδευτικού λογισμικού.*** *Μετατροπή από την ομάδα των προγραμματιστών – αναλυτών των αποφάσεων που λήφθηκαν στο 3ο βήμα της σχεδίασης του λογισμικού, σε γλώσσα υψηλού επιπέδου (γλώσσα προγραμματισμού όπως Visual C++, Visual Basic .net, Borland C κ.α.). Ανάπτυξη αρχικά ενός πρωτότυπου όπου θα δείχνει γενικά πως θα λειτουργεί το εκπαιδευτικό λογισμικό και στη συνέχεια ξεκινά η διαδικασία της κωδικοποίησης του λογισμικού.*

5. ***Αξιολόγηση του εκπαιδευτικού λογισμικού*** *από την ομάδα των «ειδικών» σε πραγματικές συνθήκες. Πρέπει στο βήμα αυτό να ελεγχθούν όλες οι λειτουργίες του εκπαιδευτικού λογισμικού και να εντοπιστούν όλα τα λογικά λάθη που μπορεί να υπάρχουν ή να προκύψουν.*

6. ***Μορφωτική αξιολόγηση του εκπαιδευτικού λογισμικού.*** *Δημιουργούμε δύο ομάδες: Η πρώτη (πειραματική ομάδα) διδάσκεται το αντικείμενο με παραδοσιακές μεθόδους σε συνδυασμό με τη χρήση του εκπαιδευτικού λογισμικού (μέσα από εργασίες για παράδειγμα). Η δεύτερη ομάδα (ομάδα ελέγχου) διδάσκεται μόνο με τη παραδοσιακή μέθοδο διδασκαλίας. Στο τέλος εξάγονται και μελετώνται οι διαφορές που προκύπτουν μεταξύ των δύο ομάδων. Κάτι που πρέπει να έχουμε υπ' όψιν μας,*

είναι να υπάρχει όσο μεγαλύτερη ομοιότητα μεταξύ των δύο ομάδων ως προς το πνευματικό επίπεδο των φοιτητών/τριών, τις γνώσεις και τη πείρα του διδάσκοντα ώστε οι διαφορές που πιθανόν να προκύψουν να οφείλονται στη χρήση του εκπαιδευτικού λογισμικού.

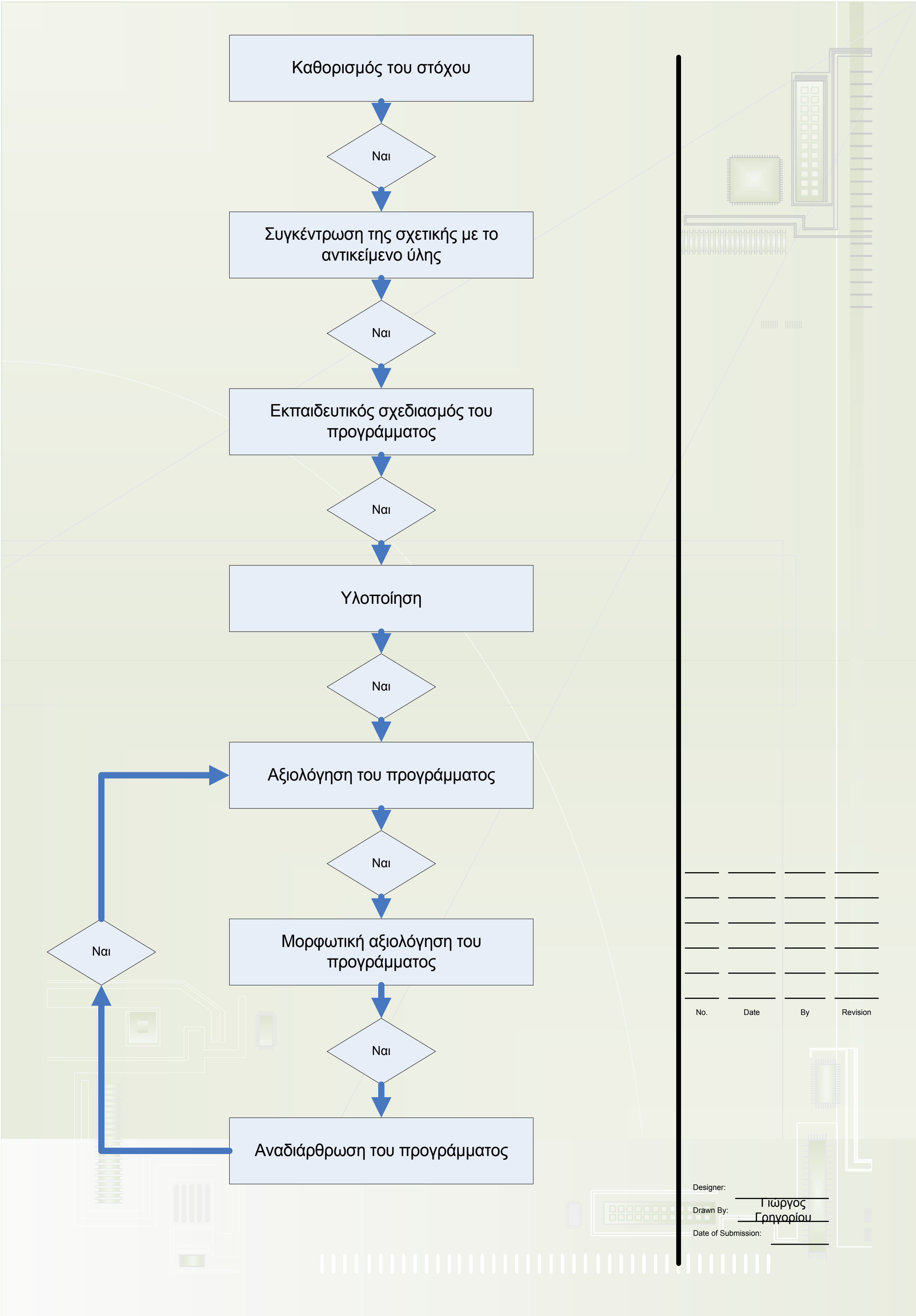
7. ***Αναδιάρθρωση του προγράμματος.*** *Με βάση το 6ο βήμα (Μορφωτική Αξιολόγηση) γίνονται οι προτάσεις για τη βελτίωση του εκπαιδευτικού λογισμικού.*

Αν το εκπαιδευτικό λογισμικό είναι σχεδιασμένο να λειτουργήσει πολλά χρόνια τότε ακολουθούν τα βήματα:

8. ***Αξιολόγηση****.*

9. ***Αναδιάρθρωση και έλεγχος****.*

Οι δύο αυτές φάσεις εναλλάσσονται για τη χρονική περίοδο που το εκπαιδευτικό λογισμικό βρίσκεται σε χρήση και λειτουργία συνήθως με ρυθμό μία φορά το χρόνο. Το εκπαιδευτικό λογισμικό καλό θα είναι να συνοδεύεται και από μια αναφορά που να λέει τον τρόπο με τον οποίο χρησιμοποιήθηκε και τεχνικές λεπτομέρειες έτσι ώστε να είναι χρήσιμο σε άλλα εκπαιδευτικά ιδρύματα.



Εικόνα 2 Διάγραμμα Ροής Δεδομένων της Μεθοδολογίας Ανάπτυξης του Εκπαιδευτικού Λογισμικού.

Η μέθοδος που εφαρμόστηκε για την υλοποίηση του εκπαιδευτικού λογισμικού βασίστηκε στη μεθοδολογία που προαναφέρθηκε και τις φάσεις του κύκλου ανάπτυξης λογισμικού όπως αναφέρονται από το National Institute of Open Schooling (2008). Τα βήματα της μεθόδου είναι τα εξής:

1. Αρχικά έγινε η μελέτη συστήματος όπου ξεκαθαρίστηκε ποιος είναι ο στόχος του εκπαιδευτικού λογισμικού. Στη μελέτη συστήματος έγινε και η σχετική έρευνα για παρόμοια εκπαιδευτικά λογισμικά όπως αυτό που θέλουμε να αναπτύξουμε.

2. Στη συνέχεια έγινε η ανάλυση του συστήματος όπου πραγματοποιήθηκε μια λεπτομερής μελέτη των διάφορων διαδικασιών που εκτελούνται από το σύστημα και των αναμεταξύ τους σχέσεων, μέσα και έξω από το σύστημα (National Institute of Open Schooling, 2008).

3. Μετά έγινε το σημαντικό κομμάτι του σχεδιασμού του συστήματος όπου σχεδιάστηκαν οι βασικές λειτουργίες του εκπαιδευτικού λογισμικού και η αρχιτεκτονική του συστήματος.

4. Ακολούθησε η κωδικοποίηση του συστήματος όπου η θεωρία έγινε πράξη.

5. Στη συνέχεια έγινε η δοκιμή του συστήματος όπου εντοπίστηκαν και διορθώθηκαν τα λογικά και τα συντακτικά λάθη.

6. Τέλος έγινε η αξιολόγηση του συστήματος.

3 Βιβλιογραφική ανασκόπηση

3.1 Εισαγωγή

Περιγράφονται στη συνέχεια το **Εκπαιδευτικό Λογισμικό** και η ιστορική του αναδρομή, τα ιστορικά στοιχεία της **Διοίκησης Έργων**, σε τι αναφέρεται η Διοίκηση Έργων και ποια είναι τα γνωστικά πεδία της. Τέλος αναλύονται τα εργαλεία της Διοίκησης Έργων και περιγράφονται με λεπτομέρεια τα εργαλεία που θα εντάξουμε στο εκπαιδευτικό λογισμικό.

3.2 Εκπαιδευτικό Λογισμικό

3.2.1 Εισαγωγή

Συγκρίνοντας το εκπαιδευτικό λογισμικό με τις παραδοσιακές μεθόδους διδασκαλίας προκύπτουν τα εξής πλεονεκτήματα (Γαρυφαλλίδου, Ιωαννίδης, Σκέλλας, & Τσιτσίρης, 2003):

- ***Με το εκπαιδευτικό λογισμικό δίνεται η δυνατότητα εξατομικευμένης διδασκαλίας.*** Ο υπολογιστής παίζει το ρόλο του προσωπικού «δασκάλου» του χρήστη όπου αναλαμβάνει το έργο της εκπαίδευσης του. Αρχικά εξηγεί τις σχετικές έννοιες, θεωρίες του αντικειμένου εκπαίδευσης, στην περίπτωση μας στη Διοίκηση Έργων. Μετά ο υπολογιστής – «δάσκαλος» εξετάζει αν το αντικείμενο έγινε κατανοητό από το χρήστη. Η εξέταση του χρήστη γίνεται με τη μορφή ασκήσεων και σεναρίων τα οποία επιδέχονται επίλυσης με συγκεκριμένες μεθόδους και τεχνικές (π.χ. Μέθοδος του Κρισίμου Δρόμου (CPM)). Το πλεονέκτημα που δίνει το εκπαιδευτικό λογισμικό στη περίπτωση αυτή είναι ότι ο χρήστης έχει ένα πλήρως αφοσιωμένο και προσωπικό «δάσκαλο».
- ***Επιλογή του χρόνου διδασκαλίας του αντικειμένου που περιλαμβάνει το εκπαιδευτικό λογισμικό.*** Στη παραδοσιακή μέθοδο διδασκαλίας ο εκπαιδευτικός παραδίδει το μάθημα και στη συνέχεια το εξετάζει με βάση κάποιο αυστηρό χρονοδιάγραμμα που πρέπει να ακολουθήσει. Με το εκπαιδευτικό λογισμικό δίνεται η δυνατότητα στο χρήστη να επιλέγει το χρόνο που αυτός θέλει ή μπορεί να διδαχθεί το αντικείμενο (στη περίπτωση μας, να διδαχθεί τις μεθόδους ή τεχνικές Διοίκησης Έργων).

- **Άμεση επίλυση και διόρθωση των λαθών.** Με τη παραδοσιακή μέθοδο διδασκαλίας ο εκπαιδευόμενος έπρεπε αφού εξεταζόταν σε ένα μάθημα να περιμένει κάποιο λογικό χρονικό διάστημα μέχρι να δει το αποτέλεσμα. Και το πρόβλημα είναι ότι πολλές φορές αυτό που ενδιαφέρει είναι μόνο το αποτέλεσμα. Σπάνια ο εκπαιδευόμενος να ανατρέξει στα λάθη που έκανε ή ο εκπαιδευτής να τα υποδείξει και να τα επισημάνει. Ο σκοπός της υπόδειξης και επισήμανσης δεν είναι άλλος από τη πλήρη κατανόηση του μαθήματος μέσα από τα λάθη. Τα εκπαιδευτικό λογισμικό με την άμεση διόρθωση και επίλυση των ασκήσεων και των σεναρίων, δίνει τη δυνατότητα στο χρήστη να βλέπει τα λάθη του άμεσα υποδεικνύοντας του το λόγο που έκανε λάθος και να του δείχνει το σωστό τρόπο επίλυσης. Δηλαδή δίνει στο χρήστη την ουσία κι όχι απλά το αποτέλεσμα.
- **Επανάληψη ίσον μάθηση.** Με την άνευ χρονικών περιορισμών διδασκαλία που προσφέρει το εκπαιδευτικό λογισμικό μαζί με την επανάληψη (όσες φορές χρειάζεται) των ασκήσεων και των σεναρίων, προσφέρεται στο χρήστη ένα πολύ σημαντικό πλεονέκτημα έναντι της παραδοσιακής μεθόδου διδασκαλίας. Στη παραδοσιακή διδασκαλία ο εκπαιδευτικός παραδίδει μια φορά το μάθημα της ημέρας και αναμένει από τον εκπαιδευόμενο να μπορεί να κατανοήσει πλήρως το μάθημα. Το εκπαιδευτικό λογισμικό κάνει το ίδιο μάθημα όσες φορές το απαιτήσει ο χρήστης και πάντα με το ίδιο τρόπο.

3.2.2 Ιστορική αναδρομή

Τα προγράμματα λογισμικού μεταφράζουν τις δυνατότητες του υπολογιστή σε αποτελέσματα. Τα καινοτόμα προγράμματα λογισμικά προσφέρουν καινούρια και απρόσμενα αποτελέσματα. Το εκπαιδευτικό λογισμικό έχει την επιπρόσθετη ευθύνη να ερεθίζει τη νοητική και γνωστική ανάπτυξη του χρήστη. Έτσι λοιπόν, καταλήγουμε στο συμπέρασμα ότι το πιο δύσκολο είδος προγράμματος λογισμικού προς ανάπτυξη αλλά παράλληλα κι αυτό που ανταμείβει περισσότερο είναι ο συνδυασμός του καινοτόμου και του εκπαιδευτικού λογισμικού (Johnson, 2003).

Η ανάπτυξη εκπαιδευτικών λογισμικών συνδέθηκε περισσότερο με τους εκπαιδευτικούς και με την έρευνα τους. Κατά τη δεκαετία του 1960, το πανεπιστήμιο της Γιούτα (Utah), το πανεπιστήμιο του Στάνφορντ (Stanford), το πανεπιστήμιο του Ιλινόις (Illinois), και το MIT ήταν οι πρώτες θερμοκοιτίδες για τη δημιουργία μιας σταθερής και στέρεας βάσης για τα εκπαιδευτικά λογισμικά. Οι αναλυτές – προγραμματιστές δημιουργούσαν λογισμικά βασιζόμενα στην εκπαιδευτική έρευνα και δούλευαν συνεχώς με τους φοιτητές και τους καθηγητές έτσι ώστε να εξετάσουν τα αποτελέσματα τους. Οι τέσσερις αυτές ομάδες καθιέρωσαν τη σημαντικότητα της απόδειξης της αποτελεσματικότητας οποιουδήποτε λογισμικού προτού καταστεί έτοιμο για αγορά. Κάποια πρώτα εκπαιδευτικά λογισμικά είχαν το όνομα της εταιρίας που τα δημιούργησε όπως: WICAT (World Institute for Computer-Assisted Teaching), CCC (Computer Curriculum Corporation), PLATO (Programmed Logic for Automated Teaching Outcomes),και η Logo (Johnson, 2003).

Το CCC λογισμικό χρησιμοποιούσε πολύ καλά σχεδιασμένες ερωτήσεις και με την επεξεργασία των απαντήσεων που έδιναν οι φοιτητές καθόριζε ποια προβλήματα θα έθετε στη συνέχεια. Αυτό το σύστημα είναι ένα προς ένα σύστημα διδασκαλίας με το φοιτητή, όπου ο εκπαιδευτής παρέχει εξάσκηση για δύσκολες να κατανοηθούν έννοιες και ο φοιτητής προχωρά αφού επιλύσει με επιτυχία τα προβλήματα (Johnson, 2003).

Το PLATO μέχρι το 1987 παρείχε περισσότερες από 12,000 ώρες οδηγιών σε περισσότερα από 100 θέματα με μεγάλο εύρος ποιοτικών και παιδαγωγικών προσεγγίσεων. Το απλό σύστημα σημειώσεων του PLATO για τη επικοινωνία, σε ότι αφορούσε τα προβλήματα του λογισμικού, ήταν ο προπομπός του σημερινού ηλεκτρονικού μηνύματος (e-mail) (Johnson, 2003).

Το MIT ήταν το σπίτι του Seymour Papert, των έξυπνων ιδεών του και μιας χελώνας που αντιδρούσε σε εντολές. Ανέπτυξε τη γλώσσα προγραμματισμού, LOGO. Στη καρδιά της Logo βρισκόταν ο τρόπος σκέψης και μάθησης των παιδιών. Ήθελε να η LOGO να είναι προσβάσιμη και εύκολη για τα παιδιά. Τη χρησιμοποίησε για να διδάξει μαθηματικά διδάσκοντας τη LOGO σε ενδιαφέροντα θέματα όπως μουσική και φυσική. Η LOGO έγινε η υπολογιστική γλώσσα του δημοτικού σχολείου (Papert, 1980).

Αργότερα, ο Seymour Papert επέκτεινε τη LOGO ώστε να δουλεύει με τη LEGO. Η προσέγγιση της κατασκευής μέσα από τη LEGO, εμφάνιζε τη μάθηση ως επαναδημιουργία της γνώσης. Ο Seymour Papert ισχυρίστηκε ότι η μάθηση είναι πιο αποτελεσματική όταν ο μαθητευόμενος κατασκευάζει ένα νοήμων προϊόν. Με τη χρήση της LEGO ο εκπαιδευόμενος μαθαίνει να καθορίζει ένα πρόβλημα και να αναπτύσσει τις απαραίτητες ικανότητες για την επίλυση του (Harel & Seymour, 1991).

Τις δεκαετίες του 1970 και 1980 σχεδόν όλοι οι υπολογιστές πωλούνταν με ενσωματωμένη κάποια έκδοση της γνωστής γλώσσας προγραμματισμού BASIC. Έτσι η BASIC έγινε ένα εργαλείο προς μάθηση κι ένα προγραμματιστικό περιβάλλον για τη δημιουργία λογισμικών από τους φοιτητές. Κάθε πρόγραμμα ήταν τόσο καλό όσο ο προγραμματιστής του που στις πλείστες περιπτώσεις ήταν ερασιτέχνης (είχε το προγραμματισμό ως χόμπι) κι όχι επαγγελματίας. Και ακόμη χειρότερα, ένα πρόγραμμα το οποίο γραφόταν σε ένα υπολογιστή, συνήθως έτρεχε μόνο σε αυτόν και σε κανένα άλλο. Η MECC άνοιξε το δρόμο για την επίλυση του προβλήματος και μεταξύ 1980 και αρχές 1990, δόθηκε άδεια στα σχολεία να μπορούν να αντιγράφουν και να δημιουργούν δισκέτες και αντίγραφα (Johnson, 2003).

Μεταξύ 1978 και 1983, δημιουργήθηκαν μερικά από τα καλύτερα δημιουργικά προγράμματα για φοιτητές. Τα περισσότερα προγράμματα αφορούσαν την επίλυση προβλημάτων και έκαναν τη βασική υπόθεση ότι οι φοιτητές ανέπτυσσαν τις ικανότητες σκέψης και είχαν μια απολαυστική εμπειρία. Οι εκπαιδευτικοί είδαν το ενδεχόμενο η τεχνολογία να συμβάλλει στη δημιουργική σκέψη των εκπαιδευόμενων κι έτσι δημιουργήθηκαν αξιόλογα προγράμματα τη περίοδο εκείνη (Johnson, 2003).

Ο Wallace Feurzeig, ανάπτυξε ένα πρόγραμμα άλγεβρας το οποίο είχε τη νοημοσύνη να επιλύει προβλήματα άλγεβρας όπως καθορίζονταν από τον φοιτητή. Ο σκοπός ήταν η δημιουργία ενός πλούσιου για εξερεύνηση περιβάλλοντος και η εισαγωγή έξυπνων υπολογιστικών ικανοτήτων και εργαλείων έτσι ώστε να δοθεί έμφαση στη ποιοτική λογική και στη λύση προβλημάτων (Feurzeig & John, 1988).

Το επόμενο βήμα ήταν η ωρίμανση των εκπαιδευτικών εργαλείων. Τα περισσότερα εκπαιδευτικά εργαλεία ήταν απλά και πολύ συχνά θεωρούνταν ως εργαλεία του καθηγητή κι όχι του φοιτητή. Αυτά τα εργαλεία επέτρεπαν στο καθηγητή να τροποποιεί το περιεχόμενο του εκπαιδευτικού λογισμικού. Τα εργαλεία που ωρίμασαν και έγιναν επιτυχία ήταν αυτά που παρείχαν αυθεντικές εμπειρίες. Παρείχαν νέους τρόπους εκμάθησης στους φοιτητές, έκφρασης της γνώσης τους, και επέκτασης της σκέψης τους. Μερικά από αυτά τα εργαλεία είναι τα εξής: Kid Pix, HyperStudio, TimeLiner, Inspiration, και Geometer's Sketchpad (Johnson, 2003).

Επιπλέον, ο Stanley (2008), μαζί με μια διεθνή ομάδα ερευνητών, που διενεργούν έρευνα πάνω στη χαώδη και κλασματική γεωμετρία, δημιούργησαν μοντέλα για φοιτητές σε ένα υπέρ-υπολογιστή. Με τη βοήθεια αυτών των φοιτητών, δημιούργησαν μαθήματα τα οποία μπορούν να κατέβουν σε κανονικούς υπολογιστές ώστε να χρησιμοποιηθούν για την εκπαίδευση. Μέσω των δικτύων των υπολογιστών μεταφέρονται τα μοντέλα και οι έννοιες στα σχολεία. Έτσι, οι νέες θεωρίες παρουσιάζονται σε όλα τα επίπεδα της εκπαίδευσης με ένα συνεκτικό τρόπο (Stanley, 2008).

Οι εργάτες σε πολλές βιομηχανίες χρειάζονται ένα νέο και διαφορετικό τρόπο τύπο εκπαίδευσης. Η Barbara Means του SRI International πειραματίζεται με την έννοια της απομακρυσμένης διδασκαλίας ως ένα εναλλακτικό εργαλείο διδασκαλίας το οποίο μπορεί να γεφυρώσει το κενό μεταξύ εννοιών που τονίζονται στην εκπαίδευση και τις δυνατότητες που προσφέρονται στο εργασιακό περιβάλλον. Έχει αναπτύξει συστήματα διδασκαλίας και συμβουλευτικής για χρήση σε υψηλής τεχνολογίας βιομηχανίες (Means, 2008).

Πολύ σημαντικό είναι τα εκπαιδευτικά εργαλεία να συνοδεύονται από υλικό οδηγιών χρήσης. Η φιλοσοφία των εκπαιδευτικών εργαλείων είναι να δίνουν τη δυνατότητα στους φοιτητές να σκέφτονται διαφορετικά με βάση τα εργαλεία που έχουν στη διάθεση τους (Johnson, 2003).

3.3 Διοίκηση Έργων

3.3.1 Ιστορικά Στοιχεία

Τα πρώτα έργα κατασκευής περιλάμβαναν την οικοδόμηση των πυραμίδων, των πόλεων, των μεσαιωνικών καθεδρικών ναών, και άλλων. Ενώ τα αποτελέσματα αυτών των έργων ήταν συχνά εντυπωσιακά, το ανθρώπινο κόστος και ο χρόνος να ολοκληρωθούν ήταν υπερβολικός (Kloppenborg, 2008).

Στον 20ό αιώνα, οι διαχειριστές έργου έπρεπε να μάθουν να διαχειρίζονται τις δαπάνες και το χρονικό προγραμματισμό πολύ καλύτερα, επιτυγχάνοντας τα επιθυμητά αποτελέσματα. Στη δεκαετία του '50 και τη δεκαετία του '60 αναπτύχθηκαν τεχνικές για και τη Διοίκηση Έργων, ξεκινώντας από τα τεράστια διαστημικά έργα. Το 1969, δημιουργήθηκε το Project Management Institute (PMI) με σκοπό να εκσυγχρονίσει και να επισημοποίηση τα εργαλείων τις τεχνικές και μεθόδους Διοίκησης Έργων. Ένα μεγάλο μέρος αυτής της ανάπτυξης βασίστηκε στην έννοια του καθορισμού των σχέσεων προτεραιότητας (δηλαδή προσδιορισμός των δραστηριοτήτων που πρέπει να ολοκληρωθούν πριν από άλλες δραστηριότητες) (Bista, 2008).

Η σύγχρονη Διοίκηση Έργου βασίζεται πάνω σε αυτό το καθορισμό της προτεραιότητας. Στη δεκαετία του '80 και τη δεκαετία του '90 πολλές εταιρίες λογισμικού πρόσφεραν ισχυρότερους και ευκολότερους τρόπους προγραμματισμό και ελέγχου των δαπανών και των χρονικών περιορισμών του έργου. Οι βιομηχανίες πληροφορικής και τηλεπικοινωνιών έχουν τροφοδοτήσει την ογκώδη ανάπτυξη και χρήση της Διοίκησης Έργου στη δεκαετία του '90. Σήμερα η Διοίκηση Έργου χρησιμοποιείται ευρέως σε όλους τους κλάδους των βιομηχανιών. Επίσης, με την αυξανόμενη τεχνολογική πρόοδο, πολλές εταιρίες χρησιμοποιούν τη Διοίκηση Έργων ως μια βασική δύναμη ολοκλήρωσης των έργων τους (Kloppenborg, 2008).

Οι επιχειρήσεις βρίσκονται αντιμέτωπες με τις προκλήσεις πιο σύνθετων προϊόντων και υπηρεσιών και απαιτούν υψηλότερα ποιοτικά αποτελέσματα. Αυτές οι προκλήσεις αναγκάζουν τις επιχειρήσεις να αναπτύσσουν γρήγορα και ανέξοδα σύνθετες λύσεις. Η Διοίκηση Έργου έχει ως σκοπό να βοηθήσει τις επιχειρήσεις να επιτύχουν στις προκλήσεις της σύγχρονης εποχής της πληροφορίας (Bista, 2008).

3.3.2 Τι είναι Διοίκηση Έργων

Το PMBoK Guide (PMI, 2004) καθορίζει ως έργο «*μια προσωρινή προσπάθεια που αναλαμβάνεται για να δημιουργήσει ένα μοναδικό προϊόν ή υπηρεσία*» και δηλώνει ότι αν και τα έργα ποικίλουν αρκετά σε τύπο και κλίμακα, είναι ένα εργαλείο στην υπηρεσία του οργανισμού για την επιτυχία του στρατηγικού σχεδίου. ***Η Διοίκηση Έργων είναι η εφαρμογή γνώσης, ικανοτήτων, εργαλείων και τεχνικών στις δραστηριότητες του έργου ώστε να επιτευχθούν οι στόχοι του έργου***.

Όλα τα έργα εκτελούνται για να επιτύχουν ένα συγκεκριμένο επιθυμητό αποτέλεσμα. Αυτό το αποτέλεσμα μπορεί να περιγραφεί κι από το ποσό, το μέγεθος, ή τα χαρακτηριστικά απόδοσής του έργου και από το πόσο καλά εξυπηρετεί το σκοπό και ευχαριστεί τους πελάτες του (ποιότητα του έργου).

Τα έργα αποτελούνται από διεργασίες. Μια διεργασία είναι μια ακολουθία πράξεων που ολοκληρώνει ένα αποτέλεσμα. Οι διεργασίες ενός έργου εκτελούνται από τους ανθρώπους και ανήκουν σε δύο κατηγορίες. Η πρώτη κατηγορία ονομάζεται **Διεργασίες Διοίκησης Έργου** όπου περιγράφονται και οργανώνονται οι εργασίες του έργου. Η δεύτερη κατηγορία ονομάζεται **Διεργασίες Προσανατολισμένες στο Προϊόν** όπου καθορίζεται και δημιουργείται το προϊόν του έργου. Οι διεργασίες αυτές καθορίζονται από το κύκλο ζωής του έργου και ποικίλουν ανάλογα με τη περιοχή εφαρμογής, (PMI, 2004).

Οι διεργασίες διοίκησης έργου μπορούν να οργανωθούν σε πέντε ομάδες μίας ή περισσοτέρων διεργασιών η κάθε μια. Οι ομάδες αυτές είναι, (PMI, 2004):

- Διεργασίες εκκίνησης.
- Διεργασίες σχεδιασμού.
- Διεργασίες εκτέλεσης.
- Διεργασίες ελέγχου.
- Διεργασίες κλεισίματος.

Οι ομάδες αυτές συνδέονται με τα αποτελέσματα που παράγουν. Η έξοδος ή το αποτέλεσμα μίας από αυτές μπορεί να γίνει είσοδος σε μια άλλη. Οι ομάδες δεν είναι διακριτές ούτε γίνονται μόνο μία φορά αλλά είναι επικαλυπτόμενες και μπορεί να συμβούν με διαφορετικά επίπεδα έντασης σε διαφορετικές φάσεις του έργου. Επίσης, οι ομάδες των διεργασιών μπορεί να είναι επαναλαμβανόμενες (PMI, 2004).

3.3.3 Γνωστικά Πεδία Διοίκησης Έργου

Σύμφωνα με το Project Management Institute (PMI, 2004), τα γνωστικά πεδία Διοίκησης Έργου είναι η διαχείριση:

- Ενοποίησης
- Εύρους
- Χρόνου
- Κόστους
- Ποιότητας
- Ανθρώπινων Πόρων
- Επικοινωνίας
- Κινδύνου
- Προμηθειών

Η διαχείριση ενοποίησης αφορά τις διεργασίες που απαιτούνται ώστε να επιβεβαιωθεί ότι τα στοιχεία ενός έργου συντονίζονται κατάλληλα.

Η διαχείριση εύρους αφορά το σχεδιασμό, το καθορισμό, την επιβεβαίωση του εύρους του έργου και τον έλεγχο.

Η διαχείριση χρόνου αφορά το καθορισμό των δραστηριοτήτων, το καθορισμό της αλληλουχίας μεταξύ των δραστηριοτήτων, την εκτίμηση της διάρκειας, την ανάπτυξη του προγράμματος και τον έλεγχο.

Η διαχείριση κόστους αφορά το προγραμματισμό των πόρων, την εκτίμηση του κόστους, το προϋπολογισμό και τον έλεγχο.

Η διαχείριση ποιότητας αφορά το σχεδιασμό, τη διασφάλιση και τον έλεγχο της ποιότητας.

Η διαχείριση ανθρώπινων πόρων αφορά την οργάνωση, τη «στρατολόγηση», την ανάπτυξη της ομάδας και την επίλυση των προβλημάτων.

Η διαχείριση επικοινωνίας αφορά το σχεδιασμό της επικοινωνίας, τη διανομή της πληροφορίας και την αναφορά της προόδου.

Η διαχείριση κινδύνου αφορά το σχεδιασμό διαχείρισης κινδύνου, τον εντοπισμό και την ανάλυση, την αντιμετώπιση και τον έλεγχο των κινδύνων.

Τέλος, η διαχείριση των προμηθειών αφορά την ανάλυση των αναγκών, το σχεδιασμό των διαδικασιών των προμηθειών, την επιλογή του προμηθευτή.

3.4 Ανάλυση Εργαλείων που χρησιμοποιούνται στη Διοίκηση Έργων

3.4.1 Εισαγωγή

Οι εννέα γνωστικές περιοχές της Διοίκησης Έργων περιλαμβάνουν συγκεκριμένα εργαλεία, μεθόδους και τεχνικές για την επίλυση των προβλημάτων που προκύπτουν.

Ξεκινώντας από τη διαχείριση ενοποίησης, τα εργαλεία που χρησιμοποιούνται είναι τα εξής, (PMI, 2004):

- Μεθοδολογία Προγραμματισμού Έργου.
- Πληροφοριακό Σύστημα Διαχείρισης Έργου.

Από τη διαχείριση εύρους, το εργαλείο που χρησιμοποιείται είναι η Δομή Ανάλυσης Εργασιών (WBS – Work Breakdown Structure), (PMI, 2006).

Από τη διαχείριση χρόνου, τα εργαλεία που χρησιμοποιούνται είναι τα εξής:

- Μέθοδος Κρισίμου Δρόμου (CPM – Critical Path Method), (O'Brien & Plotnick, 1999).
- Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique), (PMI, 2004).
- Διαγράμματα Gantt, (PMI, 2004).
- Δείκτες (SPI), (PMI, 2004).

Από τη διαχείριση κόστους, τα εργαλεία που χρησιμοποιούνται είναι τα εξής, (PMI, 2004):

- Ταμειακές Ροές (Cash Flow).
- Δείκτες (CPI).

Από τη διαχείριση ποιότητας, τα εργαλεία που χρησιμοποιούνται είναι τα εξής, (PMI, 2004):

- Συγκριτική αξιολόγηση (Benchmarking).
- Σχεδιασμός μέσω διαγραμμάτων ροής (Flowcharting).

Από τη διαχείριση ανθρώπινων πόρων, τα εργαλεία που χρησιμοποιούνται είναι:

- τον Πίνακα Αρμοδιοτήτων (RACI – Responsible Accountable Consult Informed), (Smith & Erwin, 2008) και
- η Εξισορρόπηση Πόρων. Σύμφωνα με το Project Management Institute, (PMI, 2004), η τεχνική της εξισορρόπηση πόρων, στοχεύει στην εξομάλυνση του αποθέματος των πόρων, στη μείωση των αποθεμάτων και στη πρόληψη των ελλείψεων.

Από τη διαχείριση επικοινωνίας, τα εργαλεία που χρησιμοποιούνται είναι τα εξής, (PMI, 2004):

- Ανάλυση μετόχων.
- Πληροφοριακά Συστήματα Ανάκτησης.
- Πληροφοριακά Συστήματα Διανομής.

Από τη διαχείριση κινδύνου, τα εργαλεία που χρησιμοποιούνται είναι :

- τα Φύλλα Κινδύνων, (Κηρυττόπουλος & Διαμάντας, 2006).
- ο Πίνακας Κινδύνου (Risk Matrix), (Κηρυττόπουλος & Διαμάντας, 2006).
- η προσομοίωση Monte Carlo, (PMI, 2004).

Από τη διαχείριση προμηθειών, το εργαλείο που χρησιμοποιείται είναι τα εξής, (PMI, 2004):

- Ανάλυση φτιάξε-ή-αγόρασε, (make-or-buy analysis).
- Επιλογή τύπου συμβολαίου.

Από αυτά τα γνωστικά πεδία εντάξαμε στο εκπαιδευτικό λογισμικό εργαλεία που αφορούν τη διαχείριση χρόνου, τη διαχείριση εύρους και τη διαχείριση ανθρώπινων πόρων. Συγκεκριμένα από τη διαχείριση χρόνου εντάξαμε τη **Μέθοδο Κρισίμου Δρόμου (CPM)** και τη **Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT)**. Από τη διαχείριση εύρους τη **Δομή Ανάλυσης Εργασιών (WBS)** και από τη διαχείριση ανθρώπινων πόρων τον **Πίνακα Αρμοδιοτήτων (RACI)**. Στη συνέχεια αναλύεται κάθε μια από τις μεθόδους και τεχνικές που θα συμπεριλαμβάνεται στο εκπαιδευτικό λογισμικό.

3.4.2 Μέθοδος Κρισίμου Δρόμου (CPM - Critical Path Method)

3.4.2.1 Εισαγωγή

Η **Μέθοδος του Κρισίμου Δρόμου (CPM – Critical Path Method)**, είναι ένας αλγόριθμος που βασίζεται στα μαθηματικά, για το σχεδιασμό του συνόλου των δραστηριοτήτων ενός έργου. Είναι ένα σημαντικό εργαλείο για την αποτελεσματική Διοίκηση Έργων (PMI, 2004).

Αναπτύχθηκε τη δεκαετία του '50 σε μια κοινοπραξία μεταξύ της εταιρίας DuPont και της εταιρίας Remington για τη Διοίκηση Έργων συντήρησης εγκαταστάσεων. Σήμερα, χρησιμοποιείται ευρέως με όλες τις μορφές έργων, συμπεριλαμβανομένης της κατασκευής, της ανάπτυξης λογισμικών, των ερευνητικών έργων, της ανάπτυξης προϊόντων, της εφαρμοσμένης μηχανικής και άλλων μορφών έργων. Οποιοδήποτε έργο με αλληλοεξαρτώμενες δραστηριότητες μπορεί να εφαρμόσει αυτήν την μέθοδο σχεδιασμού (O'Brien & Plotnick, 1999).

Η Μέθοδος του Κρισίμου Δρόμου (CPM), αναπτύχθηκε συγκεκριμένα για το σχεδιασμό και χρονικό προγραμματισμό των κατασκευών. Ένα από τα σημεία κλειδιά για την επιτυχία της Μεθόδου του Κρισίμου Δρόμου (CPM – Critical Path Method) είναι ότι εκμεταλλεύεται πλήρως τη γνώση του σχεδιαστή (planner) και την εμπειρία του. Με τη χρήση της Μεθόδου του Κρισίμου Δρόμου (CPM) μεθόδου μπορούμε να εξοικονομήσουμε χρόνο και χρήμα μέσω ενός καλύτερου σχεδιασμού (O'Brien & Plotnick, 1999).

Υπάρχουν δύο βασικές προσεγγίσεις σε ότι αφορά τη Μέθοδο του Κρισίμου Δρόμου (CPM). Η παλαιότερη η οποία αναπτύχθηκε από τις εταιρίες DuPont και Remington τη δεκαετία του '50 και δε χρησιμοποιείται σήμερα είναι η **Δραστηριότητα στο Βέλος** (**Activity on Arrow**) και η νεότερη η οποία χρησιμοποιείται από όλους σήμερα η **Δραστηριότητα στον Κόμβο** (**Activity on Node**) (Maxwell, 2003). Στην ***Εικόνα 3 Παράδειγμα ενός CPM Διαγράμματος*** χρησιμοποιείται η προσέγγιση **Δραστηριότητα σε Κόμβο** (**Activity on Node**). Βλέπουμε ξεκάθαρα ότι όλα τα στοιχεία της δραστηριότητας (όνομα, διάρκεια, νωρίτερη και αργότερη έναρξη) περιέχονται μέσα στο κόμβο και το βέλος δείχνει το δρόμο που ορίζεται από την ένωση των κόμβων μεταξύ τους.

3.4.2.2 Εφαρμογή

Τα βήματα που ακολουθούμε για το σχεδιασμό του διαγράμματος της Μεθόδου του Κρισίμου Δρόμου (CPM) είναι τα ακόλουθα:

1. Καθορίζουμε τις δραστηριότητες: Κάνουμε ένα κατάλογο με όλες τις δραστηριότητες του έργου.
2. Καθορίζουμε την ακολουθία των δραστηριοτήτων: Δίπλα από κάθε δραστηριότητα, σε άλλη στήλη, βάζουμε τις δραστηριότητες που προηγούνται, δηλαδή αυτές που πρέπει να ολοκληρωθούν για να ξεκινήσουν οι επόμενες.
3. Σχεδιάζουμε το διάγραμμα του δικτύου.
4. Υπολογίζουμε το χρόνο ολοκλήρωσης για κάθε δραστηριότητα: Είναι ο χρόνος που απαιτείται για την ολοκλήρωση κάθε δραστηριότητας ο οποίος προκύπτει από υπολογισμό βάσει της εμπειρίας.
5. Επιλύουμε το δίκτυο.
6. Καθορίζουμε τον κρίσιμο δρόμο: Είναι ο μακρύτερος σε διάρκεια δρόμος που διαπερνά το δίκτυο. Η σημαντικότητα του κρίσιμου δρόμου είναι ότι οι δραστηριότητες που βρίσκονται πάνω του δε μπορούν να καθυστερήσουν χωρίς να καθυστερήσει το έργο.
7. Εάν μας ενδιαφέρει η παρακολούθηση του έργου, ενημερώνουμε το διάγραμμα της Μεθόδου του Κρισίμου Δρόμου (CPM) καθώς το έργο προχωρά: Καθώς το έργο προχωρά ο πραγματικός χρόνος ολοκλήρωσης των δραστηριοτήτων γίνεται γνωστός και το διάγραμμα δικτύου μπορεί να ενημερωθεί. Έτσι ορίζεται κάθε φορά ένας νέος κρίσιμος δρόμος (ή κρίσιμη διαδρομή).

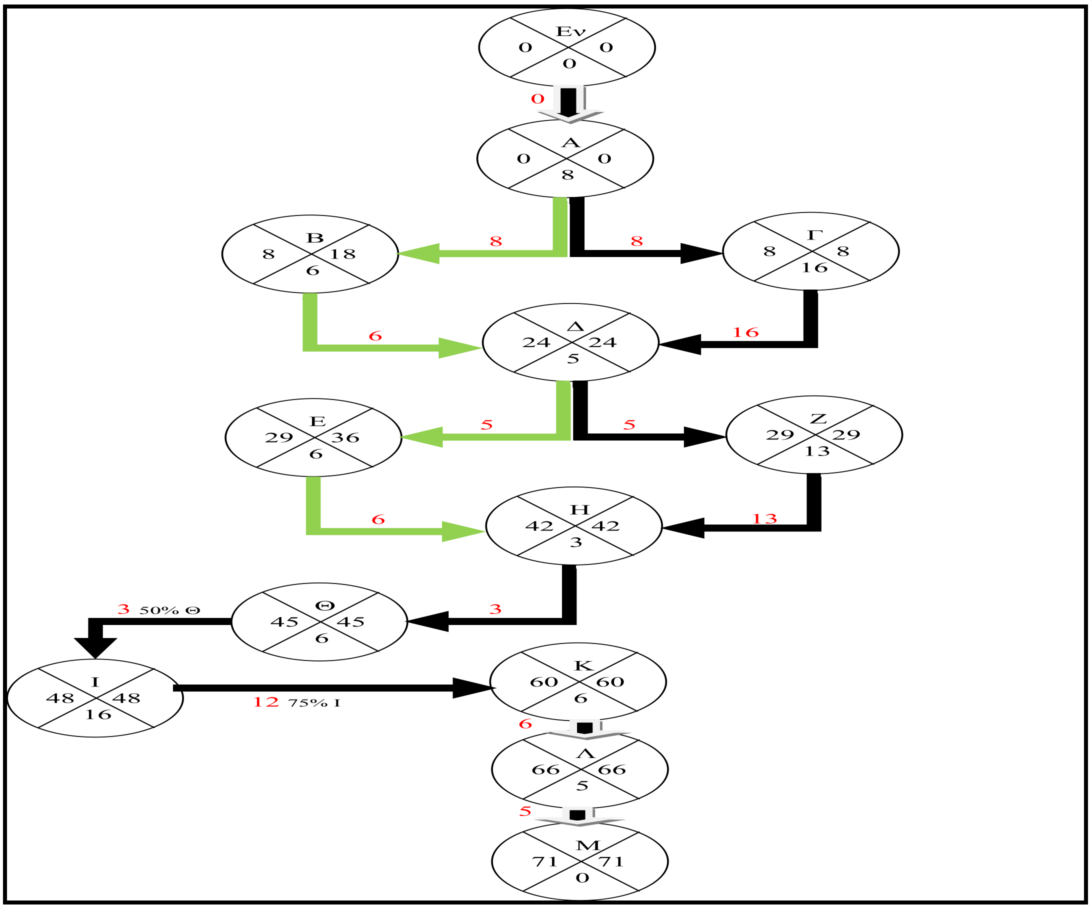
Ο κρίσιμος δρόμος ορίζεται με τον καθορισμό των ακόλουθων τεσσάρων παραμέτρων για κάθε δραστηριότητα:

- Νωρίτερη Έναρξη: ο νωρίτερος χρόνος που μπορεί να ξεκινήσει μια δραστηριότητα δεδομένου ότι η προηγούμενη δραστηριότητα ολοκληρώθηκε.
- Νωρίτερη Λήξη: ο νωρίτερος χρόνος λήξης, ίσος με το νωρίτερο χρόνο έναρξης συν το χρόνο που απαιτείται για την ολοκλήρωση της δραστηριότητας.

- Αργότερη Λήξη: ο αργότερος χρόνος στον οποίο μπορεί να τελειώσει η δραστηριότητα χωρίς να καθυστερήσει το έργο.
- Αργότερη Έναρξη: ο αργότερος χρόνος έναρξης, ίσος με τον αργότερο χρόνο λήξης μείον το χρόνο που απαιτείται για την ολοκλήρωση της δραστηριότητας.

Το συνολικό περιθώριο (total slack ή total float) για μια δραστηριότητα είναι ο χρόνος μεταξύ της νωρίτερης και αργότερης έναρξης, ή μεταξύ της νωρίτερης και αργότερης λήξης. Το συνολικό περιθώριο είναι ο χρόνος που μπορεί μια δραστηριότητα να καθυστερήσει χωρίς να καθυστερήσει το έργο.

Έτσι λοιπόν ο κρίσιμος δρόμος ορίζεται μέσα από τις δραστηριότητες του έργου όπου καμιά από αυτές δεν έχει συνολικό περιθώριο. Καθυστέρηση στον κρίσιμο δρόμο σημαίνει καθυστέρηση στο έργο. Στην ***Εικόνα 3***, ο κρίσιμος δρόμος είναι αυτό που ορίζεται ακολουθώντας τα μαύρα βέλη.



Εικόνα 3 Παράδειγμα ενός CPM Διαγράμματος

3.4.3 Τεχνική αξιολόγησης και αναθεώρησης προγράμματος (PERT – Program Evaluation and Review Technique)

3.4.3.1 Εισαγωγή

Η **Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique)**, είναι μια μέθοδος ανάλυσης των δραστηριοτήτων που χρειάζονται για την ολοκλήρωση ενός έργου και ειδικότερα του χρόνου που απαιτείται για την ολοκλήρωση κάθε δραστηριότητας, και του χρόνου που απαιτείται, στοχαστικά για την ολοκλήρωση του συνολικού έργου (PMI, 2004).

Το μοντέλο αυτό εφευρέθηκε από την εταιρία Booz Allen Hamilton, στο πλαίσιο της σύμβασης με το υπουργείο άμυνας των Ηνωμένων Πολιτειών και συγκεκριμένα του ειδικού γραφείου προγραμμάτων του ναυτικού το 1958, ως τμήμα του έργου του βαλλιστικού βλήματος του υποβρυχίου Polaris. Αυτό το έργο ήταν μια άμεση απάντηση στην κρίση που δημιουργήθηκε με την εκτόξευση του Σπούτνικ στο διάστημα από τους Ρώσους. Μερικές από τις συμβάσεις της αμερικανικής κυβέρνησης απαιτούσαν ότι η Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique) θα χρησιμοποιείτο ως τμήμα της διοικητικής επίβλεψης (PMI, 2004).

Η Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique) αναπτύχθηκε αρχικά με σκοπό να απλοποιήσει τον προγραμματισμό και σχεδιασμό των μεγάλων και σύνθετων έργων. Ήταν σε θέση να ενσωματώσει την αβεβαιότητα κάνοντας δυνατό το σχεδιασμό ενός έργου μη γνωρίζοντας ακριβώς τις λεπτομέρειες και τις διάρκειες όλων των δραστηριοτήτων. Είναι περισσότερο μια προσανατολισμένη-σε-γεγονός παρά προσανατολισμένη-σε-αρχή-και ολοκλήρωση τεχνική, και χρησιμοποιείται περισσότερο στα R&D έργα όπου ο χρόνος, έναντι του κόστους, είναι ο σημαντικότερος παράγοντας (PMI, 2004).

3.4.3.2 Ορισμοί

- Ο ***αισιόδοξος χρόνος*** (**O**ptimistic) είναι ο ελάχιστος χρόνος που απαιτείται για την ολοκλήρωση μιας δραστηριότητας.
- Ο ***απαισιόδοξος χρόνος*** (**P**essimistic) είναι ο μέγιστος χρόνος που απαιτείται για την ολοκλήρωση μιας δραστηριότητας. Αποκλείονται μέγιστες τιμές που οφείλονται σε σημαντικούς διακριτούς κινδύνους.

- Ο ***πιο πιθανός χρόνος*** (**M**ost Likely Time / Mode) είναι ο συνήθης χρόνος που απαιτείται για την ολοκλήρωση μιας δραστηριότητας.
- Ο ***αναμενόμενος χρόνος*** *ή* ***μέση τιμή*** (**E**xpected **T**ime / Mean) είναι η εκτίμηση της μέσης τιμής του χρόνου που απαιτείται για την ολοκλήρωση μιας δραστηριότητας. Με βάση τον παρακάτω τύπο εξάγουμε τον αναμενόμενο χρόνο ($\mathbf{T_E}$) για κάθε δραστηριότητα:

$$\mathbf{T_E = (O + 4M + P) \div 6}$$

- Ο ***κρίσιμος*** **δρόμος** είναι η μακρύτερη δυνατή συνεχής διαδρομή που λαμβάνεται από το αρχικό στο τελικό γεγονός. Καθορίζει το συνολικό ημερολογιακό χρόνο που απαιτείται για το έργο, και επομένως, οποτεδήποτε θα έχουμε καθυστερήσεις κατά μήκος του **κρίσιμου δρόμου** τότε θα έχουμε καθυστέρηση της επίτευξης του τελικού γεγονότος.
- *Η* ***κρίσιμη δραστηριότητα*** είναι μια δραστηριότητα που έχει το συνολικό περιθώριο ίσο με μηδέν. Κάθε δραστηριότητα με μηδέν συνολικό περιθώριο **σημαίνει** ότι είναι πάνω στο **κρίσιμο δρόμο**.

- ***Fast Tracking – Επίσπευση Εργασιών:*** εκτελώντας περισσότερα τμήματα κρίσιμων δραστηριοτήτων παράλληλα.
- ***Crashing – Δίνοντας Υπερωρίες:*** μειώνοντας τη διάρκεια των κρίσιμων δραστηριοτήτων (π.χ. με υπερωρίες).
- ***Total Float ή Slack – Συνολικό Περιθώριο:*** ο χρόνος όπου μια δραστηριότητα μπορεί να καθυστερήσει χωρίς να καθυστερήσει το έργο.
- ***Free Float – Ελεύθερο Περιθώριο:*** αφορά επακόλουθες εργασίες.

3.4.3.3 Εφαρμογή

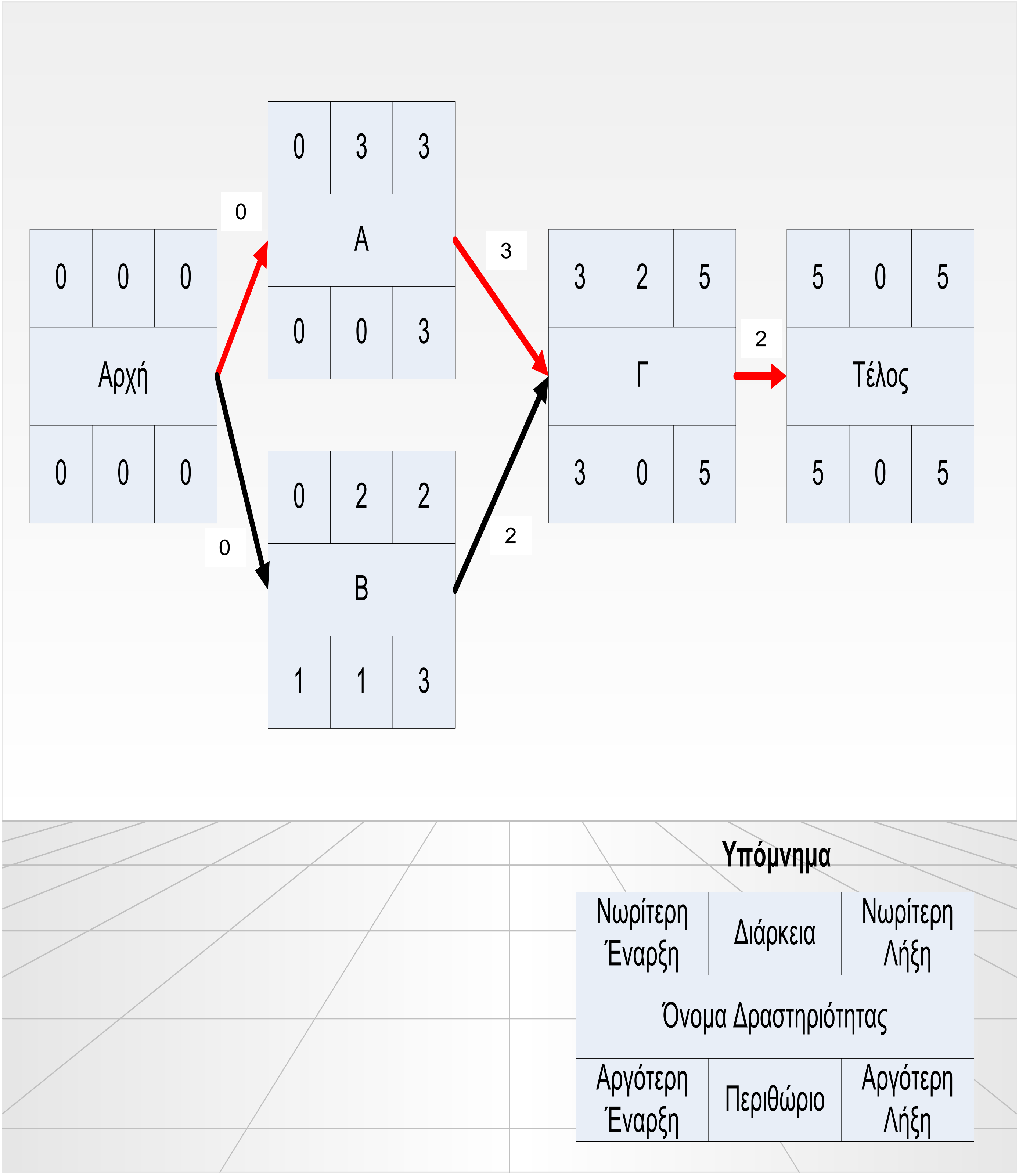
Το πρώτο βήμα στο σχεδιασμό του έργου είναι να καθοριστούν οι εργασίες που το έργο απαιτεί και η σειρά με την οποία πρέπει να ολοκληρωθούν. Η σειρά μπορεί να είναι εύκολο να καταγραφθεί για μερικές εργασίες π.χ. για την οικοδόμηση ενός σπιτιού, όπου πρέπει πρώτα να ομαλοποιηθεί το έδαφος προτού να μπορέσουν να μπουν τα θεμέλια ενώ είναι δύσκολο για άλλες εργασίες όπως π.χ. υπάρχουν δύο περιοχές που πρέπει να ομαλοποιηθούν, αλλά υπάρχουν αρκετοί εκσκαφείς για να ομαλοποιήσουν μόνο μια περιοχή. Επιπλέον, οι χρονικές εκτιμήσεις απεικονίζουν συνήθως τον πιο πιθανό χρόνο. Πολλές φορές, ο χρόνος που απαιτείται για να εκτελεστεί μια δραστηριότητα μπορεί να μειωθεί χρεώνοντας ένα επιπρόσθετο κόστος ή διαφοροποιώντας την ποιότητα.

Έστω ο ***Πίνακας 1*** ο οποίος δείχνει τις δραστηριότητες, τον αναμενόμενο χρόνο κάθε δραστηριότητας ο οποίος εξάγεται από το τύπο $T_E = (O + 4M + P) \div 6$ (όπου O=Optimistic-Αισιόδοξος, M=Mean-Πιθανότερος, P=Pessimistic-Απαισιόδοξος), τις προαπαιτούμενες δραστηριότητες, το ποσοστό ολοκλήρωσης της προαπαιτούμενης δραστηριότητας για να ξεκινήσει η επόμενη της, και τους αισιόδοξους, πιθανότερους, απαισιόδοξους χρόνους.

Α/Α	Δραστηριότητα	Αναμενόμενος Χρόνος	Προαπαιτούμενες Δραστηριότητες	Ποσοστό	Αισιόδοξος Optimistic	Πιθανότερος Mode	Απαισιόδοξος Pessimistic
1	Αρχή	0					
2	A	3	A	100%	2	3	4
3	B	2	A	100%	1	2	3
4	Γ	2	A,B	100%	1	2	3
5	Τέλος	0	Γ	100%			

Πίνακας 1 Σενάριο για επίλυση με τη τεχνική PERT.

Στη συνέχεια με βάση τον ***Πίνακας 1*** δημιουργούμε το διάγραμμα της Τεχνικής Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique) δικτύου το οποίο απεικονίζεται στην ***Εικόνα 4 Επίλυση Σεναρίου με τη τεχνική PERT.*** Τα κόκκινα βέλη απεικονίζουν το κρίσιμο δρόμο.



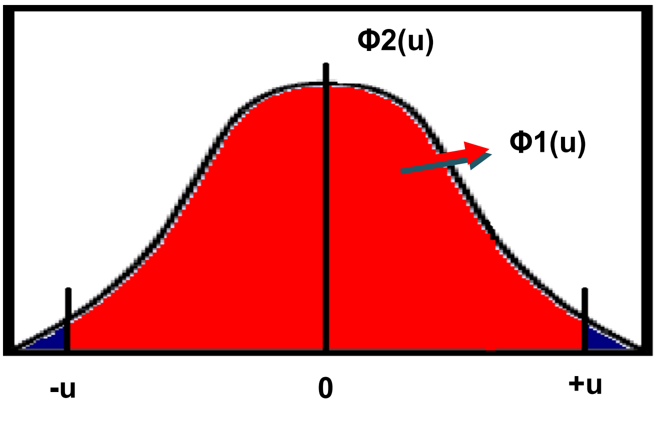
Εικόνα 4 Επίλυση Σεναρίου με τη τεχνική PERT.

Τα βήματα που ακολουθούμε για τη δημιουργία του δικτύου είναι ίδια με τη μέθοδο CPM μόνο που η διάρκεια που είχαμε στη μέθοδο CPM, θα αντικατασταθεί τώρα στο PERT με τον αναμενόμενο χρόνο ο οποίος εξάγεται από τον τύπο:

.

$$\boxed{\mathbf{T_E = (O + 4M + P) \div 6}}$$

Μετά την λύση του δικτύου του έργου, βρίσκουμε τις τυπικές αποκλίσεις των δραστηριοτήτων που βρίσκονται στο κρίσιμο δρόμο χρησιμοποιώντας τον τύπο: **σ = (bi – ai) / 4,** *<bi> είναι ο απαισιόδοξος χρόνος ενώ <ai> ο αισιόδοξος χρόνος* (για παράδειγμα για να βρούμε τη τυπική απόκλιση της δραστηριότητας Α έχουμε [σΑ = (4 – 2) / 4 => σΑ=0,5], όπου το <4> είναι ο απαισιόδοξος ενώ το <2> ο αισιόδοξος χρόνος της δραστηριότητας Α). Στη συνέχεια βρίσκουμε τη τυπική απόκλιση του έργου με το τύπο **σΕ =** $\sqrt{\sigma A^2 + \sigma \Gamma^2}$, όπου Α και Γ είναι οι δραστηριότητες που βρίσκονται πάνω στο κρίσιμο δρόμο. Αφού βρούμε το σΕ τότε βρίσκουμε το **u = (Δ – μ) / σΕ.** Η τιμή που θα πάρουμε αντιστοιχεί σε ένα ποσοστό το οποίο βρίσκουμε αν κοιτάξουμε στον πίνακα που περιέχει τις τιμές των εμβαδών της κανονικής κατανομής (**Εικόνα 5**). Για παράδειγμα για τιμή u=1,10 το αντίστοιχο Φ(u) είναι 0,7286 ή ποσοστό 72,86%. Για τιμή u=2,80, το Φ(u) είναι 0,9949 ή 99,49%. Με χρήση των παραπάνω τύπων και του πίνακα που περιέχει τις τιμές εμβαδών της κανονικής κατανομής, μπορούμε να προβλέψουμε την πιθανότητα το έργο μας να υπερβεί κάποιο χρονικό όριο. Μπορούμε να βρούμε την πιθανότητα το έργο να τελειώσει μέσα σε κάποιο χρονικό όριο. Μπορούμε ακόμα να βρούμε το ποσοστό που προκύπτει από το κίνδυνο να έρθει ρήτρα από τον πελάτη αν ξεπεραστούν για παράδειγμα οι 60 ημέρες.



Εικόνα 5 Η κανονική κατανομή.

3.4.4 Δομή Ανάλυσης Εργασιών (Work Breakdown Structure WBS)

3.4.4.1 Εισαγωγή

Η Δομή Ανάλυσης Εργασιών **(WBS)** είναι ένα θεμελιώδες εργαλείο ευρέως χρησιμοποιούμενο στη Διοίκηση Έργων. Είναι μια δομή που επιτρέπει τη σύνθεση των κατώτερων εργασιών, στις εργασίες «γονέων» πιο υψηλού επιπέδου. Για κάθε στοιχείο της Δομής Ανάλυσης Εργασιών (WBS), παράγεται μια περιγραφή της εργασίας που εκτελείται. Αυτή η τεχνική μερικές φορές αποκαλούμενη ως «Δομή ανάλυσης συστήματος» (System Breakdown Structure) χρησιμοποιείται για να καθορίσει και να οργανώσει τον τελικό σκοπό του έργου (PMI, 2006).

Μια καλά σχεδιασμένη Δομή Ανάλυσης Εργασιών (WBS) παράγει ένα προσανατολισμένο προς το προϊόν δέντρο, που αποτελείται από το υλικό, το λογισμικό, τις υπηρεσίες, τα δεδομένα, και τις εγκαταστάσεις. Αυτό σημαίνει ότι η Δομή Ανάλυσης Εργασιών (WBS) οργανώνεται γύρω από τα αρχικά προϊόντα του έργου (ή των προγραμματισμένων αποτελεσμάτων) έναντι της εργασίας που απαιτείται για να παραγάγει τα προϊόντα (προγραμματισμένες ενέργειες). Δεδομένου ότι τα προγραμματισμένα αποτελέσματα είναι οι επιθυμητές λύσεις του έργου, τα αποτελέσματα αυτά διαμορφώνουν ένα σχετικά σταθερό σύνολο κατηγοριών όπου μπορούν να συλλεχθούν οι δαπάνες των προγραμματισμένων ενεργειών που απαιτούνται. Μία καλά σχεδιασμένη Δομή Ανάλυσης Εργασιών (WBS) καθιστά εύκολο τον ορισμό κάθε δραστηριότητας του έργου σε ένα και μόνο ένα τελικό στοιχείο της Δομής Ανάλυσης Εργασιών (WBS). Εκτός από τη λειτουργία του στη χαρτογράφηση των απαιτήσεων από ένα επίπεδο συστήματος σε ένα άλλο, η Δομή Ανάλυσης Εργασιών (WBS) βοηθά επίσης και στη λογιστική του έργου (PMI, 2006).

3.4.4.2 Ο κανόνας 100%

Μια από τις σημαντικότερες αρχές της Δομής Ανάλυσης Εργασιών (WBS) καλείται ως κανόνας 100%. Το *Practice Standard for Work Breakdown Structure (δεύτερη έκδοση)*, καθορίζει τον κανόνα 100% ως εξής (Haugan, 2003): ***Ο κανόνας 100%****... δηλώνει ότι η Δομή Ανάλυσης Εργασιών (WBS) περιλαμβάνει το 100% της εργασίας που καθορίζεται από το σκοπό του έργου και περιλαμβάνει όλα τα παραδοτέα από την άποψη της ολοκλήρωσης της εργασίας, συμπεριλαμβανομένης της διαχείρισης του έργου. Ο κανόνας 100% είναι μια από τις σημαντικότερες αρχές που καθοδηγούν την ανάπτυξη, την αποσύνθεση και την αξιολόγηση της Δομής Ανάλυσης Εργασιών (WBS). Ο κανόνας ισχύει σε όλα τα επίπεδα μέσα στην ιεραρχία: το ποσό της εργασίας στο επίπεδο "παιδιών" πρέπει να είναι ίσο με 100%*

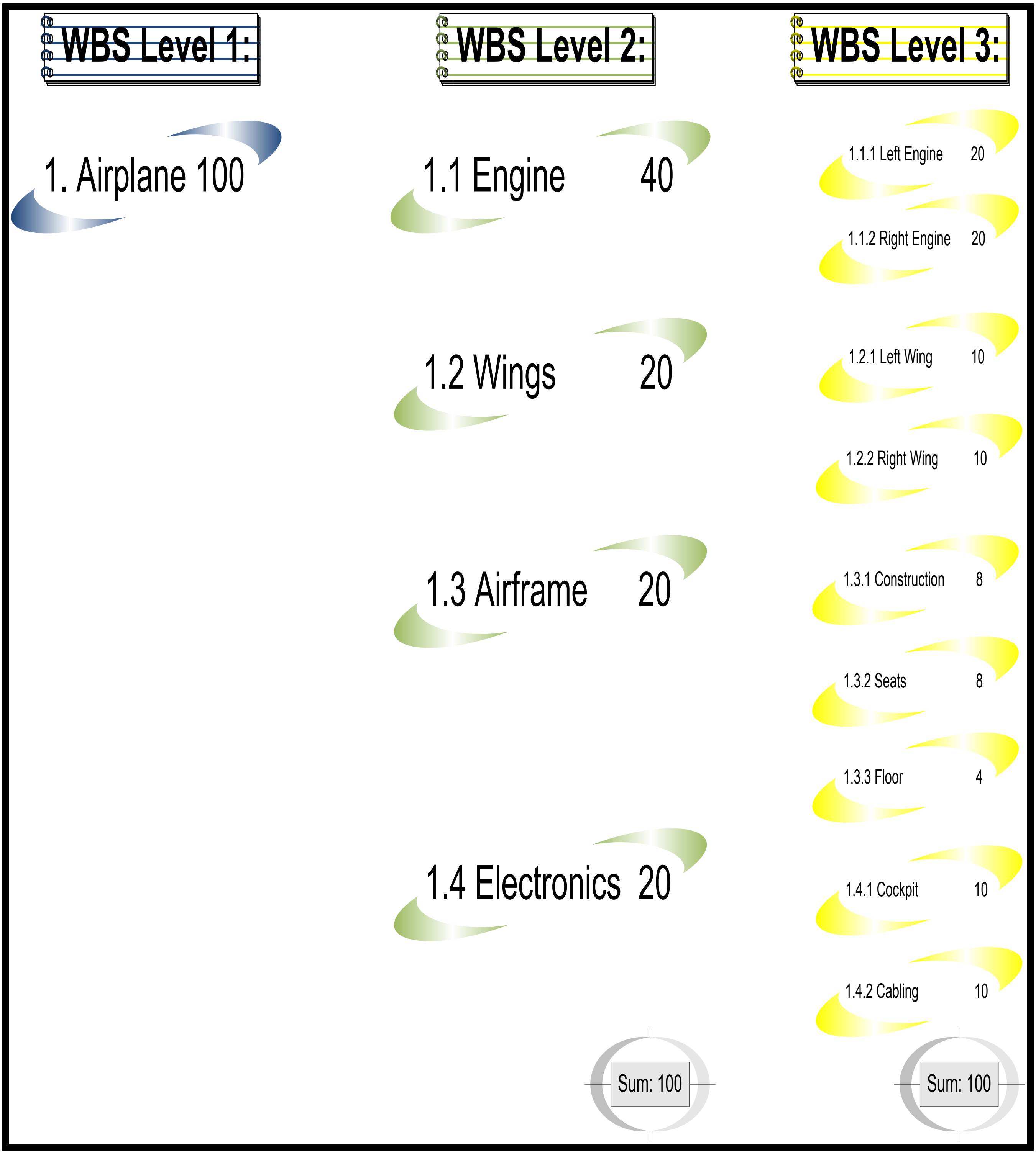
της εργασίας που αντιπροσωπεύεται από το "γονέα" και τη Δομή Ανάλυσης Εργασιών (WBS) δεν πρέπει να περιλαμβάνει οποιαδήποτε εργασία που δεν εμπίπτει στον πραγματικό στόχο του προγράμματος, δηλαδή δεν μπορεί να περιλαμβάνει περισσότερο από το 100% της εργασίας... Είναι σημαντικό να αναφερθεί ότι ο κανόνας 100% ισχύει επίσης για το επίπεδο δραστηριότητας. Η εργασία που αντιπροσωπεύεται από τις δραστηριότητες σε κάθε πακέτο εργασίας πρέπει να αθροίζει μέχρι το 100% της εργασίας που είναι απαραίτητη για την ολοκλήρωση του πακέτου εργασίας.

3.4.4.3 Σχήμα κωδικοποίησης της Δομής Ανάλυσης Εργασιών (WBS)

Όπως βλέπουμε και στο παράδειγμα στην ***Εικόνα 6***, στο επίπεδο 1 έχουμε ως κωδικό για τη δραστηριότητα «Airplane» τον αριθμό ένα (1). Καθώς προχωρούμε προς τα δεξιά πηγαίνοντας στο επίπεδο 2, η δραστηριότητα «Airplane» χωρίζεται σε άλλες τέσσερις (4) δραστηριότητες οι οποίες παίρνουν τη κωδικοποίηση 1.1, 1.2, 1.3, 1.4. Ο αριθμός πριν την υποδιαστολή ορίζει τη δραστηριότητα από το επίπεδο 1 ενώ ο αριθμός μετά την υποδιαστολή ορίζει την αύξουσα αρίθμηση των δραστηριοτήτων του επιπέδου 2. Έτσι και στο επίπεδο τρία (3), όπου π.χ. η 1.1 «Engine» δραστηριότητα χωρίζεται στις 1.1.1 «Left Engine» και 1.1.2 «Right Engine». Το μεσαίο ένα (1.**1**.1), προσδιορίζει ότι η δραστηριότητα με αύξων αριθμό (1.1.**1**) η «Left Engine» ανήκει στη δραστηριότητα ένα (1) του επιπέδου 2 με όνομα «Engine», η οποία όπως προσδιορίζει το αρχικό ένα (**1**.1.1) ανήκει στη δραστηριότητα «Airplane» του επιπέδου 1. Με τον τρόπο αυτό δημιουργείται η κωδικοποίηση μιας Δομής Ανάλυσης Εργασιών (WBS).

3.4.4.4 Τελικό Στοιχείο

Ένα τελικό στοιχείο είναι το χαμηλότερο στοιχείο (δραστηριότητα ή παραδοτέο) σε μια Δομή Ανάλυσης Εργασιών (WBS) και δεν υποδιαιρείται περαιτέρω.



Εικόνα 6 Παράδειγμα WBS για την κατασκευή ενός αεροπλάνου.

3.4.5 Πίνακας Αρμοδιοτήτων (RACI – Responsible Accountable Consulted Informed)

3.4.5.1 Εισαγωγή

Ο Πίνακας Αρμοδιοτήτων (RACI – Responsible Accountable Consulted Informed) είναι ένα σχετικά απλό εργαλείο που μπορεί να χρησιμοποιηθεί για τον προσδιορισμό των ρόλων και των ευθυνών. Αφού οι διαδικασίες μετασχηματισμού δεν υποβάλλονται από μόνες τους σε επεξεργασία, είναι προφανές ότι οι άνθρωποι πρέπει κάτι να «κάνουν» ώστε οι διαδικασίες να υλοποιηθούν. Επομένως, είναι χρήσιμο να περιγραφεί τι πρέπει να γίνει κι από ποιους, έτσι ώστε μια διαδικασία μετασχηματισμού να μπορεί να υλοποιηθεί. Στη θέση του όρου RACI, μερικές φορές χρησιμοποιούνται οι όροι RASCI ή RASIC.

Το RACI είναι η συντομογραφία για (Smith & Erwin, 2008):

- **R** = **Υπεύθυνος** – **Responsible** – υπεύθυνος του προβλήματος / έργου. Πρέπει να υπάρχει μόνο ένας «**R**» για κάθε εργασία.

- **A** = σε ποιόν ο «**R**» είναι **Υπόλογος** – **Accountable** – ποιός πρέπει να υπογράψει (να εγκρίνει) την εργασία.

- **C** = να τον συμβουλευτεί κάποιος, είναι **Σύμβουλος** – **Consultant** – Διπλό κανάλι επικοινωνίας.

- **I** = να **Ενημερωθεί** – **Informed** – πρέπει να ειδοποιηθεί / πληροφορηθεί για τα αποτελέσματα. Μονό κανάλι επικοινωνίας.

Η τεχνική υποστηρίζεται από ένα **διάγραμμα RACI** (δες ***Πίνακας 2***) που βοηθά στο να συζητηθούν και να συμφωνηθούν οι ρόλοι και οι ευθύνες.

Δραστηριότητα	Γιώργος	Κώστας	Μαρία	Έλενα
Ανάπτυξη Στρατηγικής	R	A	C	I
Καθορισμός Βασικής Ομάδας	R	A	C	C
Καθορισμός Εξωτερικής Ομάδας	R	A	I	I
Ανάλυση και Σχεδιασμός Συστήματος	I	A	C	R
Δοκιμή Συστήματος	C	A	R	I
Αξιολόγηση Συστήματος	R	A	I	C
Επανασχεδιασμός Συστήματος	R	A	C	C
Εφαρμογή Συστήματος	R	A	C	I

Πίνακας 2 Διάγραμμα RACI.

3.4.5.2 Εφαρμογή

Τα βήματα σε ένα Πίνακα Αρμοδιοτήτων (RACI – Responsible Accountable Consulted Informed) (αφορά τη διαδεδομένη και ευρέως χρησιμοποιούμενη κωδικοποίηση του RACI) είναι τα εξής (Smith & Erwin, 2008):

1. **Προσδιορίζουμε όλες τις δραστηριότητες** – Βάζουμε στην αριστερή πλευρά του διαγράμματος όλες τις δραστηριότητες που εμπλέκονται. Οι δραστηριότητες πρέπει να ορίζονται με σαφήνεια.
2. **Προσδιορίζουμε όλους τους ρόλους** – τους βάζουμε κατά μήκος της κορυφής του διαγράμματος. Οι ρόλοι μπορεί να είναι άτομα, ομάδες η ολόκληρα τμήματα.
3. Συμπληρώνουμε τα κελιά του διαγράμματος: **Προσδιορίζουμε ποιος έχει τα R, A, C, I για κάθε διαδικασία**. Ως γενικό κανόνα, καθορίζουμε πρώτα τα R στη συνέχεια τα A, τα C και τα I.
4. **Κάθε διαδικασία πρέπει να έχει ένα και μόνο ένα «R»** ως γενική αρχή. Ένα χάσμα εμφανίζεται όταν υπάρχει μια διαδικασία χωρίς «R» (κανένας

ρόλος δεν είναι υπεύθυνος), και μια επικάλυψη εμφανίζεται όταν υπάρχουν πολλαπλοί ρόλοι που έχουν «R» για μια δεδομένη διαδικασία.

5. **Επιλύουμε τις επικαλύψεις** – κάθε διαδικασία πρέπει να περιέχει ένα και μόνο ένα «Responsible» για να δείχνει σε ένα μοναδικό υπεύθυνο της διαδικασίας.
6. **Επιλύουμε τα χάσματα** – η απλούστερη περίπτωση που μπορούμε να εξετάσουμε είναι η επίλυση ενός χάσματος όπου δεν υπάρχει Responsible και ορίζουμε ένα άτομο ως Responsible.

4 Ένταξη των χρησιμοποιούμενων εργαλείων στο υπό ανάπτυξη εκπαιδευτικό λογισμικό

4.1 Εισαγωγή

Τα εργαλεία που χρησιμοποιούνται στη Διοίκηση Έργων και τα οποία θα εντάξουμε στο υπό – ανάπτυξη εκπαιδευτικό λογισμικό για την εκπαίδευση των χρηστών είναι τα εξής:

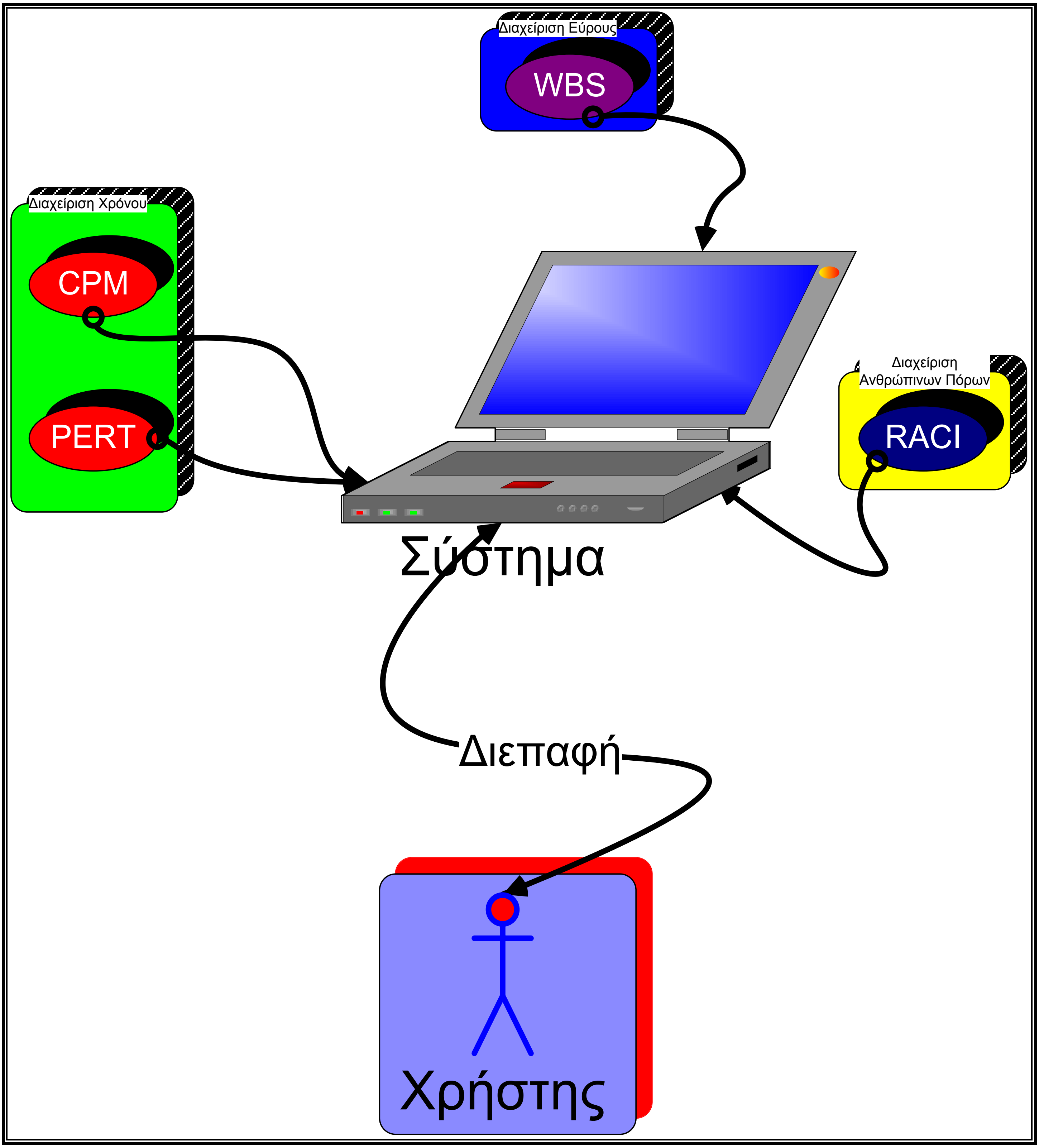
- Μέθοδος Κρισίμου Δρόμου – Critical Path Method (CPM).
- Αξιολόγηση Προγράμματος και Τεχνική Αναθεώρησης – Program Evaluation and Review Technique (PERT).
- Δομή Ανάλυσης Εργασιών – Work Breakdown Structure (WBS).
- Πίνακας Αρμοδιοτήτων (RACI).

Με βάση τα γνωστικά πεδία Διοίκησης Έργου όπως αναφέρονται από το Project Management Institute (PMI, 2004), τα πέντε παραπάνω εργαλεία ομαδοποιούνται και ταξινομούνται ως εξής:

- Διαχείριση Χρόνου:
    - Μέθοδος Κρισίμου Δρόμου – Critical Path Method (CPM),
    - Αξιολόγηση Προγράμματος και Τεχνική Αναθεώρησης – Program Evaluation and Review Technique (PERT),

- Διαχείριση Εύρους:
    - Δομή Ανάλυσης Εργασιών – Work Breakdown Structure (WBS).

- Διαχείριση Ανθρώπινων Πόρων:
    - Πίνακας Αρμοδιοτήτων (RACI).



Εικόνα 7 Ένταξη Εργαλείων στο Εκπαιδευτικό Λογισμικό

4.2 Ένταξη των εργαλείων στο Εκπαιδευτικό Λογισμικό

Η ένταξη των εργαλείων Μέθοδος Κρισίμου Δρόμου – Critical Path Method (CPM), Αξιολόγηση Προγράμματος και Τεχνική Αναθεώρησης – Program Evaluation and Review Technique (PERT), Δομή Ανάλυσης Εργασιών – Work Breakdown Structure (WBS), Πίνακας Αρμοδιοτήτων (RACI) στο Εκπαιδευτικό Λογισμικό έχει ως στόχο την εκπαίδευση του χρήστη στην εκμάθηση των μεθόδων και τεχνικών μέσω της επίλυσης των σεναρίων. Τα σενάρια με τα οποία ο χρήστης έρχεται αντιμέτωπος, ορίζουν ένα πρόβλημα το οποίο πρέπει να επιλυθεί.

Κάθε σενάριο που περιέχεται στη διεπαφή της Μεθόδου Κρισίμου Δρόμου – Critical Path Method (CPM) δίνει πληροφορίες για το πρόβλημα που αφορούν τα εξής:

- το όνομα της δραστηριότητας,
- τη διάρκεια της δραστηριότητας,
- τις προαπαιτούμενες δραστηριότητες,
- το ποσοστό ολοκλήρωσης κάθε δραστηριότητας.

Κάθε σενάριο που περιέχεται στη διεπαφή της Τεχνικής Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique) δίνει πληροφορίες για το πρόβλημα που αφορούν τα εξής:

- το όνομα της δραστηριότητας,
- τον αναμενόμενο χρόνο της δραστηριότητας ο οποίος παράγεται σύμφωνα με το τύπο $\boxed{\mathbf{T_E = (O + 4M + P) \div 6}}$
- τις προαπαιτούμενες δραστηριότητες,
- το ποσοστό ολοκλήρωσης κάθε δραστηριότητας.

Κάθε σενάριο που περιέχεται στη διεπαφή της Δομής Ανάλυσης Εργασιών – Work Breakdown Structure (WBS) δίνει πληροφορίες για το πρόβλημα που αφορούν τα εξής:

- το κωδικό (ID) της δραστηριότητας,
- το όνομα της δραστηριότητας.

Κάθε σενάριο που περιέχεται στη διεπαφή του Πίνακα Αρμοδιοτήτων (RACI) δίνει πληροφορίες για το πρόβλημα που αφορούν τα εξής:

- το όνομα της εργασίας,
- το όνομα των εμπλεκόμενων ατόμων,
- το ρόλο που παίζουν.

Κάθε φορά που ο χρήστης ξεκινά την επίλυση ενός σεναρίου θα πρέπει να ακολουθήσει τα παρακάτω βήματα σύμφωνα με το σχεδιασμό της διεπαφής και του τρόπου ένταξης των εργαλείων στο Εκπαιδευτικό Λογισμικό:

1. Επιλέγει το σενάριο.
2. Εμφανίζεται το σενάριο σε μορφή κειμένου. Στο κείμενο αναφέρονται οι βασικές πληροφορίες που αναφέραμε παραπάνω (π.χ. για το CPM: όνομα, διάρκεια, προαπαιτούμενες, ποσοστό ολοκλήρωσης).
3. Αφού ο χρήστης διαβάσει το σενάριο, ξεκινά να προσθέτει τις δραστηριότητες ή τις εργασίες ή τα στοιχεία στο χώρο που ορίζεται από τη διεπαφή ως **Χ**ώρος **Ε**ργασίας **Χ**ρήστη (**ΧΕΧ**). Αν για παράδειγμα στο σενάριο αναφέρονται τέσσερις δραστηριότητες τότε ο αριθμός των δραστηριοτήτων που πρέπει να υπάρχουν στο **ΧΕΧ** είναι τέσσερις (σε ότι αφορά το CPM, PERT). Στο Πίνακα Αρμοδιοτήτων (RACI) για παράδειγμα αν αναφέρονται τέσσερις εργασίες τότε ο αριθμός των εγγραφών του διαγράμματος που συμπληρώνει ο χρήστης πρέπει να είναι τέσσερις κ.τ.λ. Στη Δομή Ανάλυσης Εργασιών (WBS) αν έχουμε ένα στοιχείο που σπάει σε άλλα δύο και τα οποία σπάνε σε άλλα τρία στοιχεία το καθένα τότε ο αριθμός των επιπέδων που θα πρέπει να ορίσει ο χρήστης θα είναι τρία (Επίπεδο 1 – Αρχικό στοιχείο, Επίπεδο 2 – Τα δύο στοιχεία στα οποία σπάει το Αρχικό στοιχείο στο Επίπεδο 1, και Επίπεδο 3 – Τα ανά τρία στοιχεία που σπάνε τα δύο στοιχεία του Επιπέδου 2).

4. Αφού προστεθούν οι δραστηριότητες ή οι εργασίες ή τα στοιχεία, τότε πρέπει να συνδεθούν μεταξύ τους με βάση τις προαπαιτούμενες δραστηριότητες (σε ότι αφορά το CPM, RACI). Σε ότι αφορά το Πίνακα Αρμοδιοτήτων (RACI) δεν έχουμε συνδέσεις αλλά ορισμό του ρόλου κάθε ατόμου της ομάδας. Στη Δομή Ανάλυσης Εργασιών (WBS) έχουμε συνδέσεις με βάση τον αριθμό των στοιχείων που σπάει κάθε προηγούμενο στοιχείο.
5. Στη συνέχεια σε κάθε δραστηριότητα ή εργασία ή στοιχείο, ο χρήστης ξεκινά να καταγράφει τα στοιχεία που χρειάζονται με βάση τις πληροφορίες που παραθέτει κάθε σενάριο και να συμπληρώνει κατάλληλα. Για παράδειγμα σε ότι αφορά τη Μέθοδο του Κρισίμου Δρόμου (CPM) ο χρήστης καταγράφει το

όνομα της δραστηριότητας, τη διάρκεια της δραστηριότητας, τη νωρίτερη και την αργότερη έναρξη σύμφωνα με τα βήματα εφαρμογής της Μεθόδου του Κρισίμου Δρόμου (CPM). Η παραπάνω διαδικασία που αναφέραμε για τη Μέθοδο του Κρισίμου Δρόμου (CPM), εφαρμόζεται με τον ίδιο τρόπο και για τα υπόλοιπα εργαλεία.

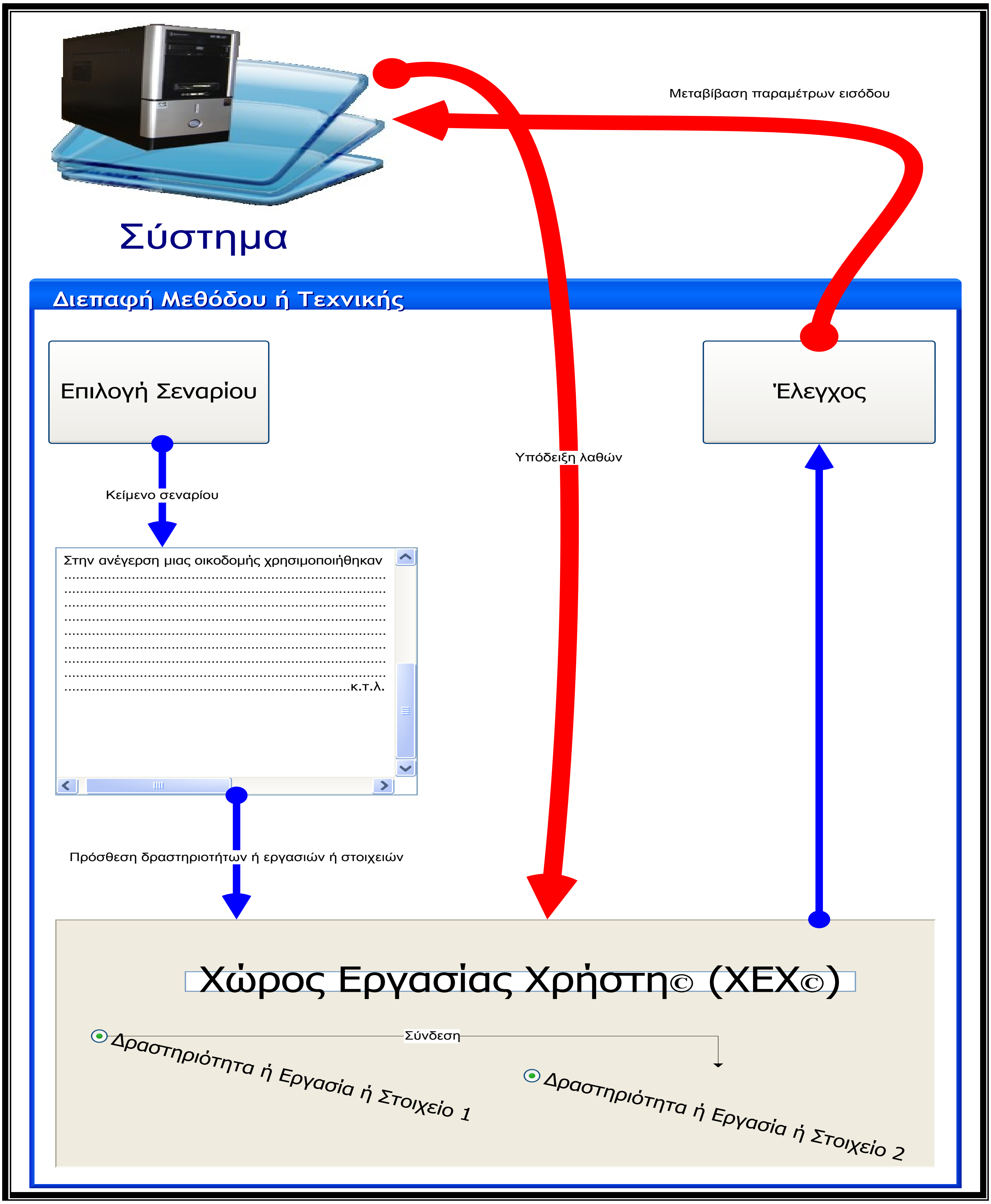
6. Το Σύστημα ελέγχει το **ΧΕΧ** (**Χ**ώρος **Ε**ργασίας **Χ**ρήστη). Για παράδειγμα, σε ότι αφορά τη Μέθοδο του Κρισίμου Δρόμου (CPM), το Σύστημα ελέγχει όλες τις εισόδους και τις συνδέσεις του διαγράμματος CPM όπως ορίστηκαν από το χρήστη, και υποδεικνύει τα λάθη έτσι ώστε να γίνουν οι απαραίτητες διορθώσεις.
7. Μετά από κάθε νέα διόρθωση που κάνει ο χρήστης σύμφωνα με τις υποδείξεις του Συστήματος, το Σύστημα επανελέγχει το ενημερωμένο διάγραμμα. Μέσα απ' αυτή τη διαδικασία επιτυγχάνουμε την *άμεση αμφίδρομη επικοινωνία μεταξύ διδάσκοντος και διδασκόμενων* . Επίσης με τον τρόπο αυτό επιτρέπεται στο/στη φοιτητή/τρια να *«αυτενεργήσει» ώστε να ανακαλύψει το αντικείμενο της μελέτης του/της* (Γαρυφαλλίδου, Ιωαννίδης, Σκέλλας, & Τσιτσίρης, 2003).
8. Για κάθε σενάριο που επιλύεται ή δεν επιλύεται ο χρήστης μπορεί να βλέπει στο πίνακα στατιστικών αποτελεσμάτων που περιέχεται στη διεπαφή των μεθόδων και των τεχνικών, τα σχετικά αποτελέσματα που αφορούν:

    - το ποσοστό ολοκλήρωσης του σεναρίου,
    - τις σωστές και λάθος απαντήσεις που έδωσε,
    - το χρόνο που χρειάστηκε για να επιλύσει το σενάριο.

9. Τέλος, ο χρήστης μπορεί να αποθηκεύσει το σημείο που σταμάτησε έτσι ώστε να το επαναφέρει κάποια άλλη χρονική στιγμή. Δίνεται με αυτό το τρόπο η δυνατότητα στο χρήστη να έχει στη διάθεση του έναν «εκπαιδευτικό» μόνο γι' αυτόν και για όποτε τον ζητήσει.

Η διαδικασία εκπαίδευσης του χρήστη στις μεθόδους και τεχνικές ξαναρχινά από το 1^{o} βήμα. Η διαδικασία επαναλαμβάνεται μέχρι ο χρήστης να κατανοήσει τον τρόπο σκέψης της επίλυσης και χρήσης κάθε μεθόδου και τεχνικής.

Το εκπαιδευτικό λογισμικό θα μπορεί να επιβραβεύει το χρήστη για τη προσπάθεια του, αφού επιλύσει όλα τα σενάρια που περιέχονται στο πρόγραμμα, δίνοντας του στο τέλος ένα τίτλο αναλόγως με το πόσο καλά τα πήγε. Για παράδειγμα αν το συνολικό ποσοστό επίλυσης των σεναρίων είναι πάνω από το 95% τότε θα του δίνει το τίτλο «Ειδήμων Διευθυντής Έργου». Αν είναι μεταξύ 85%-95% τότε μπορεί να του δίνει το τίτλο «Μέτριος Διευθυντής Έργου».



Εικόνα 8 Ένταξη Μεθόδων – Τεχνικών στο Εκπαιδευτικό Λογισμικό.

5 Παραμετροποίηση του Συστήματος

Η παραμετροποίηση του συστήματος γίνεται με σκοπό οι χρήστες να μπορούν να τροποποιούν τις παραμέτρους του συστήματος και να τις προσαρμόζουν στις δικές του απαιτήσεις και ανάγκες. Ένα στατικό και κλειστό σύστημα χωρίς δυνατότητα τροποποίησης των παραμέτρων του, δεν είναι φιλικό ούτε ευέλικτο προς τους χρήστες.

Μία παράμετρος του συστήματος η οποία θα μπορούσε να παραμετροποιηθεί έτσι ώστε να δίνεται η δυνατότητα στους χρήστες να τη τροποποιούν όπως επιθυμούν είναι η εξής:

1. **Βαθμός Δυσκολίας**: Ο βαθμός δυσκολίας του εκπαιδευτικού λογισμικού θα ορίζεται από το χρήστη αναλόγως του βαθμού ή επιπέδου δυσκολίας που θα επιλέγει. Η παράμετρος «**Βαθμός Δυσκολίας**» θα παίρνει τις τιμές:

    i. «**Αρχάριος**»: Ορίζει τα ευκολότερα προβλήματα, ασκήσεις ή σενάρια για επίλυση από το χρήστη. Το «ευκολότερα» είναι σχετικό και είναι εν συγκρίσει με τις δύο παρακάτω τιμές, προσδιορίζοντας το χαμηλότερο βαθμό δυσκολίας.

    ii. «**Επαγγελματίας**»: Ορίζει μεσαίας δυσκολίας προβλήματα, ασκήσεις ή σενάρια για επίλυση από το χρήστη. Το «μεσαίας» είναι σχετικό και είναι εν συγκρίσει με τις δύο άλλες τιμές, προσδιορίζοντας ένα μεσαίο βαθμό δυσκολίας.

    iii. «**Ειδήμων**»: Ορίζει τα δυσκολότερα προβλήματα, ασκήσεις ή σενάρια για επίλυση από το χρήστη. Το «δυσκολότερα» είναι σχετικό και είναι εν συγκρίσει με τις δύο προηγούμενες τιμές, προσδιορίζοντας το δυσκολότερο βαθμό δυσκολίας.

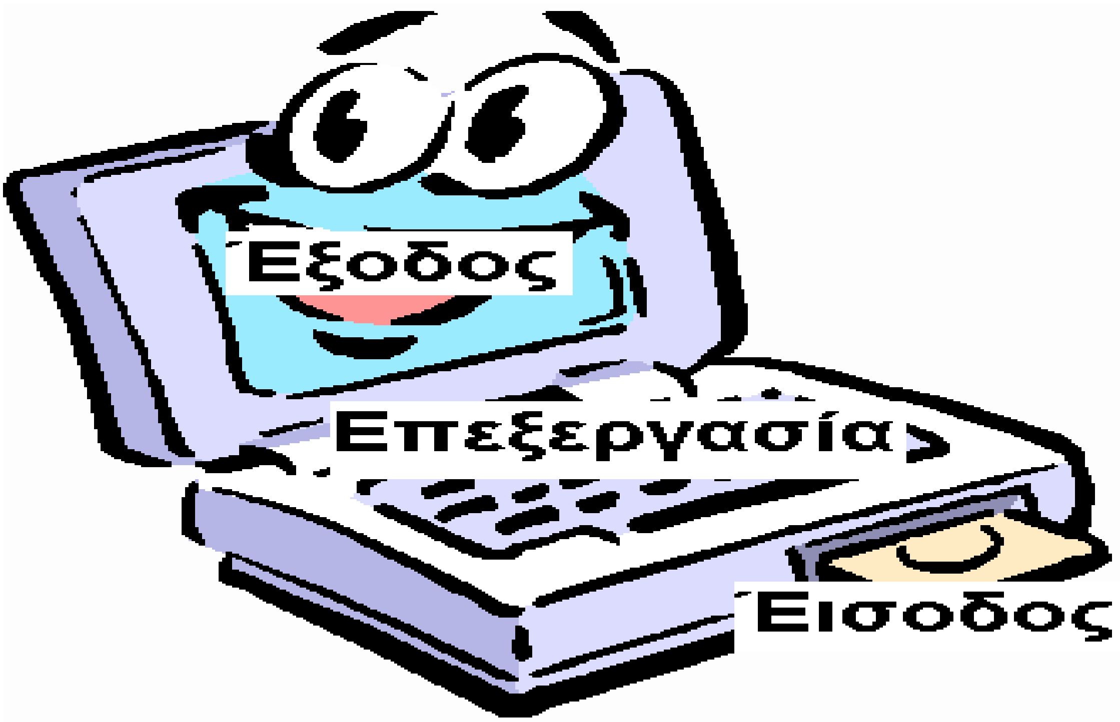
Η παραμετροποίηση «**Βαθμός Δυσκολίας**» δίνει στους χρήστες τη δυνατότητα να φέρνουν το εκπαιδευτικό λογισμικό πιο κοντά στα δικά τους μέτρα και απαιτήσεις. Το εκπαιδευτικό λογισμικό προσαρμόζεται με τον τρόπο αυτό στο προφίλ των χρηστών.

6 Ανάλυση και Σχεδιασμός του Συστήματος

6.1 Εισαγωγή

Τα Συστήματα δημιουργούνται για να επιλύουν προβλήματα. Κάποιος θα μπορούσε να σκεφτεί τη προσέγγιση των συστημάτων ως έναν οργανωμένο τρόπο διαχείρισης και αντιμετώπισης ενός προβλήματος. Στο δυναμικό κόσμο που ζούμε, η ανάλυση και ο σχεδιασμός συστημάτων, ασχολείται κυρίως με την ανάπτυξη των λειτουργιών του λογισμικού (National Institute of Open Schooling, 2008).

Μια συλλογή στοιχείων που λειτουργούν μαζί για να πραγματοποιήσουν κάποιο στόχο διαμορφώνει ένα σύστημα. Βασικά υπάρχουν τρία σημαντικά στοιχεία σε κάθε σύστημα: εισαγωγή, επεξεργασία και έξοδος.



Εικόνα 9 Είσοδος, Επεξεργασία, Έξοδος.

Σε ένα σύστημα τα διαφορετικά στοιχεία συνδέονται μεταξύ τους και είναι αλληλοεξαρτώμενα. Παραδείγματος χάριν, το ανθρώπινο σώμα αντιπροσωπεύει ένα πλήρες φυσικό σύστημα. Υπάρχουν επίσης πολλά εθνικά συστήματα όπως το πολιτικό σύστημα, οικονομικό σύστημα, εκπαιδευτικό σύστημα και ούτω καθεξής.

Ο κύκλος ζωής συστημάτων είναι μια οργανωτική διαδικασία ανάπτυξης και συντήρησης συστημάτων. Βοηθά στην καθιέρωση ενός σχεδίου του συστήματος, επειδή παραθέτει το γενικό κατάλογο διαδικασιών και υπό-διαδικασιών που απαιτούνται για την ανάπτυξη ενός συστήματος.

Ο κύκλος ζωής ανάπτυξης συστημάτων είναι ο συνδυασμός διάφορων δραστηριοτήτων. Με άλλα λόγια μπορούμε να πούμε ότι οι διάφορες δραστηριότητες συνολικά αναφέρονται ως κύκλος ζωής ανάπτυξης συστημάτων. Στην ανάλυση και σχεδιασμό συστημάτων, ο κύκλος ζωής ανάπτυξης συστημάτων είναι ο κύκλος ζωής ανάπτυξης λογισμικού (National Institute of Open Schooling, 2008).

Σύμφωνα με την εφαρμοσθείσα μέθοδο η οποία αναφέρεται στο κεφάλαιο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** ξεκινάμε την ανάλυση και τον σχεδιασμό του συστήματος.

6.2 Μελέτη Συστήματος

Η μελέτη συστήματος είναι το πρώτο στάδιο του κύκλου ζωής ανάπτυξης συστήματος. Δίνει μια σαφή εικόνα για το τι είναι στη πραγματικότητα το φυσικό σύστημα. Στην πράξη, η μελέτη συστημάτων γίνεται σε δύο φάσεις. Στην πρώτη φάση γίνεται η προκαταρκτική έρευνα για το σύστημα, που βοηθά στον προσδιορισμό του σκοπού του συστήματος. Η δεύτερη φάση της μελέτης συστήματος είναι πιο λεπτομερής και σε βάθος. Μελετώνται οι απαιτήσεις του χρήστη, οι περιορισμοί και τα προβλήματα του παρόντος συστήματος (National Institute of Open Schooling, 2008).

Τα στάδια της μελέτης του συστήματος είναι τα εξής (National Institute of Open Schooling, 2008):

- προσδιορισμός προβλήματος και έναρξη έργου,
- ανάλυση υποβάθρου,
- εξαγωγή συμπερασμάτων/ευρήματα.

Το πρόβλημα είναι η έλλειψη αντίστοιχου εκπαιδευτικού λογισμικού στο πεδίο της Διοίκησης Έργων όπου να μπορεί ο χρήστης (διαχειριστής έργου) να εκπαιδευτεί στις διάφορες μεθόδους και τεχνικές.

Το έργο της ανάπτυξης του εκπαιδευτικού λογισμικού ξεκινά με σκοπό τη δημιουργία ενός εκπαιδευτικού προγράμματος που να βασίζεται στη μεθοδολογία ανάπτυξης εκπαιδευτικού λογισμικού που αναφέραμε στο υποκεφάλαιο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..**

Με βάση την έρευνα που έγινε σε ότι αφορά την εύρεση παρόμοιων εκπαιδευτικών λογισμικών δεν βρέθηκε κάποιο σχετικό πρόγραμμα. Συγκεκριμένα τα αποτελέσματα από την έρευνα που διεξήχθη δίνονται στον ***Πίνακας 3 Αποτελέσματα Έρευνας Εκπαιδευτικών Λογισμικών.***

Κατηγορία / Όνομα Προγράμματος	Σχετικό	Πληροφορία
Open-Source desktop applications		
OpenProj	No	replacement for Microsoft Project
Open Workbench	No	alternative to Microsoft Project
TaskJuggler	No	(Reports) Linux
KPlato	No	Unix
GanttProject	No	project scheduling of tasks
Open-Source web-based applications		
Bugzilla	No	Irrelevant
dotProject	No	web-based, multi-user,**no check, non educational**
Mantis Bug Tracker	No	Irrelevant
Project.net	No	Only develop
ProjectPier	No	web-based, only develop
Trac	No	Irrelevant
SharpForge	No	Irrelevant
Proprietary desktop applications		
Artemis	Mid	tool to integrate project management processes such as planning, scheduling, aggregation, and reporting. **No check, non Educational**
Primavera Project Planner	Mid	scheduling and tracking project related activities, **No check, non Educational**
Collanos Workplace	No	Irrelevant
FastTrack Schedule	Mid	planning, tracking, and reporting project goals, **No check, non Educational**
Microsoft Project	Mid	developing plans, assigning resources to tasks, tracking progress, managing budgets and analyzing workloads, **No check, non Educational**
Merlin	Mid	developing plans,reporting system, , **No check, non Educational**
Contactizer	No	Help Contact Manager , **No check, non Educational**
OmniPlan	No	Irrelevant
OpenMind Business	Mid	brainstorming ideas and organizing information using the well-renowned Mind Map concept
Planisware OPX2 Pro	No	developing plans, **No check, non Educational**
RiskyProject	No	developing plans, risk analysis, **No check, non Educational**
Tracker Suite	No	business processes, **No check, non Educational**

Proprietary web-based applications		
@task	No	project planning, **No check, non Educational**
24SevenOffice	No	ERP , CRM
Basecamp	No	offers to-do lists, **No check, non Educational**
Central Desktop	No	Irrelevant
Daptiv	No	resource management,**No check, non Educational**
Gatherspace	No	planning, requirements management
Genius Inside	No	Like Microsoft Project
LiquidPlanner	No	Irrelevant
Mingle	No	allows project output and knowledge to be shared between an entire team at any given moment
OpenAir	No	project planning, **No check, non Educational**
Planisware OPX2/Planisware 5	No	Irrelevant
Project Insight	No	Like Microsoft Project
Projectplace	No	Web-based, **No check, non Educational**
RPlan	No	ERP
Teamwork	No	Microsoft Project
Track+	No	web-based project-management and bug tracking tool. **No check, non Educational**
VPMi	No	Project Portfolio Management, **No check, non Educational**
Wrike	No	online tool, **No check, non Educational**
Zoho Projects	No	Irrelevant

Πίνακας 3 Αποτελέσματα Έρευνας Εκπαιδευτικών Λογισμικών[2].

Με βάση τα παραπάνω αποτελέσματα, βλέπουμε ότι τα περισσότερα προγράμματα δίνουν πολλές δυνατότητες σχεδιασμού διαγραμμάτων, χρήση τεχνικών και μεθόδων Διοίκησης Έργων. Αυτό που κυρίως λείπει από τα προγράμματα αυτά είναι ο έλεγχος και κατά συνέπεια η εκπαιδευτική αγωγή που προκύπτει από τον έλεγχο.

[2] Με βάση το πίνακα στοιχείων από Wikipedia.

6.3 Ανάλυση Συστήματος

Υποθέτοντας ότι ένα νέο σύστημα πρόκειται να αναπτυχθεί, η επόμενη φάση είναι η ανάλυση του συστήματος. Η ανάλυση περιλαμβάνει μια λεπτομερή μελέτη του τρέχοντος συστήματος, που οδηγεί στις προδιαγραφές ενός νέου συστήματος. Η ανάλυση είναι μια λεπτομερής μελέτη των διάφορων διαδικασιών που εκτελούνται από το σύστημα και οι αναμεταξύ τους σχέσεις μέσα και έξω από το σύστημα. Χρησιμοποιώντας τα ακόλουθα βήματα γίνεται εύκολο να οριοθετηθεί το υπό εξέταση νέο σύστημα (National Institute of Open Schooling, 2008):

- παρακολουθώντας τα προβλήματα και τις νέες απαιτήσεις,
- εξετάζοντας τα πλεονεκτήματα και τα μειονεκτήματα συμπεριλαμβανομένων των νέων τομέων του συστήματος.

Όλες οι διαδικασίες και απαιτήσεις πρέπει να αναλυθούν και να τεκμηριωθούν υπό μορφή λεπτομερών διαγραμμάτων ροής, λεξικού στοιχείων, λογικών δομών δεδομένων και προδιαγραφών. Τα κύρια σημεία στην ανάλυση συστημάτων είναι (National Institute of Open Schooling, 2008):

- με βάση τις απαιτήσεις του χρήστη τι πρέπει το νέο σύστημα να επιτύχει,
- λειτουργική ιεραρχία που παρουσιάζει τις λειτουργίες για εκτέλεση από το νέο σύστημα και την αναμεταξύ τους σχέση.

Α/Α	Χαρακτηριστικά Εκπαιδευτικού Λογισμικού με βάση τις απαιτήσεις του χρήστη
1	Να περιλαμβάνει τις βασικές λειτουργίες (αυτά που ο μαθητευόμενος «βλέπει»)
2	Να υπάρχει σχετική ευκολία σε δοκιμές και σε αλλαγές
3	Να υπάρχει γραφικό περιβάλλον διεπαφής (Graphical User Interface, GUI)
4	HCI – Human Computer Interface (Μενού όχι γραμμή εντολών, Point & Click, Παράθυρα, Εικόνες, Pull-Down Menus)
5	Ακολουθίες από ενέργειες (μέσω των μενού)
6	Επίπεδο εμπειρίας και εξειδίκευσης χρήστη
7	Ομοιομορφία στην εμφάνιση και χρήση των λειτουργιών του συστήματος

Πίνακας 4 Χαρακτηριστικά Εκπαιδευτικού Λογισμικού.

Το σύστημα που θα αναπτύξουμε είναι ένα πρωτότυπο σύστημα το οποίο δεν έχει κάποιο άλλο παρόμοιο ώστε να το αναλύσουμε, να παρακολουθήσουμε τα προβλήματα και να εξάγουμε τις νέες απαιτήσεις. Βασιζόμαστε στις απαιτήσεις των χρηστών, εξάγοντας τα χαρακτηριστικά του εκπαιδευτικού λογισμικού, και αφού δημιουργήσουμε το πυρήνα του συστήματος, βελτιώνουμε βήμα-βήμα τις λειτουργίες του.

Στο υπό-ανάπτυξη εκπαιδευτικό λογισμικό, ο βασικός στόχος είναι η ικανοποίηση των απαιτήσεων του χρήστη. Με βάση λοιπόν, τον ***Πίνακας 4***, το εκπαιδευτικό λογισμικό πρέπει να περιλαμβάνει τις βασικές λειτουργίες οι οποίες αφορούν, την επιλογή μίας από τις πέντε τεχνικές/μεθόδους οι οποίες απαρτίζουν το εκπαιδευτικό λογισμικό, την επιλογή του σεναρίου για επίλυση από το χρήστη, έλεγχο της λύσης και υπόδειξη των λαθών από το σύστημα, διόρθωση των λαθών από το χρήστη. Η διαδικασία αυτή είναι επαναληπτική μέχρι το σενάριο να επιλυθεί ορθά από το χρήστη. Βασική λειτουργία που πρέπει να παρέχεται από το σύστημα είναι η παροχή βοήθειας στο χρήστη έτσι ώστε το περιβάλλον να είναι πιο φιλικό. Επίσης, η παροχή στατιστικών αποτελεσμάτων με βάση το ποσοστό ολοκλήρωσης των σεναρίων και το χρόνο που δαπανήθηκε, έχει ως στόχο να προσφέρει στο χρήστη μια εικόνα της προσπάθειας που κατέβαλε.

Προχωρώντας, με βάση το πίνακα, άλλη απαίτηση είναι να μπορούν να γίνονται αλλαγές και δοκιμές στο υπό-ανάπτυξη σύστημα με σχετική ευκολία και ταχύτητα. Εφόσον εντοπίζονται λάθη κυρίως λογικά λάθη, είναι σημαντικό το σύστημα να μπορεί να διορθώνεται, να βελτιώνεται και να προσφέρει στο χρήστη ένα αρτιότερο, χωρίς λάθη λογισμικό. Είναι σημαντικό η διαδικασία αυτή να γίνεται με ταχύτητα και με ευκολία.

Για να επιτύχουμε το στόχο αυτό (οι αλλαγές και βελτιώσεις να γίνονται εύκολα και γρήγορα) πρέπει να υπάρχει μια αμφίδρομη και καλή επικοινωνία μεταξύ χρηστών και διαχειριστών του συστήματος. Οι χρήστες μεταφέρουν τα προβλήματα που προκύπτουν στους διαχειριστές του συστήματος και οι διαχειριστές διορθώνουν και βελτιώνουν το σύστημα. Οι διαχειριστές του συστήματος μπορούν να εξάγουν συμπεράσματα για τη λειτουργία του συστήματος με ερωτηματολόγια χρηστών, με εξέταση συγκεκριμένων λειτουργιών του συστήματος παρακολουθώντας πως δουλεύουν οι χρήστες με αυτές τις λειτουργίες.

Μια απαραίτητη απαίτηση είναι το σύστημα να παρέχει στο χρήστη γραφικό περιβάλλον. Σήμερα, όλα σχεδόν τα λογισμικά, εκπαιδευτικά και μη, προσφέρουν γραφικό περιβάλλον. Το σύστημα γίνεται πολύ φιλικό στο χρήστη προσφέροντας ευκολία χρήσης των στοιχείων του λογισμικού, οπτική επαφή με τα αντικείμενα του λογισμικού, ευκολία σχεδιασμού της επίλυσης των σεναρίων (π.χ. στο σχεδιασμό επίλυσης ενός σεναρίου με τη Μέθοδο του Κρισίμου Δρόμου (CPM) ο χρήστης πολύ εύκολα σχεδιάζει το διάγραμμα δραστηριοτήτων και στη συνέχεια συμπληρώνει τα κενά κουτάκια με το όνομα της δραστηριότητας, τη διάρκεια και τα λοιπά στοιχεία).

Το σύστημα πρέπει να περιλαμβάνει μια ανθρωποκεντρική υπολογιστική διεπαφή αλλιώς HCI – Human Computer Interface. Βάση της διεπαφής είναι ο χρήστης και όλα γίνονται με γνώμονα την ανθρώπινη συμπεριφορά και τις ανθρώπινες ενέργειες. Αποτελείται από ευκολίες χρήσης της διεπαφής όπως:

- Μενού επιλογής και όχι γραμμή εντολών (γραμμή εντολών: συνήθως μια μαύρη οθόνη όπου ο χρήστης πρέπει να γράψει εξειδικευμένες εντολές για να κάνει συγκεκριμένες ενέργειες).

- Point & Click, επιλογή στοιχείων με ένα μόνο πάτημα του ποντικιού.

- Παράθυρα – το παραθυρικό περιβάλλον όπως αποδείχτηκε και στη πράξη (Windows, Linux – παραθυρικά περιβάλλοντα), είναι μια ανθρωποκεντρική προσέγγιση που βοηθά το χρήστη να αντιλαμβάνεται τα αντικείμενα του λογισμικού πιο εύληπτα. Όλες σχεδόν οι σημερινές εφαρμογές χρησιμοποιούν παραθυρικό περιβάλλον.

- Εικόνες – οι εικόνες κάνουν το λογισμικό πιο ενδιαφέρον προς το χρήστη και δίνουν στη διεπαφή «χρώμα» συμβάλλοντας σε μια ανθρωποκεντρική διεπαφή.

- Pull-Down Menus – Τα Pull-Down Menus προσφέρουν μια επιπλέον ευκολία στο χρήστη δίνοντας του τη δυνατότητα να βλέπει κατηγοριοποιημένες τις επιλογές και με ευκολία να επιλέγει.

Τα μενού γενικά προσφέρουν ευκολία στην επιλογή των ενεργειών που θέλει να εκτελέσει ο χρήστης.

Ένας άλλος στόχος που πρέπει το σύστημα να επιτύχει είναι η παροχή ενός επιπέδου εμπειρίας και εξειδίκευσης του χρήστη. Αυτό που θέλουμε να δημιουργήσουμε είναι το προφίλ του χρήστη με βάση το επίπεδο εμπειρίας και την εξειδίκευση του. Το σύστημα πρέπει να προσφέρει στο χρήστη με βάση το επίπεδο εμπειρίας και εξειδίκευσης, ένα διαβαθμισμένο επίπεδο δυσκολίας. Όσο πιο έμπειρος είναι ο χρήστης τόσο πιο δύσκολα πρέπει να είναι τα σενάρια που πρέπει να επιλύσει σε κάθε μέθοδο/τεχνική Διοίκησης Έργων που περιλαμβάνεται στο σύστημα. Αντίστοιχα, όσο λιγότερο έμπειρος είναι ο χρήστης τόσο πιο εύκολα πρέπει να είναι τα σενάρια προς επίλυση, προσφέροντας ακόμη βοήθεια και υποδείξεις για τη σωστή λύση των σεναρίων.

Όπως προαναφέραμε, το σύστημα θα περιλαμβάνει πέντε μεθόδους/τεχνικές Διοίκησης Έργων, όπου κάθε μία θα έχει τα δικά της σενάρια προς επίλυση. Είναι σημαντικό το σύστημα να έχει ομοιομορφία σε ότι αφορά την εμφάνιση και τη χρήση των μεθόδων/τεχνικων Διοίκησης Έργων. Η ομοιομορφία στην εμφάνιση αλλά και στη χρήση των λειτουργιών του συστήματος βοηθά το χρήστη να ακολουθεί τα ίδια βήματα, και να κάνει τις ίδιες ενέργειες, άσχετα με τη μέθοδο/τεχνική. Δεν αλλάζει ο τρόπος εμφάνισης και χρήσης των λειτουργιών κάθε διεπαφής κάθε μεθόδου/τεχνικής αλλά μόνο το σενάριο και ο τρόπος επίλυσης του σεναρίου.

6.4 Σχέδιο Συστήματος

Το νέο σύστημα πρέπει να σχεδιαστεί με βάση τις απαιτήσεις του χρήστη και τη λεπτομερή ανάλυση του συστήματος. Αυτή είναι η φάση του σχεδιασμού του συστήματος. Είναι η κρισιμότερη φάση στην ανάπτυξη ενός συστήματος. Ο σχεδιασμός χωρίζεται σε δύο στάδια (National Institute of Open Schooling, 2008):

- Προκαταρκτικός σχεδιασμός: Στο προκαταρκτικό ή γενικό σχέδιο, διευκρινίζονται τα χαρακτηριστικά γνωρίσματα του νέου συστήματος. Υπολογίζονται οι δαπάνες και τα οφέλη που απορρέουν. Εάν το έργο θεωρείται ακόμα εφικτό, κινούμαστε προς το λεπτομερές στάδιο σχεδιασμού.
- Λεπτομερές σχεδιασμός: στο λεπτομερές στάδιο σχεδιασμού, η ανάπτυξη του συστήματος ξεκινά με αποφασιστικότητα και σοβαρότητα. Σε αυτή τη φάση, ο σχεδιασμός του συστήματος γίνεται πιο δομημένος. Οι προδιαγραφές εισόδου, επεξεργασίας και εξόδου σχεδιάζονται με λεπτομέρεια.

Στο στάδιο σχεδιασμού, αποφασίζονται επίσης η γλώσσα προγραμματισμού και η πλατφόρμα στις οποίες το νέο σύστημα θα αναπτυχθεί. Για το υπό-ανάπτυξη σύστημα η γλώσσα προγραμματισμού που επιλέγηκε είναι η Visual Basic.Net η οποία ανήκει στο Visual Studio.Net. Οι λόγοι της συγκεκριμένης επιλογής είναι οι εξής (Holzner, 1998):

- Η Visual Basic.Net είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού.
- Είναι πολύ εύκολη στη χρήση και στη ανάπτυξη προγραμμάτων.
- Προσφέρει ένα περιβάλλον προγραμματισμού που συνδυάζει πολύ καλά το σχεδιασμό γραφικών παραθυρικών διεπαφών και προγραμματισμού των αντικειμένων.

Επίσης, η προηγούμενη εμπειρία που αποκτήθηκε με τη γλώσσα Visual Basic.Net, στην ανάπτυξη μικρών εφαρμογών, ήταν ακόμα ένας πιο καθοριστικός λόγος για την επιλογή της γλώσσας αυτής. Εφόσον προϋπάρχει η γνώση και η εμπειρία, τότε επιλέγοντας τη γλώσσα Visual Basic.Net κερδίζουμε πολύτιμο χρόνο στην ανάπτυξη του νέου πρωτότυπου εκπαιδευτικού λογισμικού. Δε χρειάζεται να ξανά-εφεύρουμε το τροχό. Χτίζουμε πάνω στην προϋπάρχουσα γνώση και εμπειρία.

6.4.1 Προκαταρκτικός Σχεδιασμός

Τα χαρακτηριστικά γνωρίσματα του συστήματος είναι τα εξής:

- Βασικές λειτουργίες:
  - Εισαγωγή στοιχείων του χρήστη:
    - Στοιχεία ταυτότητας.
    - Βαθμός εμπειρίας.
  - Επιλογή τεχνικής/μεθόδου για εκπαίδευση από το χρήστη.
  - Επιλογή σεναρίου για επίλυση από το χρήστη.
  - Σχεδιασμός αντίστοιχου διαγράμματος ή πίνακα σύμφωνα με τη τεχνική/μέθοδο που έχει επιλεγεί.
  - Συμπλήρωση διαγράμματος ή πίνακα από το χρήστη.
  - Έλεγχος διαγράμματος ή πίνακα από το σύστημα.
  - Υπόδειξη λαθών από το σύστημα.
  - Διόρθωση λαθών από το χρήστη (εκπαιδευτική αγωγή).

- Ευκολία σε δοκιμές και αλλαγές του συστήματος:
  - Διόρθωση λαθών του λογισμικού (debugging).
  - Δοκιμές του συστήματος με στόχο τη βελτίωση.

- Γραφικό περιβάλλον διεπαφής (Graphical User Interface, GUI).

- Human Computer Interface, HCI:

  - Μενού, για επιλογή στοιχείων από το χρήστη.
  - Point & Click, επιλογή στοιχείων με απλό πάτημα του ποντικιού.
  - Παραθυρικό περιβάλλον.
  - Εικόνες.
  - Pull-Down Menus.

- Ορισμός επιπέδου εμπειρίας και εξειδίκευσης του χρήστη.

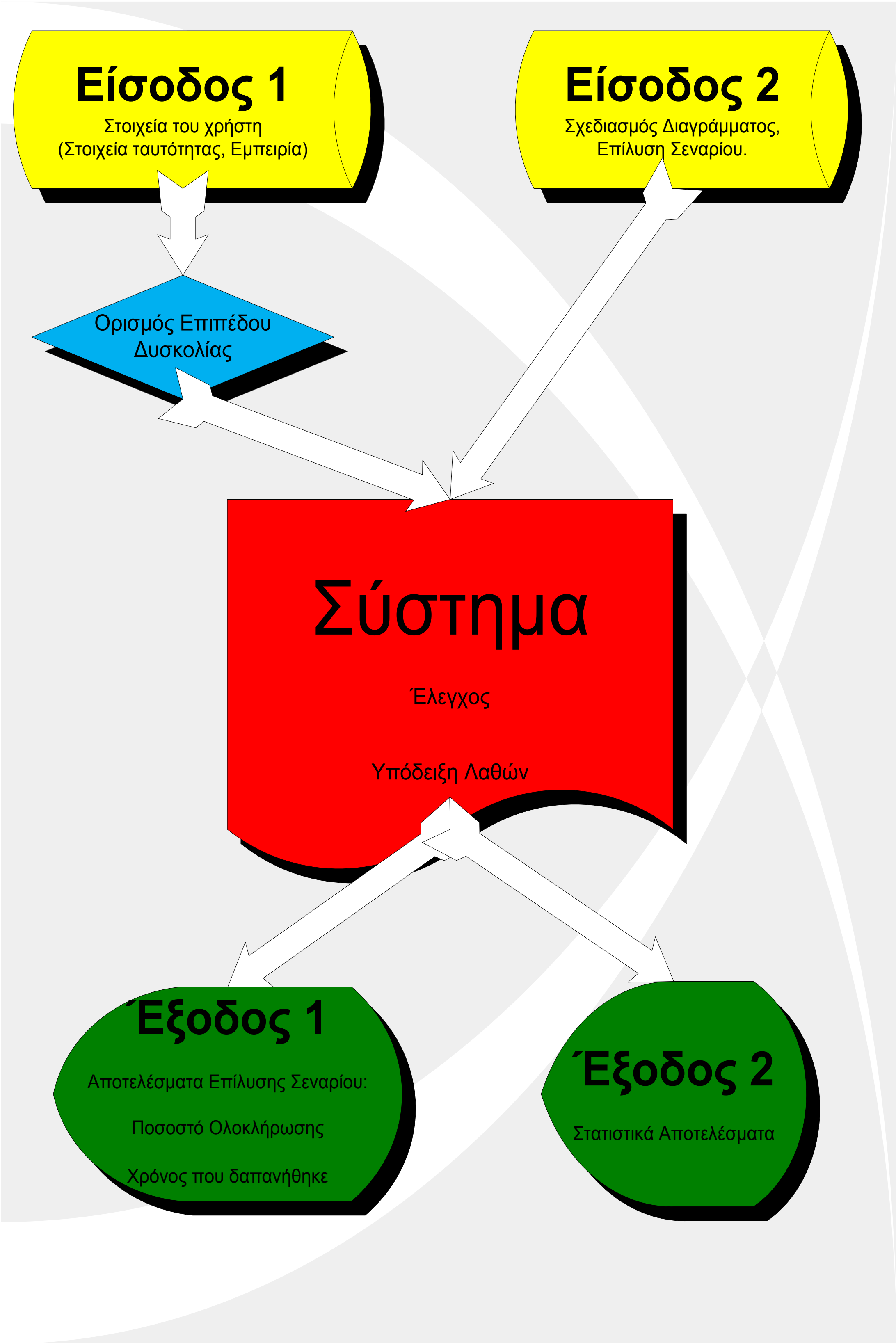
- Ομοιομορφία στην εμφάνιση και χρήση των λειτουργιών του συστήματος.

6.4.2 Λεπτομερής Σχεδιασμός

Ένα σύστημα αποτελείται από την είσοδο, την επεξεργασία και την έξοδο. Με βάση την ***Εικόνα 10 Αρχιτεκτονική του Συστήματος***, σχεδιάζουμε τις προδιαγραφές εισόδου, επεξεργασίας και εξόδου ως εξής:

- Είσοδος:
    - Είσοδος 1: Ως αρχική είσοδος στο σύστημα μπαίνουν τα στοιχεία ταυτότητας του χρήστη μαζί με το Βαθμό Εμπειρίας του (επιλογή από: «Αρχάριος», «Επαγγελματίας», «Ειδήμων»). Ο Βαθμός Εμπειρίας μετασχηματίζεται όπως φαίνεται και στην ***Εικόνα 10 Αρχιτεκτονική του Συστήματος***, σε «Βαθμό Δυσκολίας». Ο «Βαθμός Δυσκολίας» καθορίζει το επίπεδο δυσκολίας των σεναρίων καθώς επίσης και κατά πόσο το σύστημα θα παρέχει βοήθεια στο χρήστη κατά την επίλυση των σεναρίων.
    - Είσοδος 2: Στη συνέχεια ο χρήστης αφού επιλέξει σενάριο για επίλυση, σχεδιάζει το διάγραμμα ή το πίνακα και συμπληρώνει τα στοιχεία που ζητούνται.
- Επεξεργασία: Το στάδιο της επεξεργασίας του συστήματος μπορούμε να το παρομοιάσουμε με τη καρδιά που τροφοδοτεί με αίμα όλο μας το σώμα κι έτσι το κρατάει ζωντανό. Στο στάδιο αυτό, τα διάφορα στοιχεία και δεδομένα που εισάγονται στο σύστημα επεξεργάζονται για να δώσουν το αποτέλεσμα στην έξοδο που ακολουθεί την επεξεργασία. Συγκεκριμένα, αφού ο χρήστης ζητήσει να γίνει έλεγχος του επιλυμένου διαγράμματος ή πίνακα, το σύστημα επεξεργάζεται τα δεδομένα και στοιχεία που εισήχθηκαν σύμφωνα με τη τεχνική/μέθοδο που ο χρήστης έχει ήδη επιλέξει και υποδεικνύει τα λάθη ή τις παραλείψεις που πρέπει να διορθωθούν.

- Έξοδος:
    - Έξοδος 1: Αφού γίνει η επεξεργασία, έλεγχος και υπόδειξη των λαθών, το σύστημα εμφανίζει στο χρήστη το ποσοστό ολοκλήρωσης του συγκεκριμένου σεναρίου και το χρόνο που δαπανήθηκε.
    - Έξοδος 2: Επίσης εμφανίζονται τα στατιστικά αποτελέσματα του συνόλου των ποσοστών επίλυσης των σεναρίων, του συνολικού ποσοστού επίλυσης (άθροισμα των επιμέρους ποσοστών κάθε σεναρίου), του χρόνου που δαπανήθηκε για κάθε σενάριο ξεχωριστά και του συνολικού χρόνου που δαπανήθηκε (άθροισμα του επιμέρους χρόνου κάθε σεναρίου).



Εικόνα 10 Αρχιτεκτονική του Συστήματος

6.5 Κωδικοποίηση

Μετά το σχεδιασμό του συστήματος, ολόκληρο το σύστημα μεταφράστηκε στη γλώσσα προγραμματισμού την οποία ο υπολογιστής μπορεί να καταλάβει. Η κωδικοποίηση του συστήματος σε γλώσσα προγραμματισμού υλοποιεί το βήμα αυτό (National Institute of Open Schooling, 2008).

Είναι ένα σημαντικό στάδιο όπου οι καθορισμένες διαδικασίες μετασχηματίζονται σε προδιαγραφές ελέγχου με τη βοήθεια μιας γλώσσας προγραμματισμού. Καλείται επίσης φάση προγραμματισμού στην οποία ο προγραμματιστής μετατρέπει τις προδιαγραφές του προγράμματος σε εντολές και οδηγίες υπολογιστή, τις οποίες αναφέρουμε ως προγράμματα. Τα προγράμματα συντονίζουν τα δεδομένα και ελέγχουν τις διαδικασίες σε ένα σύστημα (National Institute of Open Schooling, 2008).

Όπως αναφέραμε στη φάση του σχεδιασμού του συστήματος, η γλώσσα προγραμματισμού που το εκπαιδευτικό λογισμικό υλοποιείται είναι η Visual Basic.Net. Με βάση τα χαρακτηριστικά γνωρίσματα του συστήματος που προκύπτουν από τις απαιτήσεις του χρήστη, ξεκινάμε την υλοποίηση των Βασικών Λειτουργιών του συστήματος. Αρχικά δημιουργούμε τις διεπαφές του χρήστη. Συνολικά δημιουργούμε έξι διεπαφές τις οποίες αναλύουμε παρακάτω:

- Εισαγωγική διεπαφή: Κωδικοποιήθηκε η αρχική είσοδο που δίνει ο χρήστης και αφορά την εισαγωγή των στοιχείων ταυτότητας και του βαθμού εμπειρίας. Επίσης στην εισαγωγική διεπαφή προσθέτουμε κάποιες εγκυκλοπαιδικές αναφορές για τις τεχνικές/μεθόδους που περιλαμβάνονται στο εκπαιδευτικό λογισμικό. Ο χρήστης επιλέγοντας την τεχνική/μέθοδο που θέλει, εμφανίζεται ένα παράθυρο με τις σχετικές πληροφορίες.
- Διεπαφή Μεθόδου του Κρισίμου Δρόμου (CPM): Ο χρήστης από την «εισαγωγική διεπαφή» αφού επιλέξει το κουμπί που αναγράφει «CPM» εισέρχεται στη διεπαφή της Μεθόδου του Κρισίμου Δρόμου (CPM). Κωδικοποιούμε αρχικά τη «διεπαφή CPM» που θα είναι η βάση για τις υπόλοιπες διεπαφές ως προς την ομοιομορφία και τη χρήση των αντικειμένων της διεπαφής. Κωδικοποιούμε δώδεκα κουμπιά που αντιστοιχούν σε δώδεκα σενάρια. Όταν επιλέγεται ένα σενάριο, εμφανίζεται σε ένα παράθυρο το σχετικό κείμενο που περιγράφει το σενάριο. Ο χρήστης αφού το διαβάσει, μπορεί να ξεκινήσει να σχεδιάζει το διάγραμμα της

Μεθόδου του Κρισίμου Δρόμου (CPM), στον **ΧΕΧ** (**Χ**ώρος **Ε**ργασίας **Χ**ρήστη), ένα παράθυρο όπου ο χρήστης σχεδιάζει το διάγραμμα της Μεθόδου του Κρισίμου Δρόμου (CPM) εισάγοντας τις δραστηριότητες όπως απαιτούνται από το σενάριο. Στη συνέχεια, αφού τελειώσει με το σχεδιασμό του διαγράμματος, μπορεί να ξεκινήσει να συμπληρώνει τα κενά κουτάκια που εμφανίζονται ως υποσύνολα σε κάθε δραστηριότητα. Κωδικοποιούμε επίσης τα εξής κουμπιά που αφορούν το σχεδιασμό του διαγράμματος: «εισαγωγή δραστηριότητας», «διαγραφή δραστηριότητας», «σύνδεση δραστηριοτήτων». Για να γίνει η διεπαφή περισσότερο φιλική και ανθρωποκεντρική, κωδικοποιούμε τη δυνατότητα ο χρήστης να μπορεί να προσθέσει δραστηριότητα απλά κάνοντας Επιλογή Σημείου & Αριστερό Κλικ (Point & Left Click) πάνω στον **ΧΕΧ** (**Χ**ώρος **Ε**ργασίας **Χ**ρήστη). Επίσης κάνοντας Επιλογή Σημείου & Δεξί Κλικ (Point & Right Click) πάνω σε μια δραστηριότητα εμφανίζεται μήνυμα που ζητά επιβεβαίωση διαγραφής της δραστηριότητας. Αν επιβεβαιωθεί από τη χρήστη τότε αυτόματα η δραστηριότητα διαγράφεται. Τέλος κωδικοποιούμε τη δυνατότητα ο χρήστης να μπορεί να συνδέσει δύο δραστηριότητες κάνοντας Αριστερό Κλικ (Left Click) πρώτα στη μία και ακολούθως στην άλλη. Στη συνέχεια κωδικοποιούμε την επιλογή «Έλεγχος» όπου το σύστημα ξεκινά να ελέγχει τις παραμέτρους και τις τιμές του διαγράμματος καθώς και την ακολουθία των δραστηριοτήτων (αν είναι δηλαδή με την ορθή σειρά – π.χ. η δραστηριότητα «Α» να βρίσκεται πριν από τη «Β» όπως απαιτεί το σενάριο). Σε ότι αφορά τον έλεγχο η κωδικοποίηση απαιτεί αρχικά την επίλυση του σεναρίου από το σύστημα και στη συνέχεια τη σύγκριση και έλεγχο των τιμών του χρήστη και του συστήματος. Είναι σημαντικό στο σημείο αυτό ο σύστημα να βγάζει πάντοτε σωστά αποτελέσματα. Για το λόγο αυτό η κωδικοποίηση στο σημείο αυτό, απαιτεί πολύ προσεκτικό έλεγχο αλλά και αρκετό χρόνο υλοποίησης του συγκεκριμένου κώδικα. Αφού γίνει ο έλεγχος το σύστημα υποδεικνύει στο χρήστη τα λάθη και παραλείψεις που έκανε. Ο χρήστης καλείται να τα διορθώσει και στη συνέχεια να τα επανελέγξει. Η αρχική διαδικασία σχεδιασμού και συμπλήρωσης του διαγράμματος από το χρήστη, ελέγχου και υπόδειξης από το σύστημα επαναλαμβάνεται όσες φορές χρειαστεί έως ότου το αποτέλεσμα είναι 100% ορθό. Τέλος το σύστημα εμφανίζει στο χρήστη τα στατιστικά αποτελέσματα σε ένα παράθυρο, όπου φαίνονται τα ποσοστά ολοκλήρωσης και ο χρόνος που δαπανήθηκε. Το δυσκολότερο σημείο της κωδικοποίησης και το πιο χρονοβόρο αφορούσε τον έλεγχο του διαγράμματος.

- Διεπαφή Τεχνικής Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique): Η κωδικοποίηση της «διεπαφής PERT» βασίζεται πάνω στη κωδικοποίηση της Μεθόδου του Κρισίμου Δρόμου (CPM), αφού οι δύο αυτές μέθοδοι μοιάζουν αρκετά. Το μόνο που αλλάζει στη «διεπαφή PERT» είναι ο αναμενόμενος χρόνος της δραστηριότητας ο οποίος παράγεται σύμφωνα με το τύπο:

  $$\mathbf{T_E = (O + 4M + P) \div 6}$$

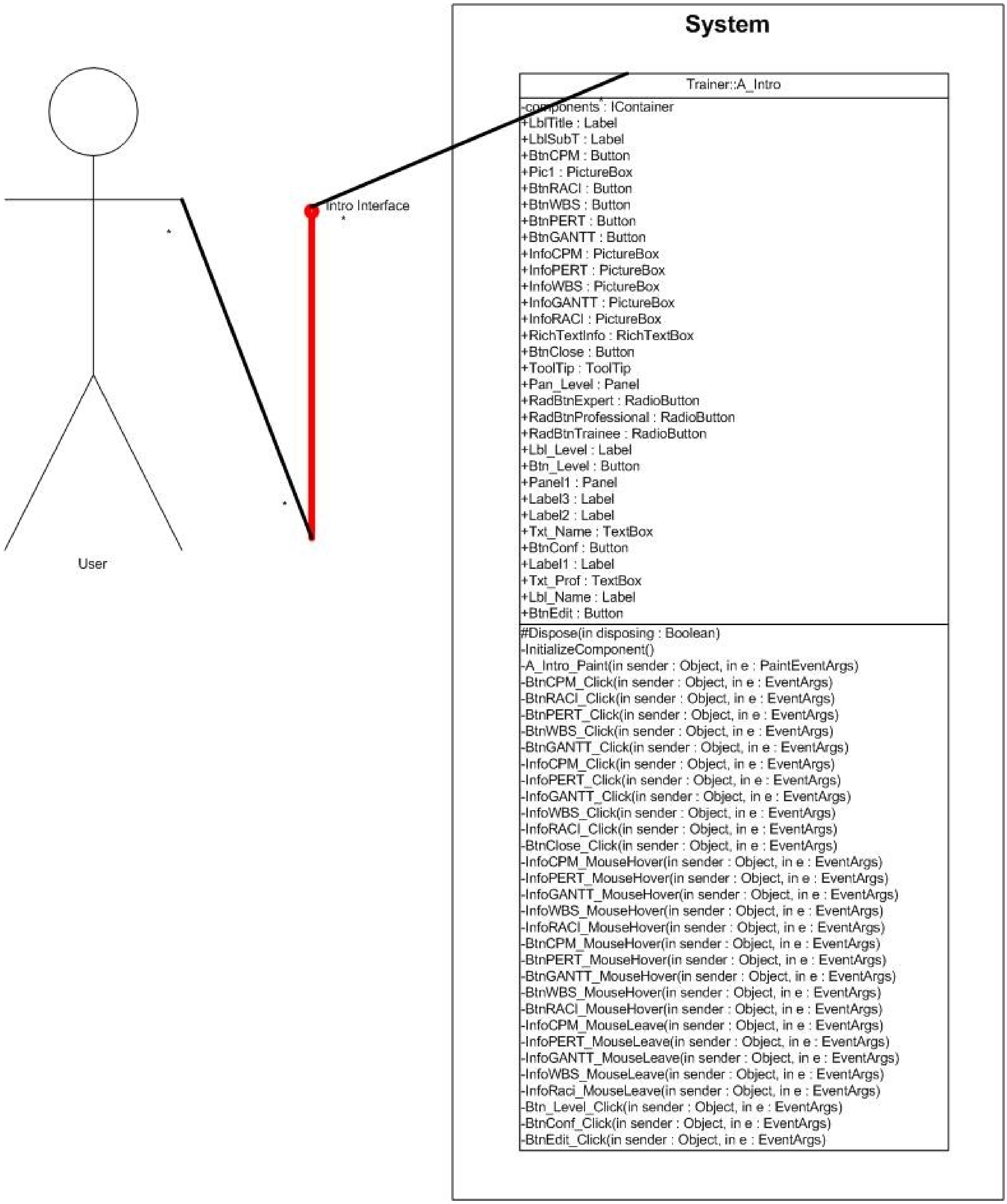
  Ο χρήστης δηλαδή δεν εισάγει τη διάρκεια της δραστηριότητας όπως στη Μέθοδο του Κρισίμου Δρόμου (CPM), αλλά τον αναμενόμενο χρόνο. Κατά τα υπόλοιπα η κωδικοποίηση παραμένει στο ίδιο πλαίσιο με τη Μέθοδο του Κρισίμου Δρόμου (CPM).
- Διεπαφή Δομής Ανάλυσης Εργασιών (WBS): Όλα όσα περιγράψαμε περί κωδικοποίησης και τρόπου λειτουργίας της «διεπαφής CPM» ισχύουν και για τη «διεπαφή WBS» με μόνη εξαίρεση το **ΧΕΧ** (**Χ**ώρος **Ε**ργασίας **Χ**ρήστη). Στο **ΧΕΧ** της «διεπαφής WBS», ο χρήστης προσθέτει κουτάκια τα οποία μπαίνουν με μια συγκεκριμένη δομή. Αρχικά επιλέγοντας ένα σενάριο, προστίθεται από το σύστημα το αρχικό κουτάκι όπου μπαίνει η κύρια δραστηριότητα του έργου π.χ. «Κατασκευή Αεροπλάνου» και στη συνέχεια σε ένα χαμηλότερο επίπεδο από το αρχικό ο χρήστης προσθέτει άλλα κουτάκια στα οποία βάζει τα υπό-βασικά μέρη της κατασκευής ενός αεροπλάνου π.χ. «Μηχανές», «Φτερά» κ.τ.λ. Με τον ίδιο τρόπο μπορεί να προσθέσει κουτάκια για κάθε ένα προηγούμενο κουτάκι, σπάζοντας ακόμα περισσότερο τη δομή του έργου σύμφωνα με το κείμενο του σεναρίου. Για τη διαγραφή απλά επιλέγει το κουμπί «Διαγραφή (Επίπεδο 2)» ή «Διαγραφή (Επίπεδο 3)» και το σύστημα διαγράφει το τελευταίο κουτάκι που υπάρχει στο επίπεδο 2 ή στο επίπεδο 3 αντίστοιχα με την επιλογή.
- Διεπαφή Πίνακα Αρμοδιοτήτων (RACI): Όλα όσα περιγράψαμε περί κωδικοποίησης και τρόπου λειτουργίας της «διεπαφής CPM» ισχύουν και για τη «διεπαφή RACI» με μόνη εξαίρεση το **ΧΕΧ** (**Χ**ώρος **Ε**ργασίας **Χ**ρήστη). Στο **ΧΕΧ** της «διεπαφής RACI» ο χρήστης συμπληρώνει το πίνακα RACI σύμφωνα με το κείμενο κάθε σεναρίου. Στην ουσία βάζει τα R, A, C, I, στα κελιά του πίνακα.

Δίνεται η επιλογή στο χρήστη της δημιουργίας σεναρίου στο οποίο εισάγει με μορφή πίνακα διάφορα στοιχεία, όπως για τη διεπαφή CPM, όνομα δραστηριότητας, διάρκεια, ποιές δραστηριότητες είναι προαπαιτούμενες, ποσοστό ολοκλήρωσης. Παρόμοια στοιχεία αφορούν και τη περίπτωση της διεπαφής PERT ενώ στη διεπαφή Δομής Ανάλυσης Εργασιών (WBS) τα στοιχεία που εισάγονται πάλι σε μορφή πίνακα αφορούν τα ονόματα των δραστηριοτήτων ανά επίπεδο. Στη διεπαφή RACI ο χρήστης εισάγει στο πίνακα το όνομα της δραστηριότητας και τα ονόματα των υπευθύνων υλοποίησης της.

Επίσης ο χρήστης μπορεί ένα υπάρχον σενάριο να το επεξεργαστεί και να αλλάξει τα δεδομένα του. Και σε αυτή τη περίπτωση η επεξεργασία γίνεται στο σχετικό πίνακα κάθε σεναρίου.

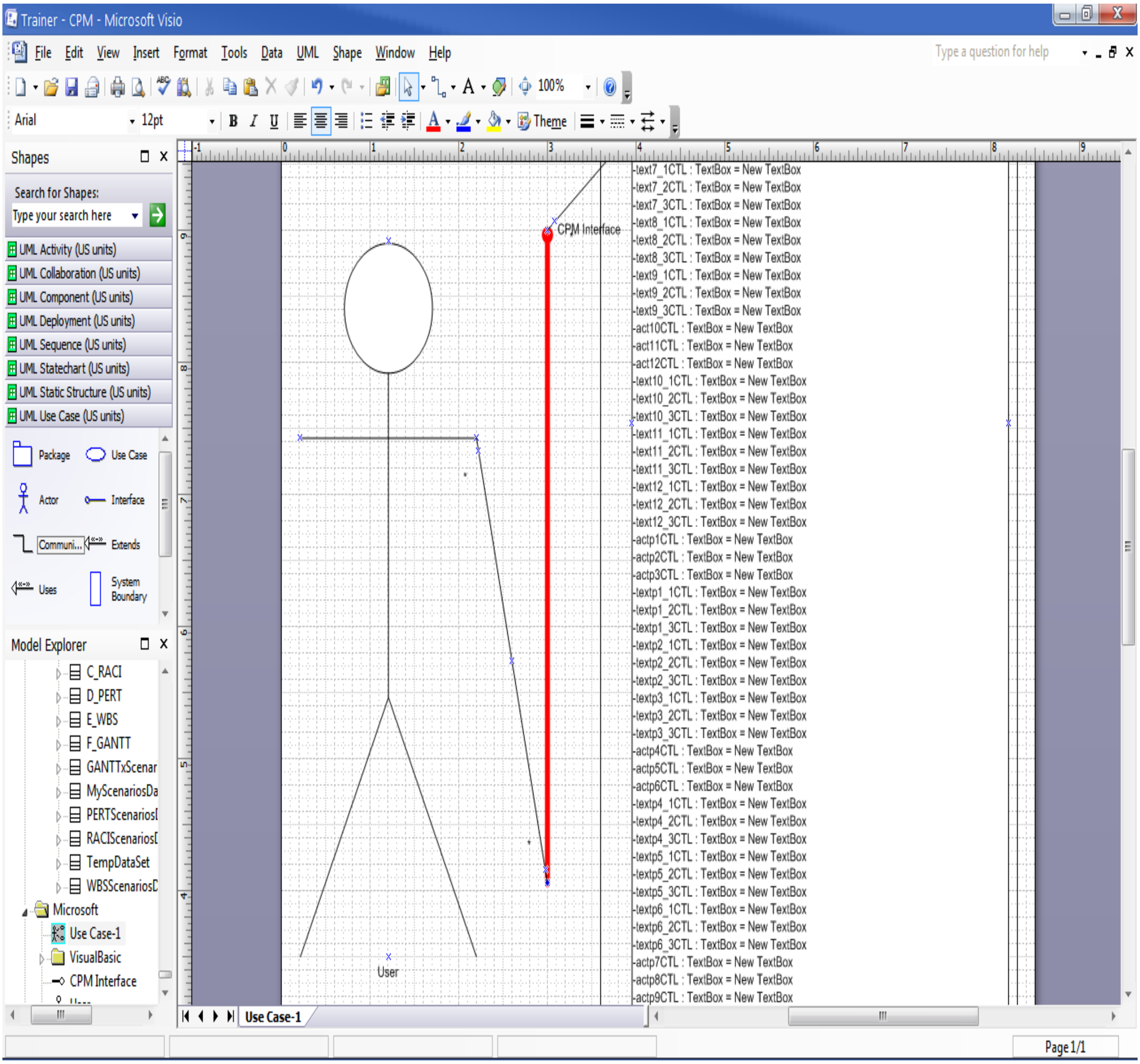
6.5.1 Διαγράμματα Ροής Δεδομένων

Τα Διαγράμματα Ροής Δεδομένων περιγράφουν τη ροή εκτέλεσης ενός αλγορίθμου ή μιας διαδικασίας. Στη συνέχεια παραθέτουμε τα διαγράμματα ροής δεδομένων που αφορούν το κώδικα και τα οποία παράχθηκαν χρησιμοποιώντας το εργαλείο της Visual Basic.Net, Reverse Engineer.

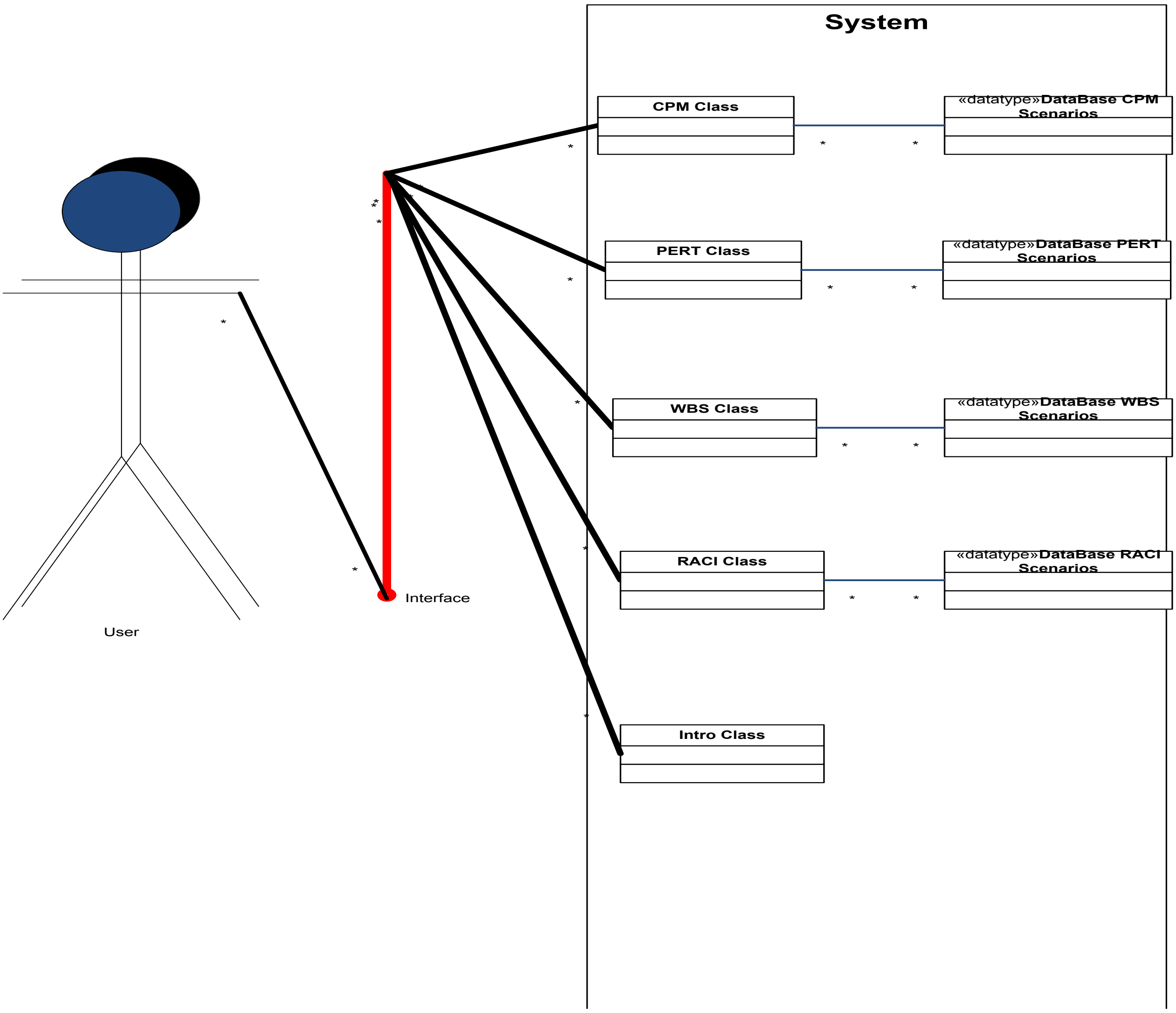


Εικόνα 11 ΔΡΔ της Διεπαφής Εισαγωγής

Στην παρακάτω εικόνα παραθέτουμε ένα δείγμα του διαγράμματος ροής δεδομένων που αφορά τη διεπαφή CPM. Λόγω των πολλών παραμέτρων της διεπαφής το μέγεθος είναι αρκετά μεγάλο και δε χωράει σε σελίδα.

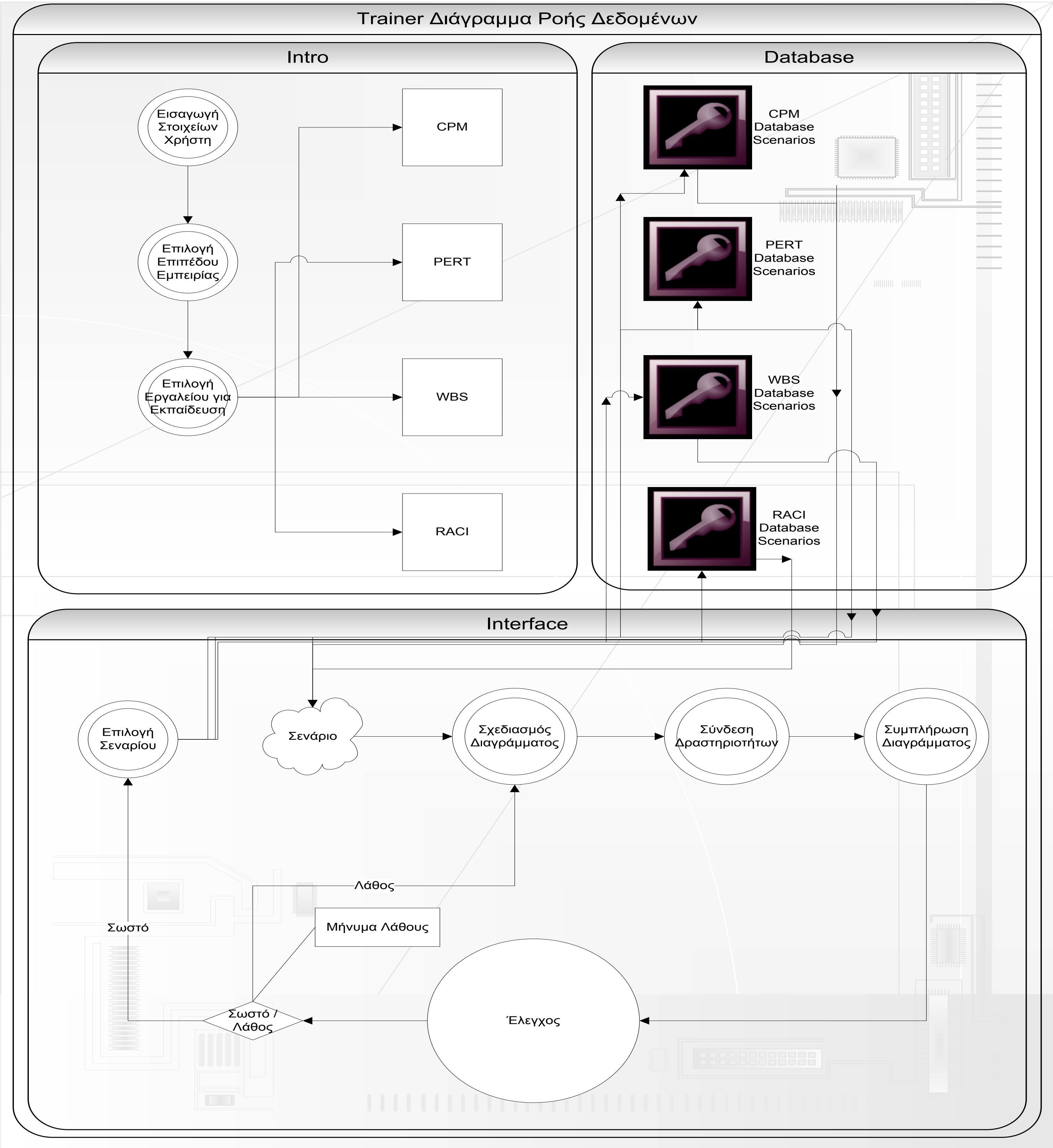


Εικόνα 12 ΔΡΔ της Διεπαφής CPM (δείγμα)



Εικόνα 13 ΔΡΔ Σύνδεση Βάσεων Δεδομένων

Στη συνέχεια παρουσιάζουμε το Διάγραμμα Ροής Δεδομένων για το λογισμικό Trainer©.



Εικόνα 14 ΔΡΔ Trainer©

6.6 Δοκιμή

Προτού το σύστημα μπει σε λειτουργία, έγινε μια δοκιμαστική λειτουργία του συστήματος αφαιρώντας όλα τα λάθη (bugs). Είναι μια σημαντική φάση ενός επιτυχούς συστήματος. Μετά την κωδικοποίηση, αναπτύχθηκε και εφαρμόστηκε ένα σχέδιο δοκιμής, σε ένα γνωστό σετ δοκιμαστικών δεδομένων. Η έξοδος που έδωσε η δοκιμαστική εφαρμογή πρέπει να ταιριάξει με τα αναμενόμενα αποτελέσματα (National Institute of Open Schooling, 2008).

Χρησιμοποιώντας τα δοκιμαστικά δεδομένα μετά από τη δοκιμαστική εφαρμογή λαμβάνουν χώρα (National Institute of Open Schooling, 2008):

- Ο έλεγχος μονάδας: Όταν τα προγράμματα κωδικοποιηθούν και λειτουργήσουν, πρέπει να εξεταστούν χωριστά με τα δεδομένα δοκιμής. Οποιοδήποτε ανεπιθύμητο συμβάν πρέπει να σημειωθεί και να διορθωθεί.
- Ο έλεγχος του συστήματος: Μετά από τον έλεγχο μονάδας και όταν αφαιρεθούν τα λάθη, πραγματοποιείται η δοκιμή του συστήματος. Σε αυτή τη φάση ο έλεγχος γίνεται με τα πραγματικά δεδομένα. Το ολοκληρωμένο σύστημα εκτελείται με τα πραγματικά δεδομένα ή με τις πραγματικές συνθήκες. Σε κάθε στάδιο της εκτέλεσης, αναλύονται τα αποτελέσματα ή η έξοδος του συστήματος. Οποιαδήποτε λάθη προκύψουν διορθώνονται.

Ξεκινώντας λοιπόν με τον έλεγχο μονάδας, εξετάζουμε κάθε μονάδα του συστήματος –στη περίπτωση μας κάθε διεπαφή, CPM, PERT, WBS, RACI– για να εντοπίσουμε τυχόν λάθη που θα προκύψουν. Κάθε μονάδα εξετάζεται ξεχωριστά. Τα λάθη που προκύπτουν από τον έλεγχο, είναι στη πλειονότητα τους λογικά λάθη. Τα συντακτικά λάθη έχουν ήδη εντοπιστεί από τη γλώσσα προγραμματισμού που χρησιμοποιούμε.

Ο έλεγχος του συστήματος ξεκινά με τη δοκιμή του συστήματος υπό πραγματικές συνθήκες. Δοκιμάζουμε το σύστημα όπως ακριβώς θα το χρησιμοποιούσε ο χρήστης για να εκπαιδευτεί. Δοκιμάζουμε όλες τις λειτουργίες του συστήματος, εισάγουμε στις παραμέτρους εισόδου διάφορες τιμές για να δούμε αν το αποτέλεσμα είναι σωστό και το σύστημα δε καταρρέει. Όπως διαπιστώνουμε από τον έλεγχο του συστήματος υπάρχουν ακόμα λογικά λάθη τα οποία προκύπτουν από την εισαγωγή διαφόρων ειδών δεδομένων στις παραμέτρους εισόδου. Παράδειγμα, εισάγουμε αντί αριθμό, ένα χαρακτήρα ως είσοδο και όταν γίνεται ο έλεγχος το σύστημα δίνει στην έξοδο λάθος αποτέλεσμα. Έτσι λοιπόν προβαίνουμε σε περιορισμό του είδους δεδομένων που ο χρήστης μπορεί να δώσει ως είσοδο –να μπορεί να δώσει ως είσοδο μόνο αριθμούς για το συγκεκριμένο παράδειγμα–.

7 Αξιολόγηση του Συστήματος

7.1 Γενικά

Αρχικά αξιολογήθηκε το σύστημα ως προς το περιεχόμενο του. Το συμπέρασμα είναι ότι το περιεχόμενο του συστήματος ανταποκρίνεται στους στόχους και τις επιδιώξεις μας. Πρέπει το σύστημα να εφαρμοστεί σε πραγματικές συνθήκες λειτουργίας για μεγάλο χρονικό διάστημα ώστε να είμαστε σε θέση να βγάλουμε ένα ασφαλές συμπέρασμα. Επίσης αξιολογήσαμε το σύστημα ως προς τους διδακτικούς του στόχους. Το σύστημα έχει βασικό στόχο την εκπαίδευση των χρηστών στη Διοίκηση Έργων με τη χρήση συγκεκριμένων εργαλείων. Βλέπουμε ότι πράγματι ο στόχος είναι συγκεκριμένος και επιτυγχάνεται μέσα από συγκεκριμένα εργαλεία που σχετίζονται άμεσα με τη Διοίκηση Έργων. Τέλος ως προς την αξιολόγηση του περιεχομένου, οι πληροφορίες που δίνονται από το σύστημα στους χρήστες, καλύπτουν σε βάθος το θέμα που εξετάζεται, στη περίπτωση μας στη Διοίκηση Έργων. Οι χρήστες λαμβάνουν από το σύστημα ικανοποιητική γνώση των εργαλείων τα οποία μπορούν στη συνέχεια να χρησιμοποιήσουν στη πράξη.

Στη συνέχεια αξιολογήσαμε το σύστημα ως προς τα παιδαγωγικά χαρακτηριστικά του. Βλέπουμε ότι μια βασική ιδιότητα του συστήματος είναι η αλληλεπίδραση χρηστών και συστήματος. Πράγματι μέσα από τη βοήθεια που προσφέρεται, τον έλεγχο και την υπόδειξη των λαθών από το σύστημα προς τους χρήστες δημιουργείται σχέση αλληλεπίδρασης. Τα γραφικά του συστήματος βοηθούν στη διατήρηση του ενδιαφέροντος των χρηστών. Το σύστημα δε δίνει τη λύση των σεναρίων στους χρήστες αλλά βοηθά το χρήστη να μάθει να χρησιμοποιεί κάθε εργαλείο σωστά κι έτσι να τον οδηγεί σε μια σωστή επίλυση. Προάγεται έτσι η κριτική σκέψη των χρηστών. Επίσης το σύστημα δίνει τη δυνατότητα στους χρήστες να προχωρούν με το δικό τους ρυθμό αφού μπορούν να αποθηκεύσουν και να φορτώσουν το σημείο που θέλουν να σταματήσουν και να συνεχίσουν αντίστοιχα.

Η αξιολόγηση συνεχίζεται ως προς την ευκολία χρήσης του συστήματος. Βλέπουμε ότι το σύστημα προσφέρει βοήθεια στους χρήστες και επεξηγήσεις όπου χρειάζονται. Το παραθυρικό και γραφικό περιβάλλον βοηθούν στην άμεση κατανόηση των λειτουργιών του συστήματος. Δε χρειάζονται ιδιαίτερες δεξιότητες από τη πλευρά του χρήστη. Μόνο οι βασικές λειτουργίες επιλογής, συμπλήρωσης παραμέτρων. Ότι δηλαδή γνωρίζει κι ένα παιδάκι της σημερινής εποχής. Το σύστημα προσφέρει εύκολη πλοήγηση μέσω των μενού αλλά και των κουμπιών που υπάρχουν σε κάθε διεπαφή. Υπάρχει ακόμα δυνατότητα αλλαγής σεναρίου σε περίπτωση δυσκολίας δίνοντας ευελιξία στους χρήστες.

Αξιολογώντας τα λειτουργικά χαρακτηριστικά του συστήματος βλέπουμε αρχικά ότι εφαρμόζεται η διαλειτουργικότητα. Δηλαδή υπάρχει δυνατότητα αντιγραφής, επικόλλησης, και αποθήκευσης κειμένου, π.χ. σεναρίου ή άλλου υλικού, π.χ. του σχεδιαγράμματος της Μεθόδου του Κρισίμου Δρόμου (CPM), ή της Τεχνικής Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT) κ.α. Όπως διαπιστώνουμε υπάρχει δυνατότητα αποθήκευσης των δεδομένων σε μια διεπαφή ώστε την επόμενη φορά ο χρήστης να μπορεί να ξεκινήσει από το σημείο που έκανε αποθήκευση. Επίσης, υπάρχει δυνατότητα αντιγραφής δεδομένων και επικόλλησης π.χ. σε ένα notepad αρχείο με συνδυασμό των Ctrl + C για αντιγραφή και Ctrl + V για επικόλληση. Δυνατότητα εκτύπωσης δεν υπάρχει μέσα από το σύστημα. Το σύστημα εγκαθίσταται με απλή και άμεση διαδικασία απλά τρέχοντας το Setup και ακολουθώντας τα βήματα, όπως άλλωστε γίνεται με οποιοδήποτε πρόγραμμα. Το σύστημα τέλος, λειτουργεί χωρίς σφάλματα κώδικα (χωρίς bugs) και είναι αξιόπιστο χωρίς προβλήματα.

7.2 Πρόταση Πειραματικής Αξιολόγησης

7.2.1 Εισαγωγή

Η πρόταση πειραματικής αξιολόγησης έχει σκοπό να αξιολογήσει την αποτελεσματικότητα του συστήματος υποβάλλοντάς το σε μια σειρά πειραμάτων (σύγκριση γνώσης που αποκτάται από τους/τις φοιτητές/τριες παραδοσιακής μάθησης και μάθησης μέσω του συστήματος). Με τον όρο «αποτελεσματικότητα» εννοούμε πόσο καλά γίνεται η μετάδοση της γνώσης και πόσο χρηστικό είναι το σύστημα. Στη συνέχεια αναπτύσσουμε την πρόταση πειραματικής αξιολόγησης την οποία προτείνουμε ξεκινώντας με τη σχεδίαση του πειράματος.

7.2.2 Σχεδίαση του πειράματος

Ορίζουμε αρχικά δύο ομάδες αξιολόγησης που χωρίζονται σε «Ομάδα Παραδοσιακής Εκπαίδευσης» και «Ομάδα Σύγχρονης Εκπαίδευσης». Η «Ομάδα Παραδοσιακής Εκπαίδευσης» παρακολουθεί το μάθημα Διοίκησης Έργων που γίνεται στη τάξη για ένα χρονικό διάστημα μέχρι να αποκτήσει επαρκή γνώση της Μεθόδου του Κρισίμου Δρόμου (CPM), της Τεχνικής Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT), της Δομής Ανάλυσης Εργασιών (WBS), και του Πίνακα Αρμοδιοτήτων (RACI). Η «Ομάδα Σύγχρονης Εκπαίδευσης» παρακολουθεί το μάθημα όπως και η «Ομάδα Παραδοσιακής Εκπαίδευσης» αλλά παράλληλα κάνει κάποιες ώρες σύγχρονης διδασκαλίας με το σύστημα **Trainer**© που αναπτύχθηκε. Οι ώρες αυτές δεν είναι επιπλέον αλλά πρέπει να ίσες ώρες διδασκαλίας και για τις δύο ομάδες. Αυτό μπορεί να γίνει με τη μεταφορά της «Ομάδας Σύγχρονης Εκπαίδευσης» στους υπολογιστές για εκπαίδευση με το σύστημα **Trainer**© όταν η «Ομάδα Παραδοσιακής Εκπαίδευσης» θα επιλύει ασκήσεις με τη Μέθοδο του Κρισίμου Δρόμου (CPM), τη Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT – Program Evaluation and Review Technique), τη Δομή Ανάλυσης Εργασιών (WBS), και το Πίνακα Αρμοδιοτήτων (RACI) στη τάξη.

Στη συνέχεια και μετά το πέρας κάποιου χρονικού διαστήματος κατανόησης της θεωρίας και τρόπου επίλυσης ασκήσεων και σεναρίων Διοίκησης Έργων (π.χ. 6 μήνες) δίνονται αναθέσεις θεμάτων Διοίκησης Έργων (επίλυση σεναρίων με τις μεθόδους/τεχνικές CPM, PERT, WBS, RACI) στις δύο ομάδες. Η «Ομάδα Παραδοσιακής Εκπαίδευσης» επιλύει τα σενάρια και τις ασκήσεις με τον παραδοσιακό τρόπο (στο χαρτί) ενώ η «Ομάδα Σύγχρονης Εκπαίδευσης» μέσω του συστήματος **Trainer**©. Για να υπάρχει πλήρης ισότητα δε θα υπάρχει δυνατότητα βοήθειας από το σύστημα **Trainer**© στους χρήστες. Το σύστημα **Trainer**© θα τροποποιηθεί ώστε να είναι ένα εξεταστικό εργαλείο κι όχι εκπαιδευτικό σε ότι αφορά το πείραμα που θα λάβει χώρα.

Αξιολογούνται και οι δύο ομάδες ως προς την ορθότητα της επίλυσης των σεναρίων και ως προς το χρόνο επίλυσης. Μελετώντας τις διαφορές στις βαθμολογίες των δύο ομάδων εξάγονται τα σχετικά συμπεράσματα.

8 Μελλοντικές Βελτιώσεις του Συστήματος

Σε ότι αφορά τις μελλοντικές βελτιώσεις του συστήματος προτείνονται τα εξής:

- Εμπλουτισμός του περιεχομένου του συστήματος με τη προσθήκη επιπλέον σεναρίων διαβαθμισμένης δυσκολίας.

- Εκτός από τον έλεγχο και υπόδειξη των λαθών επίλυσης του σεναρίου στους χρήστες από το σύστημα, θα μπορούσε να δίνεται μια επεξήγηση του λάθους. Η επεξήγηση θα έχει σκοπό την αντιμετώπιση του λάθους δημιουργικά και θα γίνεται εφόσον ζητηθεί από το χρήστη ή θα γίνεται εάν ο χρήστης είναι «Αρχάριος» όπως ορίζεται στο κεφάλαιο 5***. Παραμετροποίηση του Συστήματος.***

- Εμπλουτισμός των γραφικών του συστήματος ώστε να προκαλείται και να διατηρείται το ενδιαφέρον των χρηστών ακόμη περισσότερο.

- Να προστεθεί ήχος στις διάφορες λειτουργίες του συστήματος. Παράδειγμα, όταν ο χρήστης κάνει ένα λάθος να παίζει ένας δυσάρεστος ήχος κι όταν ολοκληρώνει με επιτυχία ένα σενάριο να παίζει ένας ευάρεστος ήχος.

- Πρόσθεση της επιλογής εκτύπωσης μέσα από το σύστημα (εκτύπωση σεναρίου, διαγράμματος επίλυσης σεναρίου, στατιστικών στοιχείων κ.α.).

9 Συμπεράσματα

Δεν είναι αρκετό να τρέξουμε ένα μοντέλο προσομοίωσης σαν ένα μαύρο κουτί, χωρίς οι χρήστες να γνωρίζουν τη λειτουργία του. Οι χρήστες πρέπει να μάθουν πως λειτουργεί ώστε να κερδίσουν εσωτερική γνώση στη διαδικασία ανάπτυξης του λογισμικού και να αποδεχτούν να αλλάξουν το δικό τους νοητικό μοντέλο. ***Οι χρήστες πρέπει να αναλύουν τα λάθη τους, έτσι ώστε να κερδίσουν επαρκή γνώση του προβλήματος*** (Phahl, Klemm, & Ruhe, 2001)**.**

Μετά τη δοκιμή του συστήματος και την αξιολόγησή του συμπεραίνουμε ότι, το σύστημα ικανοποιεί το βασικό του στόχο που είναι η εκπαίδευση των χρηστών στη Μέθοδο του Κρισίμου Δρόμου (CPM), στη Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT), στη Δομή Ανάλυσης Εργασιών (WBS), και στο Πίνακα Αρμοδιοτήτων (RACI) της Διοίκησης Έργων, τα οποία περιλαμβάνονται στη μέχρι τώρα έκδοση του εκπαιδευτικού λογισμικού **Trainer©**. Φυσικά απομένει να γίνει η πειραματική αξιολόγηση του συστήματος. Μετά την πειραματική αξιολόγηση θα είμαστε σε θέση να πούμε τελικά πόσο καλύτερα δουλεύουν οι φοιτητές με σύμμαχο το νέο εκπαιδευτικό λογισμικό.

Όπως ήδη αναφέρθηκε σημαντικό είναι οι χρήστες να αποκτήσουν εσωτερική γνώση για το πώς αναπτύχθηκε το εκπαιδευτικό λογισμικό καθώς και για τον τρόπο λειτουργίας του. Αυτό μπορεί να γίνει με μια ενημέρωση –τύπου σεμιναρίων– όπου θα αναλύονται όλες οι πτυχές του εκπαιδευτικού λογισμικού **Trainer©**. Στη συνέχεια οι χρήστες θα είναι σε θέση να χρησιμοποιούν τα διάφορα εργαλεία Διοίκησης Έργου που παρέχονται από το εκπαιδευτικό λογισμικό γνωρίζοντας τον τρόπο που δουλεύει κάθε εργαλείο.

10 Βιβλιογραφία

Bista, B. (2008, 9 19). *Project Management: History and Evolution.* Ανάκτηση 9 19, 2008, από http://EzineArticles.com/?expert=Bharat_Bista

Feurzeig, W., & John, R. (1988). *"Intelligent Tools for Algebra", Technology and Learning,* IEEE Annals of the History of Computing, vol. 28, no. 1, pp. 18-31, Jan.-Mar. 2006.

Harel, I., & Seymour, P. (1991). *Constructionism.* Norwood: Ablex Publishing Corp.

Haugan. (2003). *The Work Breakdown Structure in Government Contracting,* Management Concepts.

Holzner, S. (1998). *Visual Basic 6 Black Book.* The Coriolis Group.

Johnson, J. M. (2003). *From Lofty Beginnings to the Age of Accessibility.* U.S. & Canada: ISTE (International Society for Technology in Education), Learning and Leading with Technology, 30(7), 6-13.

Kloppenborg, T. J. (2008). *History Of Project Management.* Ανάκτηση 8 19, 2008, από http://www.referenceforbusiness.com/encyclopedia/Per-Pro/Project-Management.html#HISTORY_OF_PROJECT_MANAGEMENT

Maxwell, D. (2003, 11 2). Activity on Node Approach to CPM Scheduling.

Means, B. (2008). *Enhancing Skills Through Distant Mentoring.* SRI International. Menlo Park: SRI International.

National Institute of Open Schooling. (2008). *NOS : Certificate in Computer Applications.* Ανάκτηση από NOS : Certificate in Computer Applications: http://www.nos.org/index.htm

O'Brien, J. J., & Plotnick, F. L. (1999). *CPM in Construction Management.* McGraw-Hill Professional.

Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas.* New York: NY: Basic Books, Inc.

Phahl, D., Klemm, M., & Ruhe, G. (2001). A CBT module with integrated simulation component for software project management education and training. *The Journal of Systems and Software* , σσ. 283-298.

PMBOK. (1996). *A guide to the project management body of knowledge.* North Carolina, USA: PMI.

PMI. (2006). Project Management Institute Practice Standard for Work Breakdown Structures. *II.* Project Management Institute.

PMI. (2004). *A Guide to the Project Management Body of Knowledge* (Third Edition εκδ.). Newtown Square, Pennsylvania, USA: Project Management Institute (PMI).

Smith, M. L., & Erwin, J. (2008). *Role & Responsibility Charting (RACI).*

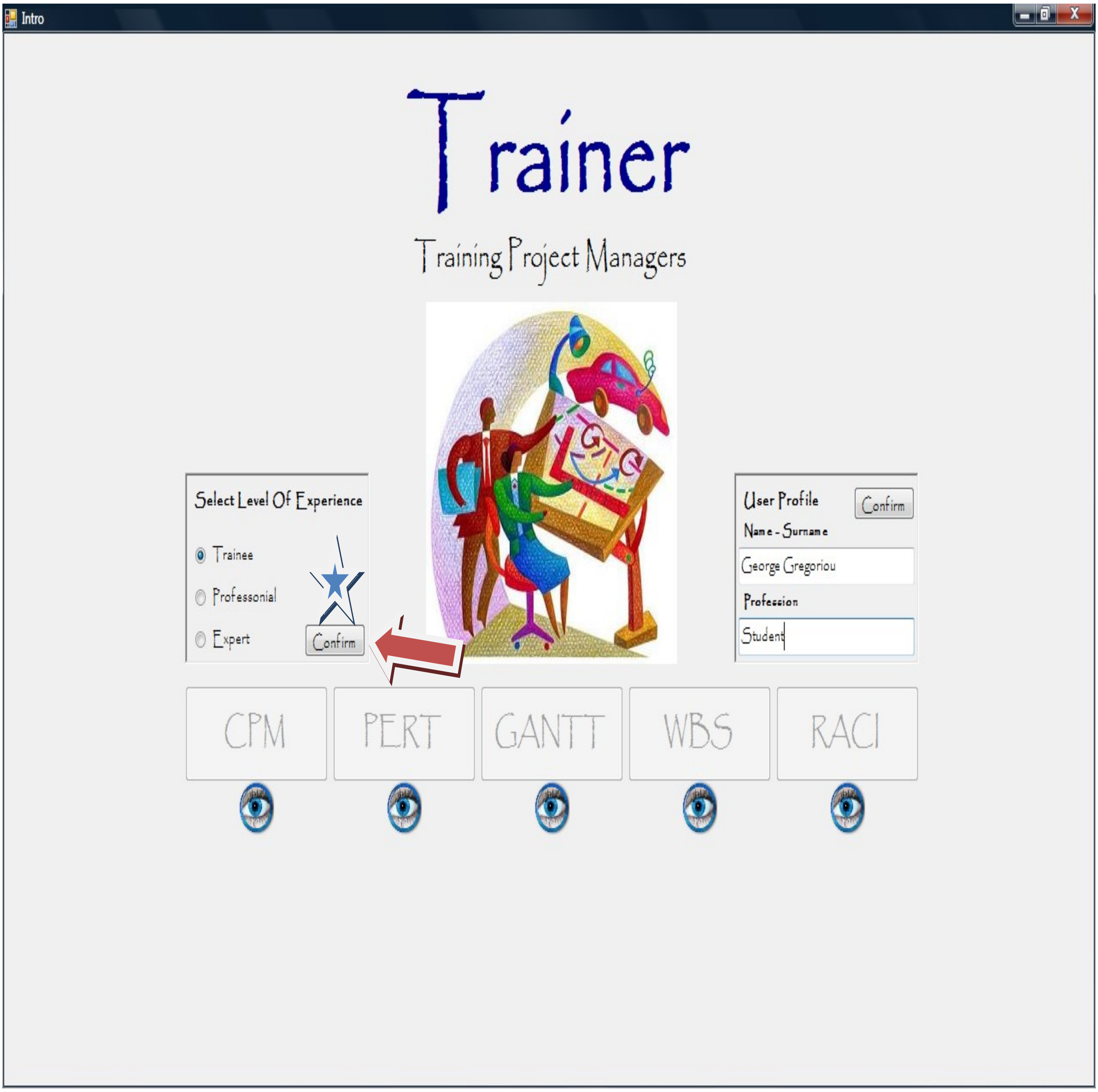
Stanley, E. (2008). *Learning Science Through Guided Discovery.* Boston University. Boston: Science and Mathematics Education Center.

Γαρυφαλλίδου, Ιωαννίδης, Σκέλλας, & Τσιτσίρης. (2003). *Εκπαιδευτικό Λογισμικό, Πολυμέσα και Internet – Σύγκριση με τις παραδοσιακές μεθόδους.* Πάτρα: Εργαστήριο Θετικών Επιστημών, Π.Τ.Δ.Ε. Πανεπιστημίου Πατρών.

Κηρυττόπουλος, Κ., & Διαμάντας, Β. (2006). *Η διαχείριση κινδύνων έργων στην κατασκευαστική βιομηχανία.* Αθήνα.

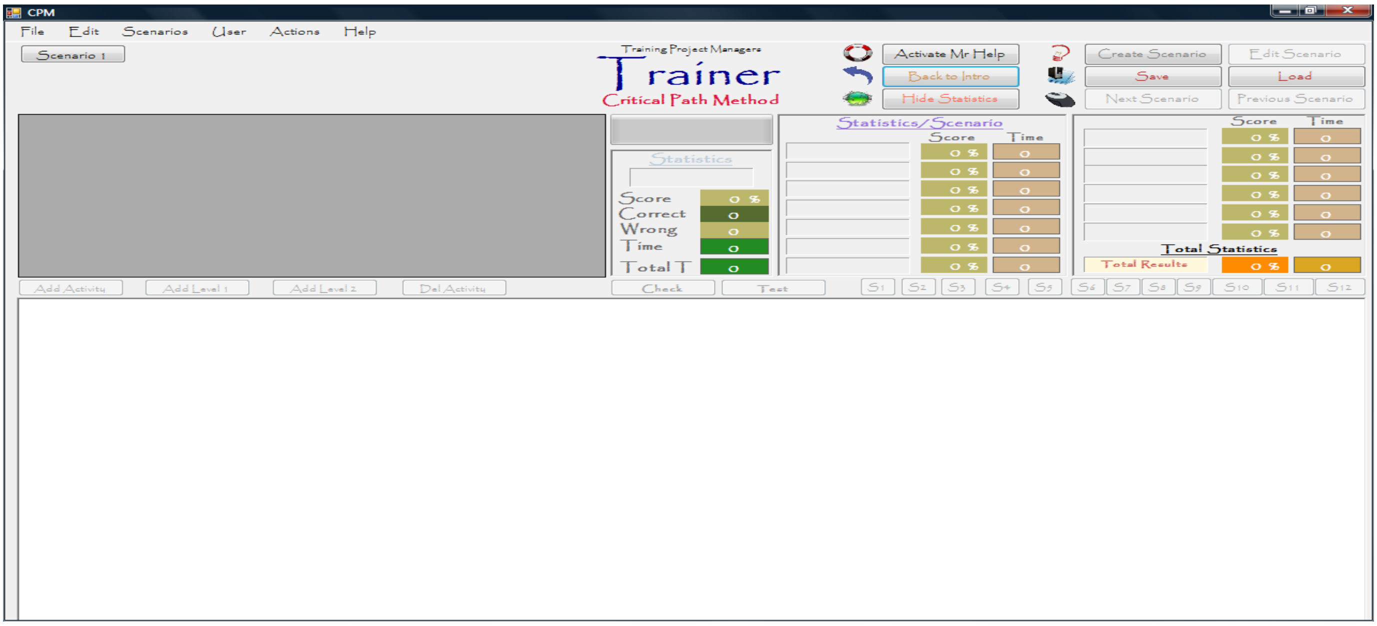
11 Παραρτήματα

Εισαγωγική Διεπαφή του εκπαιδευτικού λογισμικού **Trainer**©. Αφού ο χρήστης επιλέξει το επίπεδο εμπειρίας τότε ενεργοποιούνται οι επιλογές για τη μέθοδο/εργαλείο CPM, PERT, WBS, RACI.



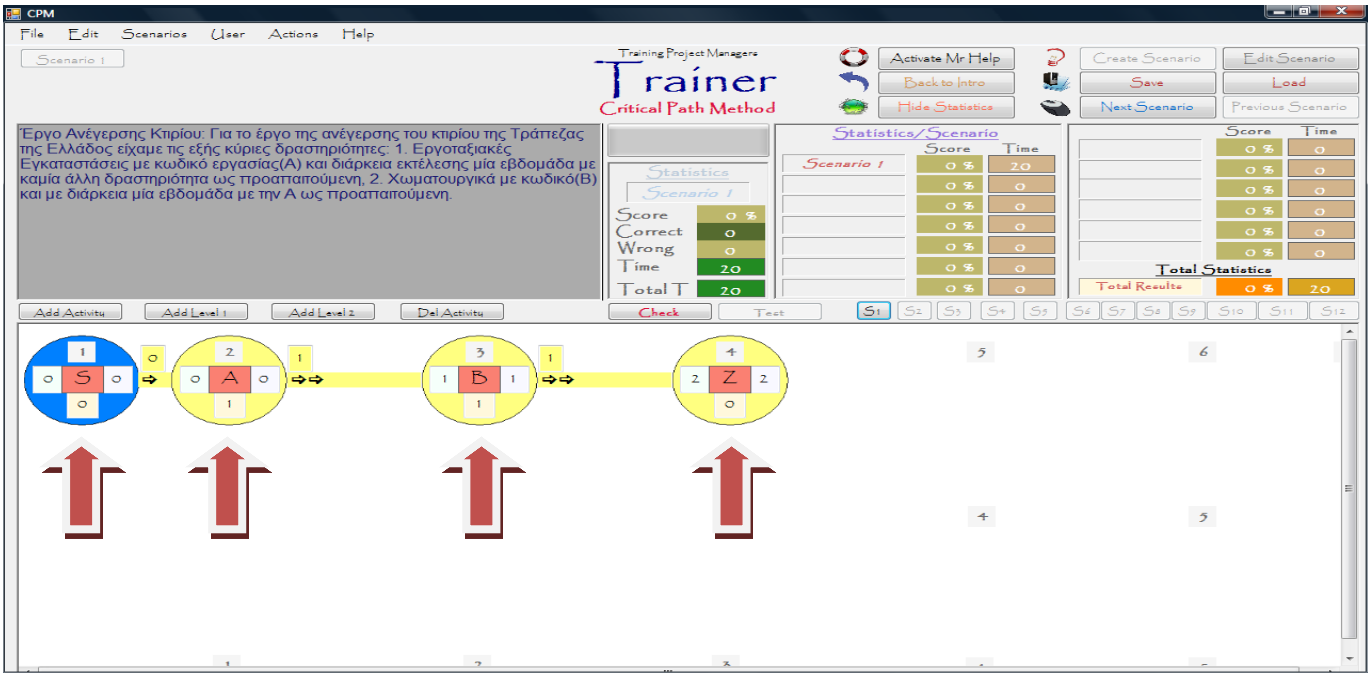
Εισαγωγική Διεπαφή του εκπαιδευτικού λογισμικού **Trainer©** μετά την επιλογή του επιπέδου εμπειρίας του χρήστη και στην συνέχεια την επιλογή παροχής πληροφορίας για τη μέθοδο CPM.

Διεπαφή CPM.

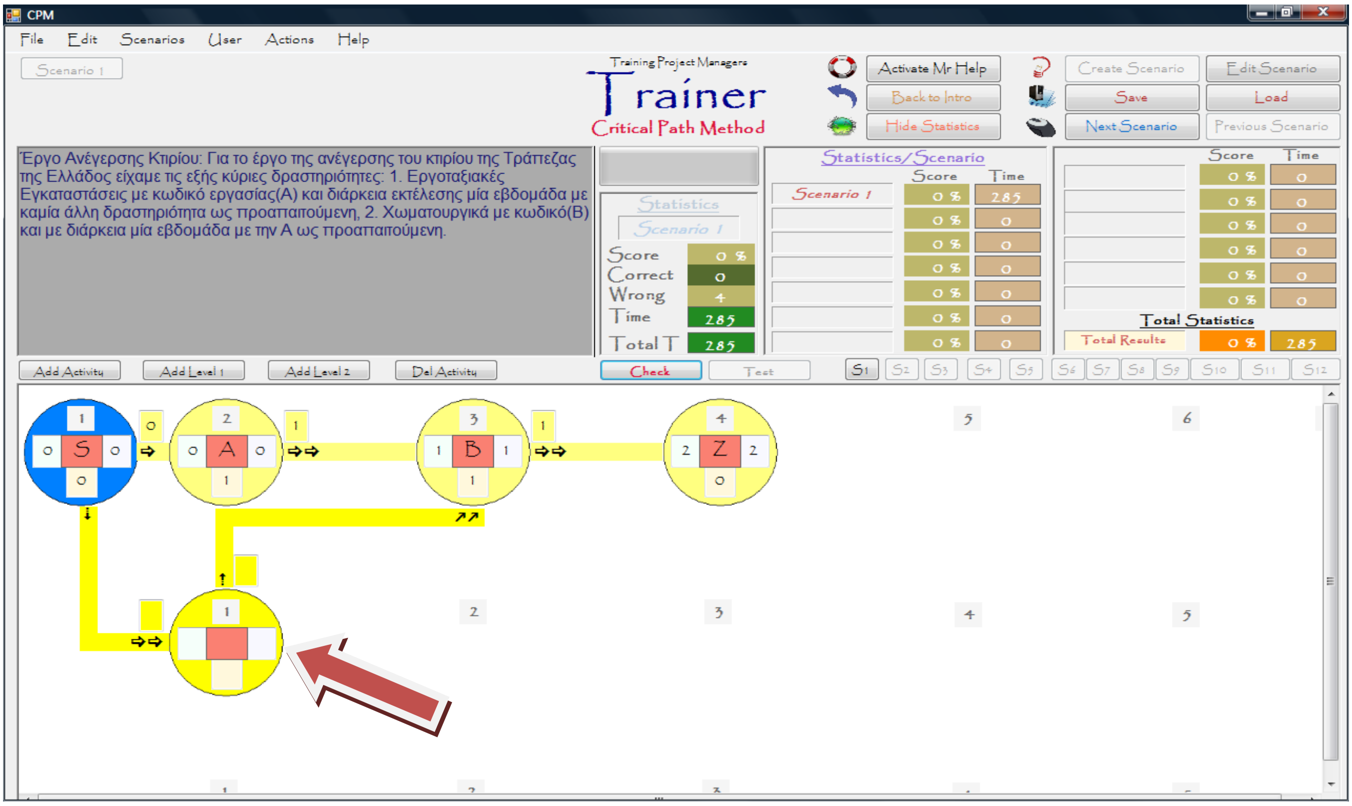


Αφού επιλέξουμε το ***Scenario 1*** (εμφανίζεται στο γκρι παράθυρο), προσθέτουμε τις αντίστοιχες δραστηριότητες (επιλέγοντας ***Add Activity*** ή πατώντας πάνω στον **ΧΕΧ**). Βλέπουμε επίσης ότι ξεκίνησε να μετράει ο χρόνος επίλυσης του σεναρίου (πράσινο κουτάκι).

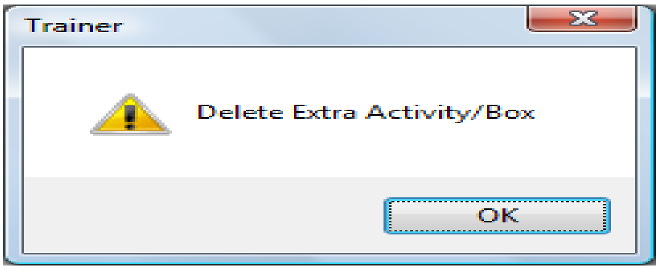
Συνδέουμε τις δραστηριότητες μεταξύ τους κάνοντας κλικ π.χ. στο κύκλο της δραστηριότητας ***S*** και στο κύκλο της δραστηριότητας ***A*** ή και αντίστροφα. Στη συνέχεια συμπληρώνουμε τα κουτάκια σύμφωνα με το σενάριο.



Προσθέτουμε μία λάθος δραστηριότητα και όταν κάνουμε έλεγχο – ***Check*** το σύστημα μας βγάζει μήνυμα λάθους.

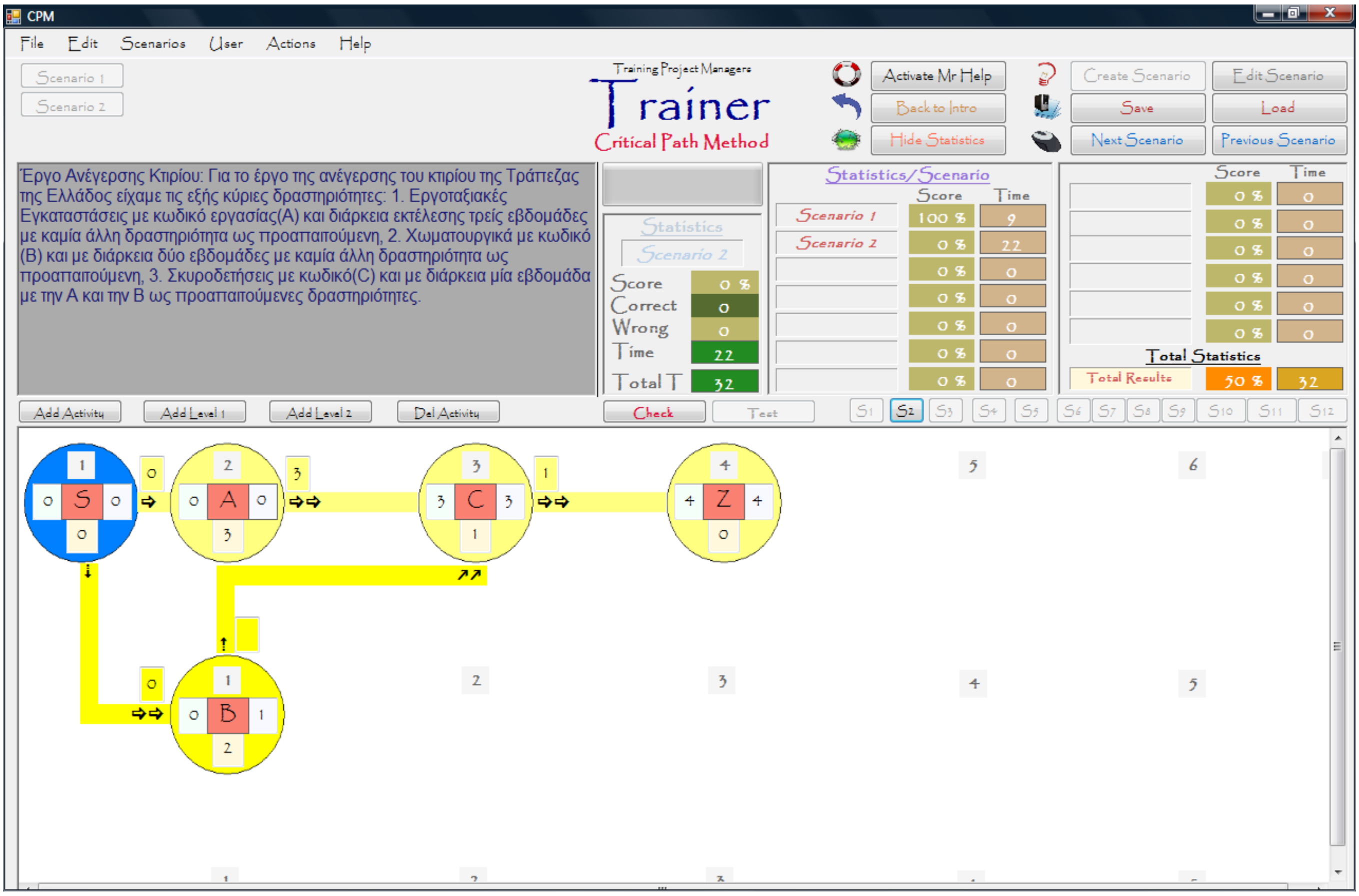


Μήνυμα λάθους για να γίνει διόρθωση από το χρήστη. Το ίδιο συμβαίνει αν ο χρήστης εισάγει λάθος δεδομένα στα κουτάκια.

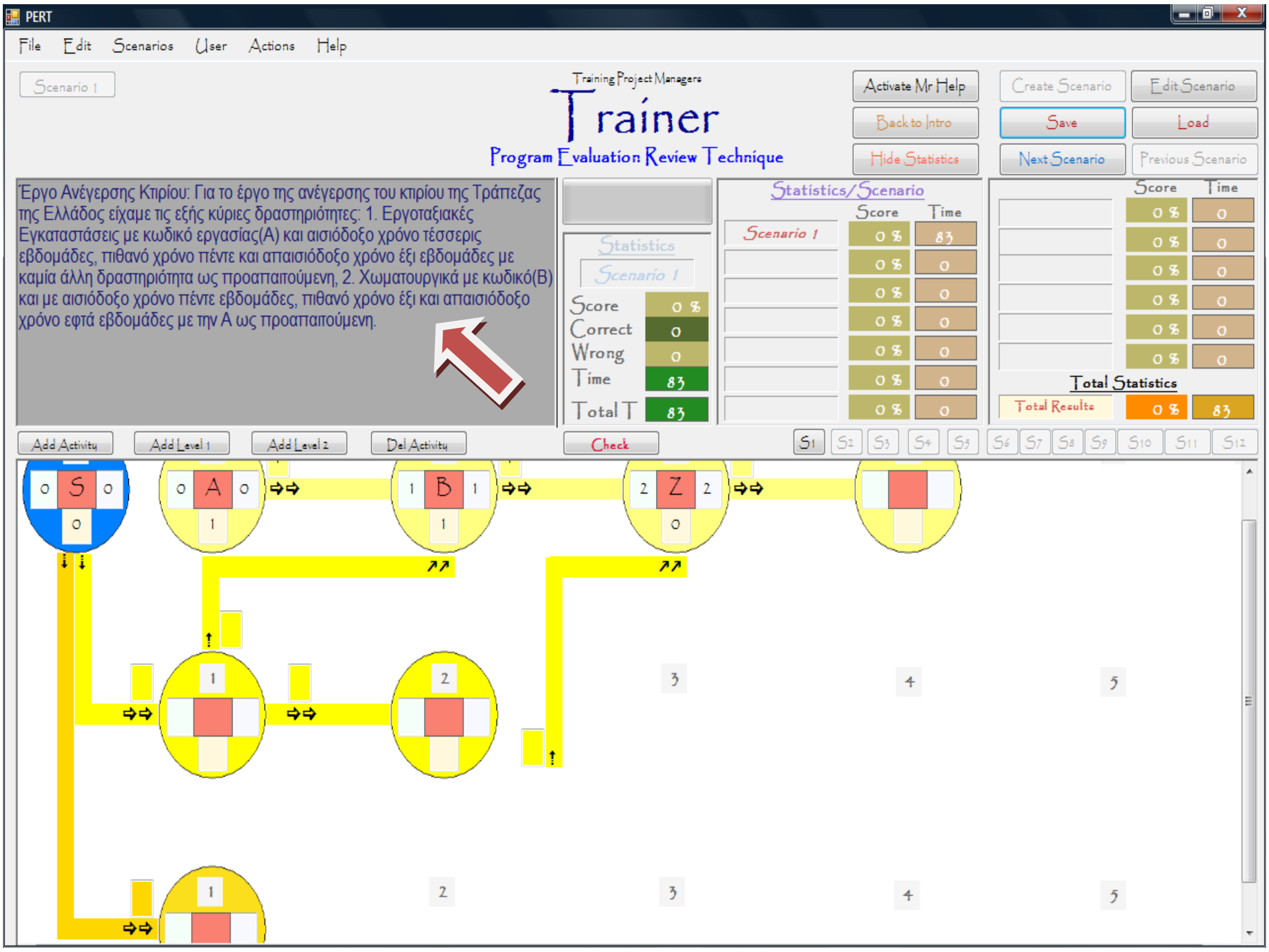


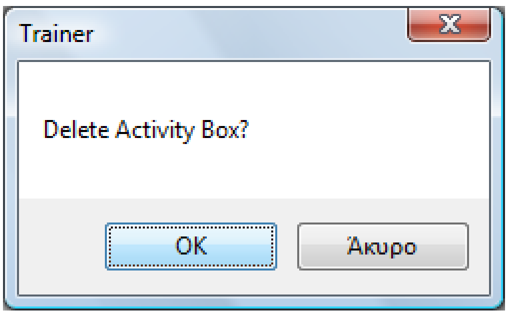
Αφού διαγράψουμε τη λάθος δραστηριότητα και ξανακάνουμε έλεγχο – ***Check*** το σύστημα βγάζει μήνυμα επιτυχίας.

Προχωρώντας στο δεύτερο σενάριο. Στα δεξιά εμφανίζονται τα στατιστικά αποτελέσματα.

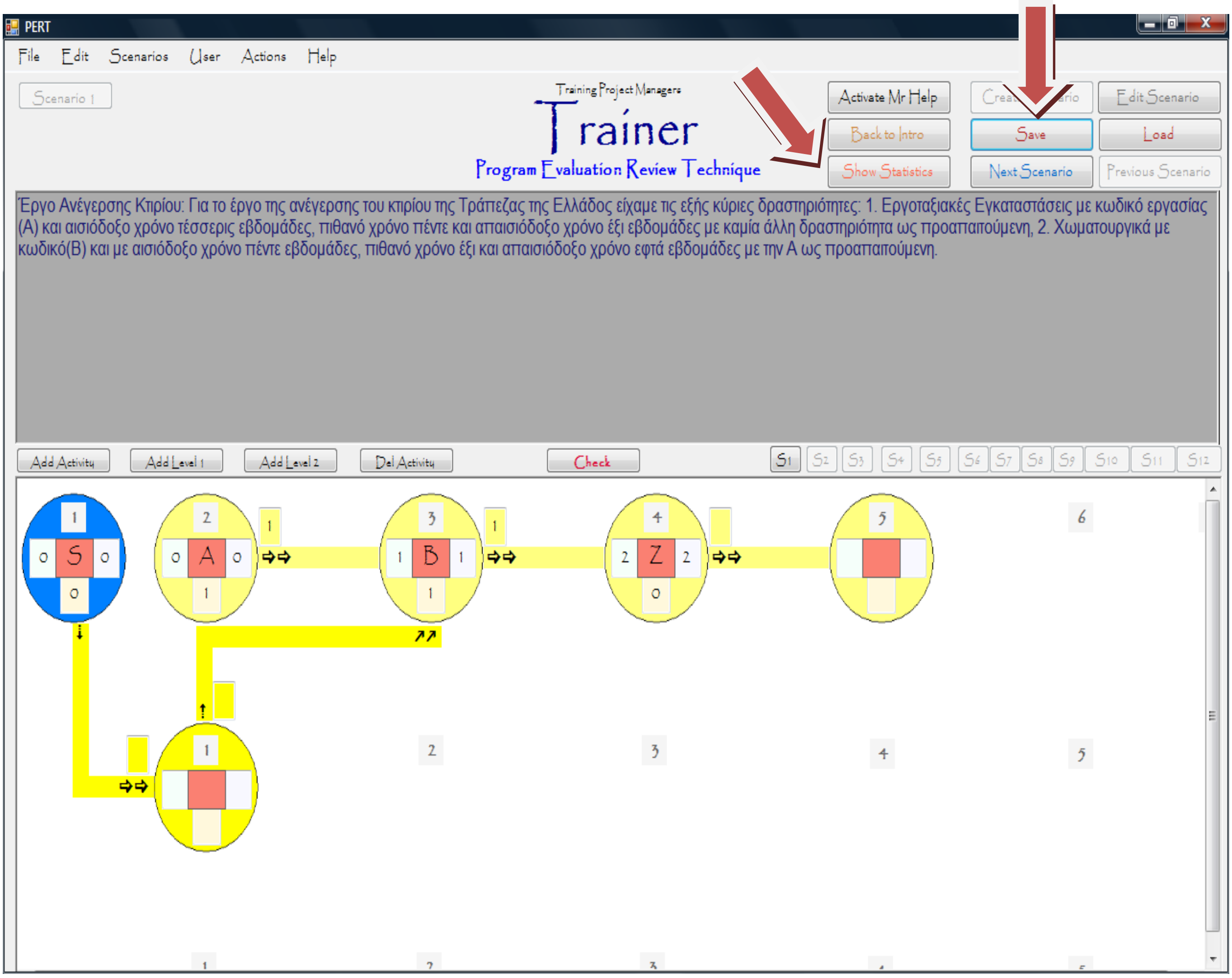


Διεπαφή PERT. Η διαφορά με τη Διεπαφή CPM είναι στο σενάριο το οποίο δεν αναφέρει διάρκεια δραστηριότητας αλλά τους χρόνους για το αισιόδοξο, πιθανό, και απαισιόδοξο σενάριο όπου με χρήση του σχετικού τύπου ο χρήστης μπορεί να εξάγει τον αναμενόμενο χρόνο της κάθε δραστηριότητας.

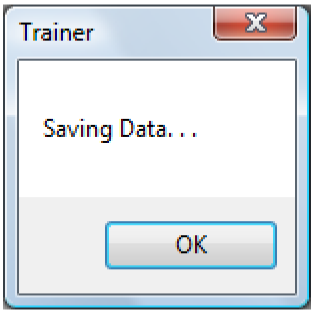




Αφού κάνουμε δεξί κλικ πάνω σε μια δραστηριότητα μας εμφανίζεται μήνυμα διαγραφής. Αν επιλέξουμε **OK** τότε η δραστηριότητα διαγράφεται μαζί με όλες τις συνδέσεις της.

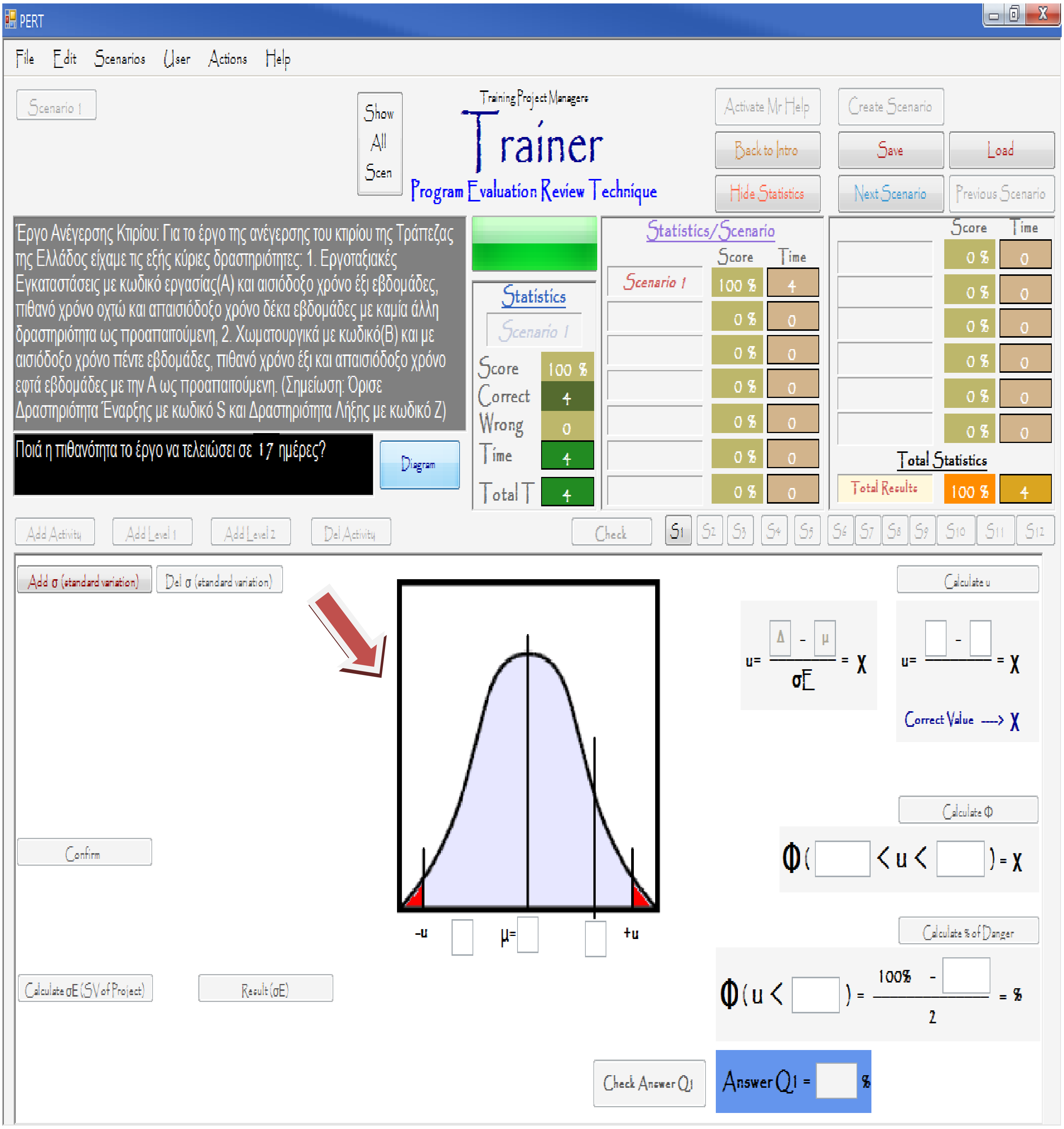


Μετά τη διαγραφή δραστηριοτήτων και την επιλογή **Hide Statistics** όπου εμφανίζεται το σενάριο σε όλο το πλάτος της οθόνης.



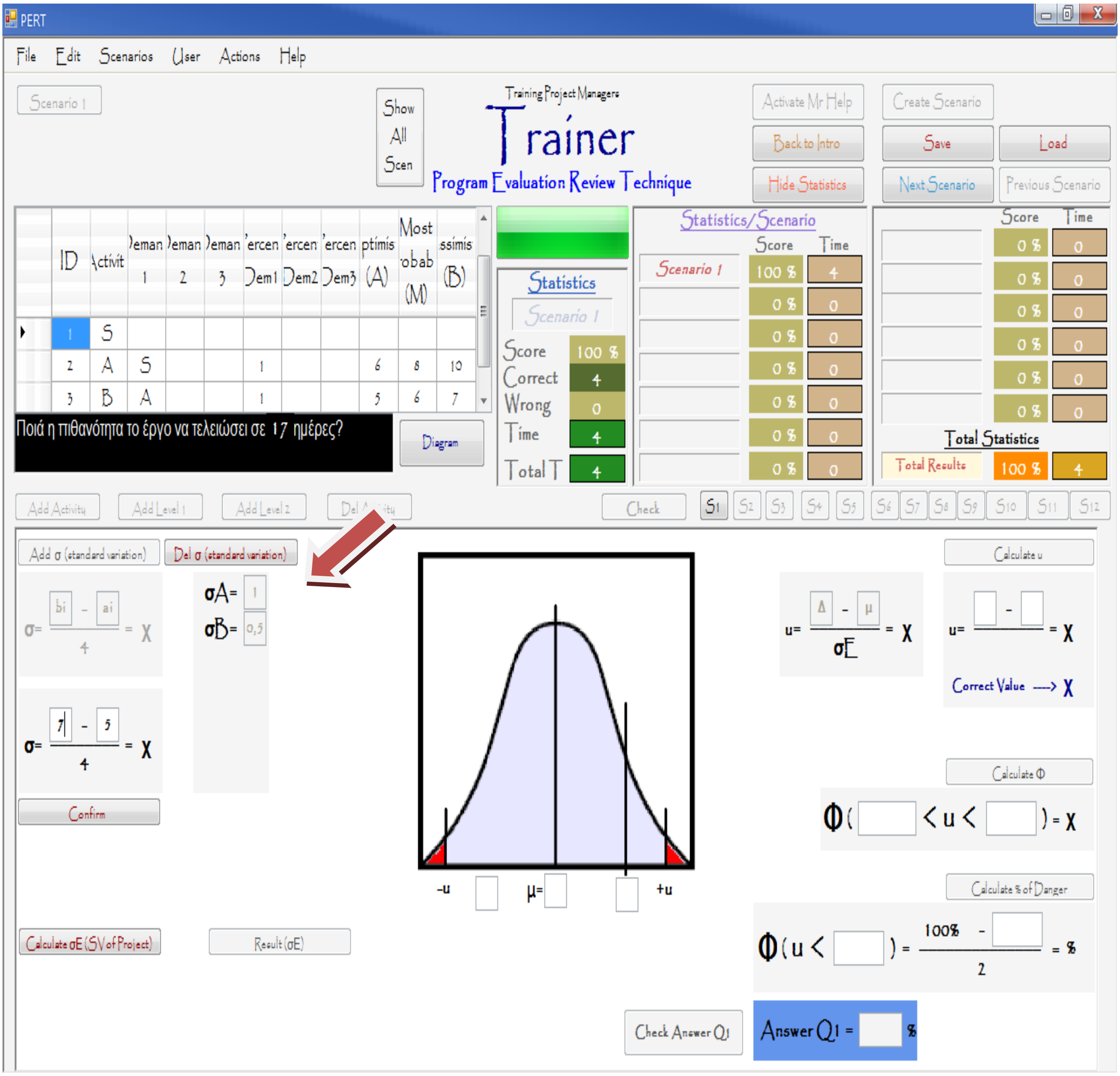
Αφού επιλέξουμε **Save** το σύστημα μας σώζει το σημείο στο οποίο βρισκόμαστε έτσι ώστε όταν κάνουμε **Load** να επανέλθουμε στο ίδιο σημείο.

Πατώντας στο κουμπί (Graphic) μας εμφανίζεται ένας καινούριος χώρος εργασίας του χρήστη, όπου μπορεί να απαντήσει στα ερωτήματα σχετικά με την πιθανότητα ολοκλήρωσης του έργου και με την πιθανότητα υπέρβασης του χρόνου αποπεράτωσης του.

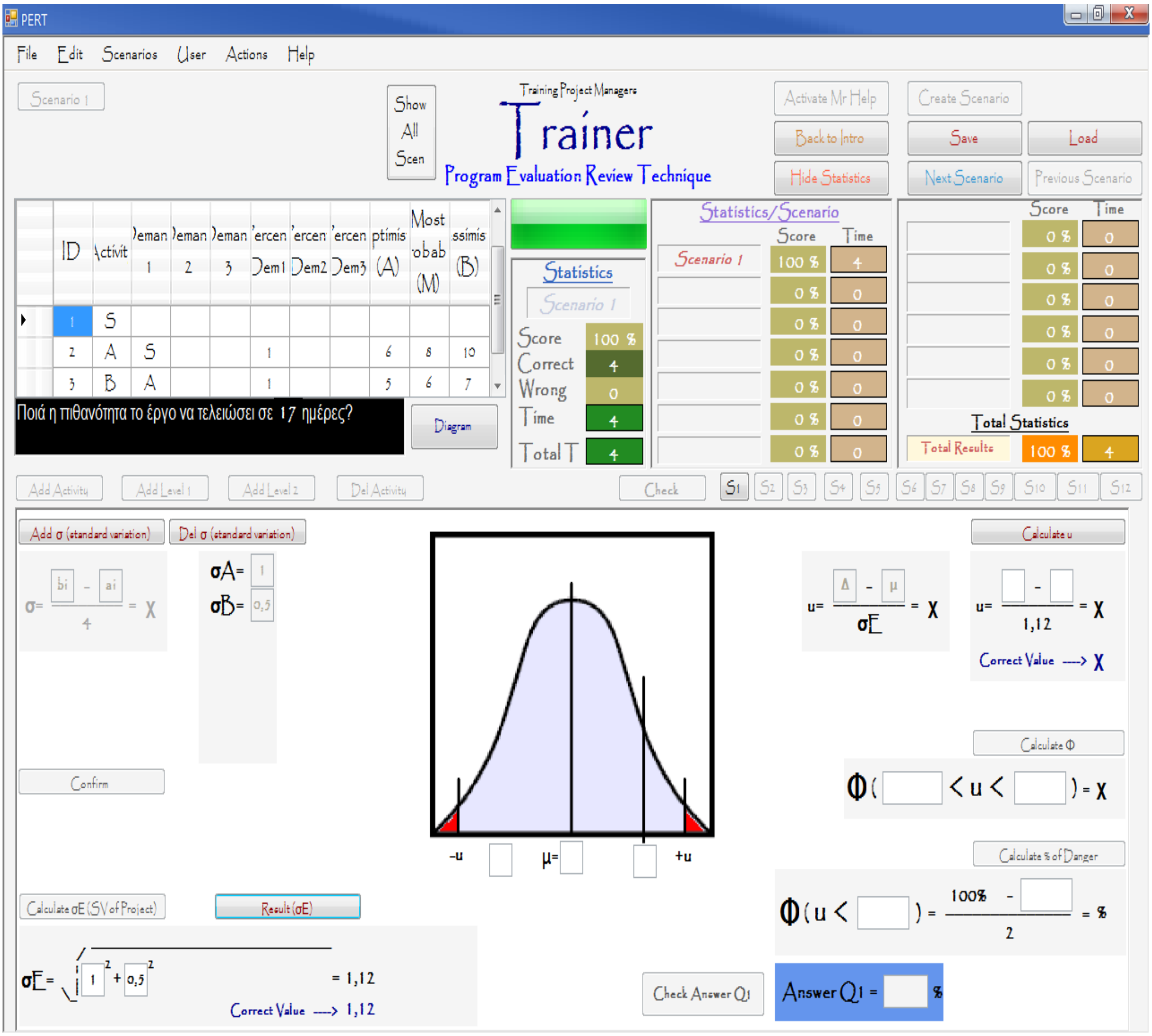


Ο χρήστης ξεκινάει απαντώντας το πρώτο ερώτημα, ακολουθώντας τα παρακάτω βήματα:

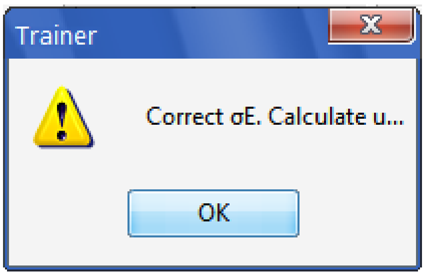
1. Εισάγει στον τύπο (bi – ai) / 4 τις αντίστοιχες τιμές με βάση το σενάριο και πατώντας στο κουμπί (Calculate σ) το σύστημα υπολογίζει το αντίστοιχο αποτέλεσμα.



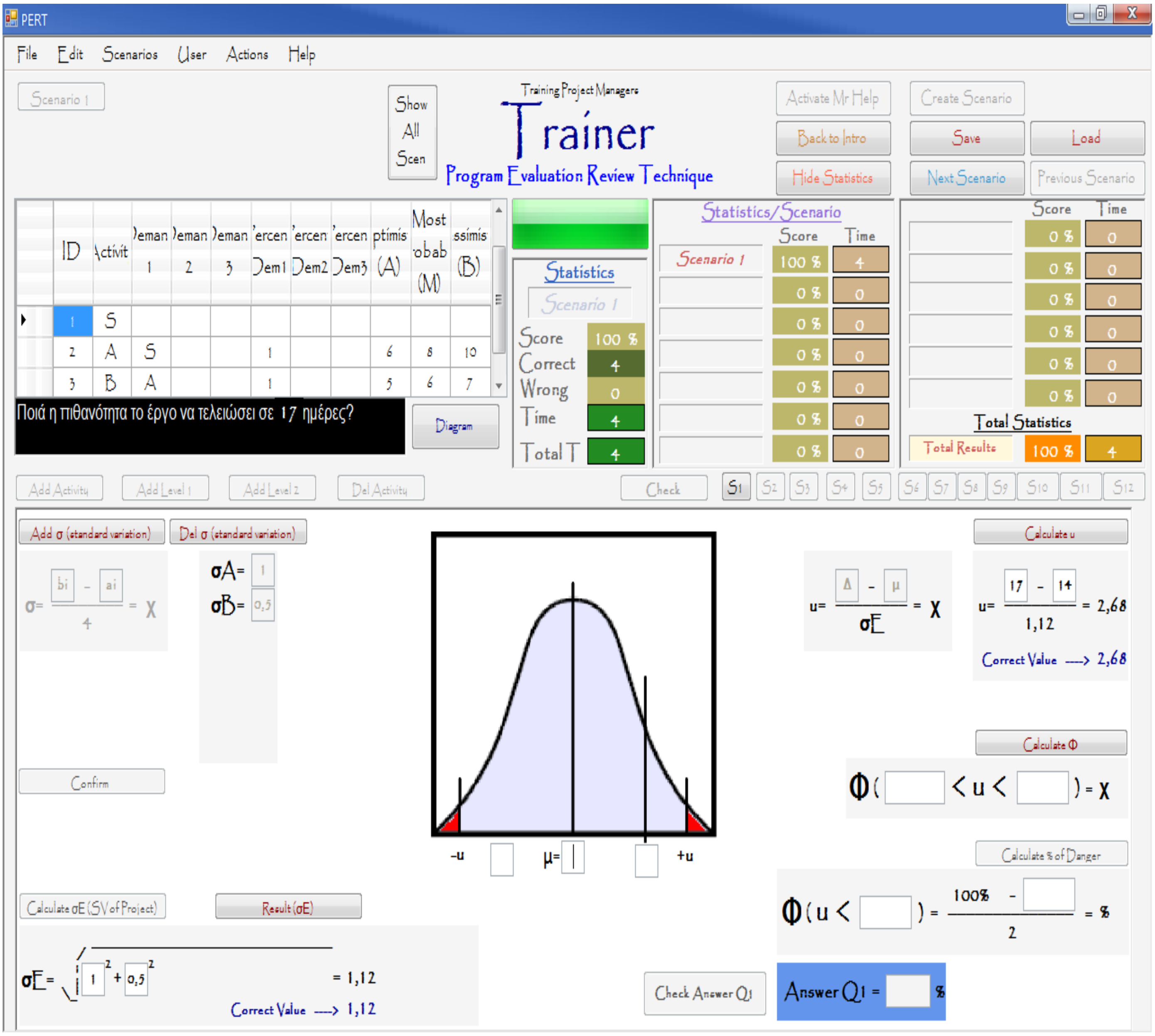
2. Στη συνέχεια ο χρήστης επιλέγει το κουμπί (Calculate σE) και εισάγει τις αντίστοιχες τιμές που βρήκε από τη χρήση του τύπου (bi – ai) / 4.

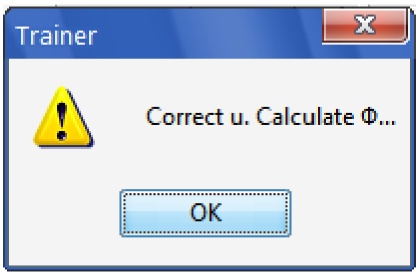


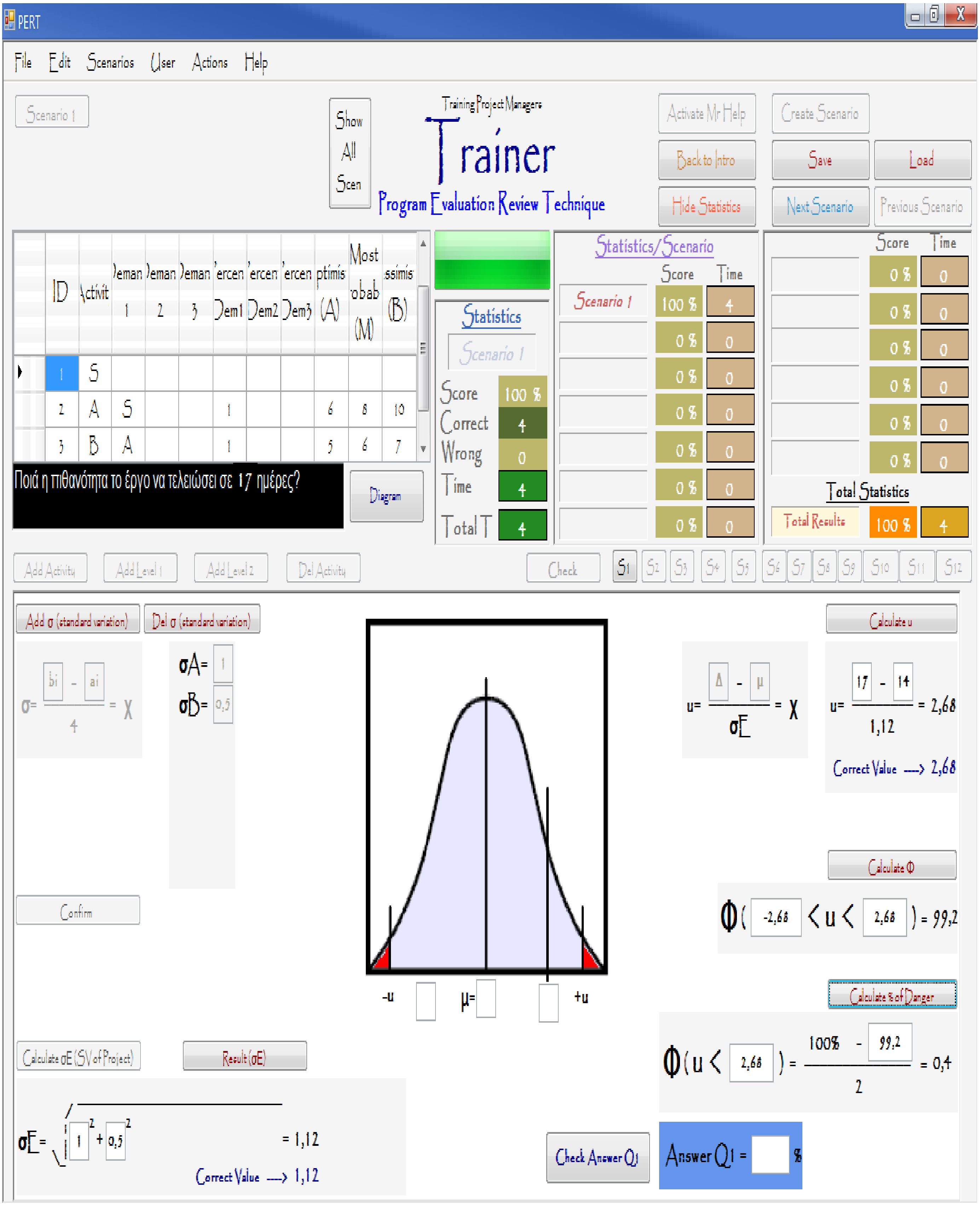
Αφού επιλέξουμε το κουμπί (Result σE) το σύστημα ελέγχει την απάντηση και μας βγάζει μήνυμα να συνεχίσουμε εφόσον το αποτέλεσμα είναι ορθό.



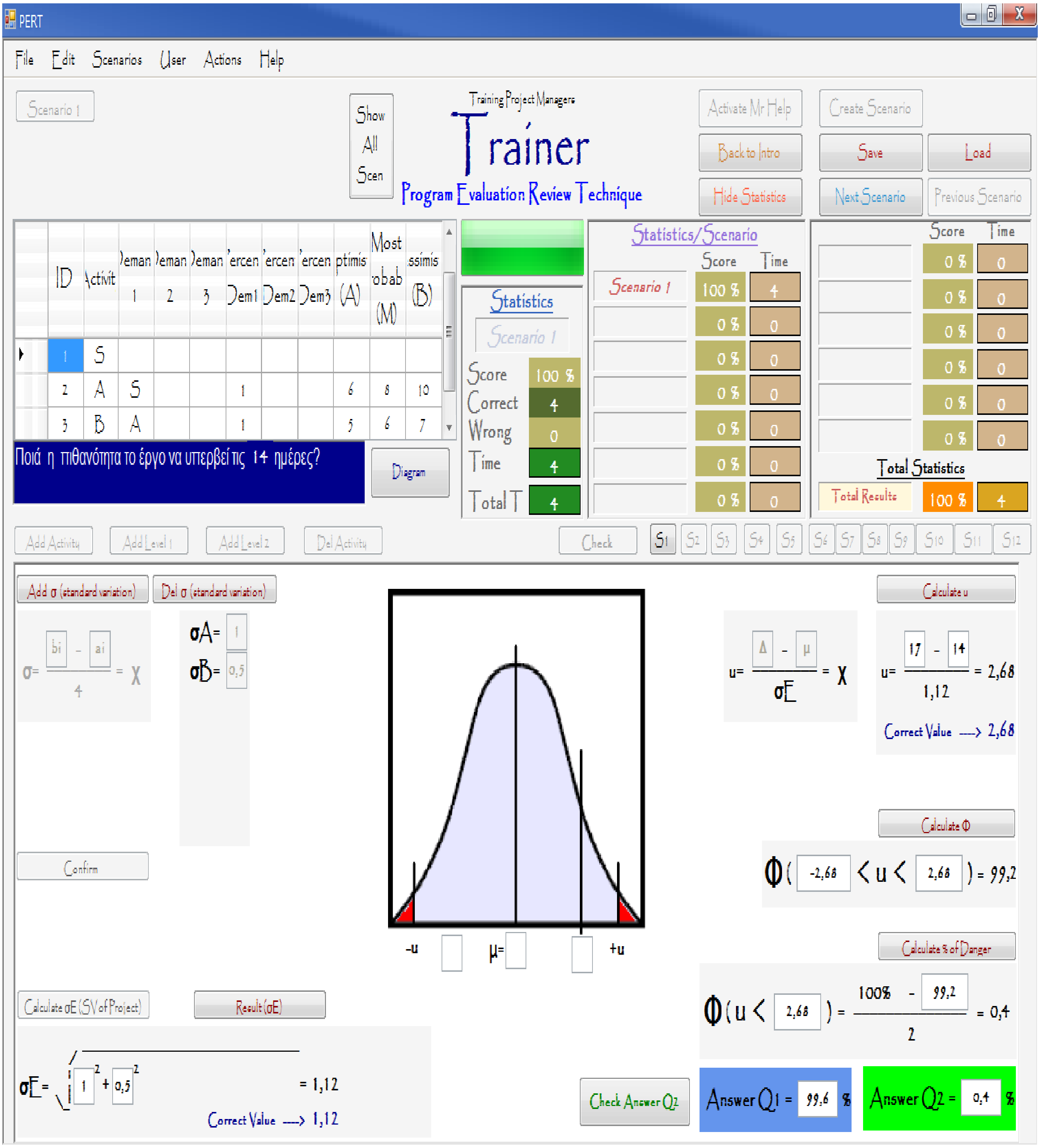
3. Υπολογίζουμε στη συνέχεια το u και μετά το Φ εισάγοντας τις αντίστοιχες τιμές στα κουτάκια.



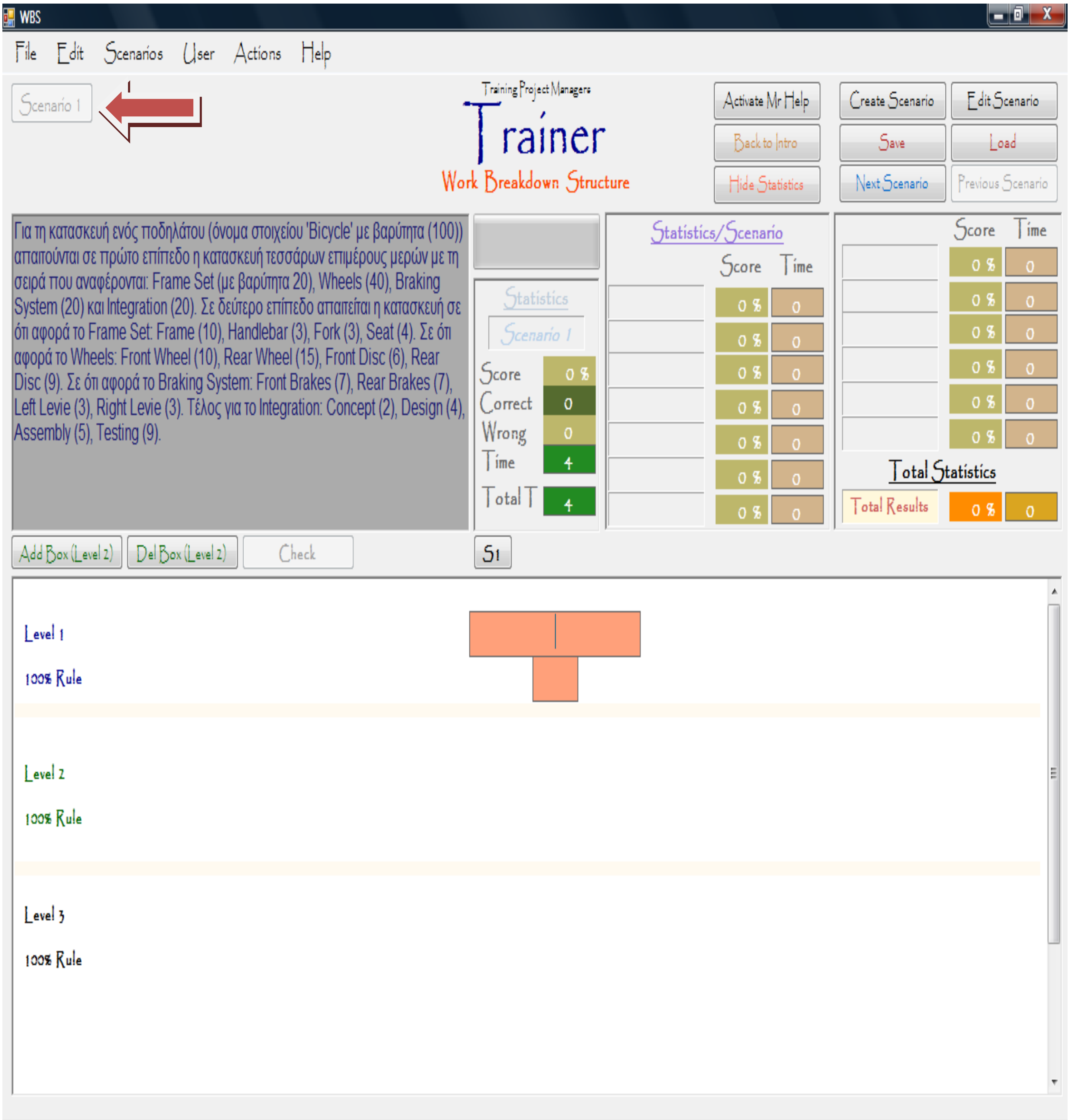


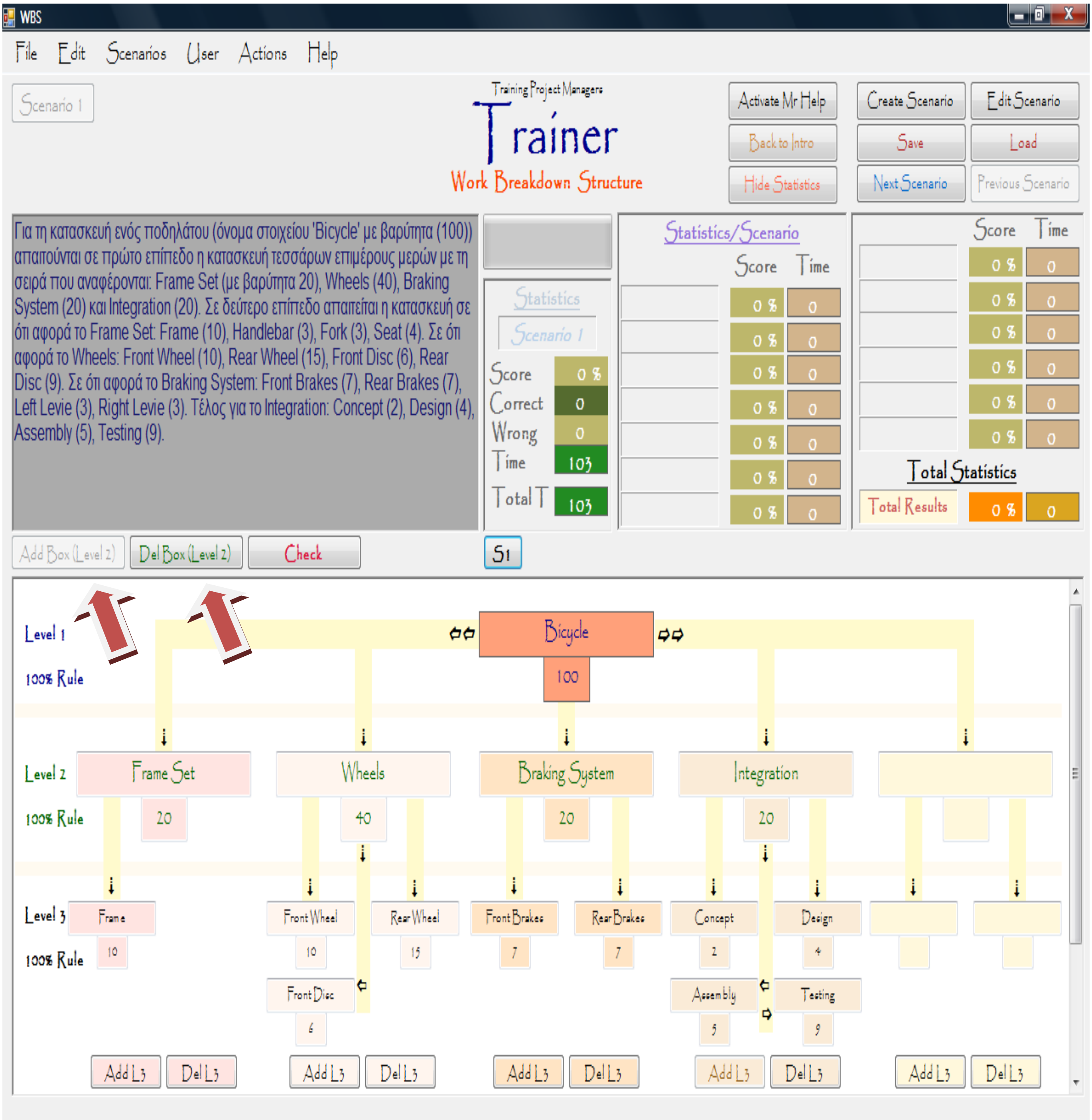

PERT
File Edit Scenarios User Actions Help
Scenario 1
Show All Scen
Training Project Managers
Trainer
Program Evaluation Review Technique
Activate Mr Help
Create Scenario
Back to Intro
Save
Load
Hide Statistics
Next Scenario
Previous Scenario
Statistics/Scenario
Statistics
Scenario 1
Score 100 %
Correct 4
Wrong 0
Time 4
Total T 4
Total Statistics
Total Results 100 % 4
Ποιά η πιθανότητα το έργο να τελειώσει σε 17 ημέρες?
Diagram
Add Activity
Add Level 1
Add Level 2
Del Activity
Check
Add σ (standard variation)
Del σ (standard variation)
σA= 1
σB= 0,5
Confirm
Calculate σE (SV of Project)
Result (σE)
σE = 1,12
Correct Value ---> 1,12
-u
μ=
+u
Calculate u
u= 17 - 14 / 1,12 = 2,68
Correct Value ---> 2,68
Calculate Φ
Φ(-2,68 < u < 2,68) = 99,2
Calculate % of Danger
Φ(u < 2,68) = 100% - 99,2 / 2 = 0,4
Check Answer Q1
Answer Q1 = %

4. Τέλος, αφού τελειώσουμε με όλους τους υπολογισμούς, δίνουμε τις αντίστοιχες απαντήσεις στα ερωτήματα. Το σύστημα ελέγχει κι αν το αποτέλεσμα είναι ορθό προχωράμε στο επόμενο σενάριο.



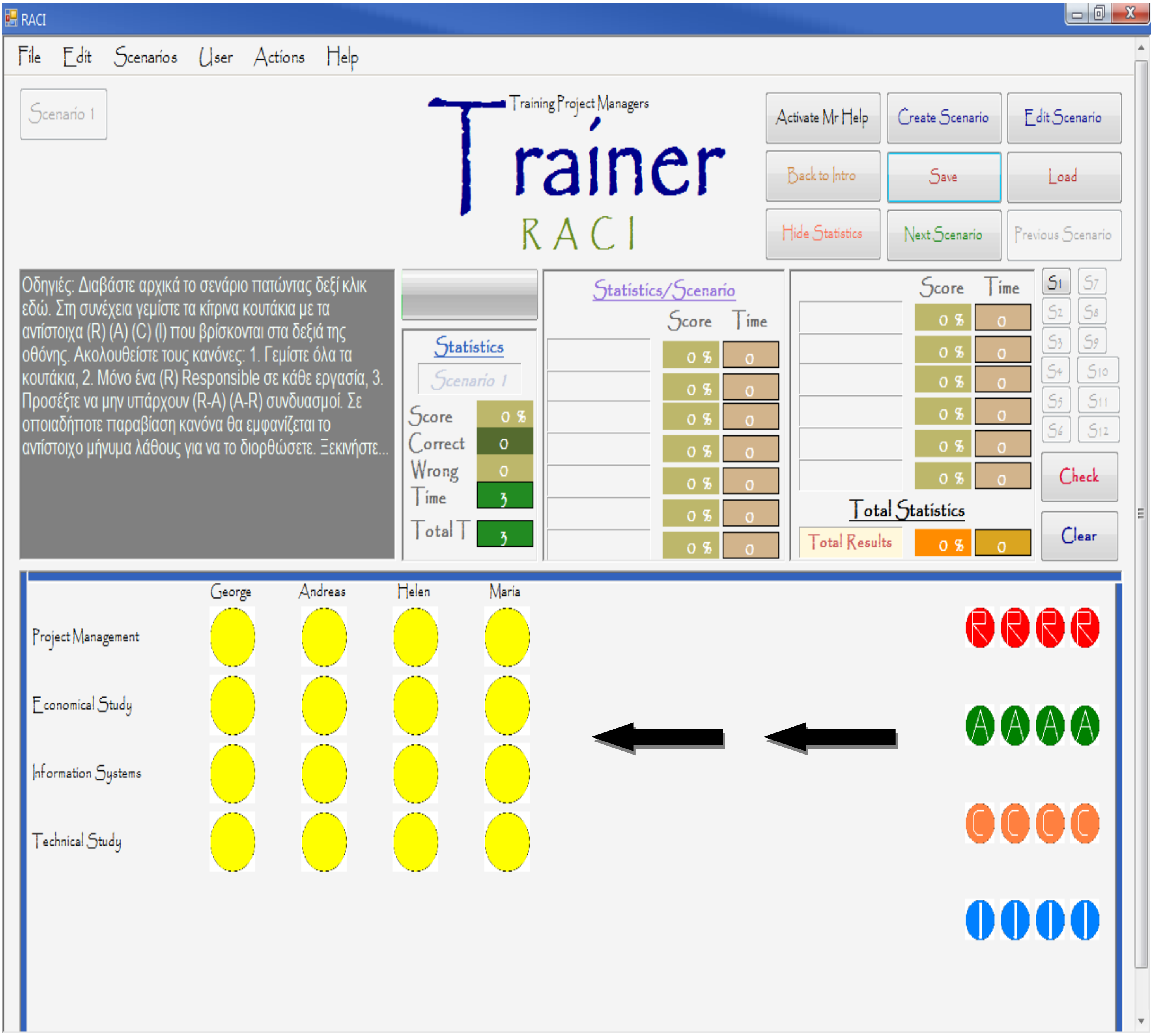
Διεπαφή WBS. Επιλογή του **Scenario 1** και εμφάνιση του σεναρίου για επίλυση.



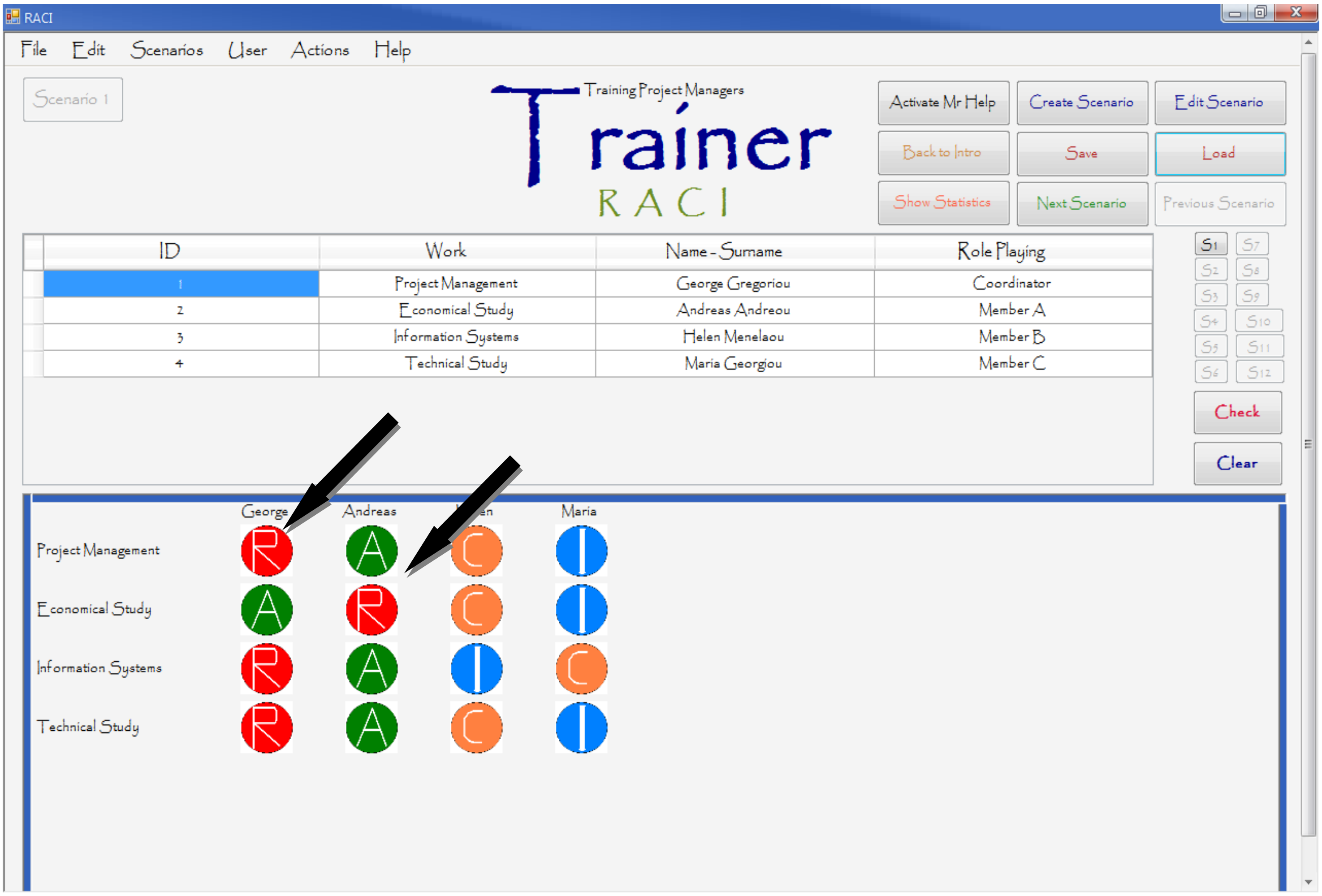


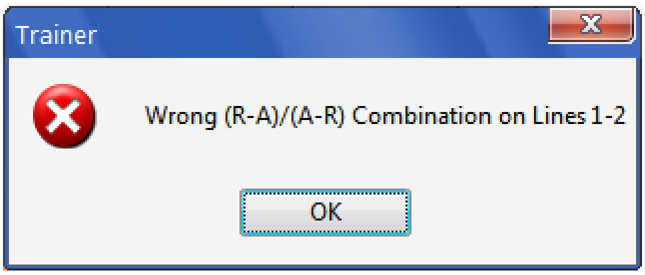
Μετά την πρόσθεση των στοιχείων και τη δημιουργία της δομής WBS.

Διεπαφή RACI. Επιλογή **Scenario 1.** Δημιουργείται ο πίνακας για συμπλήρωση από το χρήστη με βάση το σενάριο.

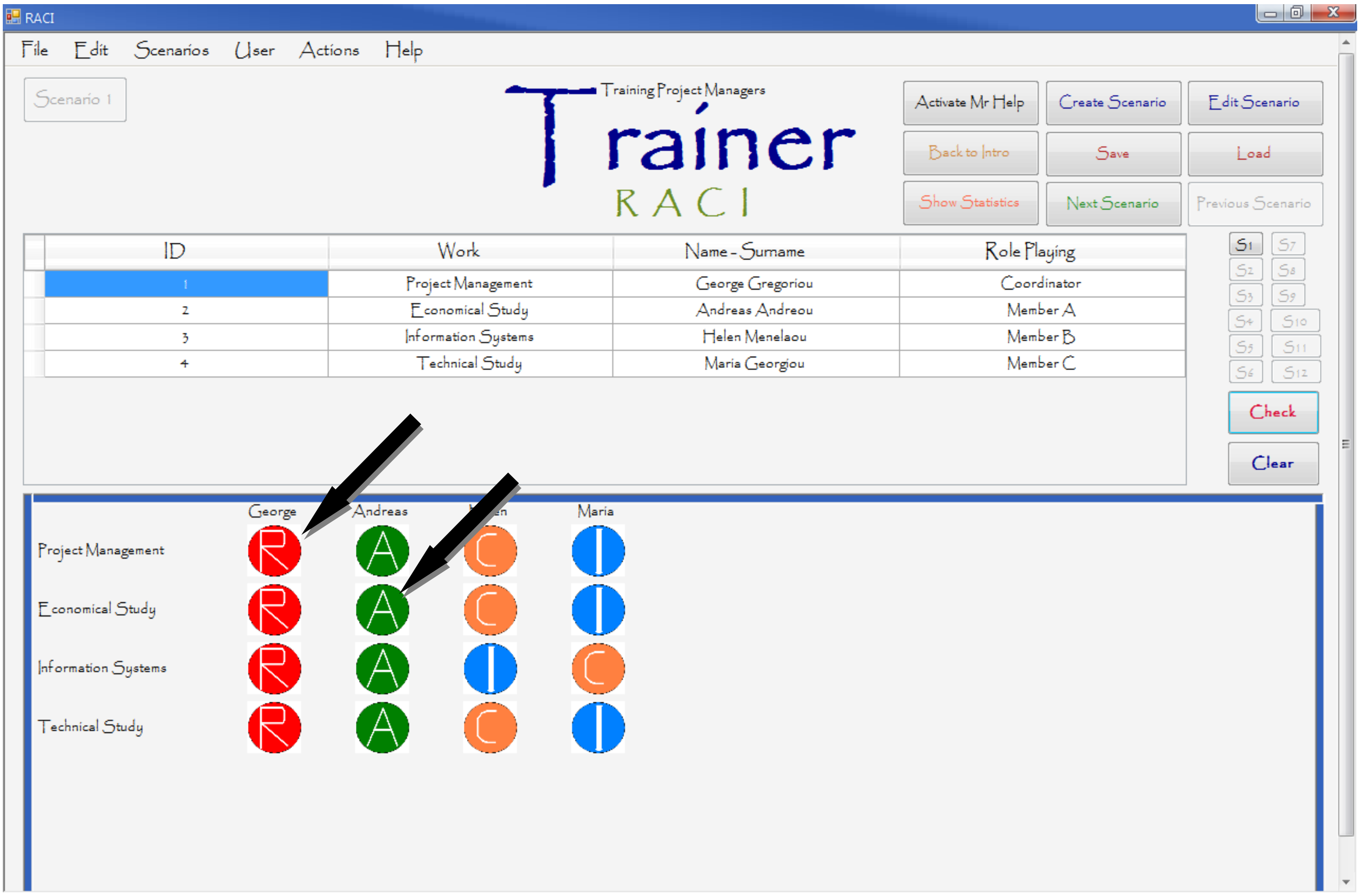


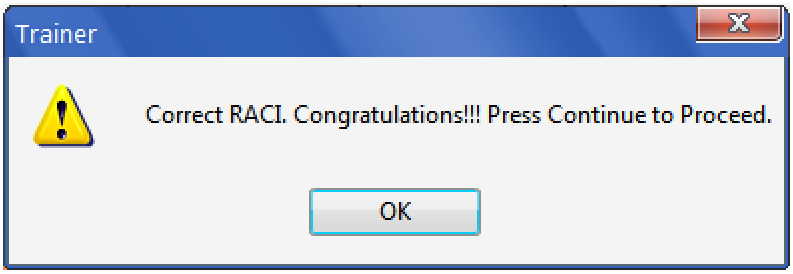
Συμπληρώνουμε (με **drag 'n' drop**) τα κίτρινα κουτάκια με τα αντίστοιχα R, A, C, I κουτάκια στα δεξιά της οθόνης. Κάνουμε τον έλεγχο (**Check**). Έχουμε πρόβλημα διότι υπάρχει R-A, A-R συνδυασμός στις γραμμές 1-2 όπως μας προειδοποιεί και το μήνυμα.





Διορθώνουμε το λάθος.





Ενδεικτικός κώδικας της Διεπαφής CPM του εκπαιδευτικού λογισμικού Trainer©.

```
Imports System.Drawing
Imports System.Drawing.Drawing2D
Imports System.Threading
Imports System.Math
Imports System
Imports System.IO
Imports System.Text
Public Class B_Main
    'Declarations
    Dim act1CTL As TextBox = New TextBox : Dim act2CTL As TextBox = New TextBox : Dim act3CTL As TextBox = New
TextBox : Dim text1_1CTL As TextBox = New TextBox : Dim text1_2CTL As TextBox = New TextBox : Dim text1_3CTL As
TextBox = New TextBox : Dim text2_1CTL As TextBox = New TextBox : Dim text2_2CTL As TextBox = New TextBox : Dim
text2_3CTL As TextBox = New TextBox : Dim text3_1CTL As TextBox = New TextBox : Dim text3_2CTL As TextBox = New
TextBox : Dim text3_3CTL As TextBox = New TextBox
    Dim act4CTL As TextBox = New TextBox : Dim act5CTL As TextBox = New TextBox : Dim act6CTL As TextBox = New
TextBox : Dim text4_1CTL As TextBox = New TextBox : Dim text4_2CTL As TextBox = New TextBox : Dim text4_3CTL As
TextBox = New TextBox : Dim text5_1CTL As TextBox = New TextBox : Dim text5_2CTL As TextBox = New TextBox : Dim
text5_3CTL As TextBox = New TextBox : Dim text6_1CTL As TextBox = New TextBox : Dim text6_2CTL As TextBox = New
TextBox : Dim text6_3CTL As TextBox = New TextBox
    Dim act7CTL As TextBox = New TextBox : Dim act8CTL As TextBox = New TextBox : Dim act9CTL As TextBox = New
TextBox : Dim text7_1CTL As TextBox = New TextBox : Dim text7_2CTL As TextBox = New TextBox : Dim text7_3CTL As
TextBox = New TextBox : Dim text8_1CTL As TextBox = New TextBox : Dim text8_2CTL As TextBox = New TextBox : Dim
text8_3CTL As TextBox = New TextBox : Dim text9_1CTL As TextBox = New TextBox : Dim text9_2CTL As TextBox = New
TextBox : Dim text9_3CTL As TextBox = New TextBox
    Dim act10CTL As TextBox = New TextBox : Dim act11CTL As TextBox = New TextBox : Dim act12CTL As TextBox =
New TextBox : Dim text10_1CTL As TextBox = New TextBox : Dim text10_2CTL As TextBox = New TextBox : Dim
text10_3CTL As TextBox = New TextBox : Dim text11_1CTL As TextBox = New TextBox : Dim text11_2CTL As TextBox =
New TextBox : Dim text11_3CTL As TextBox = New TextBox : Dim text12_1CTL As TextBox = New TextBox : Dim
text12_2CTL As TextBox = New TextBox : Dim text12_3CTL As TextBox = New TextBox
    'Declarations
    Dim actp1CTL As TextBox = New TextBox : Dim actp2CTL As TextBox = New TextBox : Dim actp3CTL As TextBox =
New TextBox : Dim textp1_1CTL As TextBox = New TextBox : Dim textp1_2CTL As TextBox = New TextBox : Dim
textp1_3CTL As TextBox = New TextBox : Dim textp2_1CTL As TextBox = New TextBox : Dim textp2_2CTL As TextBox =
New TextBox : Dim textp2_3CTL As TextBox = New TextBox : Dim textp3_1CTL As TextBox = New TextBox : Dim
textp3_2CTL As TextBox = New TextBox : Dim textp3_3CTL As TextBox = New TextBox
    Dim actp4CTL As TextBox = New TextBox : Dim actp5CTL As TextBox = New TextBox : Dim actp6CTL As TextBox =
New TextBox : Dim textp4_1CTL As TextBox = New TextBox : Dim textp4_2CTL As TextBox = New TextBox : Dim
textp4_3CTL As TextBox = New TextBox : Dim textp5_1CTL As TextBox = New TextBox : Dim textp5_2CTL As TextBox =
New TextBox : Dim textp5_3CTL As TextBox = New TextBox : Dim textp6_1CTL As TextBox = New TextBox : Dim
textp6_2CTL As TextBox = New TextBox : Dim textp6_3CTL As TextBox = New TextBox
    Dim actp7CTL As TextBox = New TextBox : Dim actp8CTL As TextBox = New TextBox : Dim actp9CTL As TextBox =
New TextBox : Dim textp7_1CTL As TextBox = New TextBox : Dim textp7_2CTL As TextBox = New TextBox : Dim
textp7_3CTL As TextBox = New TextBox : Dim textp8_1CTL As TextBox = New TextBox : Dim textp8_2CTL As TextBox =
New TextBox : Dim textp8_3CTL As TextBox = New TextBox : Dim textp9_1CTL As TextBox = New TextBox : Dim
textp9_2CTL As TextBox = New TextBox : Dim textp9_3CTL As TextBox = New TextBox
    Dim actp10CTL As TextBox = New TextBox : Dim actp11CTL As TextBox = New TextBox : Dim actp12CTL As TextBox
= New TextBox : Dim textp10_1CTL As TextBox = New TextBox : Dim textp10_2CTL As TextBox = New TextBox : Dim
textp10_3CTL As TextBox = New TextBox : Dim textp11_1CTL As TextBox = New TextBox : Dim textp11_2CTL As TextBox
= New TextBox : Dim textp11_3CTL As TextBox = New TextBox : Dim textp12_1CTL As TextBox = New TextBox : Dim
textp12_2CTL As TextBox = New TextBox : Dim textp12_3CTL As TextBox = New TextBox
    'Declarations
    Dim actpp1CTL As TextBox = New TextBox : Dim actpp2CTL As TextBox = New TextBox : Dim actpp3CTL As TextBox
= New TextBox : Dim textpp1_1CTL As TextBox = New TextBox : Dim textpp1_2CTL As TextBox = New TextBox : Dim
textpp1_3CTL As TextBox = New TextBox : Dim textpp2_1CTL As TextBox = New TextBox : Dim textpp2_2CTL As TextBox
= New TextBox : Dim textpp2_3CTL As TextBox = New TextBox : Dim textpp3_1CTL As TextBox = New TextBox : Dim
textpp3_2CTL As TextBox = New TextBox : Dim textpp3_3CTL As TextBox = New TextBox
    Dim actpp4CTL As TextBox = New TextBox : Dim actpp5CTL As TextBox = New TextBox : Dim actpp6CTL As TextBox
= New TextBox : Dim textpp4_1CTL As TextBox = New TextBox : Dim textpp4_2CTL As TextBox = New TextBox : Dim
textpp4_3CTL As TextBox = New TextBox : Dim textpp5_1CTL As TextBox = New TextBox : Dim textpp5_2CTL As TextBox
= New TextBox : Dim textpp5_3CTL As TextBox = New TextBox : Dim textpp6_1CTL As TextBox = New TextBox : Dim
textpp6_2CTL As TextBox = New TextBox : Dim textpp6_3CTL As TextBox = New TextBox
    Dim actpp7CTL As TextBox = New TextBox : Dim actpp8CTL As TextBox = New TextBox : Dim actpp9CTL As TextBox
```

```
Private Sub BtnScen1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnScen1.Click
        'Counter
        BtnNext.Tag = 1
        BtnNext.Enabled = True
        'Fill the Table
        DataGrid.DataSource = Scenario1BindingSource :
Me.Scenario1TableAdapter.Fill(Me.MyScenariosDataSet.Scenario1)
        'Scenario Text
        RTxtScen.Text = "Εργο Ανέγερσης Κτιρίου: Για το έργο της ανέγερσης του κτιρίου της Τράπεζας της
Ελλάδος είχαμε τις εξής κύριες δραστηριότητες: 1. Εργοταξιακές Εγκαταστάσεις με κωδικό εργασίας(A) και
διάρκεια εκτέλεσης μία εβδομάδα με καμία άλλη δραστηριότητα ως προαπαιτούμενη, 2. Χωματουργικά με κωδικό(B)
και με διάρκεια μία εβδομάδα με την A ως προαπαιτούμενη."
        RTxtScen.Visible = True : RTxtScen.BringToFront()
        'Enable Buttons Add,Del
        BtnAddA.Enabled = True : BtnScen1.Enabled = False : BtnScen1.Tag = 1 : LblScen.Text = "Scenario 1"
        LblCorrect.Text = 0 : LblWrong.Text = 0 : LblScore.Text = 0
        Test_S1.Enabled = True : BtnCreScen.Enabled = False : BtnEditScen.Enabled = True
        '***********************************
        'Timer
        stopLoop = 0
        sW.Start()
        sZ.Start()
        Do
            My.Application.DoEvents()
            'Calculate to get the current seconds.
            sec = sW.ElapsedMilliseconds \ 1000
            tsec = sZ.ElapsedMilliseconds \ 1000
            'Get the elapsed time in seconds.
            LblTime.Text = sec.ToString
            LblWTime.Text = tsec.ToString
            '***********************************
            'Pass Results To Total Statistics Panel
            LblScen1.Text = LblScen.Text : LblScore1.Text = LblScore.Text : LblT1.Text = LblTime.Text :
LblTScore.Text = LblScore.Text : x = LblTScore.Text : LblTotalTime.Text = LblWTime.Text
            '***********************************
            'Continue to loop until the stop button is pushed.
        Loop Until stopLoop = 1
          End Sub

Private Sub BtnScen2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnScen2.Click
        'Counter
        BtnNext.Tag = 2
        BtnNext.Enabled = True : BtnPrevious.Enabled = True
        'Fill the Table
        DataGrid.DataSource = Scenario2BindingSource :
Me.Scenario2TableAdapter.Fill(Me.MyScenariosDataSet.Scenario2)
        'Scenario Text
        RTxtScen.Text = "Εργο Ανέγερσης Κτιρίου: Για το έργο της ανέγερσης του κτιρίου της Τράπεζας της
Ελλάδος είχαμε τις εξής κύριες δραστηριότητες: 1. Εργοταξιακές Εγκαταστάσεις με κωδικό εργασίας(A) και
διάρκεια εκτέλεσης τρείς εβδομάδες με καμία άλλη δραστηριότητα ως προαπαιτούμενη, 2. Χωματουργικά με κωδικό(B)
και με διάρκεια δύο εβδομάδες με καμία άλλη δραστηριότητα ως προαπαιτούμενη, 3. Σκυροδετήσεις με κωδικό(C) και
με διάρκεια μία εβδομάδα με την A και την B ως προαπαιτούμενες δραστηριότητες."
        RTxtScen.Visible = True : RTxtScen.BringToFront()
        'Enable Buttons Add,Del
        BtnAddA.Enabled = True : BtnScen2.Enabled = False : BtnScen2.Tag = 1 : BtnScen1.Tag = 0 : LblScen.Text
= "Scenario 2"
        LblCorrect.Text = 0 : LblWrong.Text = 0 : LblScore.Text = 0 : Progress.Value = 0
        Test_S1.Enabled = False : Test_S2.Enabled = True : BtnCreScen.Enabled = False : BtnEditScen.Enabled =
True
        '***********************************
        'Timer
        stopLoop = 0
        sW.Reset()
        sW.Start()
        sZ.Start()
        Do
            My.Application.DoEvents()
            'Calculate to get the current seconds.
            sec = sW.ElapsedMilliseconds \ 1000
            tsec = sZ.ElapsedMilliseconds \ 1000
```

```
        '-----------------------------------------------------
        'Main Check (Dynamic Check)
        '----------------------------------------------------
        'Check Activity Name & Duration - Check ES & LS
        If m = 1 Then
            For i = 0 To j - 1
                LblRow1i.Text = DataGrid.Item(1, i).Value
                LblRow2i.Text = DataGridTemp.Item(1, i).Value
                LblRow3i.Text = DataGrid.Item(2, i).Value
                LblRow4i.Text = DataGridTemp.Item(2, i).Value
                '-------------------------------------------
                LblRow5i.Text = DataGridTemp.Item(7, i).Value
                LblRow6i.Text = DataGridTemp.Item(8, i).Value
                LblRow7i.Text = DataGridTemp.Item(9, i).Value
                LblRow8i.Text = DataGridTemp.Item(10, i).Value
                '--------------------------------------------
                'Real Duration = Duration * Percent
                LblRow9i.Text = DataGridTemp.Item(11, i).Value
                LblRow10i.Text = DataGridTemp.Item(12, i).Value
                '--------------------------------------------
                'Main Check Loop
                If LblRow1i.Text = LblRow2i.Text Then y = y + 1 Else MsgBox("Wrong Activity Name in Box " & i +
1, MsgBoxStyle.Critical)
                If LblRow3i.Text = LblRow4i.Text Then y = y + 1 Else MsgBox("Wrong Duration in Box " & i + 1,
MsgBoxStyle.Critical)
                If LblRow5i.Text = LblRow6i.Text Then y = y + 1 Else MsgBox("Wrong Early Start in Box " & i + 1,
MsgBoxStyle.Critical)
                If LblRow7i.Text = LblRow8i.Text Then y = y + 1 Else MsgBox("Wrong Late Start in Box " & i + 1,
MsgBoxStyle.Critical)
                If LblRow9i.Text = LblRow10i.Text Then y = y + 1 Else MsgBox("Wrong Duration in TextBox " & i +
1, MsgBoxStyle.Critical)
            Next i
            If y / 5 = j Then m = 2
        End If
        '-----------------------------------------------------
        'Check Network
        If m = 2 Then
            For i = 0 To j - 2
                'Assign Cell Values
                LblRow1i.Text = DataGrid.Item(3, i + 1).GetEditedFormattedValue(i + 1,
DataGridViewDataErrorContexts.Display)
                LblRow2i.Text = DataGridTemp.Item(3, i + 1).GetEditedFormattedValue(i + 1,
DataGridViewDataErrorContexts.Display)
                LblRow3i.Text = DataGrid.Item(4, i + 1).GetEditedFormattedValue(i + 1,
DataGridViewDataErrorContexts.Display)
                LblRow4i.Text = DataGridTemp.Item(4, i + 1).GetEditedFormattedValue(i + 1,
DataGridViewDataErrorContexts.Display)
                LblRow5i.Text = DataGrid.Item(5, i + 1).GetEditedFormattedValue(i + 1,
DataGridViewDataErrorContexts.Display)
                LblRow6i.Text = DataGridTemp.Item(5, i + 1).GetEditedFormattedValue(i + 1,
DataGridViewDataErrorContexts.Display)
                'Main Check Loop
                If LblRow1i.Text = LblRow2i.Text And LblRow3i.Text = LblRow4i.Text And LblRow5i.Text =
LblRow6i.Text Then q = q + 1 Else m = 1
            Next i
            If q = j - 1 Then m = 5 Else MsgBox("Wrong Network", MsgBoxStyle.Critical)
        End If
        '-----------------------------------------------------
        'Nullification of DataGridTemp Table
        For n = 1 To k - 1
            For i = 0 To j - 1
                'DataGridTemp.Item(n, i).Value = ""
            Next i
        Next n
        '------------------------------------------------------------------------------------------------------
--------------
        'Results / Stats
        LblCorrect.Text = (y / 5) : LblWrong.Text = (j - (y / 5)) : LblScore.Text = ((y / 5) * 100) / j :
Progress.Value = LblScore.Text
```